The Anatomy of an Edit: Mechanism-Guided Activation Steering for Knowledge Editing

Anonymous ACL submission

Abstract

Large language models (LLMs) are increasingly used as knowledge bases, but keeping them up to date requires targeted knowledge editing (KE). However, it remains unclear how edits are implemented inside the model once applied. In this work, we take a mechanistic view of KE using neuron-level knowledge attribution (NLKA). Unlike prior work that focuses on pre-edit causal tracing and localization, we use post-edit attribution – contrasting successful and failed edits – to isolate the computations that shift when an edit succeeds. Across representative KE methods, we find a consistent pattern: mid-to-late attention predominantly promotes the new target, while attention and FFN modules cooperate to suppress the original fact. Motivated by these findings, we propose **MEGA**, a **ME**chanism-**G**uided **A**ctivation steering method that performs attention-residual interventions in attribution-aligned regions without modifying model weights. On CounterFact and Popular, MEGA achieves strong editing performance across KE metrics on GPT2-XL and LLaMA2-7B. Overall, our results elevate post-edit attribution from analysis to engineering signal: by pinpointing where and how edits take hold, it powers MEGA to deliver reliable, architecture-agnostic knowledge edits.

1 Introduction

Large language models (LLMs) are increasingly used as knowledge bases, but their internal facts can be outdated or application-specific. Knowledge editing aims to update specific factual associations without retraining from scratch or degrading unrelated behavior. Existing approaches include weight-editing methods such as ROME and MEMIT (Meng et al., 2023a,b), fine-tuning and parameter-efficient adaptation (Hu et al., 2021), in-context editing (Zheng et al., 2023), and activation steering methods (Scialanga et al., 2025).

Despite progress, many methods choose edit layers based on pre-edit localization with causal tracing, i.e., where a fact appears to originate in the base model. However, such localization has been shown to be a poor predictor of effective intervention sites (Hase et al., 2023). Instead, we argue that analyzing post-edit attribution can reveal which components actually implement the updated knowledge and directly inform intervention design.

We take a mechanistic view of knowledge editing on GPT2-XL and LLaMA2-7B. Using neuron-level knowledge attribution (NLKA) (Yu and Ananiadou, 2024), we compare four representative methods—ROME, MEMIT, FT, and in-context editing—by tracking how layer- and component-wise contributions to target-token log-probabilities shift before and after editing. Across methods, we observe a consistent signature: (i) attention and FFNs jointly suppress the original fact; (ii) mid-to-late attention predominantly promotes the new target; and (iii) late residual streams act as high-leverage aggregation points, as suggested by attribution trends.

Guided by these findings, we propose **MEGA**, a mechanism-guided activation steering method that performs attention-residual interventions in attribution-aligned regions. As shown in Figure 1, rather than relying on heuristic last-layer edits, MEGA targets the mid-to-late layers where attribution and ablations indicate maximal leverage. On CounterFact and Popular, MEGA yields strong edits across GPT2-XL and LLaMA2-7B, while FFN-only steering shows limited gains.

Overall, our contributions are as follows. First, **post-edit knowledge attribution**: we present a unified post-edit NLKA analysis across four KE methods and two LLM families, revealing consistent component roles and contrasting with prior pre-edit attribution approaches. Second, **mechanism-guided activation steering**: we introduce MEGA, an attention-residual steering method aligned with attribution-identified control regions, achieving strong edits without modifying model weights. Fi-

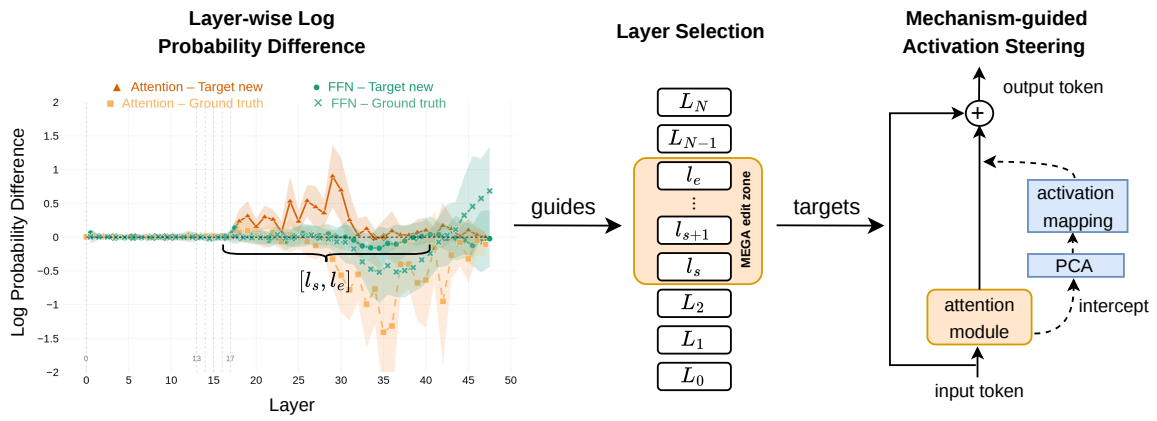


Figure 1: **Overview of MEGA. (Left):** Post-edit NLKA identifies mid-to-late attention layers with high leverage on the target vs. original token. **(Middle):** MEGA uses this signal to select an edit zone and **(Right):** applies PCA-stabilized attention–residual steering at inference time, promoting the edited fact without changing weights.

nally, **bridging analysis and intervention**: we show that post-edit attribution is not merely diagnostic but actionable, enabling mechanism-aware editing strategies that generalize across LLMs.

2 Background and Related Work

Knowledge attribution. Knowledge attribution methods aim to localize which internal units or components are responsible for a model's predictions. NLKA (Yu and Ananiadou, 2024) provides an observational, post-hoc decomposition of a token's log-probability into additive contributions across layers and components (e.g., attention and FFN). NLKA measures how each component's output, when integrated into the residual stream and projected through the final layer norm and LM head, changes the log-probability of a target token. The model itself is not modified, enabling direct comparison between edited and unedited networks and identification of promising intervention targets.

Knowledge editing. Knowledge editing aims to update factual knowledge in pretrained language models while preserving general capabilities (Meng et al., 2022, 2023b; Zhang et al., 2024). Given a fact $(x_{\text{src}}, y_{\text{tgt}})$, the objective is to increase the likelihood of y_{tgt} in relevant contexts without affecting unrelated predictions.

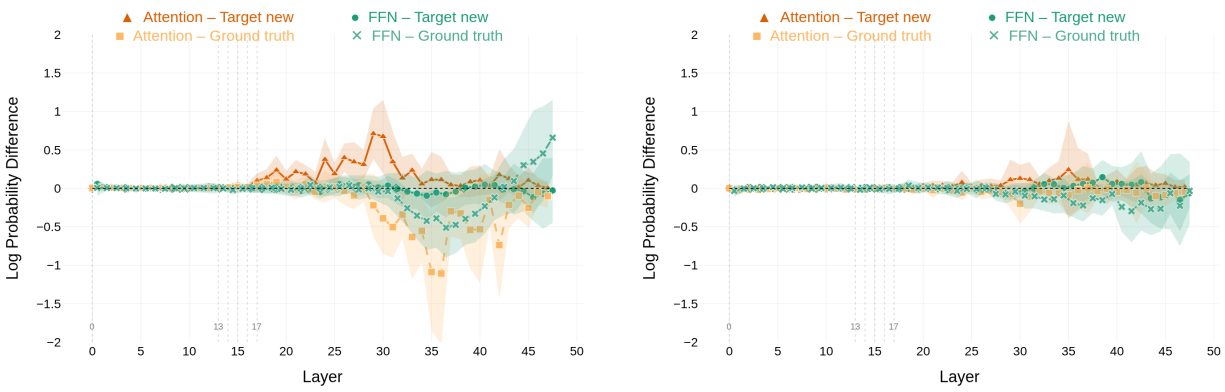
Prior work spans weight-level editors such as ROME and MEMIT (Meng et al., 2023a,b; Wang et al., 2025), FT and parameter-efficient adaptation methods (Hu et al., 2021; Houlsby et al., 2019; Wang et al., 2024a), memory-based and in-context approaches including SERAC, GRACE, and IKE (Mitchell et al., 2022; Hartvigsen et al., 2023; Zheng et al., 2023), and inference-time activation steering methods such as SAKE (Scialanga et al., 2025). SAKE learns an optimal-transport map on final-layer residual states and uses a scope detector to apply edits only in scope. Our work connects this line with NLKA and uses component-wise attribution to inform *where* and *how* to steer.

3 Experimental Setup

Models. We evaluate on GPT2-XL (Radford et al., 2019) and LLaMA2-7B (Touvron et al., 2023), two widely used LLMs in knowledge editing and mechanistic analysis (Meng et al., 2023b; Scialanga et al., 2025; Gupta et al., 2024).

Metrics. On CounterFact, we report accuracy (Acc; edit success on the rewritten fact), generalization (Gen; success on paraphrased prompts), and specificity (Spec; preservation of unrelated facts), following standard knowledge editing evaluation protocols (Meng et al., 2022). Following prior KE work (Scialanga et al., 2025), we additionally include doubt robustness metrics (DI, DII), originally introduced by Ma et al. (2024), which evaluate robustness under prompts that introduce uncertainty or challenge the edited fact, allowing us to assess the contextual robustness of edits.

On Popular, we report Subject Aliasing (SA), which evaluates generalization over subject synonyms; Compositionality I (CI) and Compositionality II (CII), which measure multi-hop reasoning; and Relation Specificity (RS), which measures logical locality by verifying that edits do not alter outputs for the same subject under different relations (Cohen et al., 2023). Together, these metrics evaluate whether edits generalize consistently to



(a) Success (L13–17)

(b) Failure (L13–17)

Figure 2: **MEMIT success vs. failure.** Mean contribution differences across multiple cases; positive promotes new target, negative suppresses original.

Table 1: Best-performing methods on CounterFact for GPT2-XL (left) and LLaMA2-7B (right). We report accuracy (Acc), generalization (Gen), specificity (Spec), and two doubt-robustness metrics (DI, DII), plus their average (Avg).

GPT2-XL

Config	Lyrs	PCA	Acc	Gen	Spec	DI	DII	Avg
FT	0	–	0.97	0.09	0.80	0.17	0.03	0.41
ROME	17	–	1.00	0.80	0.86	0.36	0.07	0.62
MEMIT	13–17	–	0.81	0.45	0.99	0.25	0.03	0.51
IKE	–	–	0.99	0.70	0.17	0.06	0.01	0.39
SAKE	47	–	0.99	0.85	0.83	0.99	0.99	0.93
MEGA	15–40	512	0.99	0.93	0.87	1.00	1.00	**0.96**

LLaMA2-7B

Config	Lyrs	PCA	Acc	Gen	Spec	DI	DII	Avg
FT	21	–	0.45	0.43	0.44	0.01	0.66	0.40
ROME	5	–	1.00	0.81	0.91	0.03	0.41	0.63
MEMIT	4–8	–	1.00	0.83	0.94	0.12	0.54	0.69
IKE	–	–	1.00	0.89	0.34	0.00	0.05	0.46
SAKE	31	–	0.98	0.83	0.84	0.92	0.96	0.91
MEGA	13–31	512	0.99	0.91	0.89	1.00	0.97	**0.95**

logical implications of the modified fact. Detailed metric definitions are provided in App. B.

Datasets. For factual rewriting, we use the CounterFact subset provided by EasyEdit (1,031 examples) (Wang et al., 2024b). Due to the computational cost of large-scale editing experiments, we evaluate on a fixed subset of 120 edits, which provides sufficient coverage for reliable comparison while maintaining practical runtime. For compositional generalization, we use the Popular subset of RippleEdits (Cohen et al., 2023). We evaluate on a fixed balanced subset of 120 edits (App. A.2)

4 Mechanistic Attribution of Edited Facts

We use NLKA (§2) to quantify how edits redistribute component-wise support for the object token. For each post-edit model produced by the evaluated KE methods (FT, ROME, MEMIT, and IKE), we compute contribution scores $C_{\ell,c}(y)$ per layer ℓ, component $c \in \{\text{attn}, \text{ffn}\}$, and token y, and analyze their changes relative to the corresponding pre-edit model:

$$\Delta C_{\ell,c}(y) \;=\; C^{\text{edit}}_{\ell,c}(y) \;-\; C^{\text{base}}_{\ell,c}(y).$$

We focus on cases where the base model predicts the original object correctly and the editor succeeds in rewriting it, and average $\Delta C_{\ell,c}(y_{\text{tgt}})$ and $\Delta C_{\ell,c}(y_{\text{orig}})$ across those edits (details in App. C).

Cross-model patterns. Figure 2 contrasts layer-wise $\Delta C_{\ell,c}$ patterns between successful and failed MEMIT edits on GPT2-XL. ROME, FT, and IKE exhibit qualitatively similar trends (App. D.1.1), and we observe comparable structures on LLaMA2-7B (App. D.1.2). Across models and editors, three consistent effects emerge: (i) mid-to-late attention layers carry most of the positive shift toward the new target; (ii) attention and FFNs jointly suppress the original fact in mid layers; and (iii) late residual streams act as high-leverage aggregation points, as suggested by attribution trends. A notable difference is that FFNs contribute minimally to target promotion in GPT2-XL but provide modest positive support in LLaMA2-7B, although attention remains the dominant driver in both. Overall, the shared pattern suggests that edited facts are routed through broadly similar circuit motifs across LLMs, motivating interventions that target these regions.

Table 2: Best-performing methods on Popular for GPT2-XL (left) and LLaMA2-7B (right). We report accuracy (Acc), compositionality (CI, CII), relation specificity (RS), subject aliasing (SA), and their average (Avg).

GPT2-XL

Config	Lyrs	PCA	Acc	CI	CII	RS	SA	Avg
FT	0	–	0.30	0.15	0.00	0.52	0.25	0.24
ROME	17	–	0.96	0.22	0.00	0.27	0.53	0.40
MEMIT	13–17	–	0.55	0.24	0.00	0.38	0.22	0.28
IKE	–	–	0.96	0.09	0.00	0.39	0.01	0.29
SAKE	47	–	0.45	0.60	0.35	0.38	0.70	**0.50**
MEGA	31–47	512	0.80	0.28	0.45	0.32	0.64	**0.50**

LLaMA2-7B

Config	Lyrs	PCA	Acc	CI	CII	RS	SA	Avg
FT	21	–	0.48	0.05	0.02	0.53	0.24	0.26
ROME	5	–	0.99	0.20	0.04	0.53	0.73	**0.50**
MEMIT	4–8	–	0.98	0.18	0.03	0.41	0.79	0.48
IKE	–	–	0.93	0.06	0.00	0.86	0.05	0.38
SAKE	31	–	0.47	0.13	0.24	0.48	0.63	0.39
MEGA	13–31	512	0.82	0.07	0.25	0.33	0.64	0.42

5 Mechanism-Guided Multi-Layer Activation Steering

Our method is inspired by SAKE (Scialanga et al., 2025), an inference-time activation steering approach that edits behavior without modifying weights. SAKE learns an affine transport map between source and target hidden-state distributions and applies it to the final residual stream via forward hooks gated by a scope detector. Given source activations h_s and target activations h_t, it fits $m(h) = Ah + b$ to align the distributions. While SAKE shows that latent distribution matching enables knowledge editing, it treats the residual stream as a monolithic target and intervenes only at the final layer. Guided by the attribution patterns in §4, we instead introduce mechanism-guided steering targeting attribution-identified control regions instead of heuristic last-layer edits.

What to steer. Post-edit attribution shows that target-promoting contributions are concentrated in mid-to-late attention layers and are integrated into the residual stream. This motivates steering along attention-residual pathways where attention outputs are incorporated into the residual state. We therefore use attention-residual steering as the primary intervention mechanism, with attention-only and residual-only variants as ablations.

How to steer. For a chosen layer range L, we collect activations from source and target prompts, project them into a low-dimensional PCA space (512 dimensions by default), and learn a transport map between the two activation distributions. The learned transformation is applied at inference time via forward hooks on the attention-residual stream for layers $\ell \in L$, gated by the same scope detector used in SAKE. Outside the detected scope, the base model remains unchanged.

Where to intervene in depth. Attribution analysis indicates that target-promoting contributions are distributed across mid-to-late attention layers rather than localized at a single depth. Motivated by this pattern, we steer over attribution-aligned mid-to-late windows instead of fixing the final layer. Attribution trends suggest that late layers act as aggregation points. However, windowed mid-to-late steering better reflects the distributed structure revealed by attribution and yields more consistent performance. Our best configurations steer attention-residual representations over attribution-aligned mid-to-late windows, with the optimal range varying by dataset and model (e.g., 15–40 for CounterFact GPT2-XL, 31–47 for Popular GPT2-XL, and 13–31 for LLaMA2-7B; Tables 1, 2).

We fix the PCA dimension to 512 across models and datasets, which provides a strong trade-off between performance and efficiency. In preliminary experiments, PCA-based steering was more stable and efficient than operating in the full space, and we therefore use it in our reported results.

Overall, our approach preserves the no-weight-update property of activation steering while aligning both the intervention target and depth with mechanistic attribution, replacing heuristic last-layer edits with attribution-guided attention-residual steering.

Conclusion

We combined NLKA with activation steering to study how edited facts are implemented inside GPT2-XL and LLaMA2-7B. Across four editors, we found a consistent pattern: mid-to-late attention promotes the new target while late residual blocks appear to aggregate belief shifts into a decisive preference. Guided by these insights, we introduce a mechanism-guided activation steering approach that targets attribution-aligned attention and residual regions, achieving strong edits on CounterFact and Popular.

Limitations

Our experiments focus on single-fact edits in GPT2-XL and LLaMA2-7B. While these models are widely used in knowledge-editing research, it remains unclear whether the same attribution patterns hold in larger or more recent architectures. In addition, we primarily evaluate single-edit scenarios; extending mechanism-guided steering to large-scale multi-edit settings remains an open challenge.

Acknowledgments

References

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2023. Evaluating the ripple effects of knowledge editing in language models. *Preprint*, arXiv:2307.12976.

Akshat Gupta, Dev Sajnani, and Gopala Anumanchipalli. 2024. A unified framework for model editing. *Preprint*, arXiv:2403.14236.

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with grace: Lifelong model editing with discrete key-value adaptors. *Preprint*, arXiv:2211.11031.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. 2023. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. *Advances in Neural Information Processing Systems*, 36:17643–17668.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. *Preprint*, arXiv:1902.00751.

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

Xinbei Ma, Tianjie Ju, Jiyang Qiu, Zhuosheng Zhang, Hai Zhao, Lifeng Liu, and Yulong Wang. 2024. On the robustness of editing large language models. In *Conference on Empirical Methods in Natural Language Processing*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 36. ArXiv:2202.05262.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2023a. Locating and editing factual associations in gpt. *Preprint*, arXiv:2202.05262.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023b. Mass-editing memory in a transformer. *Preprint*, arXiv:2210.07229.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. 2022. Memory-based model editing at scale. *Preprint*, arXiv:2206.06520.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Marco Scialanga, Thibault Laugel, Vincent Grari, and Marcin Detyniecki. 2025. Sake: Steering activations for knowledge editing. *Preprint*, arXiv:2503.01751.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Mingyang Wang, Lukas Lange, Heike Adel, Jannik Strötgen, and Hinrich Schuetze. 2024a. Better call SAUL: Fluent and consistent language model editing with generation regularization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7990–8000, Miami, Florida, USA. Association for Computational Linguistics.

Mingyang Wang, Alisa Stoll, Lukas Lange, Heike Adel, Hinrich Schuetze, and Jannik Strötgen. 2025. Bring your own knowledge: A survey of methods for LLM knowledge expansion. In *Proceedings of the First Workshop on Large Language Model Memorization (L2M2)*, pages 150–168, Vienna, Austria. Association for Computational Linguistics.

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. 2024b. Easyedit: An easy-to-use knowledge editing framework for large language models. *Preprint*, arXiv:2308.07269.

Zeping Yu and Sophia Ananiadou. 2024. Neuron-level knowledge attribution in large language models. *Preprint*, arXiv:2312.12141.

Ningyu Zhang, Yunzhi Yao, Bozhong 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, and 3 others. 2024. A comprehensive study of knowledge editing for large language models. *Preprint*, arXiv:2401.01286.

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? *Preprint*, arXiv:2305.12740.

A Dataset Construction

A.1 CounterFact Subset

For factual rewriting experiments, we use the CounterFact dataset introduced by Meng et al. (2023a), which contains knowledge-editing cases of the form $(e, r, o) \rightarrow (e, r, o^*)$. We use the EasyEdit implementation of this dataset (Wang et al., 2024b), which provides a curated version with additional evaluation prompts.

Specifically, we use the EasyEdit dataset variant `counterfact_portability_gpt4.json`, which includes portability prompts designed to test whether edited knowledge transfers to related reasoning tasks. The dataset contains 1,031 editing cases in total. From this set, we select the first 120 edits and evaluate all methods on this fixed subset. These cases include prompts for edit success, paraphrase generalization, and locality tests on unrelated facts.

A.2 RippleEdits Popular Subset

To evaluate compositional generalization, we use the Popular subset of RippleEdits (Cohen et al., 2023), which contains 885 edits involving widely known entities. RippleEdits is designed to measure ripple effects of knowledge editing across related facts and reasoning chains.

Because not all RippleEdits cases contain both Compositionality I (CI) and Compositionality II (CII) evaluation prompts, we construct a balanced 120-example subset to ensure sufficient coverage of both metrics. Specifically, we select:

- 26 edits containing both CI and CII tests,
- 47 edits containing only CI tests,
- 47 edits containing only CII tests.

This strategy ensures that both compositional metrics are well represented while maintaining a balanced evaluation set. All methods are evaluated on the same fixed subset. For each selected edit, we generate paraphrase prompts following the procedure of Scialanga et al. (2025), using the DeepSeek API as the paraphrase generation backend.

B Evaluation Metrics

B.1 CounterFact Metrics

For CounterFact, we report three standard editing metrics. **Accuracy (Acc)** measures whether the edited model produces the new target for the original rewrite prompt. **Generalization (Gen)** measures whether the edit transfers to semantically equivalent paraphrases within the intended editing scope. **Specificity (Spec)** measures locality, i.e., whether the model preserves its original behavior on unrelated prompts outside the editing scope.

Intuitively, a successful edit should change the model's output on in-scope inputs while leaving unrelated inputs unaffected.

B.2 Doubt Robustness Prompts

We additionally evaluate contextual robustness using the doubt-inducing prompts from Ma et al. (2024). Let p denote the original prompt, o the old object, and o^* the edited target object.

DI. A doubt-inducing prompt of the form:

> p, o^* ??? No, it can't be, this contradicts my understanding. It is pretty clear that the answer to p is something else. Rethink and give the correct answer:

DII. A stronger doubt-inducing prompt of the form:

> p, o^*? Really? But I am pretty sure that [old fact with o]. Am I right? Yes, you are right. I apologize for any confusion. Reconsider that the answer to p should be:

DI and DII measure whether the edited belief remains stable under uncertainty-raising or contradiction-raising contexts.

B.3 Popular / RippleEdits Metrics

Each edit has the form $(e, r, o) \rightarrow (e, r, o^*)$, where e is the subject, r the relation, and o^* the edited object. For the Popular subset of RippleEdits (Cohen et al., 2023), we evaluate four metrics designed to capture ripple effects of editing.

Compositionality I (CI). This metric tests whether the edited fact composes correctly with other facts about the edited object. For example, if we edit the sibling of *Prince* to a new person, then the model should also update composed facts such as:

Profession of sibling of Prince → profession of the new sibling

Compositionality II (CII). This metric tests whether the edit propagates through compositions involving a different subject that refers to the edited entity. For example, if the model knows:

Founder of Paisley Park Records → Prince

then changing the sibling of *Prince* should also update:

Sibling of the founder of Paisley Park Records → sibling of Prince after editing

Subject Aliasing (SA). This metric evaluates whether the edit generalizes to aliases of the subject. For example, after editing the sibling of *Prince*, the model should produce the same edited result for:

Sibling of Prince Roger Nelson → same edited sibling

Relation Specificity (RS). This metric measures locality by verifying that relations unrelated to the edited fact remain unchanged. For example, editing the sibling of *Prince* should not affect:

Mother of Prince

C Neuron-Level Attribution Procedure

We analyze how knowledge edits affect internal computation using neuron-level knowledge attribution (NLKA) (Yu and Ananiadou, 2024). The method decomposes the log-probability of a target token into additive contributions from transformer components.

Residual stream decomposition. For each layer ℓ, the residual stream evolves as

$$\mathrm{Lres}_\ell = \mathrm{Lin}_\ell + \mathrm{ATTN}_\ell,$$
$$\mathrm{Lout}_\ell = \mathrm{Lres}_\ell + \mathrm{FFN}_\ell.$$

Here Lin_ℓ denotes the input to layer ℓ, ATTN_ℓ and FFN_ℓ are the attention and feed-forward outputs at that layer, Lres_ℓ is the intermediate residual state after adding attention, and Lout_ℓ is the final layer output.

Contribution scores. Following Yu and Ananiadou (2024), the contribution of a component c at layer ℓ to a target token y is defined as the change in log-probability when the component output is added to the residual stream:

$$C_{\ell,c}(y) = \log p(y \mid h + c_\ell) - \log p(y \mid h),$$

where h denotes the residual state before the component and c_ℓ is the component output.

For each prompt, we compute contribution scores for attention and FFN components across all layers by measuring how their outputs change the log-probability of the target token. Token probabilities are obtained by projecting the residual state through the final layer normalization and LM head. This attribution procedure is purely observational and does not modify the model parameters.

Edit comparison. To isolate editing effects, we compare contribution scores between edited and base models:

$$\Delta C_{\ell,c}(y) = C^{\text{edit}}_{\ell,c}(y) - C^{\text{base}}_{\ell,c}(y).$$

Aggregation. We compute $\Delta C_{\ell,c}(y)$ for both the new target token y_{tgt} and the original token y_{orig}, and average results across all successful edits in the evaluation set.

D Additional Attribution Analysis

D.1 CounterFact

D.1.1 GPT2-XL

Figures 3, 4, and 5 show attribution patterns for successful and failed edits under IKE, ROME, and fine-tuning on GPT2-XL for CounterFact.

D.1.2 LLaMA2-7B

Figures 6, 7, 8, and 9 show attribution patterns for successful and failed edits under ROME, IKE, MEMIT, and fine-tuning on LLaMA2-7B for CounterFact.

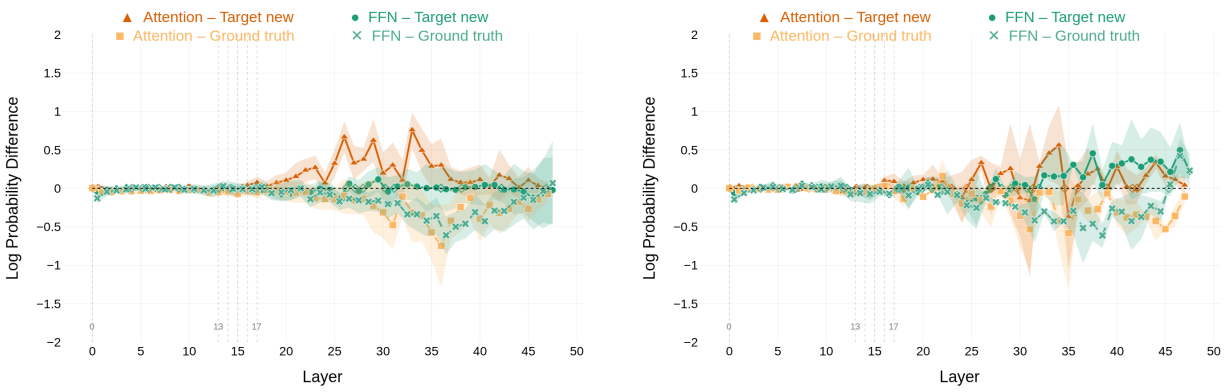
D.2 Popular

D.2.1 GPT2-XL

Figures 10, 11, 12, and 13 show attribution patterns for successful and failed edits on the Popular dataset for GPT2-XL.

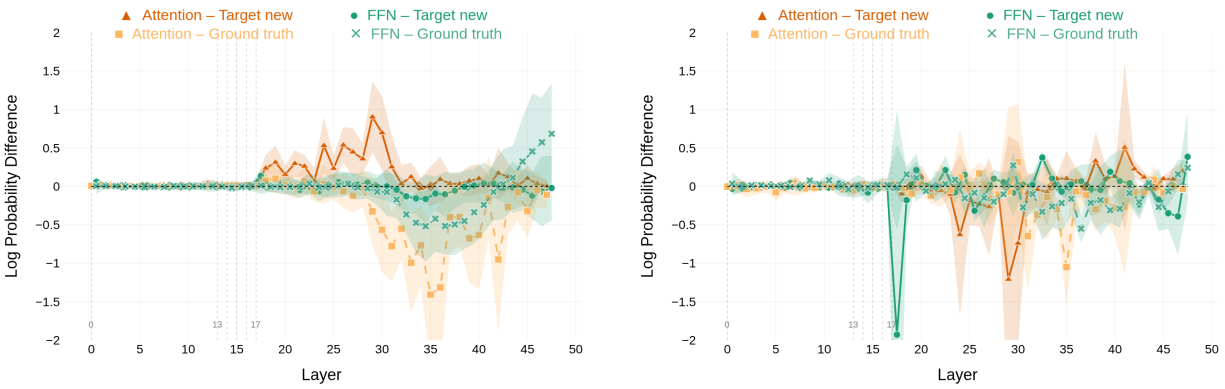
D.2.2 LLaMA2-7B

Figures 14, 15, 16, and 17 show attribution patterns for successful and failed edits on the Popular dataset for LLaMA2-7B.



(a) Success (ORI) (b) Failure (ORI)

Figure 3: **IKE success vs. failure on CounterFact (GPT2-XL).**



(a) Success (L17) (b) Failure (L17)

Figure 4: **ROME success vs. failure on CounterFact (GPT2-XL).**

E Additional Steering Results

E.1 CounterFact

E.1.1 GPT2-XL

Tables 3–7 report additional CounterFact results on GPT2-XL, including attention-window ablations, multi-layer component comparisons, and selected early-to-mid single-layer sweeps. Overall, the results support attribution-aligned mid-to-late attention-residual and residual interventions, while early-layer and FFN-only variants remain weak.

E.1.2 LLaMA2-7B

Table 8 reports additional CounterFact results on LLaMA2-7B for attribution-aligned mid-to-late window interventions.

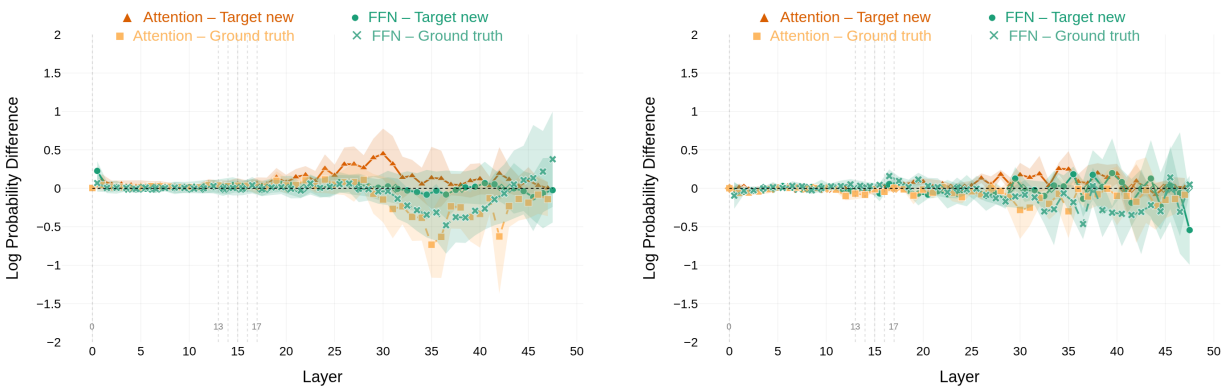
E.2 Popular

E.2.1 GPT2-XL

Table 9 reports additional Popular results on GPT2-XL for attribution-aligned mid-to-late configurations. Although residual-only variants are competitive in some Popular settings, we treat attention-residual steering as the primary instantiation of MEGA because it most directly reflects the mechanistic hypothesis suggested by attribution: target-promoting attention contributions are integrated through the residual stream.

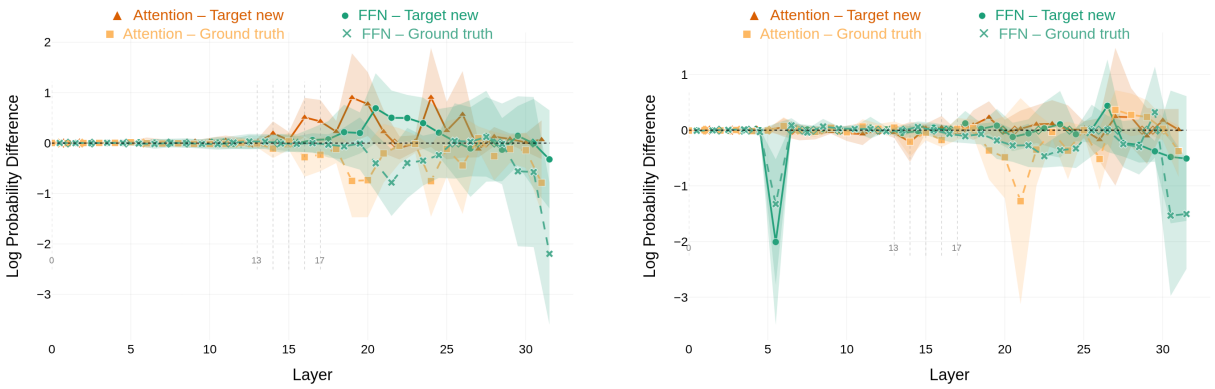
Table 3: Attention steering on CounterFact for GPT2-XL: moving a fixed-size window. Attribution-guided mid layers (15–40, 20–36) clearly outperform early or very-late windows.

Setting	Range	Acc	Gen	Spec	DI	DII	Avg
26-layer	0–25	0.16	0.08	0.99	1.00	0.47	0.54
26-layer	15–40	0.98	0.84	0.87	1.00	0.98	0.93
26-layer	22–47	0.88	0.71	0.87	0.99	0.96	0.88
17-layer	0–16	0.01	0.02	1.00	0.22	0.02	0.25
17-layer	20–36	0.86	0.73	0.88	0.98	0.93	0.88
17-layer	31–47	0.21	0.25	0.94	0.82	0.78	0.60



(a) FT Success (L0) (b) FT Failure (L0)

Figure 5: **Fine-tuning success vs. failure on CounterFact (GPT2-XL).**



(a) Success (L5) (b) Failure (L5)

Figure 6: **ROME success vs. failure on CounterFact (LLaMA2-7B).**

E.2.2 LLaMA2-7B

Table 10 reports additional Popular results on LLaMA2-7B for attribution-aligned mid-to-late layer configurations.

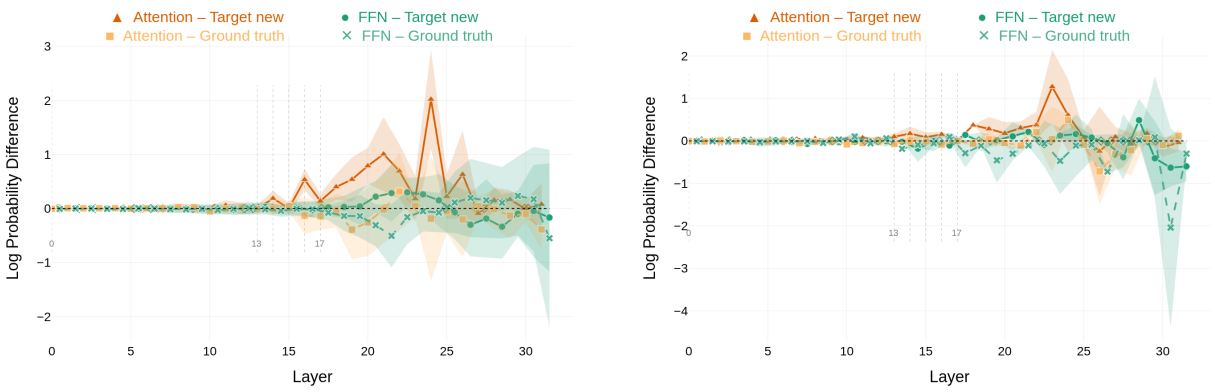
F Implementation Details

F.1 Knowledge Editing Baselines

All baselines are implemented using EasyEdit (v1.0) (Wang et al., 2024b) with default hyperparameters. We perform single independent edits (sequential_edit=False), resetting model weights between edits. ROME targets layer 17 (GPT2-XL) and layer 5 (LLaMA2-7B); MEMIT targets layers 13–17 (GPT2-XL) and layers 4–8 (LLaMA2-7B); FT fine-tunes the FFN at layer 0 (GPT2-XL) and layer 21 (LLaMA2-7B); IKE retrieves k=16 in-context demonstrations using `all-MiniLM-L6-v2` as the sentence encoder.

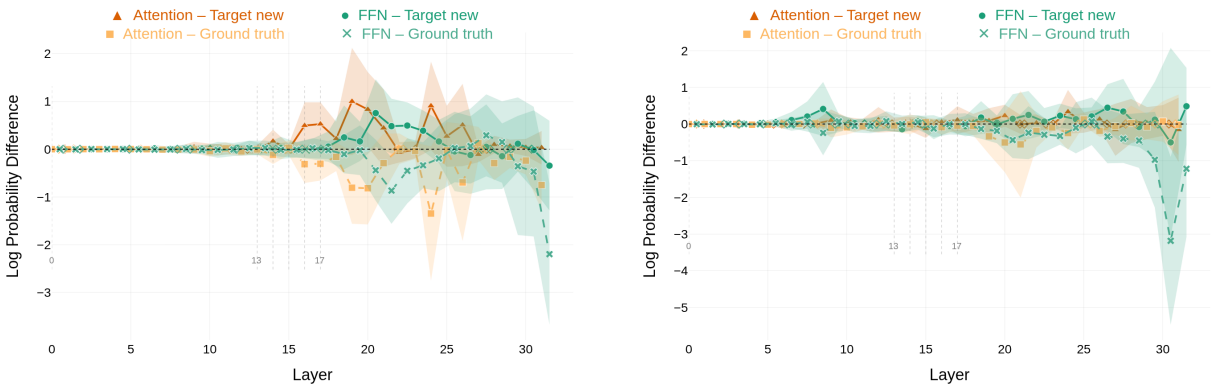
Success and failure classification. An edit is classified as *successful* if $\text{acc}_{\text{post}} > \text{acc}_{\text{pre}}$ and $\text{acc}_{\text{post}} \geq 0.9$, and as *failed* otherwise. Cases where the model already predicted the target correctly before editing are excluded as outliers. We retain around 50 successful and 50 failed cases per method for NLKA attribution analysis, restricting to cases where the unedited model correctly predicts the original ground-truth answer, ensuring a clean baseline for attribution.

NLKA Attribution. We apply the neuron-level knowledge attribution framework of Yu and Ananiadou (2024) to analyze post-edit model behavior. For each component $c \in \{\text{attn}, \text{FFN}\}$ at layer ℓ, the attribution score $\Delta C_{\ell,c}(y)$ measures the log-probability increase of token y when the component's output is added to the residual stream baseline at the last token position. We track two target tokens per case: `target_new` (the intended edit target) and `ground_truth` (the original answer), and average scores across 50 successful and 50 failed cases per method. For IKE, since no model param-



(a) Success (ORI) (b) Failure (ORI)

Figure 7: **IKE success vs. failure on CounterFact (LLaMA2-7B).**



(a) Success (L4–8) (b) Failure (L4–8)

Figure 8: **MEMIT success vs. failure on CounterFact (LLaMA2-7B).**

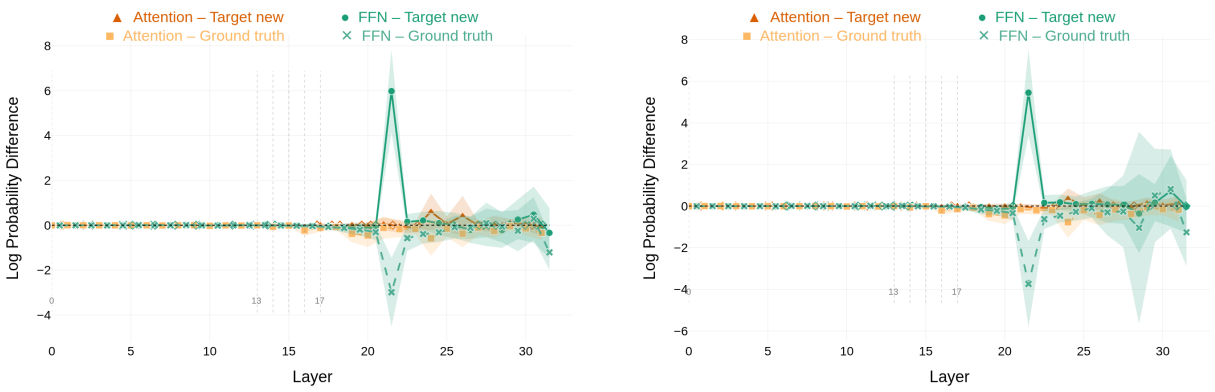
eters are modified, we instead apply attribution to the full in-context input: the retrieved demonstrations are prepended to the original prompt following the standard IKE format (i.e., `[ICL examples] + New Fact: [prompt + target] + Prompt: [prompt]`), and attribution scores are computed at the last token position of this extended sequence.

F.2 MEGA Steering Method

MEGA implements mechanism-guided activation steering based on attribution patterns identified by NLKA. For each edit, source and target activations are collected at the last token position from paraphrased prompts, projected into a 512-dimensional PCA subspace, and used to learn a linear optimal-transport map (PythonOT `LinearTransport`) between the two activation distributions. During inference, the current activation is transformed through this learned map and injected back into selected components via forward hooks, gated by a scope detector that determines whether the input prompt falls within the edit scope.

Multi-layer steering. Unlike single-layer steering approaches, MEGA supports interventions across layer ranges identified through attribution analysis. For a chosen layer set L, we hook the attention module at each layer $\ell \in L$ and modify its output so that the resulting attention-residual state aligns with the mapped target activation. For CounterFact, we steer layers 15–40 in GPT2-XL and 13–31 in LLaMA2-7B. For Popular, we steer layers 31–47 in GPT2-XL and 13–31 in LLaMA2-7B.

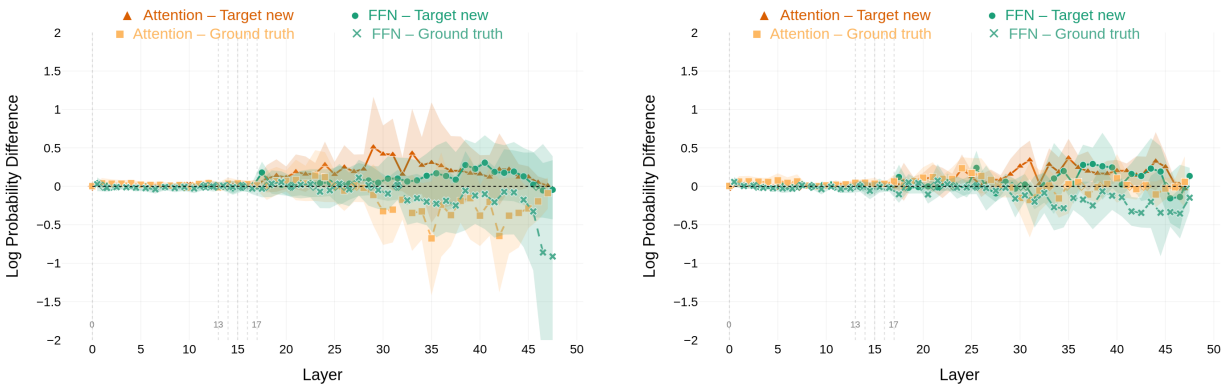
Component support. Steering can be applied to multiple internal components, including attention outputs, FFN outputs, and attention-residual representations. This allows interventions to target the internal pathways identified as most influential by the attribution analysis.



(a) FT Success (L21)

(b) FT Failure (L21)

Figure 9: **Fine-tuning success vs. failure on CounterFact (LLaMA2-7B).**



(a) Success (L17)

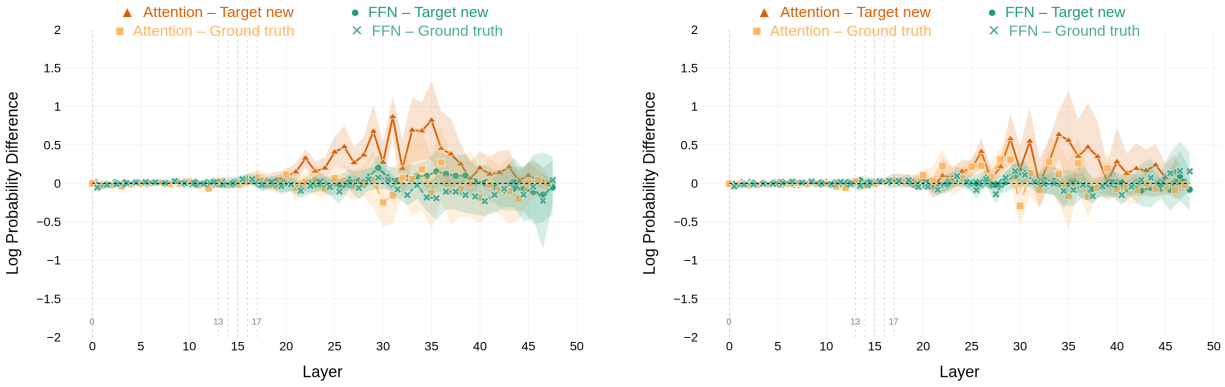
(b) Failure (L17)

Figure 10: **ROME success vs. failure on Popular (GPT2-XL).**

Dimensionality reduction. Before learning the transport map, activations are projected into a 512-dimensional PCA subspace. Operating in this reduced space improves numerical stability and produces more consistent steering directions compared to using the full hidden-state space.

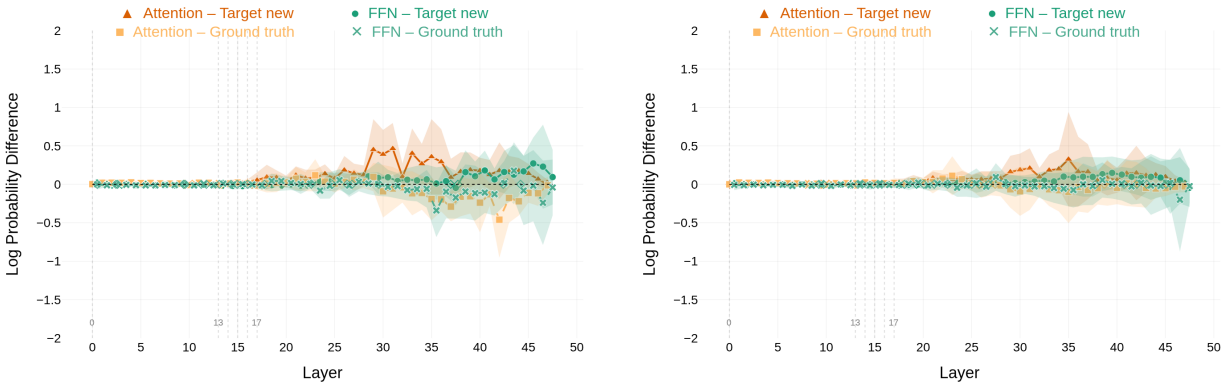
Hyperparameters. The Euclidean prompt-distance threshold is set to 6.5 for CounterFact and 5.0 for Popular for both GPT2-XL and LLaMA2-7B. The optimal transport regularization coefficient is set to $\lambda = 10^{-2}$ across both models. For Popular, we generate 100 paraphrases of the edit prompt together with 10 logical implication prompts, each augmented with four Gaussian perturbations ($\sigma = 0.1$). A fixed random seed (42) is used to ensure reproducibility.

Computational setup. GPT2-XL experiments are run on a single NVIDIA RTX A6000 (49GB VRAM). LLaMA2-7B experiments are run on a single NVIDIA A100-SXM4-80GB (80GB VRAM).



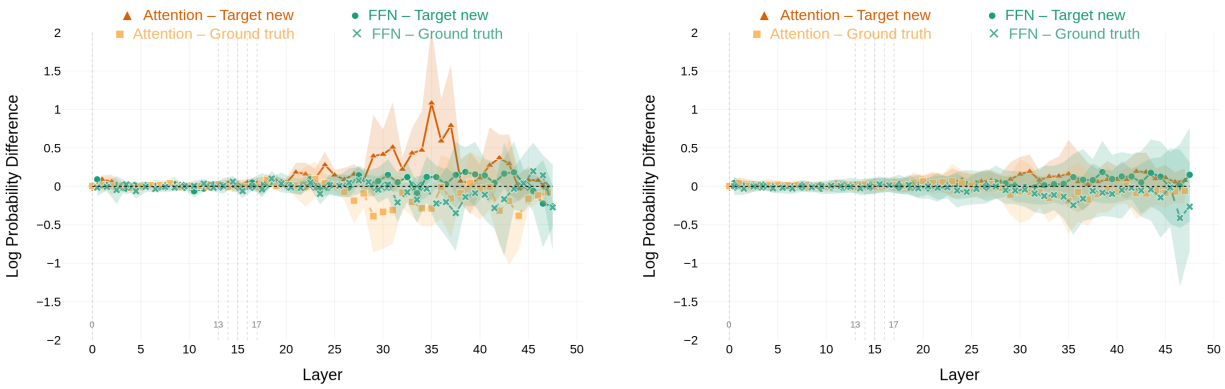
(a) Success (ORI) (b) Failure (ORI)

Figure 11: **IKE success vs. failure on Popular (GPT2-XL).**



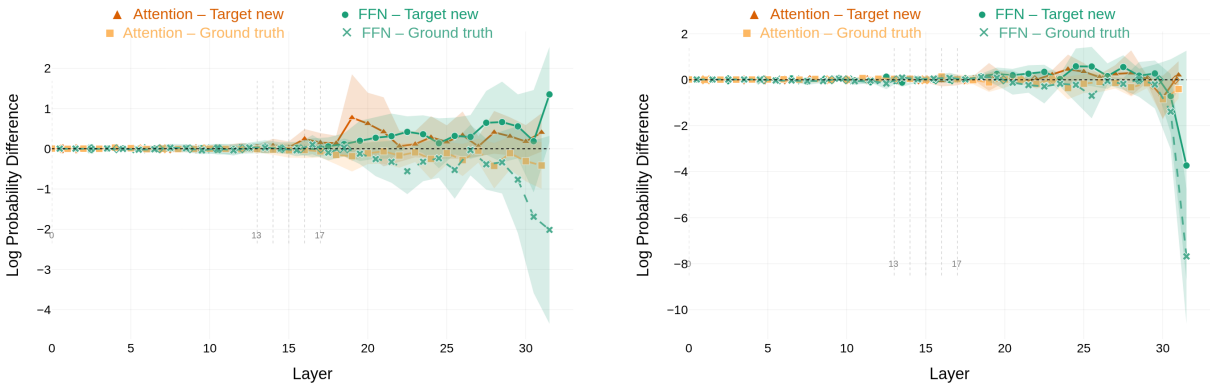
(a) Success (L13–17) (b) Failure (L13–17)

Figure 12: **MEMIT success vs. failure on Popular (GPT2-XL).**



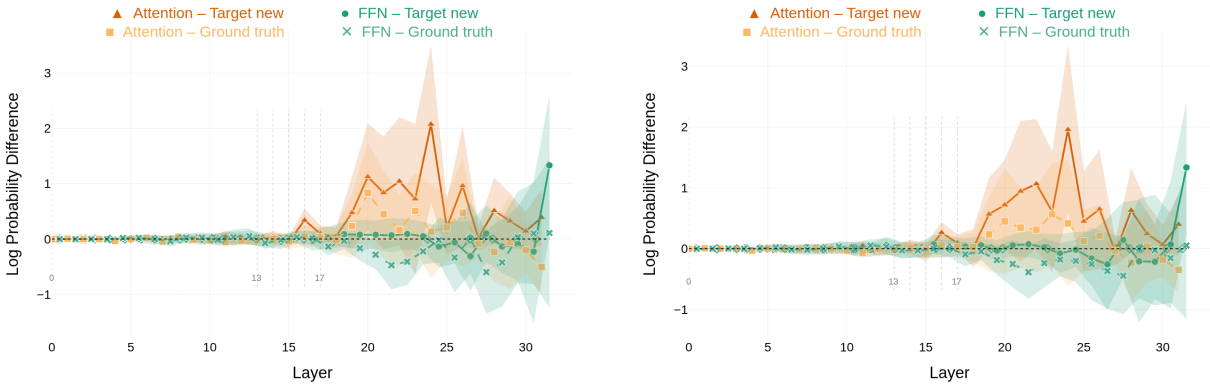
(a) FT Success (L0) (b) FT Failure (L0)

Figure 13: **Fine-tuning success vs. failure on Popular (GPT2-XL).**



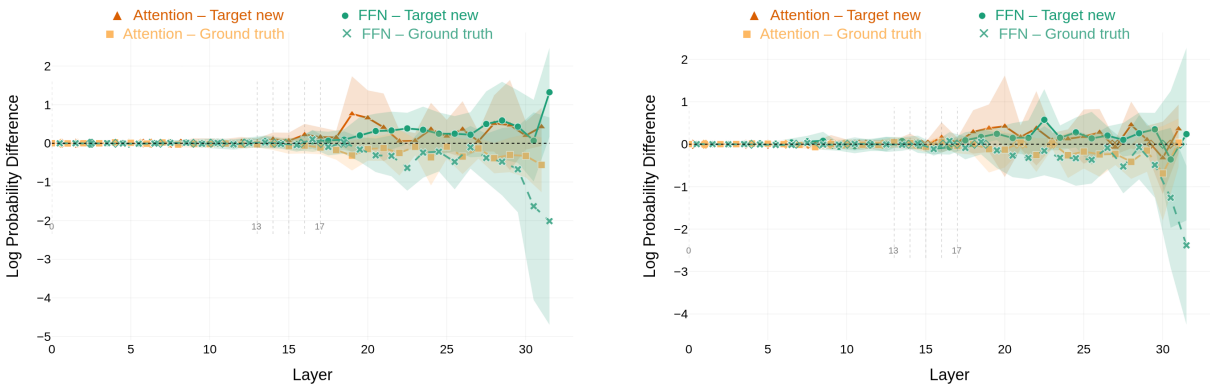
(a) Success (L5) (b) Failure (L5)

Figure 14: **ROME success vs. failure on Popular (LLaMA2-7B).**



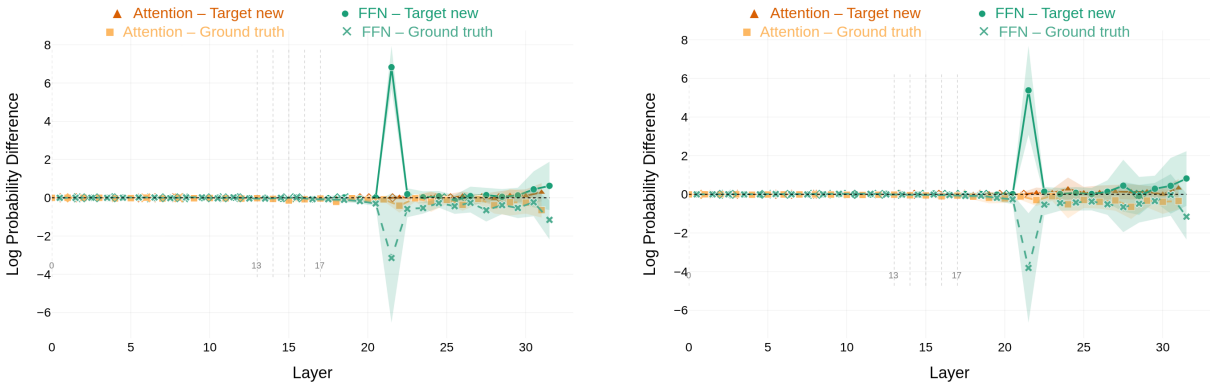
(a) Success (ORI) (b) Failure (ORI)

Figure 15: **IKE success vs. failure on Popular (LLaMA2-7B).**



(a) Success (L4–8) (b) Failure (L4–8)

Figure 16: **MEMIT success vs. failure on Popular (LLaMA2-7B).**



(a) FT Success (L21)

(b) FT Failure (L21)

Figure 17: **Fine-tuning success vs. failure on Popular (LLaMA2-7B).**

Table 4: Multi-layer steering on CounterFact (GPT2-XL), 26-layer windows.

Comp	PCA	Acc	Gen	Spec	DI	DII	Avg
Range 0–25							
ATTN	128	0.158	0.079	0.992	1.000	0.467	0.539
ATTN	512	0.183	0.075	0.987	0.925	0.533	0.541
ATTN-res	128	0.092	0.075	0.915	0.708	0.217	0.401
ATTN-res	512	0.217	0.233	0.897	0.858	0.783	0.598
FFN	128	0.008	0.004	0.998	0.008	0.017	0.207
FFN	512	0.008	0.008	0.999	0.067	0.008	0.218
Res	128	0.158	0.100	0.994	1.000	0.408	0.532
Res	512	0.183	0.079	0.992	0.975	0.483	0.543
Range 15–40							
ATTN	128	0.983	0.838	0.871	1.000	0.975	0.933
ATTN	512	0.983	0.792	0.872	0.942	0.900	0.898
ATTN-res	128	0.983	0.917	0.871	0.975	0.967	0.943
ATTN-res	512	0.992	0.925	0.871	1.000	1.000	0.958
FFN	128	0.033	0.029	0.992	0.158	0.208	0.284
FFN	512	0.033	0.025	0.994	0.217	0.175	0.289
Res	128	0.983	0.892	0.873	1.000	1.000	0.950
Res	512	0.983	0.892	0.873	0.983	1.000	0.946
Range 22–47							
ATTN	128	0.883	0.713	0.872	0.992	0.958	0.884
ATTN	512	0.892	0.704	0.883	0.925	0.908	0.862
ATTN-res	128	0.992	0.925	0.871	0.975	0.983	0.949
ATTN-res	512	0.992	0.925	0.871	1.000	0.992	0.956
FFN	128	0.033	0.029	0.995	0.183	0.067	0.262
FFN	512	0.042	0.025	0.994	0.183	0.100	0.269
Res	128	0.875	0.763	0.893	0.975	0.917	0.884
Res	512	0.967	0.817	0.880	1.000	0.992	0.931

Table 5: Multi-layer steering on CounterFact (GPT2-XL), 17-layer windows.

Comp	PCA	Acc	Gen	Spec	DI	DII	Avg
Range 0–16							
ATTN	128	0.008	0.017	1.000	0.217	0.017	0.252
ATTN	512	0.008	0.013	1.000	0.167	0.008	0.239
ATTN-res	128	0.025	0.008	0.969	0.533	0.017	0.310
ATTN-res	512	0.017	0.017	0.978	0.283	0.092	0.277
FFN	128	0.008	0.013	1.000	0.000	0.033	0.211
FFN	512	0.008	0.008	1.000	0.033	0.008	0.212
Res	128	0.008	0.013	1.000	0.308	0.017	0.269
Res	512	0.008	0.017	1.000	0.142	0.017	0.237
Range 20–36							
ATTN	128	0.858	0.733	0.875	0.983	0.925	0.875
ATTN	512	0.858	0.717	0.877	0.892	0.850	0.839
ATTN-res	128	0.983	0.896	0.871	0.950	0.958	0.932
ATTN-res	512	0.983	0.921	0.871	1.000	0.983	0.952
FFN	128	0.025	0.033	0.995	0.200	0.183	0.287
FFN	512	0.025	0.025	0.996	0.233	0.133	0.283
Res	128	0.933	0.875	0.887	1.000	0.992	0.937
Res	512	0.958	0.888	0.874	0.983	1.000	0.941
Range 31–47							
ATTN	128	0.208	0.254	0.935	0.817	0.775	0.598
ATTN	512	0.225	0.267	0.933	0.783	0.758	0.593
ATTN-res	128	1.000	0.925	0.871	0.983	0.992	0.954
ATTN-res	512	1.000	0.925	0.871	1.000	0.992	0.958
FFN	128	0.025	0.025	0.996	0.058	0.033	0.228
FFN	512	0.017	0.021	0.995	0.067	0.033	0.227
Res	128	0.883	0.758	0.894	0.983	0.908	0.885
Res	512	0.967	0.817	0.880	1.000	0.992	0.931

Table 6: Single-layer steering on CounterFact (GPT2-XL), layers 0–15.

Comp	PCA	Acc	Gen	Spec	DI	DII	Avg
Layer 0							
ATTN	1600	0.008	0.008	1.000	0.067	0.000	0.217
ATTN	128	0.008	0.008	1.000	0.067	0.000	0.217
ATTN	512	0.008	0.008	1.000	0.067	0.000	0.217
ATTN-res	1600	0.008	0.008	1.000	0.067	0.000	0.217
ATTN-res	128	0.008	0.008	0.998	0.000	0.000	0.203
ATTN-res	512	0.008	0.008	1.000	0.000	0.000	0.203
FFN	1600	0.008	0.008	1.000	0.067	0.000	0.217
FFN	128	0.008	0.008	1.000	0.000	0.000	0.203
FFN	512	0.008	0.008	1.000	0.000	0.000	0.203
Res	1600	0.008	0.008	1.000	0.067	0.000	0.217
Res	128	0.008	0.008	1.000	0.000	0.000	0.203
Res	512	0.008	0.008	1.000	0.000	0.000	0.203
Layer 5							
ATTN	1600	0.008	0.008	1.000	0.067	0.000	0.217
ATTN	128	0.008	0.008	1.000	0.067	0.000	0.217
ATTN	512	0.008	0.008	1.000	0.067	0.000	0.217
ATTN-res	1600	0.008	0.013	0.999	0.058	0.000	0.216
ATTN-res	128	0.008	0.004	0.998	0.000	0.000	0.202
ATTN-res	512	0.008	0.008	0.999	0.000	0.000	0.203
FFN	1600	0.008	0.008	1.000	0.067	0.000	0.217
FFN	128	0.008	0.008	1.000	0.075	0.000	0.218
FFN	512	0.008	0.008	1.000	0.075	0.000	0.218
Res	1600	0.008	0.013	0.999	0.067	0.000	0.217
Res	128	0.008	0.004	0.999	0.000	0.000	0.202
Res	512	0.008	0.008	0.999	0.000	0.000	0.203
Layer 10							
ATTN	1600	0.008	0.008	1.000	0.075	0.000	0.218
ATTN	128	0.008	0.008	1.000	0.067	0.000	0.217
ATTN	512	0.008	0.008	1.000	0.067	0.000	0.217
ATTN-res	1600	0.008	0.013	1.000	0.075	0.000	0.219
ATTN-res	128	0.008	0.017	1.000	0.067	0.017	0.222
ATTN-res	512	0.008	0.013	1.000	0.133	0.008	0.233
FFN	1600	0.008	0.008	1.000	0.067	0.000	0.217
FFN	128	0.008	0.008	1.000	0.075	0.000	0.218
FFN	512	0.008	0.008	1.000	0.075	0.000	0.218
Res	1600	0.008	0.013	1.000	0.067	0.000	0.218
Res	128	0.008	0.017	1.000	0.067	0.025	0.223
Res	512	0.008	0.013	1.000	0.117	0.017	0.231
Layer 15							
ATTN	1600	0.008	0.008	1.000	0.067	0.000	0.217
ATTN	128	0.008	0.008	1.000	0.075	0.000	0.218
ATTN	512	0.008	0.008	1.000	0.067	0.000	0.217
ATTN-res	1600	0.008	0.013	1.000	0.092	0.000	0.223
ATTN-res	128	0.008	0.017	1.000	0.150	0.017	0.238
ATTN-res	512	0.008	0.017	1.000	0.167	0.033	0.245
FFN	1600	0.008	0.008	1.000	0.067	0.000	0.217
FFN	128	0.008	0.008	1.000	0.067	0.000	0.217
FFN	512	0.008	0.008	1.000	0.067	0.000	0.217
Res	1600	0.008	0.013	1.000	0.083	0.000	0.221
Res	128	0.008	0.017	1.000	0.267	0.017	0.262
Res	512	0.008	0.017	1.000	0.183	0.033	0.248

Table 7: Single-layer steering on CounterFact (GPT2-XL), layers 20–30.

Comp	PCA	Acc	Gen	Spec	DI	DII	Avg
Layer 20							
ATTN	1600	0.008	0.013	1.000	0.125	0.000	0.229
ATTN	128	0.008	0.013	1.000	0.142	0.000	0.233
ATTN	512	0.008	0.013	1.000	0.133	0.000	0.231
ATTN-res	1600	0.025	0.029	1.000	0.467	0.067	0.318
ATTN-res	128	0.025	0.038	0.999	0.825	0.042	0.386
ATTN-res	512	0.025	0.033	0.999	0.725	0.108	0.378
FFN	1600	0.008	0.008	1.000	0.050	0.000	0.213
FFN	128	0.008	0.008	1.000	0.025	0.000	0.208
FFN	512	0.008	0.008	1.000	0.050	0.000	0.213
Res	1600	0.025	0.029	1.000	0.450	0.075	0.316
Res	128	0.025	0.038	0.999	0.842	0.033	0.387
Res	512	0.025	0.038	1.000	0.650	0.117	0.366
Layer 25							
ATTN	1600	0.008	0.008	1.000	0.150	0.008	0.235
ATTN	128	0.008	0.008	1.000	0.192	0.025	0.247
ATTN	512	0.008	0.008	1.000	0.092	0.017	0.225
ATTN-res	1600	0.192	0.088	0.991	0.892	0.467	0.526
ATTN-res	128	0.167	0.104	0.994	1.000	0.425	0.538
ATTN-res	512	0.175	0.079	0.992	0.975	0.517	0.548
FFN	1600	0.008	0.008	1.000	0.067	0.000	0.217
FFN	128	0.008	0.008	1.000	0.067	0.000	0.217
FFN	512	0.008	0.008	1.000	0.067	0.000	0.217
Res	1600	0.183	0.088	0.990	0.825	0.425	0.502
Res	128	0.158	0.096	0.994	1.000	0.400	0.530
Res	512	0.167	0.083	0.992	0.967	0.492	0.540
Layer 30							
ATTN	1600	0.008	0.013	0.996	0.167	0.008	0.238
ATTN	128	0.008	0.008	0.998	0.125	0.008	0.230
ATTN	512	0.008	0.008	0.998	0.125	0.008	0.230
ATTN-res	1600	0.625	0.500	0.936	0.942	0.842	0.769
ATTN-res	128	0.517	0.529	0.956	1.000	0.925	0.785
ATTN-res	512	0.633	0.563	0.937	0.983	0.917	0.807
FFN	1600	0.008	0.008	1.000	0.083	0.008	0.222
FFN	128	0.008	0.008	1.000	0.100	0.017	0.227
FFN	512	0.008	0.008	1.000	0.092	0.017	0.225
Res	1600	0.633	0.488	0.936	0.950	0.833	0.768
Res	128	0.500	0.504	0.955	1.000	0.925	0.777
Res	512	0.608	0.550	0.935	0.975	0.892	0.792

Table 8: Multi-layer steering on CounterFact (LLaMA2-7B), layer windows.

Comp	PCA	Acc	Gen	Spec	DI	DII	Avg
Range 13–31							
ATTN	512	0.942	0.825	0.888	0.917	0.950	0.904
ATTN-res	512	0.992	0.913	0.888	1.000	0.967	0.952
FFN	512	0.000	0.008	0.888	0.250	0.433	0.316
Res	512	0.908	0.779	0.888	0.767	0.875	0.844
Range 15–27							
ATTN	512	0.875	0.717	0.888	0.833	0.950	0.853
ATTN-res	512	0.992	0.913	0.888	0.975	0.883	0.930
FFN	512	0.008	0.029	0.888	0.483	0.608	0.404
Res	512	0.967	0.904	0.888	1.000	1.000	0.952

Table 9: Multi-layer steering on Popular (GPT2-XL), layer windows.

Comp	PCA	Acc	CI	CII	RS	SA	Avg
Range 15–40							
ATTN	128	0.667	0.275	0.400	0.382	0.703	0.485
ATTN	512	0.392	0.250	0.400	0.441	0.658	0.428
ATTN-res	128	0.642	0.225	0.400	0.324	0.679	0.454
ATTN-res	512	0.817	0.200	0.450	0.324	0.648	0.488
FFN	128	0.050	0.410	0.000	0.660	1.000	0.424
FFN	512	0.017	0.375	0.000	0.714	1.000	0.421
Res	128	0.817	0.450	0.250	0.412	0.682	0.522
Res	512	0.717	0.650	0.300	0.456	0.689	0.562
Range 15–47							
ATTN	128	0.733	0.300	0.400	0.338	0.696	0.494
ATTN	512	0.525	0.250	0.450	0.353	0.655	0.447
ATTN-res	128	0.683	0.200	0.450	0.324	0.670	0.465
ATTN-res	512	0.817	0.250	0.450	0.324	0.640	0.496
FFN	128	0.058	0.235	0.000	0.588	1.000	0.376
FFN	512	0.017	0.276	0.000	0.681	1.000	0.395
Res	128	0.433	0.462	0.200	0.569	0.765	0.486
Res	512	0.533	0.600	0.250	0.515	0.675	0.515
Range 20–45							
ATTN	128	0.658	0.275	0.350	0.426	0.709	0.484
ATTN	512	0.458	0.275	0.300	0.426	0.662	0.424
ATTN-res	128	0.692	0.225	0.450	0.324	0.667	0.471
ATTN-res	512	0.817	0.225	0.450	0.324	0.649	0.493
FFN	128	0.067	0.278	0.000	0.589	0.000	0.187
FFN	512	0.000	0.300	0.000	0.681	0.000	0.196
Res	128	0.675	0.385	0.150	0.375	0.745	0.466
Res	512	0.658	0.475	0.300	0.485	0.671	0.518
Range 31–47							
ATTN	128	0.375	0.575	0.150	0.683	0.738	0.504
ATTN	512	0.308	0.450	0.200	0.657	0.660	0.455
ATTN-res	128	0.725	0.225	0.450	0.324	0.674	0.480
ATTN-res	512	0.800	0.275	0.450	0.324	0.638	0.497
FFN	128	0.000	0.417	0.000	0.698	0.938	0.410
FFN	512	0.017	0.296	0.000	0.696	1.000	0.402
Res	128	0.433	0.462	0.200	0.569	0.765	0.486
Res	512	0.533	0.600	0.250	0.515	0.675	0.515

Table 10: Multi-layer steering on Popular (LLaMA2-7B), layer windows.

Comp	PCA	Acc	CI	CII	RS	SA	Avg
Range 13–31							
ATTN	512	0.517	0.082	0.284	0.354	0.637	0.375
ATTN-res	512	0.817	0.074	0.247	0.329	0.637	0.421
Res	512	0.692	0.081	0.148	0.399	0.651	0.394
Range 15–31							
ATTN	512	0.500	0.082	0.284	0.367	0.630	0.372
ATTN-res	512	0.808	0.074	0.247	0.329	0.637	0.419
Res	512	0.692	0.081	0.148	0.399	0.651	0.394