Revealing the Deceptiveness of Knowledge Editing: A Mechanistic Analysis of Superficial Editing

Jiakuan Xie[1,2], Pengfei Cao[1,2,*], Yubo Chen[1,2], Kang Liu[1,2], Jun Zhao[1,2,*]

[1]School of Artificial Intelligence, University of Chinese Academy of Sciences
[2]The Key Laboratory of Cognition and Decision Intelligence for Complex Systems, Institute of Automation, Chinese Academy of Sciences

xiejiakuan2023@ia.ac.cn, {pengfei.cao, yubo.chen, kliu, jzhao}@nlpr.ia.ac.cn

Abstract

Knowledge editing, which aims to update the knowledge encoded in language models, can be deceptive. Despite the fact that many existing knowledge editing algorithms achieve near-perfect performance on conventional metrics, the models edited by them are still prone to generating original knowledge. This paper introduces the concept of "**superficial editing**" to describe this phenomenon. Our comprehensive evaluation reveals that this issue presents a significant challenge to existing algorithms. Through systematic investigation, we identify and validate two key factors contributing to this issue: (1) the residual stream at the last subject position in earlier layers and (2) specific attention modules in later layers. Notably, certain attention heads in later layers, along with specific left singular vectors in their output matrices, encapsulate the original knowledge and exhibit a causal relationship with superficial editing. Furthermore, we extend our analysis to the task of superficial unlearning, where we observe consistent patterns in the behavior of specific attention heads and their corresponding left singular vectors, thereby demonstrating the robustness and broader applicability of our methodology and conclusions. Our code is available at https://github.com/jiakuan929/superficial-editing.

1 Introduction

The inherent static nature of knowledge embedded within a pretrained large language model (LLM) poses a fundamental limitation as the real world evolves. To address this issue, the concept of knowledge editing has been proposed to modify specific knowledge in LLMs while ensuring that unrelated knowledge remains unaffected (Zhu et al., 2020). To date, numerous studies have been conducted on knowledge editing, encompassing diverse methodologies (Zhu et al., 2020; De Cao et al., 2021; Mitchell et al., 2022a; Mitchell et al., 2022b; Meng et al., 2022; Meng et al., 2023; Zheng et al., 2023), paradigms (Hartvigsen et al., 2023; Fang et al., 2024; Jiang et al., 2024; Cai and Cao, 2024; Xu et al., 2023; Wang et al., 2024c; Wang et al., 2024a; Wu et al., 2024), evaluation strategies (Zhong et al., 2023; Cohen et al., 2024; Rosati et al., 2024; Yang et al., 2024; Ma et al., 2024), and applications (Wang et al., 2024d; Uppaal et al., 2024; Chen et al., 2024a). Although significant progress has been made in these endeavors, a critical challenge persists: *models that appear to have been successfully edited may unexpectedly revert to their original knowledge when exposed to specific contextual inputs*. As shown in Figure 1, the edited model demonstrates the capability to appropriately respond to the query "The President of the United States is". However, when the context "Is Joe Biden the President of the U.S.?" is incorporated into the query, the updated model reverts to generating responses based on its original knowledge. The phenomenon reveals the potential deceptiveness of knowledge editing: edited models may revert to their original knowledge, undermining the goal of enabling continuous knowledge updates in

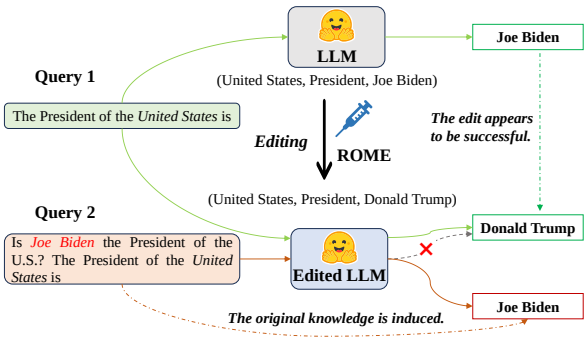


Figure 1: An example of superficial editing with the LLaMA3-8B-Instruct model. Following the editing process, the model accurately responds to Query 1. However, when presented with Query 2 as input, the edited model reverts to generating the original answer.

*Corresponding authors.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 17756–17780
July 27 - August 1, 2025

LLMs. This limitation severely hinders the practical utility and reliability of knowledge editing.

In this paper, we define a knowledge editing process as "**superficial editing**" when the resulting model appears to successfully integrate new knowledge, yet reverts to its original knowledge when exposed to carefully crafted prompts. To quantitatively evaluate this issue, we introduce "attack probe", which is a specifically designed prompt consisting of an attack prefix and a baseline prompt (as exemplified by Query 2 in Figure 1). We develop three attack types based on two widely used datasets and assess several editing algorithms across three models. Empirical results demonstrate that while the majority of editing algorithms exhibit strong performance on conventional evaluation metrics, models edited through these approaches remain vulnerable to attack probes. For instance, both PMET (Li et al., 2024) and AlphaEdit (Fang et al., 2024) demonstrate near-optimal performance in terms of editing efficacy; however, they exhibit superficial editing in over 70% of the cases. This finding suggests that current parameter-editing algorithms are fundamentally inadequate in addressing the challenge of superficial editing.

To elucidate the underlying mechanisms of this phenomenon, we focus on the core components of the Transformer architecture (Vaswani et al., 2017). We initially conduct intervention experiments on the residual stream at two token positions. First, our intervention at the last subject position in earlier layers reveals shifts in prediction probabilities. Second, we intervene at the last position and find that this intervention exerts a significant effect in the later layers, where the probability of the original answer exceeds that of the new answer. This shift is a prerequisite for superficial editing, a phenomenon we term the "Reversal of the Residual Stream" (RRS). Additionally, our preliminary analysis of Multi-Layer Perceptron (MLP) and Multi-Head Attention reveals that specific attention modules in later layers play a significant role. Based on the observations, we formulate two hypotheses: (**H1**) **The enrichment of new knowledge at the last subject position in earlier layers is impeded, and the accumulation of the original knowledge at this position is relatively limited.** (**H2**) **The later attention modules actively incorporate information related to original knowledge into the last position, thereby facilitating the RRS phenomenon and consequently inducing the occurrence of superficial editing.** To validate **H1**, we project the representation of the last subject position in each layer into the vocabulary space. We observe greater suppression of the new answer when the attack probe is used as input, compared to the baseline prompt. However, despite this suppression effect, the ranking of the original answer consistently lags behind that of the new answer in the earlier layers, indicating minimal enrichment of the original knowledge. To validate **H2**, we first establish that the later attention modules exhibit a causal relationship with the RRS phenomenon, highlighting their critical role. Subsequently, we analyze the attention heads and confirm that a causal correlation exists between certain heads in later layers and superficial editing. Furthermore, we investigate the internal mechanisms of attention heads through singular value decomposition (SVD) and demonstrate that specific left singular vectors are responsible for encoding the original knowledge and contributing to superficial editing. These findings provide robust evidence in support of **H2**.

To demonstrate the broader applicability of our interpretability analysis framework, we extend our investigation to a distinct task: **superficial unlearning**, wherein the unlearned model fails to truly forget the target information. Our experimental results reveal a strong correlation between this phenomenon and specific attention heads along with their corresponding singular vectors, substantiating the generalizability of both our analytical methodology and conclusions.

The primary contributions of this paper are as follows: **(1)** We formally define superficial editing and provide corresponding evaluation datasets and metrics, thereby completing the assessment of multiple algorithms. **(2)** We identify and validate two critical factors contributing to superficial editing: the residual stream in earlier layers and specific attention modules in later layers. Additionally, we explore the internal mechanisms of the attention module and reveal that specific attention heads and their corresponding left singular vectors are responsible for superficial editing. **(3)** We apply our analytical approach to superficial unlearning. The consistent finding across both phenomena validates the robustness and broader applicability of both our methodology and conclusions.

2 Problem Formulation

Knowledge editing, which aims to adjust the knowledge of a language model, can generally be ex-

Methods	Wiki					Rep					Que				
	Eff.	Gen.	Loc.	OM ↓	OP ↓	Eff.	Gen.	Loc.	OM ↓	OP ↓	Eff.	Gen.	Loc.	OM ↓	OP ↓
FT	100	80.51	52.37	49.45	51.65	100	70.54	44.46	30.68	35.98	100	87.90	33.87	29.07	31.40
MEND	98.31	65.25	47.63	35.16	39.56	100	50.99	51.29	34.47	38.36	100	81.45	39.52	33.73	38.37
ROME	100	94.92	85.08	54.95	58.24	100	97.52	84.75	61.74	64.02	100	99.19	82.74	38.37	38.37
MEMIT	100	94.07	86.10	52.75	54.95	100	98.27	87.18	40.15	42.42	100	100	82.58	37.21	37.21
PMET	94.92	85.59	90.00	70.33	72.43	99.50	93.32	91.88	66.67	71.97	96.67	89.17	88.17	39.29	41.67
r-ROME	96.61	92.37	86.78	54.95	57.14	99.01	97.28	89.11	64.39	68.18	98.33	97.50	84.50	40.48	40.48
AlphaEdit	100	83.90	88.98	72.53	73.62	100	92.33	92.23	68.18	71.97	100	88.33	87.67	34.52	35.71

Table 1: Evaluation results of superficial editing conducted on LLaMA3-8B-Instruct using the CF-a dataset. **Wiki**, **Rep**, and **Que** represent the three attack types defined in Section 2. Experimental results for other models and datasets are available in Appendix B.

pressed as follows:

$$(s, r, o) \xrightarrow{e} (s, r, o^*), \tag{1}$$

where s is subject (e.g., United States), r is relation (e.g., President), o is the pre-editing object (e.g., Joe Biden), o^* is the post-editing object (e.g., Donald Trump), and e is a prompt used for editing (e.g., "The President of the United States is"). We define the following attack prefixes for o:

$$a \in \mathcal{A} = \{\text{Wiki}(o), \text{Rep}(o), \text{Que}(o)\}, \tag{2}$$

where a is an attack prefix, $\text{Wiki}(o)$ denotes the Wikipedia summary of o, $\text{Rep}(o)$ denotes the repetition of o, and $\text{Que}(o)$ represents a question incorporating s, r, and o simultaneously (e.g., Is Joe Biden the President of the U.S.?). The set of all queries derivable from s and r is denoted as $\mathcal{I} = \{x \mid s, r \Rightarrow x\}$. According to the editing operation defined in Equation 1, the edit is classified as **superficial editing** if the edited model f' satisfies the following conditions:

$$\begin{cases} f'(x) = o^* & x \in \mathcal{I} \\ f'(a \oplus x) = o & a \in \mathcal{A} \quad x \in \mathcal{I}, \end{cases} \tag{3}$$

where $\oplus$ denotes text concatenation. To quantify the extent of superficial editing, we define the following metrics:

$$\begin{aligned} \text{OM} &= \mathbb{E}_x \left[f'(a \oplus x) = o \right] \\ \text{OP} &= \mathbb{E}_x \left[P(o \mid a \oplus x) > P(o^* \mid a \oplus x) \right], \end{aligned} \tag{4}$$

where OM indicates whether the model's prediction matches the original answer o, and OP measures whether the output probability of o exceeds that of o^*. Higher values of OM and OP reflect a greater degree of superficial editing.

3 Evaluation of Superficial Editing

This section evaluates multiple representative parameter-editing algorithms for superficial editing. We first describe the evaluation setup of our experiment (§3.1), followed by a comprehensive assessment of various methods (§3.2).

3.1 Evaluation Setup

Data Collection. To construct our evaluation dataset for superficial editing, we employ two widely used datasets in knowledge editing: CounterFact (Meng et al., 2022) and ZsRE (Zhu et al., 2020). First, we select cases where the model has already acquired the corresponding knowledge. Next, based on the definition, we generate three attack prefixes and concatenate them with the baseline prompts from the original dataset to construct attack probes. Finally, we evaluate all instances, filtering the cases that meet the definition to create two enhanced datasets, designated as **CF-a** and **ZsRE-a**, respectively. The detailed construction procedure is provided in Appendix A.

Baselines. We employ the following knowledge editing methods as baselines: FT (Zhu et al., 2020), MEND (Mitchell et al., 2022a), ROME (Meng et al., 2022), MEMIT (Meng et al., 2023), PMET (Li et al., 2024), r-ROME (Gupta et al., 2024), and AlphaEdit (Fang et al., 2024).

Models & Metrics. We conduct experiments using three powerful language models: LLaMA3-8B-Instruct[1], Qwen2.5-7B-Instruct, and Qwen2.5-14B-Instruct[2]. In addition to the metrics specifically defined for superficial editing in Equation (4), we also report three conventional knowledge editing metrics: Efficacy (Eff.), Generalization (Gen.), and Locality (Loc.), respectively. The formal definitions for them are detailed in Appendix B.

[1] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

[2] https://qwenlm.github.io/blog/qwen2.5-llm/

3.2 Evaluation Results

Table 1 presents our evaluation results. Notably, while the models edited using various methods demonstrate near-perfect performance on conventional metrics, particularly Efficacy, they exhibit significant vulnerability to attack probes. For instance, under the Wiki attack scenario, both PMET and AlphaEdit achieve superior Efficacy scores, yet simultaneously maintain high OM metrics of 70.33% and 72.53%, respectively. This underscores the severity of superficial editing. The results also highlight the limitations of conventional evaluation frameworks, which inadequately capture the practical effectiveness of knowledge editing algorithms. The experimental findings motivate our subsequent investigation into the underlying mechanisms of superficial editing.

4 Mechanistic Analysis of Superficial Editing

This section presents a comprehensive investigation into the underlying mechanisms responsible for superficial editing. We initiate our analysis by examining the influence of the three fundamental components within the Transformer architecture: Residual Stream (He et al., 2016; Elhage et al., 2021), Multi-Layer Perceptron (MLP), and Multi-Head Attention. Building upon the observations, we formulate two key hypotheses (§4.1). Subsequently, we conduct rigorous empirical validation of these hypotheses, systematically elucidating the causal factors underlying superficial editing (§4.2 and §4.3). Furthermore, we extend our analysis to investigate the related task of superficial unlearning, thereby demonstrating the generalizability of our approach and conclusions (§4.4).

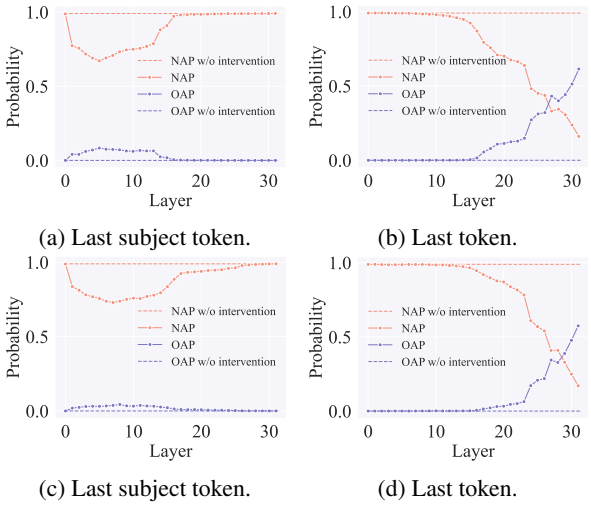
4.1 Effects of Transformer Components

4.1.1 Effect of the Residual Stream

To investigate the influence of the Residual Stream, we implement two distinct forward propagation procedures: a "clean run" using the baseline prompt e as input and a "corrupted run" using the attack probe $a \oplus e$ as input. Through these two forward passes, we can obtain the outputs of each layer in the model:

$$\boldsymbol{H} = \{\boldsymbol{h}_i^{(l)} \mid i \in [0, T), l \in [0, L)\} \quad (5)$$

$$\hat{\boldsymbol{H}} = \{\hat{\boldsymbol{h}}_i^{(l)} \mid i \in [0, \hat{T}), l \in [0, L)\}, \quad (6)$$



(a) Last subject token. (b) Last token.

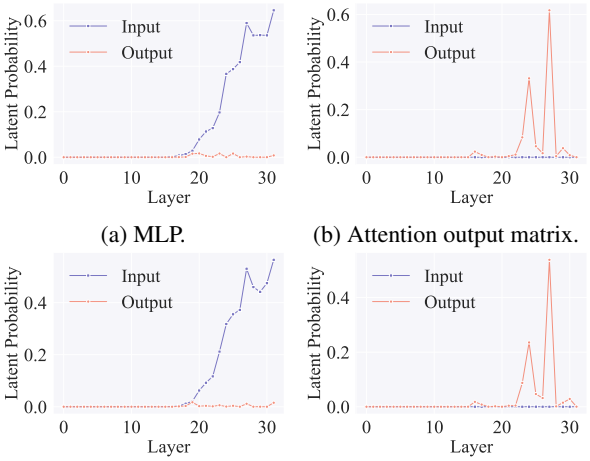
(c) Last subject token. (d) Last token.

Figure 2: Intervention results of LLaMA3-8B-Instruct edited by ROME (2a, 2b) and MEMIT (2c, 2d) at different tokens. The final probabilities without any intervention are depicted by dashed lines in the respective colors. Results for other models are provided in Appendix C.1.

where $\boldsymbol{H}$ and $\hat{\boldsymbol{H}}$ denote the hidden states of the clean and corrupted runs, respectively. T and $\hat{T}$ represent the sequence lengths of the two inputs, and L is the number of layers. Subsequently, we introduce an intervention within the residual stream of the clean run. More precisely, we replace the representation of a specific token at layer l with its corresponding representation from the corrupted run at the same layer:

$$\boldsymbol{h}_{t_0}^{(l)} \leftarrow \hat{\boldsymbol{h}}_{t_1}^{(l)}, \quad (7)$$

where t_0 and t_1 denote the indices of the same token in e and $a \oplus e$, respectively. In this study, we concentrate on two distinct positions: (1) the last position of the subject, which has been identified as crucial for a specific process (Geva et al., 2023); (2) the last position of the sentence, which serves as the primary basis for the model to predict the next token. Following intervention at each layer, we can determine the original answer probability (OAP) and the new answer probability (NAP) of the edited model. To establish a baseline for comparison, we additionally compute the mean OAP and NAP of the clean run without any interventions. The results are illustrated in Figure 2. As shown in the figure, the residual streams at these two positions exert a causal effect on the model's predictions. The residual stream at the last subject position predominantly influences the earlier layers (Figures 2a, 2c), whereas the residual stream at the last position primarily affects the later layers (Figures 2b, 2d). The



(a) MLP. (b) Attention output matrix.

(c) MLP. (d) Attention output matrix.

Figure 3: Latent probabilities of the original answer for the input and output of the MLP and Attention output matrix in LLaMA3-8B-Instruct edited by ROME (3a, 3b) and MEMIT (3c, 3d). Results for other models are presented in Appendix C.1.

latter's impact is more pronounced, as the OAP exceeds the NAP, which is a critical prerequisite for superficial editing. We formally designate this observed pattern in the later layers as the "Reversal of the Residual Stream" (RRS) phenomenon.

4.1.2 Effect of MLP and Attention

The impact of both the MLP and Multi-Head Attention on model predictions arises from their iterative refinement of the vector at the last position, thereby enhancing its predictive capacity for generation. To investigate their effects, we extract both the input and output vectors at the last position from the MLP and the attention output matrix $\boldsymbol{W}_o$. These vectors are projected into the vocabulary space using the "logit lens" technique (nostalgebraist, 2020; Geva et al., 2022; Dar et al., 2023; Halawi et al., 2024), enabling us to observe the probability of the original answer o within each latent probability distribution. A detailed explanation of this technique is provided in Appendix D.

The findings are illustrated in Figure 3. Our analysis demonstrates that the probability of o within the latent probability distribution of each MLP layer's output is consistently lower than that of its input. In contrast, certain attention modules exhibit an inverse pattern in later layers, where the probability of o in the output distribution is significantly higher than that of the input. This observation suggests that the RRS phenomenon is likely driven by attention modules in later layers.

4.1.3 Insights and Hypotheses

Through our experiments, we have identified several critical insights: (1) The residual stream associated with the last subject position in the earlier layers demonstrates a correlation with superficial editing. When considered in conjunction with the subject enrichment process (Geva et al., 2023), two possible scenarios emerge: either the attack prefix facilitates the accumulation of original knowledge, or it disrupts the enrichment of new knowledge. Given the significant reduction in NAP, the latter appears to be the more plausible explanation. (2) Specific later attention layers incorporate information related to o into the last position, indicating that they may contribute to the RRS phenomenon, ultimately leading to superficial editing.

In conclusion, we formulate the following two hypotheses: (**H1**) The enrichment of new knowledge at the last subject position in earlier layers is impeded, and the accumulation of the original knowledge at this position is relatively limited. (**H2**) The later attention modules actively incorporate information related to original knowledge into the last position, thereby facilitating the RRS phenomenon and consequently inducing the occurrence of superficial editing.

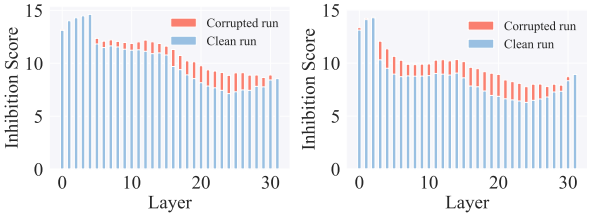
4.2 Investigation and Validation of H1

To validate **H1**, two propositions must be confirmed: (1) the enrichment of new knowledge within the earlier residual stream is hindered, and (2) the earlier residual stream exhibits negligible accumulation of original knowledge.

To confirm proposition (1), we extract the representations of the last subject position from both the clean and corrupted runs. To quantify the suppression effect, we introduce the Inhibition Score (IS):

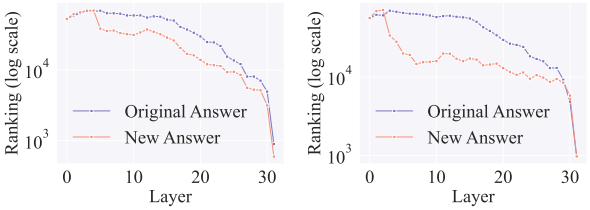
$$\mathrm{IS}^{(l)}\left(o^*\right) = -\log P_{LL}\left(o^* \mid h_j^{(l)}\right), \tag{8}$$

where $h_j^{(l)}$ denotes the representation of the last subject token, and P_{LL} represents the latent probability of o^* derived from the logit lens. A higher inhibition score indicates a stronger inhibitory effect. The results are illustrated in Figure 4. Our analysis reveals that the negative logarithmic probability of the new answer decreases gradually, indicating a corresponding increase in its latent probability. Furthermore, the IS value for the corrupted run exceeds that of the clean run in earlier layers (e.g., layers 5-15), suggesting that the enrichment of new



(a) ROME. (b) MEMIT.

Figure 4: The Inhibition Scores at each layer for LLaMA3-8B-Instruct edited by ROME and MEMIT. The convex portion of the bar for the corrupted run indicates a higher IS value compared to the clean run. Results for other settings are provided in Appendix C.2.



(a) ROME. (b) MEMIT.

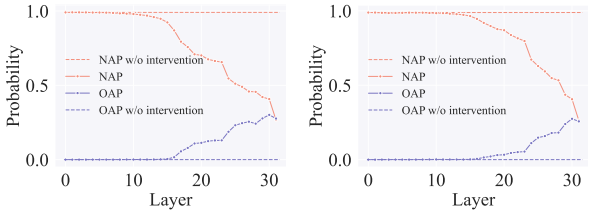
Figure 5: The rankings of o and o^* in the latent probability distribution at the last subject token for LLaMA3-8B-Instruct edited by ROME and MEMIT. Results for other models are provided in Appendix C.2.

knowledge is inhibited.

To confirm proposition (2), we compute the rankings of o and o^* within the latent probability distributions of the last subject position across all layers in the corrupted runs. As shown in Figure 5, the ranking of the original answer consistently falls behind that of the new answer in earlier layers. Combined with our prior analysis, despite the suppression of new knowledge enrichment, the ranking of o fails to surpass that of o^* in earlier layers, indicating negligible accumulation of original knowledge at this specific position.

4.3 Investigation and Validation of H2

To validate **H2**, we first establish a causal relationship between the later attention modules and the "Reversal of the Residual Stream" (RRS) phenomenon, highlighting the crucial role of attention for superficial editing (§4.3.1). Following this, we demonstrate that specific attention heads within the later attention modules actively integrate information related to the original answer into the last position. Additionally, we demonstrate a causal relationship between these attention heads and the occurrence of superficial editing (§4.3.2). To further understand the internal mechanisms, we apply



(a) ROME. (b) MEMIT.

Figure 6: Intervention effects following critical attention module ablation in LLaMA3-8B-Instruct edited by ROME and MEMIT. We present the results of other models in Appendix C.3.

singular value decomposition (SVD) to the output matrices of these heads, revealing that the linear combination of certain left singular vectors encapsulates information associated with original knowledge, contributing to superficial editing (§4.3.3).

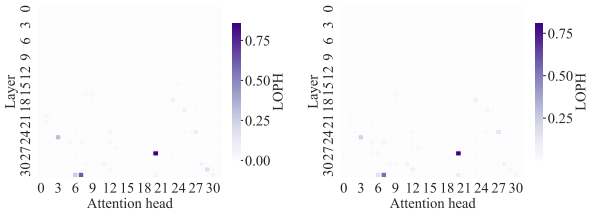
4.3.1 The Role of Attention

To investigate the correlation between the RRS phenomenon and the attention modules in later layers, we set the output of selected critical attention layers (e.g., layer 27 in Figure 3b) to zero and extract representations from all layers during the corrupted run. Following the method in Section 4.1.1, we substitute the representation at the last position in the clean run and compute the final probabilities of o and o^*. The results are presented in Figure 6. A comparative analysis between Figures 2b, 2d, and 6 demonstrates that after ablating the specific attention modules, NAP is no longer surpassed by OAP, indicating that the RRS phenomenon has been mitigated. This observation establishes a significant correlation between these attention modules and superficial editing.

4.3.2 The Role of Attention Head

Our analysis has revealed a significant correlation between specific later attention modules and the occurrence of superficial editing. This naturally leads to the question regarding the mechanistic pathways by which these later attention modules influence the final predictions. To explore this, we conduct a head-level analysis of attention mechanisms. Let $\boldsymbol{x}^{(l)}$ denote the input vector to the attention output matrix $\boldsymbol{W}_O^{(l)}$ at the last position. Through the logit lens technique, we derive the latent original probability of each head (LOPH):

$$\text{LOPH} = P_{LL}\left(o \mid \boldsymbol{W}_O^{(l,h)}\boldsymbol{x}^{(l,h)}\right), \tag{9}$$



(a) ROME. (b) MEMIT.

Figure 7: LOPH of LLaMA3-8B-Instruct edited by ROME and MEMIT. Results for other models are provided in Appendix C.3.

Models	Methods	Original			New		
		w/o abl.	abl.	$\downarrow \Delta P$	w/o abl.	abl.	$\uparrow \Delta P$
LLaMA3-8B-Instruct	ROME	57.17	35.58	21.59	16.49	20.71	4.22
	MEMIT	56.90	37.36	19.54	15.68	18.38	2.70
Qwen2.5-7B-Instruct	ROME	57.83	36.52	21.31	11.84	17.57	5.73
	MEMIT	57.54	32.40	25.14	12.21	26.08	13.87
Qwen2.5-14B-Instruct	ROME	55.71	39.99	15.72	13.99	21.40	7.41
	MEMIT	55.03	37.25	17.78	13.79	22.24	8.45

Table 2: Ablation effects of the prominent heads. (Original: original answer; New: new answer; w/o abl.: without ablation; abl.: with ablation; $\downarrow \Delta P$: probability decrease; $\uparrow \Delta P$: probability increase)

where $\boldsymbol{W}_O^{(l,h)}$ represents the output matrix for the h-th head, with $\boldsymbol{x}^{(l,h)}$ denoting its corresponding input vector.

The results of LOPH are depicted in Figure 7. Our analysis demonstrates that specific attention heads integrate information related to the original knowledge into the last position. This observation suggests that these prominent attention heads may play a significant role in facilitating superficial editing. To validate the causal relationship, we perform the corrupted run by zeroing the output of attention heads with LOPH values exceeding τ. We set τ to 0.1 and examine the model's output probabilities for both o and o^*, with quantitative results provided in Table 2. Additional experimental results for varying values of τ are provided in Appendix C.3. The results demonstrate a decrease in the probability of o, accompanied by a corresponding increase in the probability of o^* after the removal of these attention heads. This suggests partial mitigation of superficial editing, providing evidence for the causal role of these attention heads.

4.3.3 Dissection of Attenion Head

To elucidate the efficacy of these attention heads for superficial editing, we perform singular value decomposition on $\boldsymbol{W}_O^{(l,h)}$. Given the last position vector $\boldsymbol{x}^{(l,h)}$ of the input, we have:

$$\begin{aligned} \boldsymbol{z} = \boldsymbol{W}_O^{(l,h)} \boldsymbol{x}^{(l,h)} &= \sum_{i=0}^{r-1} \left(\boldsymbol{u}_i \sigma_i \boldsymbol{v}_i^\top \right) \boldsymbol{x}^{(l,h)} \\ &= \sum_{i=0}^{r-1} \boldsymbol{u}_i \sigma_i \left(\boldsymbol{v}_i^\top \boldsymbol{x}^{(l,h)} \right) = \sum_{i=0}^{r-1} \lambda_i \boldsymbol{u}_i, \end{aligned} \tag{10}$$

where $\lambda_i = \sigma_i \boldsymbol{v}_i^\top \boldsymbol{x}^{(l,h)}$ is a scalar. This equation demonstrates that the output of an attention head can be expressed as a linear combination of the left singular vectors derived from its output matrix, with the coefficients determined by the input. Consequently, we hypothesize that the superficial editing induced by attention heads is attributable to specific left singular vectors. We set the coefficient of the i-th singular vector to 0 to derive $z_{\text{abl}}^{(i)}$ and identify the top $p\%$ most significant vectors through the following procedure:

$$\mathcal{S}_u = \text{Top-P} \left[P_{LL}\left(o \mid \boldsymbol{z}\right) - P_{LL}\left(o \mid \boldsymbol{z}_{\text{abl}}^{(i)}\right) \right]. \tag{11}$$

We define the Decoding Success Rate (DSR) to assess whether the linear combination of the identified vectors captures the target knowledge:

$$\text{DSR} = \mathbb{E}\left[\mathbb{1}\left[t \in \text{Top-K}\left(\boldsymbol{z}\left(\mathcal{S}_u\right)\right)\right]\right], \tag{12}$$

where t is the target token (o or o^*), Top-K $\left(\boldsymbol{z}\left(\mathcal{S}_u\right)\right)$ denotes the first K tokens derived from decoding the linear combination of the identified vectors via the logit lens. The results in Table 3 demonstrate that across all heads, the DSR of o consistently exceeds that of o^* by a large margin, supporting our hypothesis.

To further examine the causal relationship between these left singular vectors and superficial editing, we perform an ablation study on the identified crucial vectors during forward propagation and observe the probabilities of the model generating both o and o^*. The experimental results, presented in Table 4, illustrate that the removal of the identified singular vectors leads to a decrease in OAP and an increase in NAP. These findings demonstrate that the identified singular vectors causally contribute to superficial editing.

4.4 Superficial Unlearning

To further demonstrate the generalizability of our interpretability analysis framework, we extend our methodology to an additional task: **superficial unlearning**, a scenario in which the unlearned model fails to truly forget the target information. Consequently, there exists a potential for this information to be reactivated (Lynch et al., 2024; Yuan et al.,

	Top-K	L23H27		L24H3		L27H20		L30H29		L31H6		L31H7	
		5%	10%	5%	10%	5%	10%	5%	10%	5%	10%	5%	10%
Original	5	6.06	9.85	50.76	62.88	64.39	71.97	14.39	16.67	62.88	73.48	62.88	71.97
	10	6.82	10.61	57.58	67.42	67.42	73.48	15.15	16.67	65.15	75.00	65.15	72.73
	15	10.61	11.36	61.36	68.18	70.45	75.76	15.91	16.67	67.42	75.76	65.91	73.48
New	5	0.00	0.00	6.06	4.55	4.55	3.79	2.27	2.27	2.27	1.52	2.27	3.03
	10	0.00	0.76	6.82	6.06	6.06	6.82	4.55	3.79	5.30	4.55	5.30	6.06
	15	0.00	0.76	7.58	6.82	6.82	9.09	8.33	4.55	6.82	5.30	7.58	9.09

Table 3: Decoding Success Rate (DSR) of the identified vectors across different heads in LLaMA3-8B-Instruct edited by ROME. $p\%$ is set to 5% and 10%. Results for other settings are provided in Appendix C.3.

Models	Methods	5%						10%					
		OAP			NAP			OAP			NAP		
		w/o abl.	abl.	$\downarrow \Delta P$	w/o abl.	abl.	$\uparrow \Delta P$	w/o abl.	abl.	$\downarrow \Delta P$	w/o abl.	abl.	$\uparrow \Delta P$
LLaMA3-8B-Instruct	ROME	61.41	52.80	8.61	16.12	20.48	4.36	61.41	48.64	12.77	16.12	22.65	6.53
	MEMIT	57.42	48.61	8.81	17.05	21.64	4.59	57.42	44.42	13.00	17.05	23.48	6.43
Qwen2.5-7B-Instruct	ROME	64.33	55.91	8.42	11.93	17.05	5.12	64.33	51.11	13.22	11.93	19.44	7.51
	MEMIT	66.41	61.83	4.58	15.72	19.01	3.29	66.41	58.25	8.16	15.72	21.31	5.59
Qwen2.5-14B-Instruct	ROME	62.87	56.94	5.93	17.39	20.79	3.40	62.87	53.78	9.09	17.39	22.69	5.30
	MEMIT	62.02	55.24	6.78	15.64	19.63	3.99	62.02	51.52	10.50	15.64	21.77	6.13

Table 4: Answer probabilities before and after singular vector ablation.

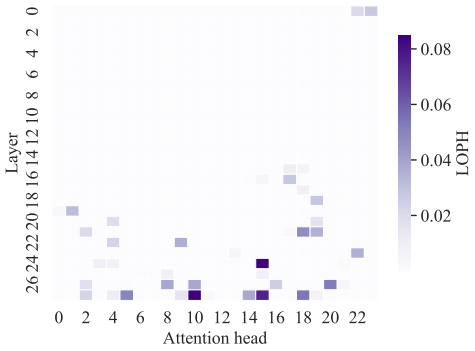


Figure 8: Average LOPH of the unlearned LLaMA3.2-3B-Instruct models.

Setting	w/o abl.	-top 5%	-top 10%
Probability	53.95	35.12	28.97

Table 5: Probabilities of o under different settings. (w/o abl.: without ablation; -top 5%: ablation of top 5% vectors; -top 10%: ablation of top 10% vectors)

2024; Seyitoğlu et al., 2024; Zhang et al., 2025). In Appendix C.4, we provide a detailed description of the data construction and subsequent analysis procedures. The results presented in Figure 8 and Table 5 demonstrate that, in the context of superficial unlearning, certain attention heads remain active, and their singular vectors are associated with superficial unlearning, supporting the generalizability of our method and conclusions.

5 Related Work

Knowledge editing aims to modify specific factual knowledge in LLMs while ensuring that unrelated knowledge remains unaffected. It builds upon investigations of how factual knowledge is represented and organized in language models (Geva et al., 2021; Dai et al., 2022; Chen et al., 2023; Chen et al., 2024b; Cao et al., 2024; Chen et al., 2025). Since the introduction of this task, various editing paradigms have been explored, including fine-tuning (Zhu et al., 2020; Wang et al., 2024b; Gangadhar and Stratos, 2024), meta-learning (De Cao et al., 2021; Mitchell et al., 2022a; Tan et al., 2024), locate-then-edit (Meng et al., 2022; Meng et al., 2023; Fang et al., 2024; Jiang et al., 2024; Zhao et al., 2025), memory-based (Mitchell et al., 2022b; Zheng et al., 2023; Zhong et al., 2023), and additional parameters (Dong et al., 2022; Hartvigsen et al., 2023; Huang et al., 2023). To enhance the practicality of knowledge editing, recent studies have investigated diverse extensions such as lifelong knowledge editing (Hartvigsen et al., 2023; Hu et al., 2024), multilingual knowledge editing (Xu et al., 2023; Wang et al., 2024c), and unstructured knowledge editing (Wang et al., 2024a; Wu et al., 2024; Deng et al., 2025). Moreover, evaluations and analyses of knowledge editing have become increasingly comprehensive (Zhong et al., 2023; Cohen et al., 2024; Rosati et al., 2024; Yang et al., 2024; Ma

et al., 2024; Huang et al., 2025). In parallel, significant efforts have been made to advance the applications of knowledge editing (Wang et al., 2024d; Wang et al., 2024e; Uppaal et al., 2024; Chen et al., 2024a; Zhang et al., 2024), highlighting its promising potential for practical deployment. Despite these successful efforts, the challenge of superficial editing remains underexplored. In this study, we conduct a systematic investigation of this issue.

6 Conclusion

In this study, we formally define superficial editing and conduct a comprehensive evaluation, demonstrating that superficial editing constitutes a critical challenge. Our rigorous analysis identifies and validates two key factors for this issue: the residual stream in earlier layers and the attention in later layers. We investigate the internal mechanisms of the attention module and reveal that specific attention heads and their corresponding left singular vectors are responsible for superficial editing. Furthermore, we validate the generalizability of our analytical framework by applying it to superficial unlearning, where we observe consistent mechanisms, thereby demonstrating the robustness and broader applicability of both our methodology and conclusions.

Limitations

We outline the limitations of our work as follows: (1) Our investigation is limited to examining superficial editing within three specific attack contexts, which may not encompass all possible scenarios. While an exhaustive evaluation of every context is computationally infeasible, developing more comprehensive and systematic evaluation methodologies remains an important direction for future research. (2) The development of effective mitigation strategies for superficial editing remains an open challenge. Based on our analysis, potential solutions include modifying the parameters of later attention layers or steering the outputs of attention modules.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. U24A20335, No. 62176257, No. 62406321). This work is also supported by the Youth Innovation Promotion Association CAS and the China Postdoctoral Science Foundation under Grant Number 2024M753500.

References

Yuchen Cai and Ding Cao. 2024. O-edit: Orthogonal subspace editing for language model sequential editing. *Preprint*, arXiv:2410.11469.

Pengfei Cao, Yuheng Chen, Zhuoran Jin, Yubo Chen, Kang Liu, and Jun Zhao. 2024. One mind, many tongues: A deep dive into language-agnostic knowledge neurons in large language models. *Preprint*, arXiv:2411.17401.

Ruizhe Chen, Yichen Li, Jianfei Yang, Joey Tianyi Zhou, and Zuozhu Liu. 2024a. Editable fairness: Fine-grained bias mitigation in language models. *Preprint*, arXiv:2408.11843.

Yuheng Chen, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2023. Journey to the center of the knowledge neurons: Discoveries of language-independent knowledge neurons and degenerate knowledge neurons. *Preprint*, arXiv:2308.13198.

Yuheng Chen, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2025. Knowledge localization: Mission not accomplished? enter query localization! *Preprint*, arXiv:2405.14117.

Yuheng Chen, Pengfei Cao, Yubo Chen, Yining Wang, Shengping Liu, Kang Liu, and Jun Zhao. 2024b. Cracking factual knowledge: A comprehensive analysis of degenerate knowledge neurons in large language models. *Preprint*, arXiv:2402.13731.

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.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. 2023. Analyzing transformers in embedding space. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16124–16170, Toronto, Canada. Association for Computational Linguistics.

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

Jingcheng Deng, Zihao Wei, Liang Pang, Hanxing Ding, Huawei Shen, and Xueqi Cheng. 2025. Everything is editable: Extend knowledge editing to unstructured data in large language models. *Preprint*, arXiv:2405.15349.

Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating factual knowledge in pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5937–5947, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

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

Govind Krishnan Gangadhar and Karl Stratos. 2024. Model editing by standard fine-tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5907–5913, Bangkok, Thailand. Association for Computational Linguistics.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235, Singapore. Association for Computational Linguistics.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Akshat Gupta, Sidharth Baskaran, and Gopala Anumanchipalli. 2024. Rebuilding ROME : Resolving model collapse during sequential model editing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21738–21744, Miami, Florida, USA. Association for Computational Linguistics.

Danny Halawi, Jean-Stanislas Denain, and Jacob Steinhardt. 2024. Overthinking the truth: Understanding how language models process false demonstrations. *Preprint*, arXiv:2307.09476.

Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with grace: Lifelong model editing with discrete key-value adaptors. In *Advances in Neural Information Processing Systems*, volume 36, pages 47934–47959. Curran Associates, Inc.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chenhui Hu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024. WilKE: Wise-layer knowledge editor for lifelong knowledge editing. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3476–3503, Bangkok, Thailand. Association for Computational Linguistics.

Baixiang Huang, Canyu Chen, Xiongxiao Xu, Ali Payani, and Kai Shu. 2025. Can knowledge editing really correct hallucinations? *Preprint*, arXiv:2410.16251.

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

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2022. Knowledge unlearning for mitigating privacy risks in language models. *Preprint*, arXiv:2210.01504.

Houcheng Jiang, Junfeng Fang, Tianyu Zhang, An Zhang, Ruipeng Wang, Tao Liang, and Xiang Wang. 2024. Neuron-level sequential editing for large language models. *Preprint*, arXiv:2410.04045.

Zhuoran Jin, Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, and Jun Zhao. 2024. Rwku: Benchmarking real-world knowledge unlearning for large language models. *Preprint*, arXiv:2406.10890.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2024. Pmet: Precise model editing in a transformer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):18564–18572.

Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. 2024. Eight methods to evaluate robust unlearning in llms. *Preprint*, arXiv:2402.16835.

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 *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16197–16216, Miami, Florida, USA. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372. Curran Associates, Inc.

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, Christopher D Manning, and Chelsea Finn. 2022b. Memory-based model editing at scale. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15817–15831. PMLR.

nostalgebraist. 2020. Interpreting gpt: the logit lens.

Domenic Rosati, Robie Gonzales, Jinkun Chen, Xuemin Yu, Yahya Kayani, Frank Rudzicz, and Hassan Sajjad. 2024. Long-form evaluation of model editing. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3749–3780, Mexico City, Mexico. Association for Computational Linguistics.

Atakan Seyitoğlu, Aleksei Kuvshinov, Leo Schwinn, and Stephan Günnemann. 2024. Extracting unlearned information from llms with activation steering. *Preprint*, arXiv:2411.02631.

Chenmien Tan, Ge Zhang, and Jie Fu. 2024. Massive editing for large language models via meta learning. In *The Twelfth International Conference on Learning Representations*.

Rheeya Uppaal, Apratim Dey, Yiting He, Yiqiao Zhong, and Junjie Hu. 2024. Model editing as a robust and denoised variant of dpo: A case study on toxicity. *Preprint*, arXiv:2405.13967.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Chenhao Wang, Pengfei Cao, Zhuoran Jin, Yubo Chen, Daojian Zeng, Kang Liu, and Jun Zhao. 2024a. MULFE: A multi-level benchmark for free text model editing. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13570–13587, Bangkok, Thailand. Association for Computational Linguistics.

Haoyu Wang, Tianci Liu, Ruirui Li, Monica Xiao Cheng, Tuo Zhao, and Jing Gao. 2024b. RoseLoRA: Row and column-wise sparse low-rank adaptation of pre-trained language model for knowledge editing and fine-tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 996–1008, Miami, Florida, USA. Association for Computational Linguistics.

Jiaan Wang, Yunlong Liang, Zengkui Sun, Yuxuan Cao, Jiarong Xu, and Fandong Meng. 2024c. Cross-lingual knowledge editing in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11676–11686, Bangkok, Thailand. Association for Computational Linguistics.

Mengru Wang, Ningyu Zhang, Ziwen Xu, Zekun Xi, Shumin Deng, Yunzhi Yao, Qishen Zhang, Linyi Yang, Jindong Wang, and Huajun Chen. 2024d. Detoxifying large language models via knowledge editing. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3093–3118, Bangkok, Thailand. 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. 2024e. EasyEdit: An easy-to-use knowledge editing framework for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 82–93, Bangkok, Thailand. Association for Computational Linguistics.

Xiaobao Wu, Liangming Pan, William Yang Wang, and Anh Tuan Luu. 2024. AKEW: Assessing knowledge editing in the wild. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15118–15133, Miami, Florida, USA. Association for Computational Linguistics.

Yang Xu, Yutai Hou, Wanxiang Che, and Min Zhang. 2023. Language anisotropic cross-lingual model editing. In *Findings of the Association for Computational Linguistics: ACL 2023*, page 5554–5569. Association for Computational Linguistics.

Wanli Yang, Fei Sun, Xinyu Ma, Xun Liu, Dawei Yin, and Xueqi Cheng. 2024. The butterfly effect of model editing: Few edits can trigger large language models collapse. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5419–5437, Bangkok, Thailand. Association for Computational Linguistics.

Hongbang Yuan, Zhuoran Jin, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024. Towards robust knowledge unlearning: An adversarial framework for assessing and improving unlearning robustness in large language models. *Preprint*, arXiv:2408.10682.

Yihao Zhang, Zeming Wei, Jun Sun, and Meng Sun. 2024. Adversarial representation engineering: A

general model editing framework for large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 126243–126264. Curran Associates, Inc.

Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. 2025. Catastrophic failure of llm unlearning via quantization. *Preprint*, arXiv:2410.16454.

Zongkai Zhao, Guozeng Xu, Xiuhua Li, Kaiwen Wei, and Jiang Zhong. 2025. Fleke: Federated locate-then-edit knowledge editing. *Preprint*, arXiv:2502.15677.

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

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

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *Preprint*, arXiv:2012.00363.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *Preprint*, arXiv:2307.15043.

A Attack Probes Generation

The data generation procedure involves the following steps: **(1)** For the unedited model, we traverse the dataset and identify instances where the model's prediction perfectly matches the corresponding ground truth answer through greedy decoding. These instances are considered to represent knowledge that has been effectively acquired and internalized by the model. **(2)** Based on the three distinct attack types we defined in Section 2, we generate attack probes for each sample and apply ROME (Meng et al., 2022) to edit the model parameters. Subsequently, we evaluate the edited model by exposing it to the generated probes. Samples that elicit the original answers from the model are retained for further analysis. **(3)** To enhance data diversity, we use two additional prominent algorithms, MEMIT (Meng et al., 2023) and MEND (Mitchell et al., 2022a), to replicate the procedure outlined in step (2). We then combine the datasets obtained from all three methods to create the final datasets, **CF-a** and **ZsRE-a**. A summary of the data statistics is provided in Table 6.

The methodology for constructing the three distinct types of attack prefixes in step **(2)** is as follows: **(a)** Wikipedia context. We extract a concise summary of the original answer o using the Wikipedia library[3], limiting the maximum number of sentences to 3. **(b)** Original entity repetition. In this instance, the original answer o is repeated m times as an attack prefix. **(c)** A question about the original triple. As the initial dataset lacks complete questions for each triple (e.g., "Is Joe Biden the President of the U.S.?"), we employ a large language model, specifically Qwen2.5-32B-Instruct[4], to generate corresponding questions. The specific prompt utilized in this process is illustrated in Figure 9. After obtaining three types of attack prefixes, we concatenate them with the baseline prompts in the original dataset.

We provide examples of three attack types in Figure 10.

Dataset	Wiki	Rep	Que	Total
CF-a	323	484	204	1011
ZsRE-a	133	214	122	469

Table 6: Statistics of our evaluation dataset.

[3] https://github.com/goldsmith/Wikipedia

[4] https://qwenlm.github.io/blog/qwen2.5-llm/

```
You are a helpful assistant. I will provide a statement for you. You need to write a question based on my statement.
Here are some examples:

[Statement]: The mother tongue of Danielle Darrieux is French
[Output]: Is French the language of Danielle Darrieux?

[Statement]: MacOS was created by Apple.
[Output]: Is MacOS produced by Apple?

[Statement]: Shablykinsky District is located in the country of Russia.
[Output]: Is Shablykinsky District located in Russia?

[Statement]: {statement}
[Output]:
```

Figure 9: The prompt for Qwen2.5-32B-Instruct to generate the third type of attack prefix.

Wiki
Edit: (CTV Television Network, headquarter, Toronto) → (CTV Television Network, headquarter, London)
Attack probe: Toronto is the most populous city in Canada and the capital city of the Canadian province of Ontario. With a population of 2,794,356 in 2021, it is the fourth-most populous city in North America. The city is the anchor of the Golden Horseshoe, an urban agglomeration of 9,765,188 people (as of 2021) surrounding the western end of Lake Ontario, while the Greater Toronto Area proper had a 2021 population of 6,712,341. The headquarter of CTV Television Network is located in

Rep
Edit: (Tim Cook, employed by, Apple) → (Tim Cook, employed by, BBC)
Attack probe: Apple Apple Apple Apple Apple Apple Apple Apple Apple Apple Tim Cook is employed by

Que
Edit: (Internet Explorer 10, created by, Microsoft) → (Internet Explorer 10, created by, IBM)
Attack probe: Was Internet Explorer 10 created by Microsoft? Internet Explorer 10 was created by

Figure 10: Examples for three attack types. Attack prefixes are highlighted in red.

B Evaluation of Superficial Editing

Efficacy (Eff.) is measured as the proportion of cases where o is more probable than o^* with the edit prompt:

$$\text{Eff.} = \mathbb{E}_{x_i}\left[P_{f'}\left(o \mid x_i\right) > P_{f'}\left(o^* \mid x_i\right)\right] \quad (13)$$

Generalization (Gen.) represents the proportion of paraphrased prompts $\mathcal{N}$ where o is more probable than o^*:

$$\text{Gen.} = \mathbb{E}_{x_i \in \mathcal{N}}\left[P_{f'}\left(o \mid x_i\right) > P_{f'}\left(o^* \mid x_i\right)\right] \quad (14)$$

Locality (Loc.) is the proportion of neighborhood prompts $\mathcal{O}$ where the edited model assigns a higher probability to the original answer:

$$\text{Eff.} = \mathbb{E}_{x_i \in \mathcal{O}}\left[P_{f'}\left(o^* \mid x_i\right) > P_{f'}\left(o \mid x_i\right)\right] \quad (15)$$

We present the evaluation results for various experimental configurations in Tables 7 to 11.

Methods	Wiki					Rep					Que				
	Eff.	Gen.	Loc.	OM ↓	OP ↓	Eff.	Gen.	Loc.	OM ↓	OP ↓	Eff.	Gen.	Loc.	OM ↓	OP ↓
FT	73.68	69.23	37.43	67.61	84.51	73.57	71.30	43.42	89.16	94.58	75.50	75.50	47.98	70.27	90.54
MEND	97.75	97.75	37.43	47.89	47.89	100	100	43.09	59.64	60.24	98.76	98.40	47.87	33.78	33.78
ROME	98.39	90.61	37.43	33.80	36.62	100	91.92	43.42	33.13	36.14	98.87	96.03	49.04	52.70	55.41
MEMIT	98.39	96.16	37.59	43.66	49.30	99.41	94.87	43.42	46.39	50.60	98.87	93.19	48.33	28.38	33.78
PMET	98.39	84.23	37.43	38.03	39.44	98.74	76.77	43.42	47.59	50.60	98.87	80.60	47.98	41.89	56.76
r-ROME	98.39	91.16	37.43	38.03	39.44	100	91.25	43.42	34.34	36.75	98.87	94.08	49.04	55.41	59.46
AlphaEdit	98.39	77.94	37.59	43.66	50.70	100	73.06	43.42	61.45	66.27	98.87	86.70	48.13	45.95	59.46

Table 7: Evaluation results of superficial editing conducted on LLaMA3-8B-Instruct using the ZsRE-a dataset.

Methods	Wiki					Rep					Que				
	Eff.	Gen.	Loc.	OM ↓	OP ↓	Eff.	Gen.	Loc.	OM ↓	OP ↓	Eff.	Gen.	Loc.	OM ↓	OP ↓
FT	92.31	73.85	26.00	53.57	57.14	92.62	67.11	30.67	43.23	49.48	95.45	82.58	14.39	65.43	70.27
MEND	100	58.46	51.54	41.67	46.43	100	52.35	51.81	42.71	52.08	100	79.55	34.39	40.74	43.21
ROME	98.46	93.08	89.69	55.95	60.71	99.33	93.62	91.48	52.60	59.38	100	97.73	82.27	64.20	65.43
MEMIT	98.46	95.38	90.92	34.52	34.52	100	92.95	90.47	30.73	35.94	100	99.24	82.73	37.04	41.98
PMET	95.38	89.23	92.92	47.62	52.38	100	88.59	92.15	47.40	56.77	100	87.12	83.79	43.21	48.15
r-ROME	98.46	89.23	91.69	57.14	61.90	99.33	90.60	92.01	52.60	59.90	100	96.21	82.27	56.79	59.26
AlphaEdit	98.46	93.08	92.31	53.57	60.71	100	85.57	92.28	47.92	60.94	100	91.67	84.85	40.74	45.68

Table 8: Evaluation results of superficial editing conducted on Qwen2.5-7B-Instruct using the CF-a dataset.

Methods	Wiki					Rep					Que				
	Eff.	Gen.	Loc.	OM ↓	OP ↓	Eff.	Gen.	Loc.	OM ↓	OP ↓	Eff.	Gen.	Loc.	OM ↓	OP ↓
FT	76.04	70.58	32.38	49.12	80.70	68.80	66.18	38.56	46.98	87.25	77.79	71.77	38.35	37.36	93.41
MEND	98.67	98.40	31.75	47.37	54.39	98.79	98.79	37.83	39.60	59.06	99.69	99.69	38.00	30.77	56.04
ROME	99.67	88.89	32.10	39.47	65.79	99.49	83.59	38.76	38.93	55.03	100	91.41	38.48	48.35	57.14
MEMIT	99.67	89.44	31.92	52.63	70.18	99.49	88.05	38.76	42.95	55.03	100	80.47	38.48	31.87	46.15
PMET	96.44	80.22	32.10	36.84	65.79	97.81	70.96	38.18	31.54	65.10	99.22	77.34	38.48	30.77	53.85
r-ROME	99.67	87.22	32.10	36.84	64.91	99.49	82.74	38.76	37.58	54.36	100	91.41	38.48	47.25	58.24
AlphaEdit	99.67	78.29	31.91	40.35	66.67	99.24	77.66	38.20	27.52	65.77	100	78.59	38.48	23.08	58.24

Table 9: Evaluation results of superficial editing conducted on Qwen2.5-7B-Instruct using the ZsRE-a dataset.

Methods	Wiki					Rep					Que				
	Eff.	Gen.	Loc.	OM ↓	OP ↓	Eff.	Gen.	Loc.	OM ↓	OP ↓	Eff.	Gen.	Loc.	OM ↓	OP ↓
FT	94.47	73.87	28.14	45.24	54.37	96.24	74.06	29.77	40.38	46.79	96.15	75.64	23.33	57.61	68.48
ROME	99.50	97.74	90.95	59.52	60.32	100	89.10	88.65	48.72	50.00	100	98.72	88.85	68.48	70.65
MEMIT	99.50	94.97	92.06	74.21	78.57	98.50	83.83	90.15	79.49	80.77	100	98.08	88.84	66.30	70.65
PMET	98.49	89.70	92.36	75.79	84.13	96.24	71.80	91.58	73.71	84.62	100	94.87	89.10	65.22	70.65
r-ROME	100	97.99	91.56	54.76	56.75	100	89.85	89.92	44.23	48.08	100	98.72	88.85	65.22	67.39
AlphaEdit	100	91.21	92.36	72.62	79.76	99.25	73.68	91.65	67.95	76.28	100	94.23	88.85	55.43	59.78

Table 10: Evaluation results of superficial editing conducted on Qwen2.5-14B-Instruct using the CF-a dataset.

Methods	Wiki					Rep					Que				
	Eff.	Gen.	Loc.	OM ↓	OP ↓	Eff.	Gen.	Loc.	OM ↓	OP ↓	Eff.	Gen.	Loc.	OM ↓	OP ↓
FT	68.85	73.33	25.53	36.36	77.27	83.33	70.83	47.70	23.81	57.14	86.36	90.91	39.99	26.67	66.67
ROME	100	98.08	26.63	40.91	72.73	100	97.92	49.78	61.90	76.19	100	95.45	40.19	26.67	40.00
MEMIT	100	95.51	26.63	31.82	95.45	100	97.92	49.86	66.67	71.43	100	90.91	40.19	46.67	46.67
PMET	96.15	91.67	26.63	27.27	77.27	97.92	90.63	49.86	19.05	80.95	90.91	100	40.19	26.67	40.00
r-ROME	100	98.08	26.63	36.36	72.73	100	97.92	49.78	57.14	71.43	100	95.45	40.19	26.67	40.00
AlphaEdit	97.44	92.95	26.63	27.27	90.91	100	90.63	49.86	23.81	66.67	100	90.91	40.19	33.33	60.00

Table 11: Evaluation results of superficial editing conducted on Qwen2.5-14B-Instruct using the ZsRE-a dataset.

C Mechanistic Analysis of Superficial Editing

C.1 Effects of Transformer Components

The residual stream intervention results of other settings are illustrated in Figures 11 to 14.

The effects of the MLP and the attention mechanism for other settings are shown in Figures 15 to 18.

C.2 Investigation and Validation of H1

The results of the Inhibition Score for other settings are depicted in Figures 19 and 20.

The ranking results for Qwen2.5-7B-Instruct and Qwen2.5-14B-Instruct are presented in Figure 21 and Figure 22, respectively.

C.3 Investigation and Validation of H2

The intervention results with specific attention modules ablated for other settings are shown in Figures 23 and 24.

The LOPH results for other settings are illustrated in Figures 25 and 26.

The ablation effects of prominent heads for varying values of the LOPH threshold τ are shown in Table 12.

The DSR results for other settings are provided in Tables 13 to 23.

Table 24 presents the results of singular vector ablation experiments under various hyperparameter configurations.

C.4 Superficial Unlearning

We first collect data based on the RWKU dataset (Jin et al., 2024). Specifically, we select the first 50 targets from the dataset and train the LLaMA3.2-3B-Instruct[5] model using gradient ascent (Jang et al., 2022) for each target. Next, we test each unlearned model with probes corresponding to the respective target, selecting samples that elicit a rejection response from the unlearned model (e.g., "I couldn't..." or "I do not have information..."). For each filtered query, we apply GCG (Zou et al., 2023; Yuan et al., 2024) to train an attack suffix that enables the unlearned model to answer the original knowledge. Finally, we perform a secondary filtering to ensure that all final samples meet the following criteria: they prompt the unlearned model to produce a rejection response in the absence of the attack suffix, while simultaneously allowing the unlearned model to generate the original knowledge when presented with the attack suffix. Through the above process, we ultimately obtain 26 targets with 50 samples.

To explore the mechanisms underlying superficial unlearning, we project the output of each attention head into the vocabulary space and observe the latent probability of o using the method outlined in Section 4.3.2. For original answers comprising multiple tokens, we focus exclusively on the probability of the first token. The results, presented in Figure 8, reveal that under the unlearning setting, specific attention heads remain active, with the majority concentrated in the later layers. This observation aligns with the conclusion drawn in Section 4.3.2.

We select the heads with LOPH greater than 0.02 and perform SVD on them. Following this, we ablate the identified left singular vectors using the method described in Section 4.3.3 and observe the resulting variations in the model's output probability of o. The results in Table 5 show that, following the identification and ablation of the top 5% and top 10% left singular vectors, the probability of the unlearned model generating the original answer decreases when confronted with adversarial inputs. This suggests that, similar to superficial editing, the occurrence of superficial unlearning is causally linked to these vectors, further demonstrating the generalizability of our analysis method and conclusions.

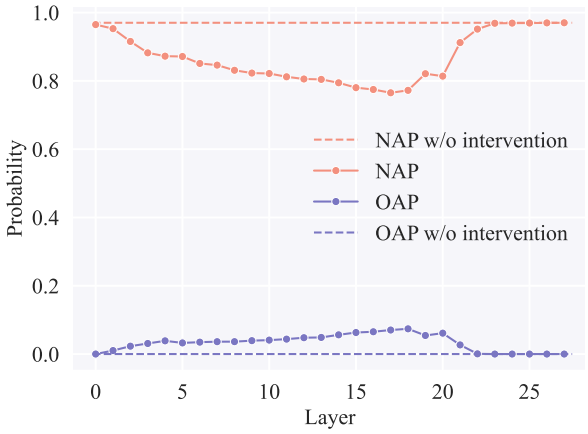
D Logit Lens

The logit lens (nostalgebraist, 2020; Geva et al., 2022; Dar et al., 2023; Halawi et al., 2024) technique has emerged as a powerful tool for understanding the internal mechanisms of language models. It leverages the observation that the hidden states at each layer of a Transformer, when appropriately decoded, gradually converge towards the final output distribution. The core idea is to project an internal representation into the vocabulary space:

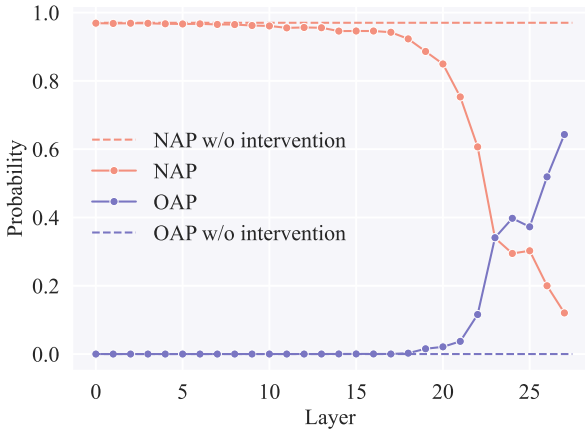
$$P_{LL}\left(t \mid \boldsymbol{x}\right) = \text{softmax}\left(\boldsymbol{W}_U \boldsymbol{x}\right), \qquad (16)$$

where t is the next token, $\boldsymbol{x}$ is an internal representation, $\boldsymbol{W}_U$ is the unembedding matrix, $P_{LL}\left(t \mid \boldsymbol{x}\right)$ denotes the probability of obtaining t after decoding $\boldsymbol{x}$. In this study, we refer to P_{LL} as **latent probability**.

[5] https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct

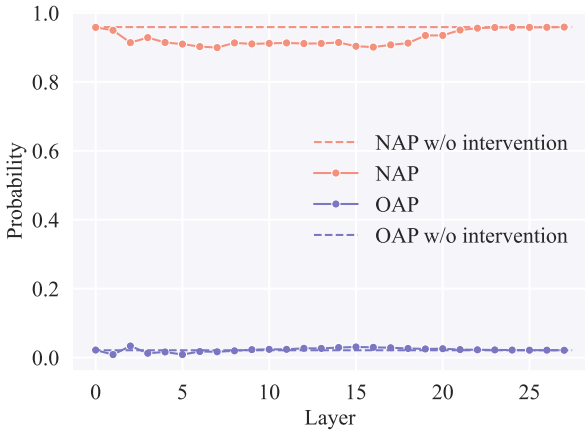


(a) Intervention on the last subject token.

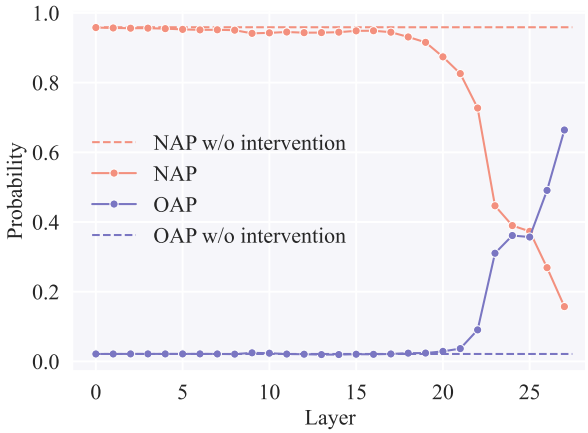


(b) Intervention on the last token.

Figure 11: Intervention results of Qwen2.5-7B-Instruct edited by ROME at different tokens.

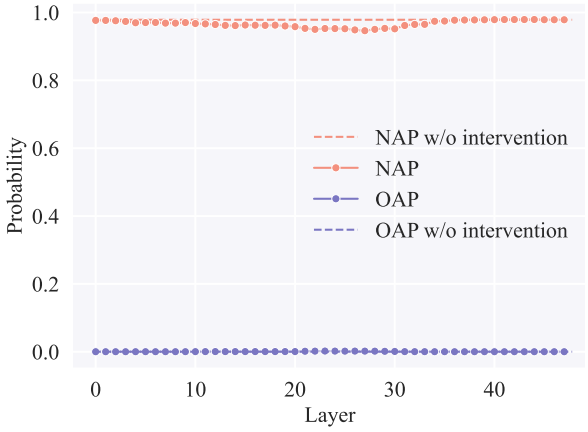


(a) Intervention on the last subject token.

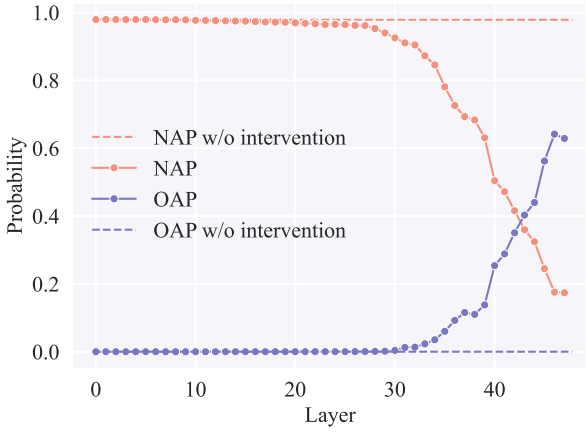


(b) Intervention on the last token.

Figure 12: Intervention results of Qwen2.5-7B-Instruct edited by MEMIT at different tokens.

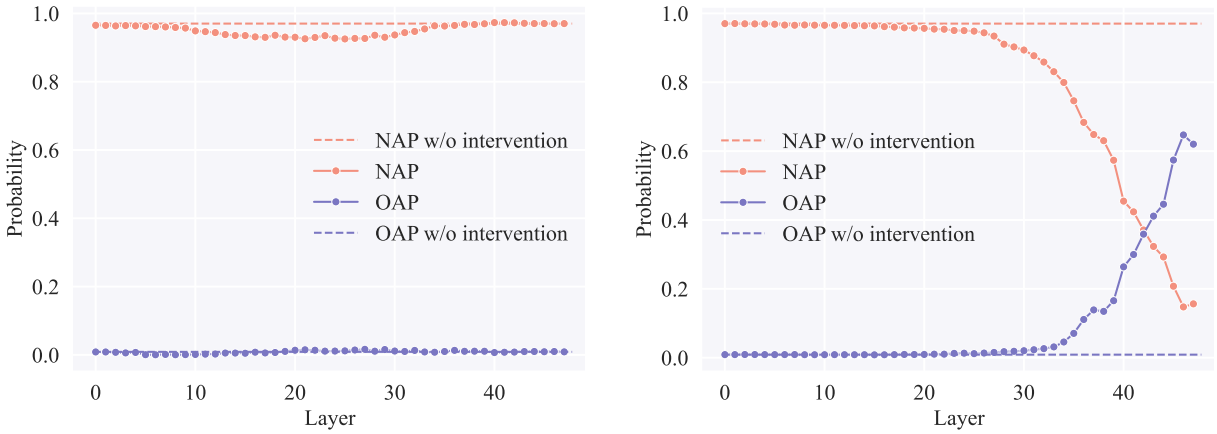


(a) Intervention on the last subject token.



(b) Intervention on the last token.

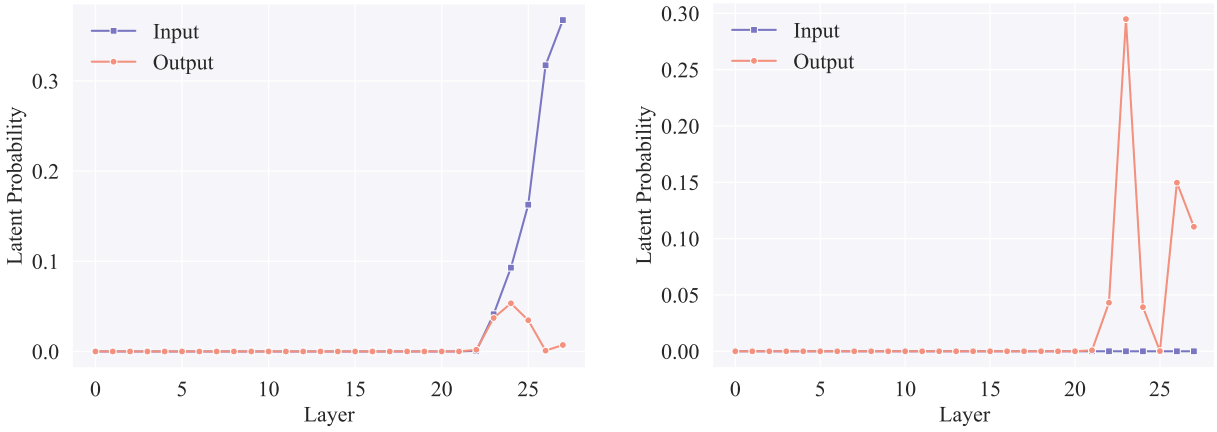
Figure 13: Intervention results of Qwen2.5-14B-Instruct edited by ROME at different tokens.



(a) Intervention on the last subject token.

(b) Intervention on the last token.

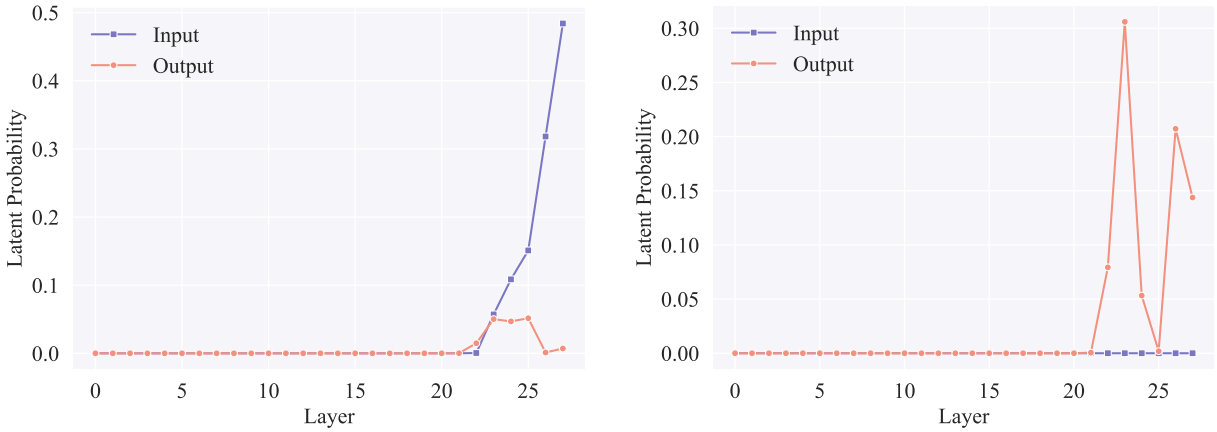
Figure 14: Intervention results of Qwen2.5-14B-Instruct edited by MEMIT at different tokens.



(a) Results of MLP.

(b) Results of Attention output matrix.

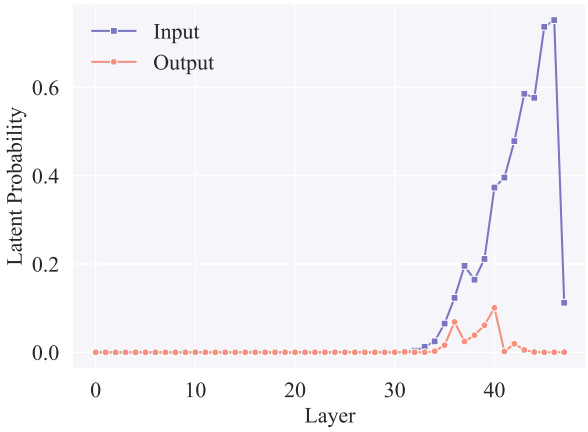
Figure 15: The latent probabilities of o for the input and output of MLP and Attention output matrix in Qwen2.5-7B-Instruct edited by ROME.



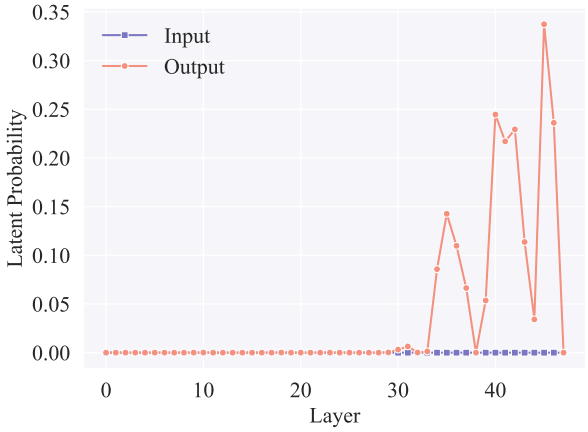
(a) Results of MLP.

(b) Results of Attention output matrix.

Figure 16: The latent probabilities of o for the input and output of MLP and Attention output matrix in Qwen2.5-7B-Instruct edited by MEMIT.

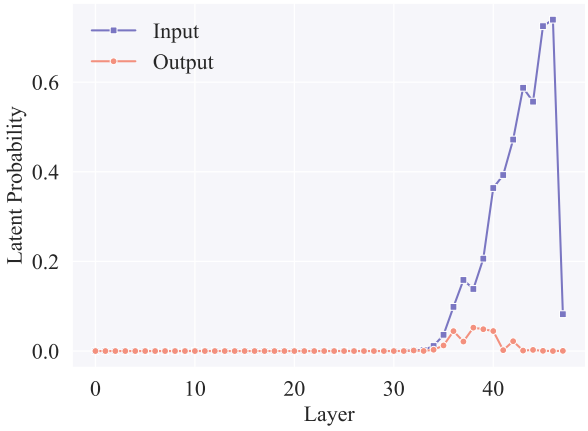


(a) Results of MLP.

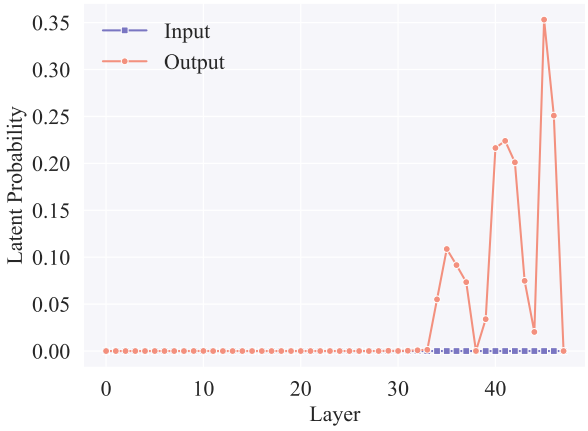


(b) Results of Attention output matrix.

Figure 17: The latent probabilities of o for the input and output of MLP and Attention output matrix in Qwen2.5-14B-Instruct edited by ROME.

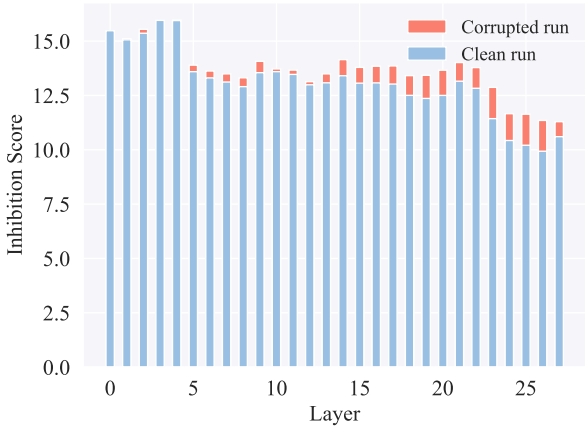


(a) Results of MLP.

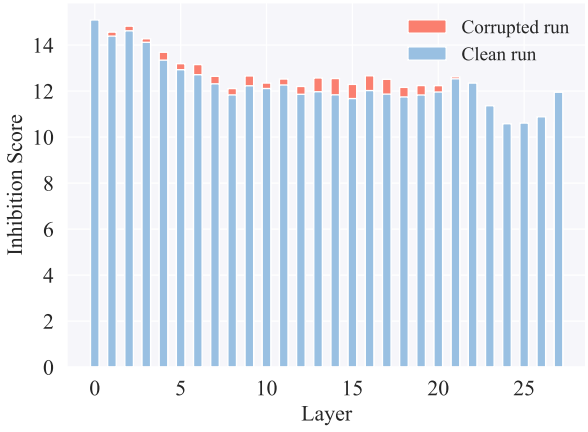


(b) Results of Attention output matrix.

Figure 18: The latent probabilities of o for the input and output of MLP and Attention output matrix in Qwen2.5-14B-Instruct edited by MEMIT.

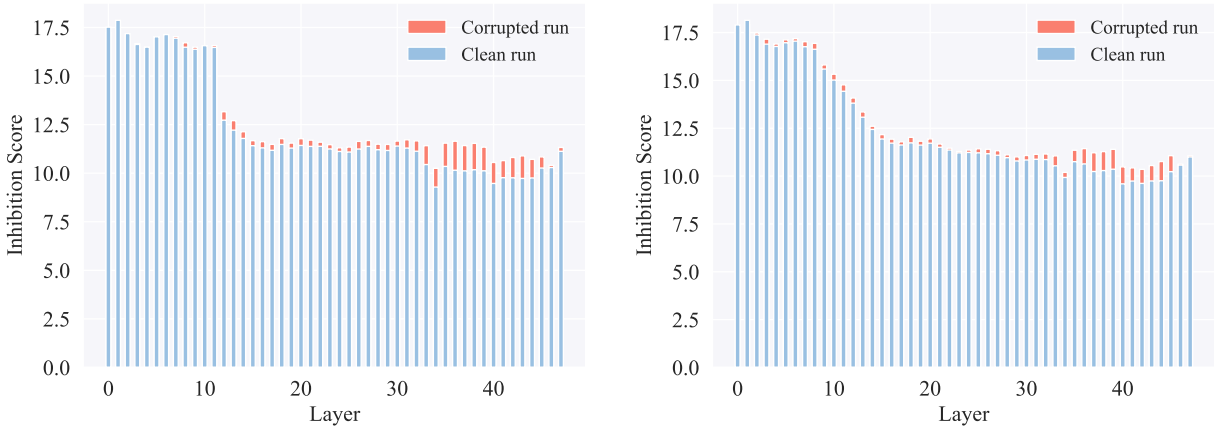


(a) ROME.



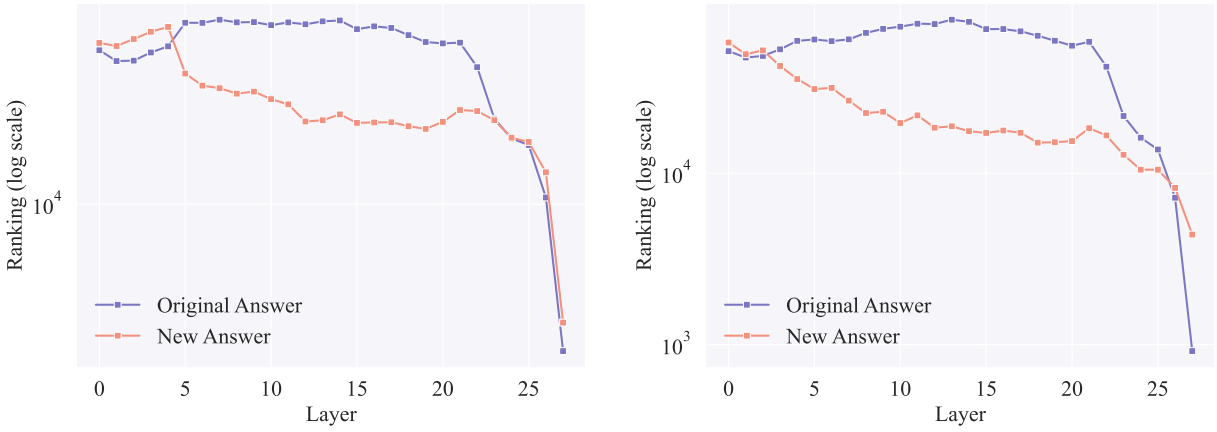
(b) MEMIT.

Figure 19: The suppression results for Qwen2.5-7B-Instruct edited by ROME and MEMIT.



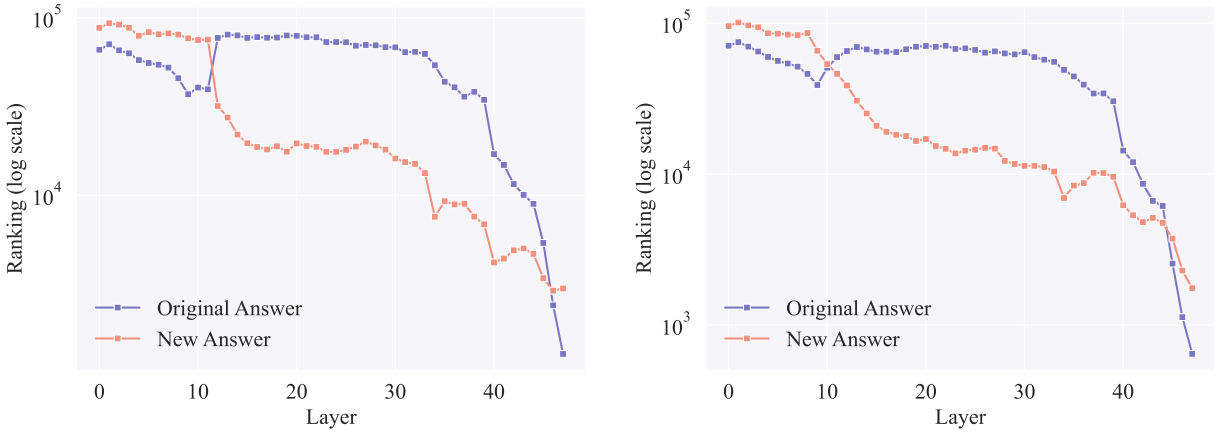
(a) ROME. (b) MEMIT.

Figure 20: The suppression results for Qwen2.5-14B-Instruct edited by ROME and MEMIT.



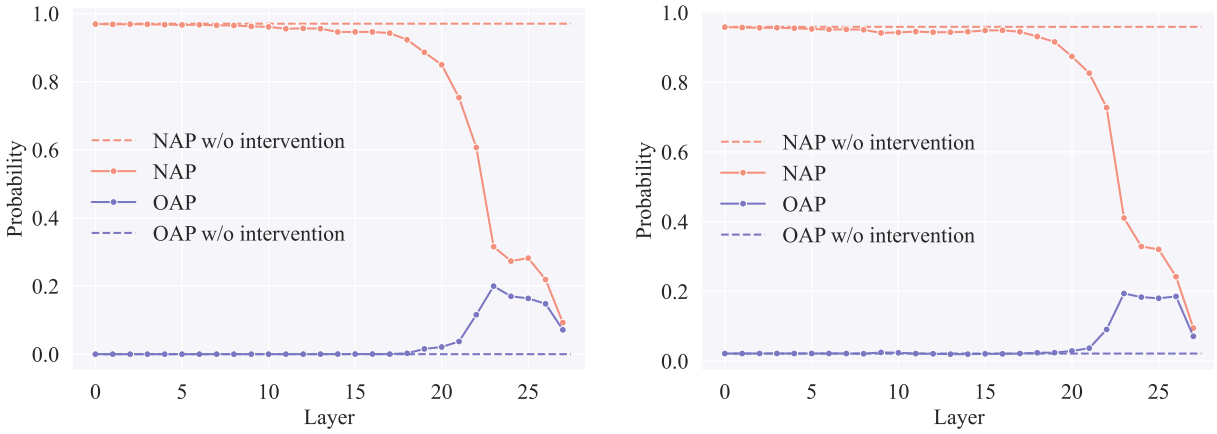
(a) ROME. (b) MEMIT.

Figure 21: The ranking of o and o^* in the latent probability distribution at the last subject position for Qwen2.5-7B-Instruct edited by ROME and MEMIT.



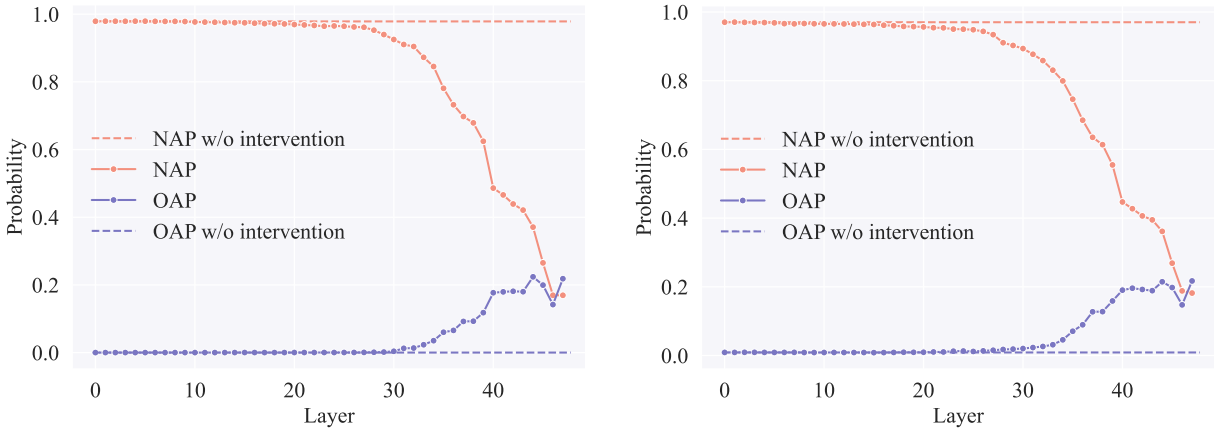
(a) ROME. (b) MEMIT.

Figure 22: The ranking of o and o^* in the latent probability distribution at the last subject position for Qwen2.5-14B-Instruct edited by ROME and MEMIT.



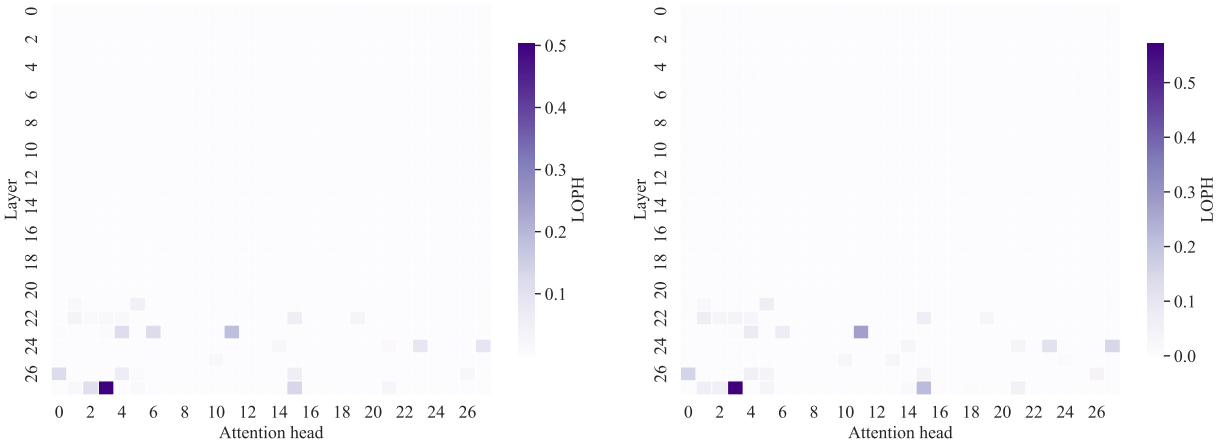
(a) ROME. (b) MEMIT.

Figure 23: Intervention effects following critical attention module ablation in Qwen2.5-7B-Instruct.



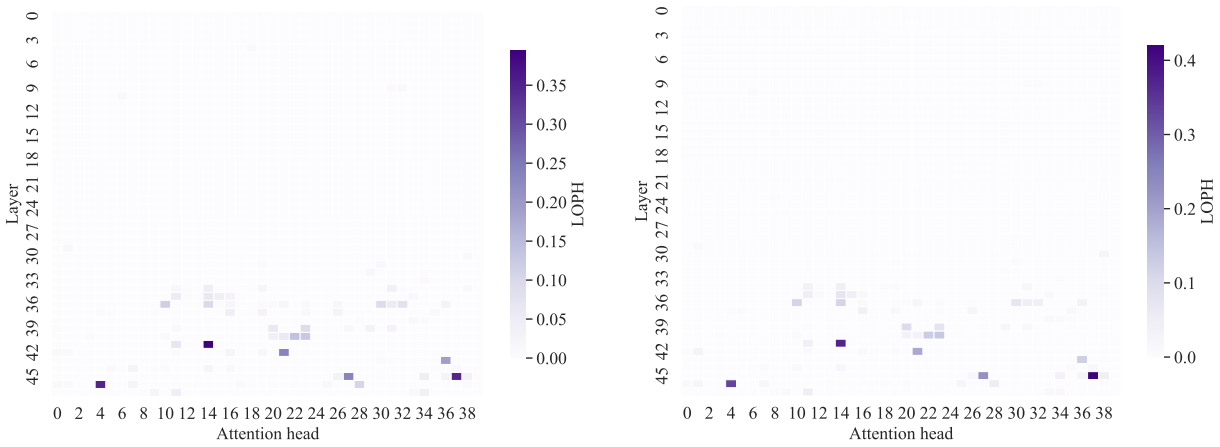
(a) ROME. (b) MEMIT.

Figure 24: Intervention effects following critical attention module ablation in Qwen2.5-14B-Instruct.



(a) ROME. (b) MEMIT.

Figure 25: LOPH of Qwen2.5-7B-Instruct edited by ROME and MEMIT.



(a) ROME. (b) MEMIT.

Figure 26: LOPH of Qwen2.5-14B-Instruct edited by ROME and MEMIT.

LOPH threshold	Models	Methods	Original			New		
			w/o abl.	abl.	$\downarrow \Delta P$	w/o abl.	abl.	$\uparrow \Delta P$
$\tau = 0.2$	LLaMA3-8B-Instruct	ROME	57.17	36.96	20.21	16.49	20.50	4.01
		MEMIT	56.90	38.72	18.18	15.68	18.80	3.12
	Qwen2.5-7B-Instruct	ROME	57.83	48.84	8.99	11.84	16.13	4.29
		MEMIT	57.54	41.94	15.60	12.21	19.69	7.48
	Qwen2.5-14B-Instruct	ROME	55.71	46.71	9.00	13.99	18.43	4.44
		MEMIT	55.03	46.87	8.16	13.79	17.40	3.61
$\tau = 0.3$	LLaMA3-8B-Instruct	ROME	57.17	37.58	19.59	16.49	20.90	4.41
		MEMIT	56.90	42.99	13.91	15.68	19.88	4.20
	Qwen2.5-7B-Instruct	ROME	57.83	48.84	8.99	11.84	16.13	4.29
		MEMIT	57.54	47.64	9.90	12.21	17.48	5.27
	Qwen2.5-14B-Instruct	ROME	55.71	50.15	5.56	13.99	16.39	2.40
		MEMIT	55.03	49.10	5.93	13.79	16.29	2.50

Table 12: Ablation effects of the prominent heads across different values of τ. (Original: original answer; New: new answer; w/o abl.: without ablation; abl.: with ablation; $\downarrow \Delta P$: probability decrease; $\uparrow \Delta P$: probability increase)

	Top-K	L23H27		L24H3		L27H20		L30H29		L31H6		L31H7	
		15%	20%	15%	20%	15%	20%	15%	20%	15%	20%	15%	20%
Original	5	12.12	12.12	65.91	67.42	75.76	76.52	16.67	16.67	77.27	78.03	73.48	74.24
	10	12.88	14.39	68.94	71.21	76.52	76.52	16.67	16.67	78.03	78.03	73.48	75.00
	15	14.39	15.15	70.45	71.21	76.52	76.52	16.67	16.67	78.03	78.79	73.48	75.00
New	5	0.76	0.00	5.30	6.06	5.30	3.79	1.52	2.27	2.27	1.52	3.03	3.79
	10	0.76	1.52	8.33	6.82	9.85	7.58	5.30	4.55	6.06	7.58	9.09	8.33
	15	0.76	2.27	9.09	7.58	12.12	11.36	6.82	9.09	8.33	9.85	9.09	12.12

Table 13: Decoding Success Rate (DSR) of the identified vectors across different heads in LLaMA3-8B-Instruct edited by ROME. $p\%$ is set to 15% and 20%.

	Top-K	L23H27		L24H3		L27H20		L31H6		L31H7	
		5%	10%	5%	10%	5%	10%	5%	10%	5%	10%
Original	5	5.45	8.18	30.00	35.45	36.36	47.27	37.27	47.27	37.27	42.73
	10	8.18	10.00	33.64	42.73	39.09	47.27	41.82	50.00	39.09	44.55
	15	9.09	10.91	38.18	42.73	40.00	49.09	43.64	51.82	40.91	44.55
New	5	0.91	0.91	0.00	0.00	1.82	0.91	2.73	3.64	3.64	2.73
	10	0.91	0.91	0.00	0.00	3.64	4.55	3.64	4.55	3.64	4.55
	15	0.91	0.91	0.00	0.00	5.45	6.36	3.64	7.27	3.64	6.36

Table 14: Decoding Success Rate (DSR) of different heads in LLaMA3-8B-Instruct edited by MEMIT. $p\%$ is set to 5% and 10%.

	Top-K	L23H27		L24H3		L27H20		L31H6		L31H7	
		15%	20%	15%	20%	15%	20%	15%	20%	15%	20%
Original	5	12.73	15.45	39.09	41.82	50.91	50.91	51.82	51.82	44.55	45.45
	10	16.36	18.18	43.64	44.55	50.91	50.91	51.82	51.82	44.55	45.45
	15	17.27	18.18	44.55	45.45	50.91	50.91	51.82	51.82	45.45	45.45
New	5	0.91	0.91	0.00	0.00	1.82	0.91	3.64	5.45	1.82	1.82
	10	1.82	2.73	0.91	1.82	6.36	3.64	9.09	10.00	5.45	5.45
	15	2.73	2.73	0.91	1.82	6.36	7.27	12.73	14.55	6.36	7.27

Table 15: Decoding Success Rate (DSR) of different heads in LLaMA3-8B-Instruct edited by MEMIT. $p\%$ is set to 15% and 20%.

	Top-K	L23H4		L23H6		L23H11		L26H0		L27H2		L27H3		L27H15	
		5%	10%	5%	10%	5%	10%	5%	10%	5%	10%	5%	10%	5%	10%
Original	5	25.17	34.69	34.01	49.66	28.57	59.86	8.16	17.01	21.77	43.54	33.33	41.50	3.40	17.01
	10	31.29	42.18	41.50	54.42	37.41	61.90	10.88	25.17	27.89	46.94	38.10	44.90	7.48	19.73
	15	36.05	43.54	43.54	55.78	40.82	62.59	12.93	27.21	34.69	47.62	38.78	46.94	10.20	21.77
New	5	0.68	1.36	1.36	1.36	0.00	0.00	0.00	0.00	0.00	0.00	1.36	0.00	0.00	0.00
	10	0.68	2.04	2.72	2.72	0.00	0.00	0.00	0.00	0.00	0.00	3.40	2.04	0.00	0.00
	15	0.68	3.40	4.76	4.76	0.00	0.00	0.00	0.00	0.00	0.00	4.08	2.04	0.00	0.68

Table 16: Decoding Success Rate (DSR) of different heads in Qwen2.5-7B-Instruct edited by ROME. $p\%$ is set to 5% and 10%.

	Top-K	L23H4		L23H6		L23H11		L26H0		L27H2		L27H3		L27H15	
		15%	20%	15%	20%	15%	20%	15%	20%	15%	20%	15%	20%	15%	20%
Original	5	43.54	49.66	57.82	62.59	66.67	70.75	30.61	31.97	48.30	52.38	45.58	48.30	20.41	25.85
	10	48.98	52.38	63.95	65.31	70.07	71.43	32.65	36.73	51.02	55.78	46.26	51.02	25.17	28.57
	15	52.38	55.10	67.35	69.39	70.75	72.11	36.05	37.41	53.06	56.46	49.66	52.38	27.89	32.65
New	5	1.36	0.68	4.76	4.08	0.00	0.00	0.00	0.00	0.00	0.00	1.36	0.68	0.68	0.68
	10	2.04	2.04	5.44	5.44	0.00	0.00	0.00	0.00	0.00	0.00	1.36	1.36	1.36	1.36
	15	3.40	3.40	6.12	7.48	0.00	0.00	0.00	0.00	0.00	0.00	2.04	3.40	1.36	1.36

Table 17: Decoding Success Rate (DSR) of different heads in Qwen2.5-7B-Instruct edited by ROME. $p\%$ is set to 15% and 20%.

	Top-K	L23H11		L24H23		L24H27		L26H0		L27H3		L27H15	
		5%	10%	5%	10%	5%	10%	5%	10%	5%	10%	5%	10%
Original	5	33.93	63.39	30.36	56.25	22.32	49.11	9.82	22.32	37.50	48.21	7.14	28.57
	10	42.86	67.86	40.18	63.39	30.36	58.93	14.29	25.00	41.96	50.00	16.96	31.25
	15	48.21	68.75	41.96	63.39	33.93	59.82	16.07	27.68	49.11	51.79	23.21	33.04
New	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.79	1.79	0.00	0.00
	10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.57	4.46	0.89	0.89
	15	0.00	0.00	0.00	0.00	0.00	0.00	0.89	0.00	5.36	6.25	0.89	0.89

Table 18: Decoding Success Rate (DSR) of different heads in Qwen2.5-7B-Instruct edited by MEMIT. $p\%$ is set to 5% and 10%.

	Top-K	L23H11		L24H23		L24H27		L26H0		L27H3		L27H15	
		15%	20%	15%	20%	15%	20%	15%	20%	15%	20%	15%	20%
Original	5	70.54	73.21	63.39	69.64	59.82	62.50	24.11	27.68	52.68	54.46	33.04	33.93
	10	73.21	75.00	66.96	71.43	64.29	70.54	29.46	30.36	54.46	54.46	33.93	38.39
	15	74.11	75.00	69.64	72.32	67.86	71.43	31.25	33.04	55.36	56.25	36.61	43.75
New	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.68	3.57	0.00	0.89
	10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.46	6.25	0.89	0.89
	15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.25	6.25	0.89	0.89

Table 19: Decoding Success Rate (DSR) of different heads in Qwen2.5-7B-Instruct edited by MEMIT. $p\%$ is set to 15% and 20%.

	Top-K	L36H10		L40H22		L40H23		L41H14		L42H21	
		5%	10%	5%	10%	5%	10%	5%	10%	5%	10%
Original	5	29.06	52.22	29.56	40.39	29.06	46.80	26.11	37.44	10.34	22.17
	10	35.96	60.10	36.45	45.32	34.97	50.74	33.00	45.81	15.27	28.57
	15	40.89	64.04	39.41	45.81	39.41	56.16	36.95	51.23	18.72	32.02
New	5	0.00	0.00	0.99	1.48	0.49	0.49	1.48	2.46	0.00	0.00
	10	0.00	0.00	2.46	2.96	0.49	0.49	2.46	4.43	0.00	0.99
	15	0.00	0.00	3.94	3.94	0.49	0.99	4.43	8.87	0.00	0.99
	Top-K	L43H36		L45H27		L45H37		L46H4		L46H28	
		5%	10%	5%	10%	5%	10%	5%	10%	5%	10%
Original	5	15.27	21.67	11.33	26.11	19.70	33.99	20.20	36.95	13.30	22.66
	10	17.24	25.12	17.24	29.56	24.63	37.44	26.11	40.89	14.78	29.56
	15	20.20	28.08	18.23	32.02	27.59	38.42	29.56	43.84	15.76	34.48
New	5	1.48	1.48	0.00	0.00	0.00	0.49	0.00	0.49	0.00	0.00
	10	1.97	1.48	0.00	0.49	0.49	0.49	0.49	0.49	0.00	0.49
	15	2.46	1.97	0.00	0.49	0.49	0.49	0.99	0.49	0.00	0.99

Table 20: Decoding Success Rate (DSR) of different heads in Qwen2.5-14B-Instruct edited by ROME. $p\%$ is set to 5% and 10%.

	Top-K	L36H10		L40H22		L40H23		L41H14		L42H21	
		15%	20%	15%	20%	15%	20%	15%	20%	15%	20%
Original	5	64.04	68.97	45.32	46.31	49.75	59.11	48.28	52.22	33.50	38.42
	10	67.49	70.44	47.29	49.75	56.16	62.07	55.17	60.10	40.39	44.33
	15	69.95	71.43	49.75	50.25	60.10	63.55	59.11	61.08	44.33	46.80
New	5	0.00	0.00	2.46	3.45	1.48	1.48	3.94	2.96	0.00	0.49
	10	0.00	0.00	5.91	5.91	1.97	2.46	4.43	5.42	1.48	0.49
	15	0.00	0.00	6.90	6.90	2.46	3.45	6.90	8.37	1.97	0.99
	Top-K	**L43H36**		**L45H27**		**L45H37**		**L46H4**		**L46H28**	
		15%	20%	15%	20%	15%	20%	15%	20%	15%	20%
Original	5	25.62	28.08	33.00	39.90	40.89	42.36	45.81	48.77	33.50	37.44
	10	29.56	29.56	39.41	43.35	41.87	43.35	47.78	53.20	36.95	41.87
	15	30.54	31.53	40.89	43.84	42.86	45.32	51.23	55.67	39.90	42.36
New	5	2.46	1.97	0.49	0.00	0.49	0.49	0.99	0.99	0.49	0.99
	10	2.96	2.96	0.49	0.00	0.49	0.49	0.99	1.48	1.48	1.48
	15	2.96	3.45	0.49	0.00	0.49	0.99	1.97	2.46	1.97	2.46

Table 21: Decoding Success Rate (DSR) of different heads in Qwen2.5-14B-Instruct edited by ROME. $p\%$ is set to 15% and 20%.

	Top-K	L36H14		L39H20		L40H22		L40H23		L41H14	
		5%	10%	5%	10%	5%	10%	5%	10%	5%	10%
Original	5	30.61	55.78	32.31	55.78	29.59	37.07	28.91	41.50	26.53	36.05
	10	37.07	58.84	42.52	58.16	34.35	41.50	34.69	46.60	34.69	42.86
	15	41.16	61.22	45.58	58.84	36.73	43.20	36.05	48.64	38.10	46.94
New	5	0.00	0.00	0.34	1.02	0.34	0.68	0.34	1.02	1.02	1.36
	10	0.00	0.00	1.70	1.36	1.02	1.70	0.34	1.36	2.72	3.74
	15	0.00	0.00	1.70	2.38	1.70	2.38	0.68	1.36	4.76	6.46
	Top-K	**L42H21**		**L43H36**		**L45H27**		**L45H37**		**L46H4**	
		5%	10%	5%	10%	5%	10%	5%	10%	5%	10%
Original	5	8.84	19.73	10.20	15.99	7.82	19.05	22.11	39.80	18.71	37.41
	10	11.22	26.19	11.90	18.03	10.88	22.79	26.87	43.20	24.49	40.48
	15	12.93	29.93	12.93	19.39	13.61	26.87	30.27	44.56	27.89	42.52
New	5	0.00	0.00	0.68	1.70	0.34	0.34	0.00	0.00	0.68	0.68
	10	0.00	0.00	1.36	2.04	0.34	0.34	0.00	0.00	1.02	1.36
	15	0.00	0.68	1.70	2.72	0.34	0.34	0.00	0.34	1.02	2.04

Table 22: Decoding Success Rate (DSR) of different heads in Qwen2.5-14B-Instruct edited by MEMIT. $p\%$ is set to 5% and 10%.

	Top-K	L36H14		L39H20		L40H22		L40H23		L41H14	
		15%	20%	15%	20%	15%	20%	15%	20%	15%	20%
Original	5	63.27	66.67	62.24	62.59	41.84	43.54	48.98	52.38	41.84	47.28
	10	65.31	68.03	63.27	63.95	44.56	47.62	52.72	55.44	48.64	52.72
	15	67.01	68.03	63.27	63.95	46.94	48.30	53.06	56.46	52.04	54.08
New	5	0.00	0.00	0.68	0.68	1.02	0.68	0.34	0.68	1.36	1.36
	10	0.00	0.34	1.02	1.02	2.04	3.74	1.36	1.70	4.08	6.12
	15	0.00	0.34	2.04	2.38	3.06	5.10	2.04	2.38	8.16	8.50
	Top-K	**L42H21**		**L43H36**		**L45H27**		**L45H37**		**L46H4**	
		15%	20%	15%	20%	15%	20%	15%	20%	15%	20%
Original	5	32.65	37.07	19.73	21.09	29.25	37.76	44.56	45.24	43.54	45.58
	10	39.12	43.88	21.43	23.13	35.37	41.50	46.60	46.60	46.60	49.66
	15	42.18	46.26	23.13	25.17	38.10	43.54	46.94	47.96	47.96	52.04
New	5	0.34	0.34	1.36	1.70	0.34	0.34	0.00	0.00	0.68	1.02
	10	0.34	0.34	2.38	3.40	0.34	0.68	0.00	0.34	1.36	1.70
	15	0.68	1.02	2.72	4.08	0.34	0.68	0.00	0.34	3.06	3.06

Table 23: Decoding Success Rate (DSR) of different heads in Qwen2.5-14B-Instruct edited by MEMIT. $p\%$ is set to 15% and 20%.

Models	Methods	15%						20%					
		OAP			NAP			OAP			NAP		
		w/o abl.	abl.	$\downarrow \Delta P$	w/o abl.	abl.	$\uparrow \Delta P$	w/o abl.	abl.	$\downarrow \Delta P$	w/o abl.	abl.	$\uparrow \Delta P$
LLaMA3-8B-Instruct	ROME	61.41	45.45	15.96	16.12	24.21	8.09	61.41	43.41	18.00	16.12	25.11	8.99
	MEMIT	57.42	41.37	16.05	17.05	24.68	7.63	57.42	39.30	18.12	17.05	25.84	8.79
Qwen2.5-7B-Instruct	ROME	64.33	47.52	16.81	11.93	21.26	9.33	64.33	45.46	18.87	11.93	22.26	10.33
	MEMIT	66.41	54.89	11.52	15.72	23.41	7.69	66.41	52.98	13.43	15.72	24.62	8.90
Qwen2.5-14B-Instruct	ROME	62.87	51.14	11.73	17.39	24.40	7.01	62.87	49.62	13.25	17.39	25.28	7.89
	MEMIT	62.02	48.85	13.17	15.64	23.35	7.71	62.02	47.06	14.96	15.64	24.41	8.77

Table 24: Answer probabilities before and after singular vector ablation across varying values of $p\%$.