One Mask to Rule Them All: On Hidden Facts after Editing and How to Find Them

Anonymous ACL submission

Abstract

Knowledge editing methods such as ROME and MEMIT update factual associations in transformer models by modifying MLP weights. While evaluated mainly by output behavior, their internal mechanism remains underexplored. We investigate whether edits rely on a common mechanism, regardless of which fact is modified. Despite fact-specific weight changes, we argue that ROME and MEMIT target the same subset of weights critical for maintaining edits. To isolate this subset, we train a compact binary mask (<10%) over the edited weights. The mask reverses 80% of edits on the training set and over 70% on the test set, confirming that diverse edits share a common functional structure. Our analysis reveals that the mask reverses edits by eliminating overattention in later layers. Additionally, we show that injecting the mask during editing drops editing success from 98% to 38%, demonstrating that this mechanism is necessary for edits to succeed. Our finding that edits suppress rather than overwrite knowledge explains why ROME and MEMIT fail to propagate changes to related facts. The identified common functional subspace informs detection and defense against unwanted edits.

1 Introduction

Knowledge Editing (KE) aims to update specific facts in transformer models without expensive retraining. Among KE methods, locate-and-edit approaches such as ROME (Meng et al., 2022) and MEMIT (Meng et al., 2023) have gained traction for their efficiency. These methods identify parameters associated with target facts and directly modify MLP weights to overwrite factual associations. These methods are motivated by the hypothesis that MLP layers act as associative memories (Geva et al., 2021), and that targeted updates can overwrite specific facts.

Editing success is evaluated solely based on whether the model outputs the new fact. Thus, the claim that knowledge is overwritten rests on this behavioral change alone. It remains unexplored, however, how edits affect the model's internal representations. Additionally, there is a conceptual puzzle: While transformers are said to retrieve knowledge through redundant pathways (McGrath et al., 2023; Hase et al., 2023), KE methods successfully update facts by modifying a single layer or a small range of connected layers. If factual knowledge is distributed, how can modifying a single layer or a small range of connected layers successfully override knowledge retrieval?

We hypothesize that ROME and MEMIT succeed not by overwriting original factual knowledge, but by suppressing it. We propose that beyond introducing fact-specific changes, these methods rely on a common subset of weights critical for maintaining any edit. By modifying these weights, ROME and MEMIT inject amplified signals that force the model to output edited facts and suppress the downstream propagation of original knowledge without erasing it. These amplified signals cause the later layers to attend disproportionately to the edited signal.

To verify this hypothesis, we isolate the subset of weights critical for maintaining edits by training a compact binary mask over the edited weight matrices. The mask identifies which weight changes are necessary for the edit to persist, allowing us to neutralize edits by pruning parts of the edited weights. If a compact mask suffices to remove the edit, this confirms that only a small subset of weights is critical for maintaining the edit. If the same mask removes diverse edits across different facts, this reveals a shared structure that all edits exploit. Conversely, if edits introduced only fact-specific changes, no single mask could generalize across semantically different edits.

We find that a single mask removing fewer than 10% of edited weights reverses over 80% of edits on the training set and over 70% on unseen edits.

This generalization confirms that diverse edits rely on a common functional structure. Analyzing what the mask targets, we find that it eliminates amplified attention signals in later layers while preserving MLP pathways, which continue encoding original knowledge. This reveals that edits succeed by hijacking attention rather than overwriting stored facts. Furthermore, injecting the mask prior to the editing process drops success from 98% to 38%, confirming that this mechanism is not just sufficient for reversal but necessary for edits to succeed. These findings suggest that ROME and MEMIT are fundamentally limited: they cannot truly overwrite knowledge, only suppress its retrieval. This explains their known failure to propagate changes to related facts (Hsueh et al., 2024) and creates pathways for detecting and defending against unwanted edits.

2 Related Work

Editing Methods. Knowledge editing methods (Wang et al., 2024c; Mazzia et al., 2024) fall into two categories: parameter-modifying and parameter-preserving. Parameter-modifying methods include locate-and-edit approaches such as ROME (Meng et al., 2022) and MEMIT (Meng et al., 2023), which locate and update parameters responsible for the facts, and meta-learning approaches such as MEND (Mitchell et al., 2022a) and MALMEN (Tan et al., 2023), which train hypernetworks to predict necessary parameter shifts to update facts. Parameter-preserving methods add memory modules (Mitchell et al., 2022b; Hartvigsen et al., 2023; Wang et al., 2024a; Guo et al., 2025) or leverage in-context learning (Zheng et al., 2023). In this work, we focus on locate-and-edit KEs such as ROME and MEMIT due to their wide usage, and to better understand how these KEs adapt LLMs.

Detecting and Reversing Knowledge Edits. Research on detecting and reversing edits has emerged in response to potential malicious uses. Work includes distinguishing edited from unedited facts via internal representations (Youssef et al., 2025b; Li et al., 2024), reversing in-context edits (Youssef et al., 2025a), and tracing ROME edits through weight analysis (Youssef et al., 2025c). We extend this line by identifying a minimal set of common weights that maintain edits across diverse facts.

Knowledge Representation in Transformers. Transformers exhibit emergent self-repair capabilities: when a layer is ablated, downstream layers compensate by increasing their contributions, partially restoring original outputs (McGrath et al., 2023). This redundancy implies that factual knowledge can be retrieved through multiple pathways. Work on knowledge-critical subnetworks (Bayazit et al., 2024) demonstrates that sparse subnetworks spanning multiple layers are responsible for maintaining specific factual associations. These findings create a paradox for locate-and-edit methods: if knowledge is distributed across redundant pathways, how can modifying a single layer or a small range of connected layers successfully override factual retrieval? We show that ROME and MEMIT resolve this by inducing overattention - a common mechanism that suppresses redundant pathways without erasing them.

Attention Phenomena in Knowledge Editing. Recent work documents attention-related phenomena in edited models. Wang et al. (2025) identify *attention drift*: excessive attention scores assigned to edited entities causing specificity failure, where edits corrupt unrelated knowledge. They propose to selectively constrain drifting attention heads via regularization during editing. Xie et al. (2025) analyze *superficial editing*: the tendency of edited models to revert to original knowledge under adversarial prompts. They identify two contributing factors: the residual stream at the last subject position in earlier layers, and specific attention heads in later layers. Both papers document how attention mechanisms relate to editing failures. Neither addresses what makes edits *succeed* in the first place: the structural mechanism that suppresses original knowledge during normal retrieval. Our work identifies this mechanism and shows it is shared across edits.

3 Method

Background. Let M be the original model. A factual prompt x consists of a subject s and relation r (e.g., "Marie Curie discovered"), with a corresponding object o (e.g., "radium"). A knowledge edit replaces the original object o with a new object o^* (e.g., "krypton"), producing an edited model M_e.

Hypothesis. We hypothesize that ROME and MEMIT rely on a common mechanism crucial to

inject and maintain edits. Specifically, beyond introducing fact-specific updates, these methods target the same subset of weights critical for the edit to persist.

Mask Training. To verify this hypothesis, we identify whether such a shared subset exists by training a binary mask $K = (k_{a,b})$, where $k_{a,b} \in \{0, 1\}$ for all a, b, over a set of edited weights $\hat{W}$ to remove the edits. If only a small subset of weights is critical for maintaining an edit, then selectively removing those weights should be sufficient to restore the original model's behavior. The mask is applied element-wise to the edited weight matrix $\hat{W}$, producing a pruned model M_p with weights $\hat{W} \odot K$, where $\odot$ denotes the element-wise product operation. A mask value of 1 retains the edited weight; a value of 0 removes its contribution. If the mask successfully reverses an edit, the pruned model M_p assigns a higher probability to the original object o than the edited object o^*.

We train a single mask K across diverse edits spanning semantically different facts (see Figure 1). If this single mask generalizes, reversing not only training edits but also unseen edits, this constitutes evidence for a shared mechanism, i.e., the same weight positions are responsible for the edit across different facts. Conversely, if each edit introduces fact-specific changes, a single mask cannot generalize well, i.e., different edits would modify different weight positions, and a mask trained on one set of facts would fail to reverse edits on new (unseen) facts.

Loss Function. We train the mask to satisfy three constraints. First, *restoration*: the pruned model M_p should prefer the original object o over the edited object o^*. This means that after applying the mask to the edited weights, the probability $P_{M_p}(o \mid x)$ should exceed $P_{M_p}(o^* \mid x)$. If the mask successfully restores the probabilities across diverse edits, it demonstrates that the original knowledge was never erased. Second, *minimality*: the mask should remove as few weights as possible. We want most mask values to be 1, pruning only the small subset of weights responsible for maintaining the edit. This constraint ensures that we identify the specific weights that enable the edit, rather than broadly disrupting the layer. Third, *behavior preservation*: the behavior of the pruned model should remain close to the original model. This ensures that the mask does not introduce arbitrary changes that happen to flip the prediction but also damage the model's general language capabilities.

We formalize each constraint as a loss term. The *restoration* loss measures whether the pruned model prefers the original object over the edited one:

$$\mathcal{L}_{\text{restoration}} = -[\log P_{M_p}(o \mid x) - \log P_{M_p}(o^* \mid x)] \tag{1}$$

This loss is negative when $P_{M_p}(o \mid x) > P_{M_p}(o^* \mid x)$, i.e., when the original fact is restored. We require $\mathcal{L}_{\text{restoration}} \leq -\delta$, where the margin δ encourages confident restoration rather than marginal preference.

The sparsity loss measures the fraction of weights pruned:

$$\mathcal{L}_{\text{sparsity}} = \frac{1}{|K|} \sum_{a=1}^{m} \sum_{b=1}^{n} 1 - k_{a,b} \tag{2}$$

where $k_{a,b} \in \{0, 1\}$ is the mask value at position (a, b). The constraint $\mathcal{L}_{\text{sparsity}} \leq S_{\text{max}}$ limits pruning to at most S_{max} of the layer's weights. The *behavior preservation* loss measures divergence between the output distributions of the original model M and the pruned model M_p using KL divergence. We combine these terms into a constrained optimization problem, minimizing the KL divergence subject to constraints on restoration and sparsity:

$$\begin{aligned} \min_{\theta} \beta\, \mathcal{L}_{\text{KL}}^{T} \text{ s.t. } \mathcal{L}_{\text{sparsity}} \leq S_{\text{max}} \\ \text{and } \mathcal{L}_{\text{restoration}} \leq 0 - \delta \end{aligned} \tag{3}$$

We convert these constraints into penalty terms, yielding the combined loss:

$$\begin{aligned} \mathcal{L}(\theta) = \beta \mathcal{L}_{\text{KL}}^{T} + \max(0, \mathcal{L}_{\text{sparsity}} - S_{\text{max}}) \\ + \max(0, \mathcal{L}_{\text{restoration}} - \delta) \end{aligned} \tag{4}$$

The penalty terms activate only when constraints are violated.

4 Experiments

Training Details. Since a binary mask itself is non-differentiable, we initialize trainable parameters Θ and apply a sigmoid function to obtain soft mask $K \in (0, 1)$ (Louizos et al., 2018; Maddison et al., 2017). At inference, we binarize the mask using a threshold $\gamma \in (0, 1)$.

As illustrated in Figure 1, we train the mask across multiple edits simultaneously. Each training sample corresponds to a different edit with its

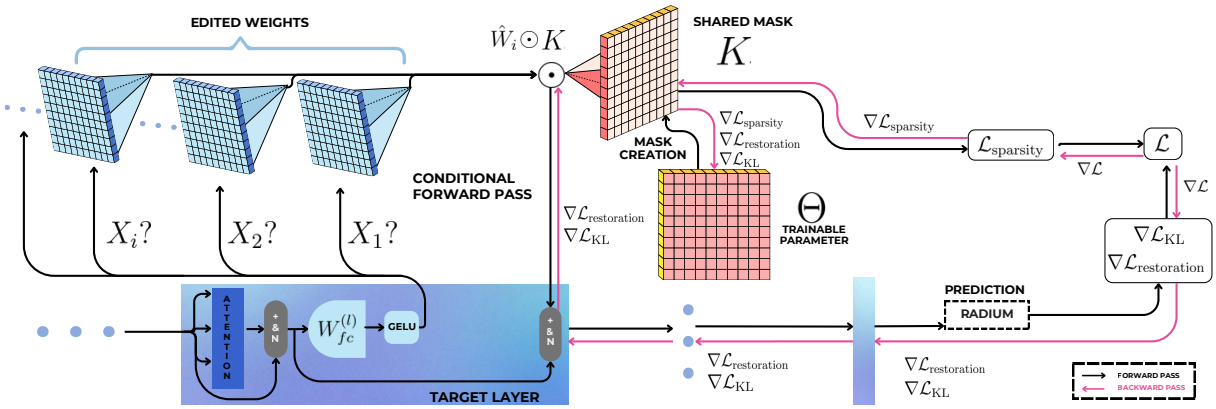

Figure 1: **Shared-mask training.** A single mask K is learned across batches of diverse edits to restore the original knowledge. During a conditional forward pass (black arrows), the mask is applied elementwise to the edited weights $\hat{W}_i$. Gradients (pink arrows) update only the mask parameters Θ to minimize the loss from Equation 4 while edited weights remain fixed.

own frozen edited weight matrix $\hat{W}_i$. During the forward pass, we swap in the appropriate $\hat{W}_i$ for each sample and apply the shared mask elementwise: $\hat{W}_i \odot K$. Gradients flow only through the mask parameters Θ, while all edited weights remain fixed. This conditional forward pass ensures the mask learns patterns shared across edits rather than memorizing a single edit's structure.

Setup. We use the CounterFact dataset introduced as part of the ROME study (Meng et al., 2022). CounterFact contains counterfactual triples (see Table 1 for examples). We train on 3,000 samples stratified across 10 relations and test on 1,700 held-out samples from the same relations. We use EasyEdit (Wang et al., 2024b) to edit models.

We evaluate on GPT-2 XL (1.5B), and LLaMA-3.2 (3B) (Dubey et al., 2024). For ROME, we train the mask across single edits. For MEMIT, which supports batch editing, we edit 1,000 facts simultaneously and train the mask on a single edited layer. For testing, we apply 1,000 different edits from the test set.

In our experiments, we aim to answer three questions: (a) Can a single mask reverse diverse edits? (b) Does the mask generalize to unseen edits from different facts? (c) Does applying the mask preserve general language modeling capabilities?

4.1 Evaluation Metrics

Reversal Success Rate (RSR). To assess whether original knowledge persists after editing, we measure if the pruned model prefers the original object over the edited one. We define $\Delta r_i := P_{M_p}(o_i \mid x_i) - P_{M_p}(o_i^* \mid x_i)$ and compute RSR as the proportion of samples where $\Delta r_i > 0$. High RSR indicates that original facts survive the edit and can be recovered.

Subject	Relation	Object (True → Edit)
Prydz Bay	is located in	Antarctica → Africa
Blowin' Up	premieres on	MTV → NBC

Table 1: Sample cases from CounterFact.

Top-1 Overlap (Top-1). While RSR measures relative preference, it does not guarantee exact behavioral restoration. We report Top-1 Overlap: the percentage of samples where the pruned model's top prediction matches the original model's. This verifies whether the model's generation path has been successfully reverted.

Perplexity (PPL). *Perplexity (PPL)* is one of the most widely used metrics for evaluating language models and is commonly applied to the WikiText datasets (Radford et al., 2019; Huang et al., 2024). We measure perplexity of M, M_e, and M_p on a random 70k-token subset of WikiText-2 (Merity et al., 2017) to assess whether reversals affect general language modeling capabilities.

4.2 Results

Reversal Performance. Table 2 shows that a single mask successfully reverses the majority of edits across both methods and architectures. For ROME on GPT-2 XL, pruning only 10% of the edited layer's weights achieves a Reversal Success Rate (RSR) of 83% on training set edits, with 78% Top-1

Method	Model	Pruned	Train		Test		PPL (Wikitext-2) ↓		
			RSR ↑	Top-1 ↑	RSR ↑	Top-1 ↑	M	M_e	M_p
ROME	GPT-2 XL	10.0%	83%	78%	82%	77%	17.80	$44.51_{\pm 6.8}$	$25.68_{\pm 0.4}$
	LLaMA-3 (3B)	10.0%	*90%*	*75%*	79%	72%	9.46	$9.59_{\pm 0.0}$	$10.14_{\pm 0.0}$
MEMIT	GPT-2 XL	4.5%	82%	81%	74%	78%	17.80	17.90	19.00
	LLaMA-3 (3B)	8.8%	87%	67%	78%	65%	9.46	10.78	12.53

Table 2: **Evaluation of edit reversal via the shared mask.** We report reversal performance (RSR and Top-1 Overlap) alongside model perplexity (PPL) for the original (M), edited (M_e), and pruned (M_p) models.

Overlap indicating that the pruned model's top prediction matches the original model in most cases. MEMIT requires even less intervention: pruning just 4.5% of the single edited layer's weights (0.9% of total edited weights) yields 82% RSR with 81% Top-1 Overlap.

Crucially, these masks generalize to unseen edits. On the held-out test set, the ROME mask maintains 82% RSR and 77% Top-1 Overlap; the MEMIT mask achieves 74% RSR with 78% Top-1 Overlap. The consistency across GPT-2 XL and LLaMA-3 suggests that the shared mechanism is a general property of ROME and MEMIT rather than an artifact of a specific architecture.

Perplexity After Pruning. Beyond reversal performance, we verify that our masks do not degrade general language modeling. Table 2 shows perplexity on WikiText-2 for the original (M), edited (M_e), and pruned (M_p) models.

ROME edits can substantially harm model performance. On GPT-2 XL, perplexity increases from 17.8 to 44.5 after editing yielding a 2.5× degradation. Applying the mask reduces perplexity to 25.7, recovering much of the lost performance without modifying the edited weights. In extreme cases, ROME edits cause model collapse (Yang et al., 2024) with a perplexity score exceeding one million. Our mask reduces these closer to the baseline levels (see Appendix B).

LLaMA-3 proves more robust to ROME edits, with perplexity rising only from 9.46 to 9.59. The mask introduces minimal additional degradation (10.14), confirming that our intervention is targeted rather than destructive.

MEMIT tells a different story. Its edits barely affect perplexity on both models (17.8 → 17.9 for GPT-2 XL; 9.46 → 10.78 for LLaMA-3), yet our mask still reverses over 80% of edits by pruning under 9% of weights without substantial damage to the performance. This indicates that MEMIT's edits, while less disruptive to general capabilities, still rely on a small subset of functionally critical weights to maintain the edit. Once these are suppressed, original knowledge resurfaces.

5 Analysis

The reversal results suggest that a single mask can reverse diverse edits across methods and architectures by targeting a shared mechanism anchored in the edited weights. In this section, we analyze *what constitutes this mechanism* by answering two questions: 1) How do ROME and MEMIT alter information flow within the model?; 2) What exactly does the trained mask target to reverse edits?

Edits induce overattention. Both ROME and MEMIT force edited models to assign dramatically higher probabilities to edited facts. As shown in Table 3, ROME on GPT-2 XL increases mean probability from 0.045 to 0.87 (Cohen's $d = 3.78$); LLaMA-3.2 shows the same pattern at lower absolute values ($d = 0.34$). These artificially elevated probabilities suggest that editing fundamentally alters information flow through the model. If knowledge can be retrieved through multiple pathways (McGrath et al., 2023; Hase et al., 2023), *how does editing succeed in producing such dominant output probabilities?*

Recent work has documented attention-related phenomena in edited models: excessive attention to edited entities (Wang et al., 2025) and later-layer attention modules that cause the residual stream to revert toward original knowledge (Xie et al., 2025). While previous work treats overattention and retention of original knowledge as side effects , we hypothesize that this **overattention is not merely a side effect but the mechanism by which edits succeed**: amplified signals hijack downstream attention, suppressing original facts without erasing them. To test this, we decompose the residual

Model	Method	M	M_e	p-value	Cohen's d
GPT-2 XL	ROME	$0.045_{\pm 0.11}$	$0.866_{\pm 0.19}$	2.9e-165	3.78
	MEMIT	$0.046_{\pm 0.11}$	$0.614_{\pm 0.36}$	3.5e-156	1.54
LLaMA-3 3B	ROME	$2.46\text{e-}5_{\pm 6\text{e-}5}$	$1.96\text{e-}4_{\pm 5\text{e-}4}$	1.6e-68	0.34
	MEMIT	$2.48\text{e-}5_{\pm 6\text{e-}5}$	$1.17\text{e-}4_{\pm 3\text{e-}4}$	1.3e-73	0.35

Table 3: **Statistical analysis of probabilities strength on 1,000 samples across models**. We compare the mean output probability of the original model (M) on original facts versus the edited model (M_e) on edited facts.

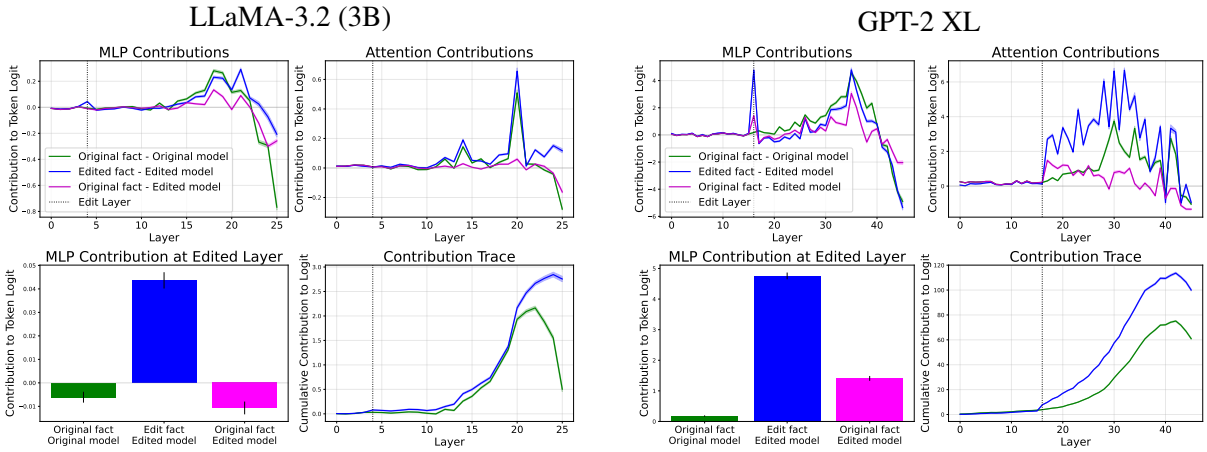

Figure 2: **Decomposition of residual stream** (mean and standard error over 1,000 samples) across LLaMA-3.2 (3B) and GPT-2 XL. For each model: Top Left: MLP contributions per layer. Top Right: Attention contributions per layer. Bottom Left: Comparison of MLP contributions at the edited layer. Bottom Right: Overall logit trace across layers. Edits amplify signals at the edited layer, causing downstream attention spikes while MLP pathways continue to follow broadly similar trajectories for original and edited facts.

stream into MLP and attention contributions using the Logit Lens method (nostalgebraist, 2020), measuring each component's contribution to the target token's logit (Figure 2).

The two architectures differ in magnitude but share the same functional pattern. In GPT-2 XL, ROME injects a signal $35\times$ larger than baseline at the edit layer (0.13 vs 4.55), producing immediate attention spikes downstream. LLaMA-3.2 shows modest amplification at the edit layer, but attention contributions increase sharply in later layers (19-21) causing a delayed overattention effect. The critical observation is that *MLP contributions beyond the edit layer follow broadly similar trajectories for edited and original facts* in both architectures. Downstream MLPs continue processing original knowledge; the main divergence occurs in the attention layers. The cumulative logit traces confirm this shared mechanism. In both models, edited facts accumulate substantially higher contributions than original facts, with the gap widening through downstream layers. MEMIT exhibits the same pattern despite distributing edits across layers (Appendix C).

These observations indicate that overattention is central to how edits produce dominant outputs. However, a key question remains: do different edits induce overattention through independent pathways, or do they share a common mechanism anchored in the edited weights? Is amplified attention a side-effect of editing or its core mechanism?

Mask eliminates overattention. The mask's high reversal performance on unseen edits (cf. Table 2) suggests that a shared mechanism critical to maintaining edits exists. To identify what the mask targets, we decompose the residual stream of the pruned model (Figure 3) and compare it with the edited and original models from Figure 2.

In GPT-2 XL, the MLP spike at the edit layer is eliminated. Critically, attention contributions in downstream layers are substantially reduced, while MLP contributions remain largely unaffected, following trajectories similar to the original model.

LLaMA-3.2 shows the same pattern. The late-layer attention spikes (layers 19-21) that dominate in the edited model are fully eliminated in the pruned model, and the subsequent MLP spike (layers 20-22) disappears along with it. This confirms

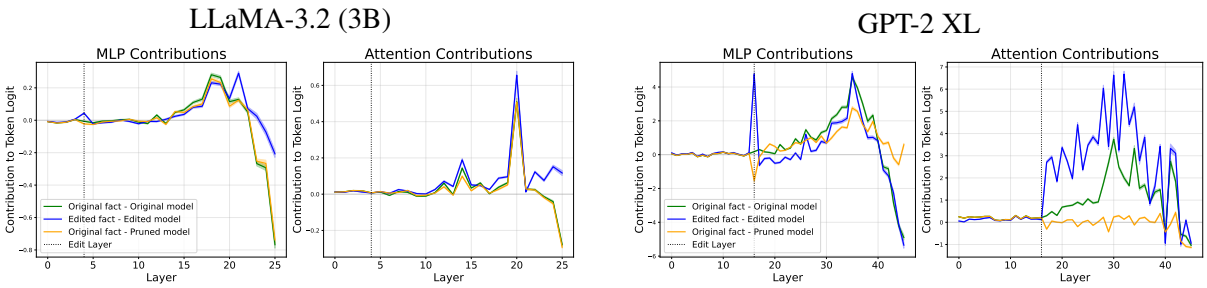

Figure 3: **Decomposition of residual stream for original, edited and pruned models** (mean and standard error over 1,000 samples) across both models.

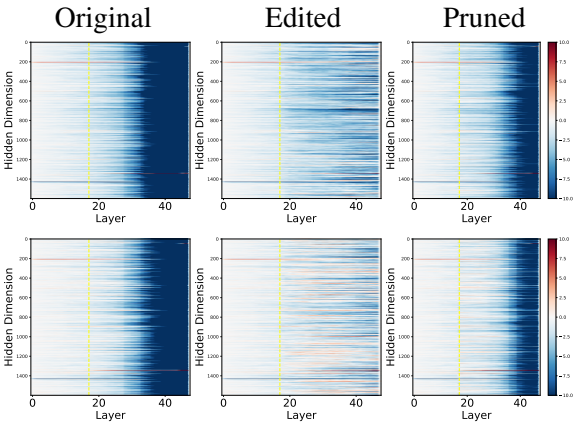

Figure 4: **Activations across hidden dimensions and layers.** Residual stream activations for original (left), edited (middle), and pruned (right) GPT-2 XL models. Yellow dashed line marks the edited layer.

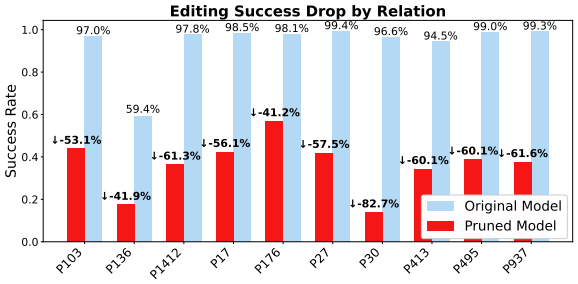

Figure 5: **Editing Success Drop.** Comparison of standard ROME editing efficacy (blue bars) versus editing the model while the shared mask is active (red bars).

that the late MLP activity was a consequence of overattention.

In both architectures, the masks trained on semantically diverse edits converge to a common solution: eliminating overattention while preserving MLP pathways. The mask trained to restore original knowledge independently discovers that targeting overattention is necessary and sufficient for reversal. The fact that the same weight positions are masked across different edits suggests a shared update pattern in how ROME and MEMIT induce overattention. This convergence is direct evidence that overattention is not a side effect but the shared structural mechanism that ROME and MEMIT exploit.

Additionally, Figure 4 illustrates this restoration: the pruned model recovers activation patterns closely resembling the original, confirming that the mask reverses the edit's effect on the residual stream. Further analysis of mask structure and other pruning experiments can be found in Appendix D.

Mask blocks new edits. The mask's ability to reverse edits demonstrates that targeting the shared subspace is sufficient for reversal. But is this subspace also *necessary* for edits to succeed or can it be avoided by taking other computational pathways inside the model? To test this, we inject the learned mask into the forward pass *during* the editing process itself.

If the subspace targeted by the mask is merely associated with successful edits but not required for them, editing should succeed through alternative pathways. Instead, editing success rate drops from 98% to 38% (Figure 5). This drop is consistent across relation types, indicating that the mask does not target relation-specific structure but a general mechanism. This result confirms that the weight subspace identified by our mask is not incidental to editing – it is structurally necessary. ROME struggles to bypass this shared mechanism to inject new facts.

6 Discussion

Our findings have several implications for understanding knowledge editing and for AI safety.

Overattention as mechanism. Prior work documents overattention in edited models as a problem:

a source of specificity failure where edits affect unrelated knowledge (Wang et al., 2025), or a reason an edited model reverts to original facts under adversarial prompts (Xie et al., 2025). Our findings reframe overattention as the core mechanism that makes editing work. A single mask recovers over 80% of training edits and 70% of unseen edits (Table 2) by targeting the same weight positions across semantically diverse facts. This would be impossible if edits introduced only fact-specific updates without a shared mechanism critical for maintaining them. The mask converges on eliminating overattention – the shared target that sustains edits across diverse facts. Moreover, injecting the mask during the editing process confirms that ROME cannot route around the blocked weights. This demonstrates that overattention is not a side effect but the structural basis ROME and MEMIT exploit.

Knowledge is not erased. On a conceptual level, these findings challenge the current knowledge editing paradigm. Knowledge is densely interconnected: a single fact relates to thousands of others. What does modifying a factual association actually mean? In a rigorous sense, it should keep the model's knowledge base consistent: modifying one fact should propagate to related associations, spanning a coherent counterfactual world. If we change the capital of France from Paris to Lyon, the model should reflect an alternative reality where this is true across all related queries. Current editing methods fail to achieve this. They do not propagate edits to related knowledge, creating ripple effects (Li et al., 2023; Cohen et al., 2024; Qin et al., 2024; Hsueh et al., 2024) that distort both related and unrelated facts. Our findings suggest why: ROME and MEMIT do not modify the knowledge graph - they hijack attention to suppress its retrieval. The original facts remain encoded; overattention simply prevents them from surfacing. This explains both why edits fail to propagate and why they can be reversed by targeting a small shared subspace.

Fundamental limitations of locate-and-edit methods. If ROME and MEMIT succeed by hijacking attention rather than modifying stored knowledge, this points to a fundamental limitation of the ROME/MEMIT paradigm that may extend to other locate-and-edit methods. The associative memory hypothesis (Geva et al., 2021) motivating ROME and MEMIT may be incomplete: even if facts are stored in MLP layers, retrieval involves attention, and current methods exploit this dependency rather than updating the stored associations. A shared functional subspace - one that generalizes across semantically diverse edits - may suggest an upper bound on what locate-and-edit methods can achieve. They may be inherently limited to suppressing knowledge rather than modifying it.

Implications for AI safety. On the defensive side, edits are more reversible than previously assumed: a sparse mask trained on a small set of edits can recover original knowledge across unseen facts. The same mask can also lock the model against future edits, as our blocking experiment demonstrates. Prior work has shown that edited facts are detectable from internal representations (Youssef et al., 2025b,c). Our findings may explain why: if all edits exploit a shared mechanism that produces distinctive attention patterns, they leave a common signature that classifiers can learn to recognize. Overall, understanding this shared mechanism brings us closer to defending against unwanted edits, by both locking models against future interventions or reversing the existing edits.

7 Conclusion

We demonstrated that ROME and MEMIT exploit a shared mechanism to maintain edits across semantically diverse facts. A single sparse mask generalizes to unseen edits with over 70% success, revealing that majority of edits rely on the same functional structure. Residual stream decomposition shows that the mask converges on eliminating overattention while preserving MLP pathways that continue to encode original knowledge. Injecting the mask during editing drops success from 98% to 38%, confirming that this mechanism is not merely sufficient for reversal but necessary for edits to succeed.

These findings highlight a fundamental limitation of locate-and-edit methods: rather than modifying stored knowledge, ROME and MEMIT act by suppressing its retrieval. Our analysis provides a foundation for both detecting and defending against unwanted edits. The same mask that reverses edits can also block future editing attempts, suggesting practical applications for locking models against malicious editing attacks. Future work should extend this analysis to other locate-and-edit methods such as AlphaEdit (Fang et al., 2025) and meta-learning approaches (Mitchell et al., 2022a; Tan et al., 2023).

Limitations

In this work, we focused on locate-and-edit KEs like ROME and MEMIT because of their widespread use and computational efficiency. Meta-learning KEs also adapt the model's parameters, and might be changing facts retrieval in LLMs in a different way. We did not consider these methods in our work because of the high computational costs associated with training editing hypernetworks.

References

Deniz Bayazit, Negar Foroutan, Zeming Chen, Gail Weiss, and Antoine Bosselut. 2024. Discovering knowledge-critical subnetworks in pretrained language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6549–6583, Miami, Florida, USA. Association for Computational Linguistics.

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.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 82 others. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2025. Alphaedit: Null-space constrained model editing for language models. In *The Thirteenth International Conference on Learning Representations*.

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.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1321–1330. JMLR.org.

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

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with grace: Lifelong model editing with discrete key-value adaptors. In *Advances in Neural Information Processing Systems*.

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. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Cheng-Hsun Hsueh, Paul Kuo-Ming Huang, Tzu-Han Lin, Che Wei Liao, Hung-Chieh Fang, Chao-Wei Huang, and Yun-Nung Chen. 2024. Editing the mind of giants: An in-depth exploration of pitfalls of knowledge editing in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9417–9429, Miami, Florida, USA. Association for Computational Linguistics.

Wei Huang, Xingyu Zheng, Xudong Ma, Haotong Qin, Chengtao Lv, Hong Chen, Jie Luo, Xiaojuan Qi, Xianglong Liu, and Michele Magno. 2024. An empirical study of llama3 quantization: from llms to mllms. *Visual Intelligence*, 2.

Yongchang Li, Yujin Zhu, Tao Yan, Shijian Fan, Gang Wu, and Liang Xu. 2024. Knowledge editing for large language model with knowledge neuronal ensemble. *Preprint*, arXiv:2412.20637.

Zichao Li, Ines Arous, Siva Reddy, and Jackie Cheung. 2023. Evaluating dependencies in fact editing for language models: Specificity and implication awareness. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7623–7636, Singapore. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Christos Louizos, Max Welling, and Diederik P. Kingma. 2018. Learning sparse neural networks through l0 regularization. In *International Conference on Learning Representations*.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*.

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

Thomas McGrath, Matthew Rahtz, Janos Kramar, Vladimir Mikulik, and Shane Legg. 2023. The hydra effect: Emergent self-repair in language model computations. *Preprint*, arXiv:2307.15771.

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

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*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

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

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

nostalgebraist. 2020. Interpreting gpt: the logit lens. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens. LessWrong. Accessed: 2025-12-18.

Jiaxin Qin, Zixuan Zhang, Chi Han, Pengfei Yu, Manling Li, and Heng Ji. 2024. Why does new knowledge create messy ripple effects in LLMs? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12602–12609, Miami, Florida, USA. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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

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

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. 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.

Pinzheng Wang, Zecheng Tang, Keyan Zhou, Juntao Li, Qiaoming Zhu, and Min Zhang. 2025. Revealing and mitigating over-attention in knowledge editing. In *The Thirteenth International Conference on Learning Representations*.

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

Jiakuan Xie, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2025. Revealing the deceptiveness of knowledge editing: A mechanistic analysis of superficial editing. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17756–17780, Vienna, Austria. 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 and virtual meeting. Association for Computational Linguistics.

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

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

Paul Youssef, Zhixue Zhao, Christin Seifert, and Jörg Schlötterer. 2025c. Tracing and reversing rank-one model edits. In *ICML 2025 Workshop on Machine Unlearning for Generative AI*.

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

A Training Details

We train a single shared mask K across all edits using gradient-based optimization. The mask is parameterized by learnable parameters $\Theta \in \mathbb{R}^{(m \times n)}$,

Statistic	Train	Test
Facts	3,000	1,700
Relations	10	10
Unique objects (o_{true})	236	204
Unique subjects	2,986	1,697
Unique mappings $o_{\text{true}} \rightarrow o^*$	1,284	866

Table 4: Dataset statistics for ROME experiments using CounterFact samples.

Statistic	Train	Test
Facts	1,000	1,000
Relations	10	10
Unique objects (o_{true})	174	172
Unique subjects	1,000	1,000
Unique mappings $o_{\text{true}} \rightarrow o^*$	564	580

Table 5: Dataset statistics for MEMIT experiments using CounterFact samples.

with the soft mask computed as $K = \sigma(\frac{\Theta}{\tau})$ (Guo et al., 2017), where σ is the sigmoid function and τ controls the sharpness of binarization.

We initialize $\Theta \sim N(0.85, 0.1)$, biasing the mask toward retaining weights initially. The temperature parameter τ starts at 6.0 and decays with rate 3.0 over training to encourage binary mask values. We use AdamW (Loshchilov and Hutter, 2017) with learning rate $1e-3$ and $\beta = (0.9, 0.999)$ for 300 epochs. At inference, we binarize the mask using threshold $\gamma = 0.7$ for GPT-2 XL and 0.9 for LLaMa-3. The thresholds are determined as a trade-off of sparsity and the reversal success rate (RSR) for a specific model.

Loss hyperparameters. For the restoration loss, we set the margin $\delta = 3.0$. For the KL divergence term, we use $\beta_{KL} = 3.26$ with temperature annealing from $T = 1.64$ to $T_{max} = 4.30$ following a linear schedule. The sparsity constraint is set to $S_{max} = 0.10$ for both model architectures and editing methods.

Dataset Statistics. We use the CounterFact dataset (Meng et al., 2022) for training and evaluating the shared mask. Tables 4 and 5 summarize the statistics for ROME and MEMIT experiments respectively. For ROME, we train on 3,000 single-edit samples and evaluate on 1,700 held-out samples, both stratified across 10 relation types. For MEMIT, which supports batch editing, we use 1,000 samples for training and 1,000 for testing. In the MEMIT setting, all facts within each split are edited simultaneously as a single batch edit, and the mask is trained on the last edited layer.

Computational Resources. All experiments were conducted on an HPC cluster using NVIDIA A100 (80GB) GPUs. Mask training for ROME required approximately 40 GPU hours, while MEMIT mask training required approximately 15 GPU hours.

B Mask Results

B.1 ROME

Figure 6 presents detailed reversal performance for ROME edits on GPT-2 XL. The Reversal Success Rate (RSR) remains consistently high across all relation types, ranging from approximately 75% to 90%, demonstrating that the learned mask generalizes across semantically diverse facts. Top-1 Overlap follows a similar pattern, confirming that the pruned model not only prefers the original fact but also recovers the exact prediction behavior of the unedited model in most cases.

The perplexity analysis reveals that ROME edits substantially degrade language modeling capabilities, with mean perplexity increasing from 17.80 to 44.51. Applying the shared mask reduces perplexity to 25.68, recovering much of the lost performance. The KL-divergence distribution further confirms that the pruned model's output distribution is substantially closer to the original model than the edited model, with the majority of samples showing lower divergence after mask application.

In more extreme cases, ROME edits can cause model collapse (Yang et al., 2024), leading to perplexity spikes ranging from hundreds to tens of millions. As illustrated in Table 6, our learned mask is able to substantially recover performance in these scenarios, reducing perplexity by several orders of magnitude without modifying the edited weights themselves.

B.2 MEMIT

Figure 7 presents the corresponding analysis for MEMIT edits. Despite MEMIT distributing edits across multiple connected layers, the shared mask trained only on the last edited layer achieves comparable reversal performance to ROME. RSR remains above 70% for most relation types, with Top-1 Overlap showing similar consistency.

A notable difference from ROME is that

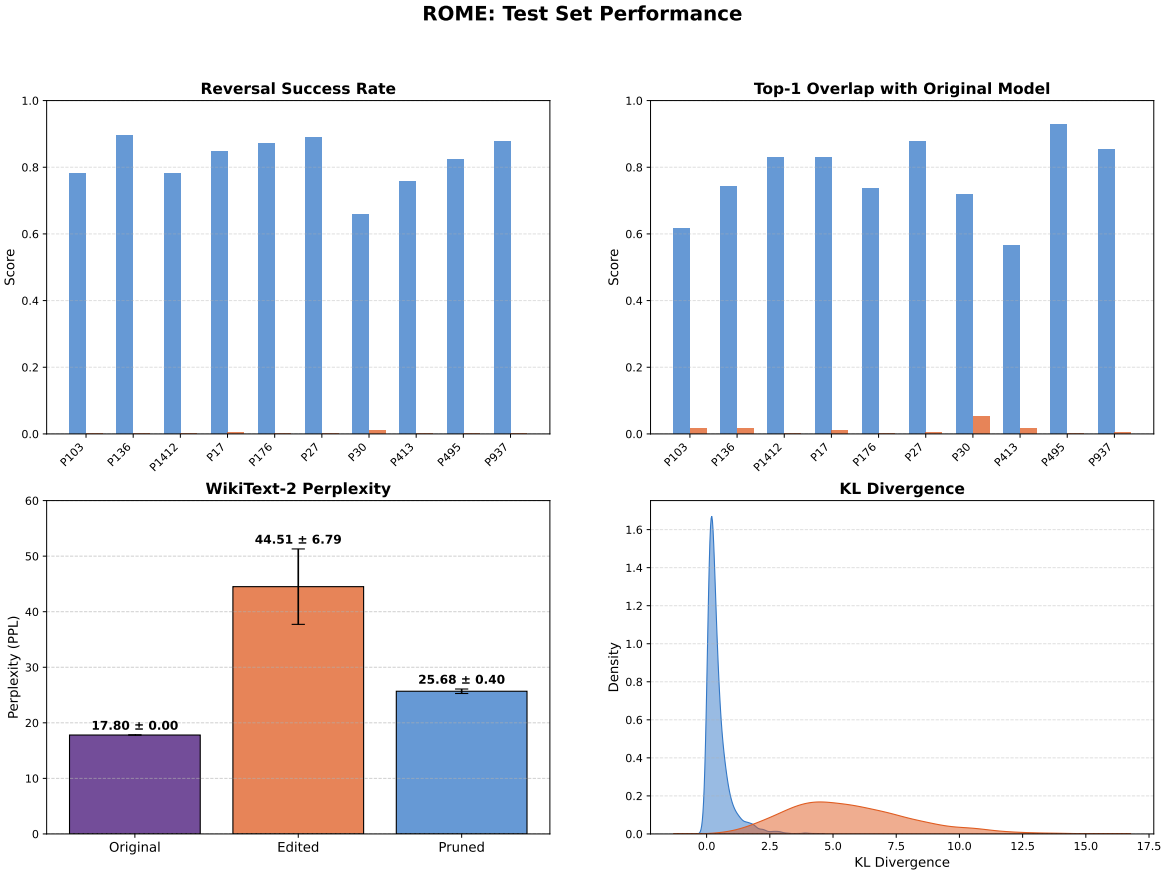

Figure 6: **ROME's Reversal performance.** First Row: We report the Reversal Success Rate (left) and Top-1 Overlap with the original model (right) across different relation types (e.g., P103, P136). Second Row: The leftmost plot compares WikiText-2 Perplexity, demonstrating that the pruned model (M_p) significantly reduces the perplexity degradation caused by the edit (M_e), recovering capabilities closer to the original model (M). On the right, we report the KL-divergence between 2 pairs of model states: 1) the original M and the edited M_e (orange); 2) the original M and the pruned M_p (blue), showing that pruned model is closer to the original model than the edited one.

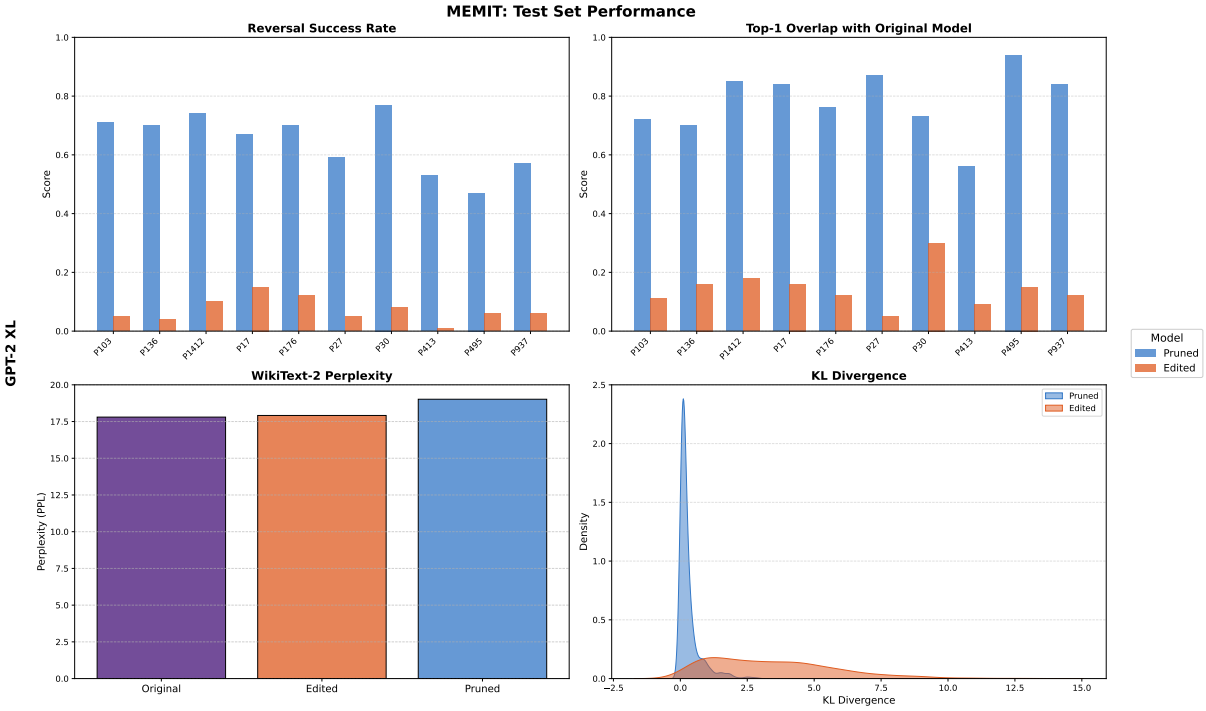

Figure 7: **MEMIT's Reversal performance.** First Row: Reversal Success Rate (left) and Top-1 Overlap (right) across relation types, showing consistent reversal performance despite MEMIT's multi-layer editing. Second Row: WikiText-2 Perplexity (left) remains close to the original after both editing and pruning, indicating MEMIT causes less collateral damage than ROME. KL-divergence (right) confirms the pruned model distribution is closer to the original than the edited model.

Case ID	PPL (M_e)	PPL (M_p)
3877	21,840,210.0	55.0
13259	1,195.8	32.0
16110	1,091.5	31.5
102	969.1	34.3
20421	748.3	35.0

Table 6: **Examples of perplexity recovery.** Selected cases where the initial edit (M_e) caused catastrophic perplexity spikes, which were significantly repaired by the shared mask (M_p).

MEMIT edits cause minimal perplexity degradation: WikiText-2 perplexity increases only marginally from 17.80 to 17.90 on GPT-2 XL. This suggests that MEMIT's distributed editing strategy is less disruptive to general language modeling capabilities. However, the KL-divergence analysis reveals that despite this stability, MEMIT edits still shift the model's output distribution away from the original, and the mask successfully reduces this divergence.

C MEMIT Analysis: Decompostion of Residual Stream

We extend the residual stream analysis from Section 5 to MEMIT edits. While ROME modifies a single layer, MEMIT distributes edits across multiple consecutive layers. This raises the question: does MEMIT exploit the same overattention mechanism as ROME, or does its distributed editing strategy produce fundamentally different internal dynamics?

C.1 Edits Induce Overattention

Figure 8 presents the residual stream decomposition for MEMIT edits. Despite the distributed nature of MEMIT, we observe a similar overattention pattern identified for ROME.

GPT-2 XL. The MLP contributions across the edited layers (13–17) show modest amplification compared to the original model, with the edited model giving higher contributions for edited (blue) and original (purple) facts than the original fact in the unedited model (green). However, the dominant effect emerges in the attention contributions: a sharp spikes appear in the downstream layers, sub-

stantially exceeding the original model's attention pattern. The cumulative contribution trace confirms this: edited facts accumulate dramatically higher logit contributions, with the gap widening primarily through downstream attention layers.

LLaMA-3.2 (3B). The pattern differs in timing but not in mechanism. MLP contributions across the edited layers (4–8) show minimal differentiation between conditions. Instead, attention contributions exhibit a pronounced spike in later layers (17–22), consistent with ROME's delayed overattention effect on this architecture. The cumulative trace shows edited facts reaching substantially higher final contributions.

Key difference from ROME. Unlike ROME, which injects a large signal at a single layer (35× amplification in GPT-2 XL), MEMIT's distributed edits produce smaller per-layer perturbations. However, these perturbations compound through downstream attention, ultimately producing comparable overattention effects.

C.2 Mask Eliminates Overattention

Figure 9 shows the effect of applying the learned mask to MEMIT-edited models. The mask, trained only on a single edited layer, successfully eliminates the overattention pattern while preserving MLP dynamics.

GPT-2 XL. The attention spike in layers 25–35 is substantially reduced in the pruned model (red), returning toward the original model's trajectory (green). Critically, MLP contributions remain largely unaffected by the mask, following similar paths for both the original and pruned models. The key difference is that mask significantly reduces the contribution in the edited MLP block, counterbalancing the effects of editing.

LLaMA-3.2 (3B). The late-layer attention spike (layers 17–22) is eliminated in the pruned model. The MLP contributions show close alignment between the original and pruned models throughout all layers, including those beyond the edit site.

Consistency with ROME. The mask's effect on MEMIT mirrors its effect on ROME: eliminating downstream overattention while preserving MLP pathways. This convergence provides strong evidence that **ROME and MEMIT exploit the same shared mechanism**, and that this mechanism can be targeted by a single sparse mask regardless of whether edits are concentrated in one layer or distributed across several.

D Mask analysis and Pruning

D.1 Mask Analysis

To understand how the mask reverses edits, we analyze which components it targets. Figure 10 visualizes the learned mask as a heatmap over the edited weight matrix. The mask does not uniformly prune weights or remove entire neurons; instead, it exhibits a structured sparsity pattern, concentrating on specific output dimensions (columns) of the MLP weight matrix while leaving others largely intact.

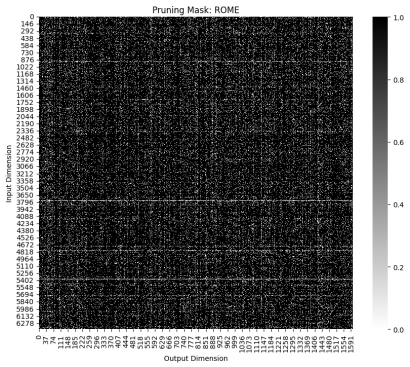

Figure 10: **Learned mask structure.** Heatmap of the binary mask over the edited MLP weight matrix in GPT-2 XL. White denotes pruned weights (mask value 0), black denotes retained weights (mask value 1). The mask exhibits column-wise sparsity to a limited extent, targeting specific output dimensions.

Figure 11 quantifies this column-wise concentration. The distribution is highly skewed: the majority of output dimensions have only a small subset of weights pruned, while some dimensions are heavily targeted. The top-5 most pruned dimensions have over 40% of their weights pruned, with dimension 214 reaching 73.1%. This concentrated pruning pattern suggests the mask identifies specific functional pathways critical for maintaining edits.

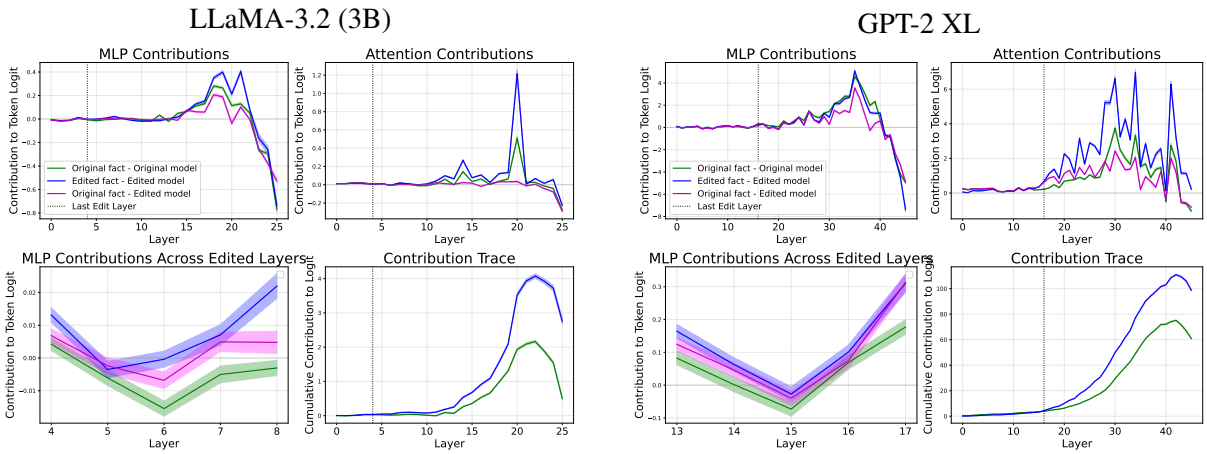

Figure 8: **Residual stream decomposition for MEMIT edits.** Top row: MLP (left) and attention (right) contributions per layer. Bottom row: MLP contributions across the edited layers only (left) and cumulative logit contribution trace (right). Green: original fact in original model; blue: edited fact in edited model; purple: original fact in edited model. Both architectures show amplified attention contributions in downstream layers, consistent with the overattention mechanism identified for ROME.

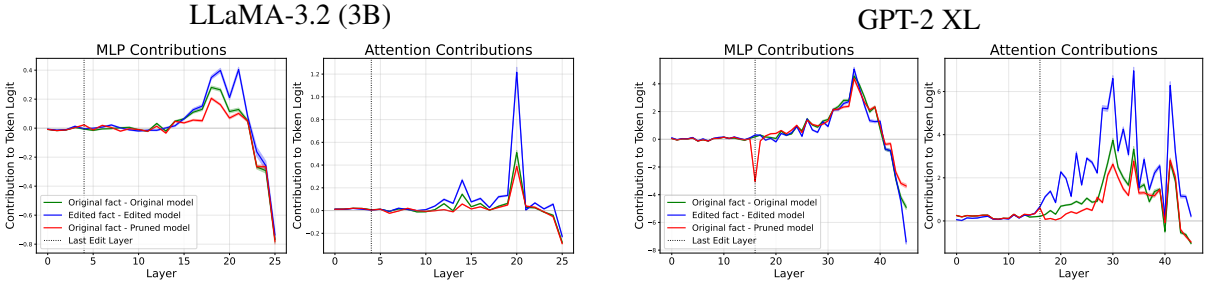

Figure 9: **Effect of the learned mask on MEMIT edits.** MLP contributions (left) and attention contributions (right) for the original model (green), edited model (blue), and pruned model (red). The mask eliminates the amplified attention contributions visible in Figure 8 while preserving MLP trajectories close to the original model.

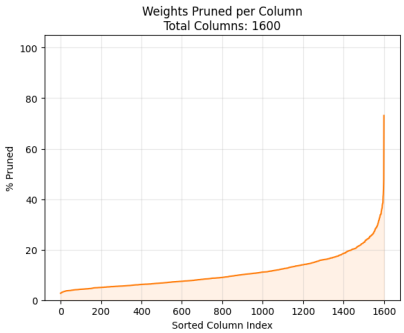

Figure 11: **Distribution of pruned weights per output dimension.** Each point represents one of the 1,600 output dimensions (columns) of the edited MLP weight matrix, sorted by pruning percentage. The steep rise on the right indicates that pruning is concentrated in a small subset of dimensions, with most dimensions retaining over 90% of their weights.

Notably, only 18% of pruned weights correspond to high-magnitude ROME updates, suggesting the mask targets functional pathways rather than simply reversing the largest weight changes.

Tracking pruned dimensions across layers. To verify that the heavily pruned dimensions are indeed functionally relevant, we track their activation trajectories across layers in the original, edited, and pruned models (Figure 12).

Across all five dimensions, the edited model (blue) diverges substantially from the original GPT2-XL trajectory (green) after the edit layer. Dimension 214, the most heavily pruned (73.1%), shows the clearest effect: its activation is suppressed by the edited model. The remaining dimensions (506, 1134, 292, 572) each show distinct divergence patterns where the edited model deviates from the original baseline.

The same pattern holds for LLaMA-3.2. Dimension 1659, the most heavily pruned (62.0%), shows clear divergence after the edit layer, with the edited model suppressing its activation relative to the original. The remaining dimensions (2205, 1480, 1085, 2825) similarly exhibit distinct divergence patterns, confirming that the mask targets functionally relevant pathways across architectures.

In all cases, applying the mask (red dashed) restores the activation trajectories toward the original model. This consistent restoration across the most heavily pruned dimensions in both GPT-2 XL and LLaMA-3 confirms that the mask precisely targets the activation changes induced by editing, reversing their effect on downstream computation rather than introducing arbitrary perturbations.

D.2 Comparison with Traditional Pruning Methods

To contextualize our learned mask approach, we compare against traditional pruning methods. We evaluate four pruning criteria: (1) unstructured magnitude pruning on the weight difference $\Delta W = W - \hat{W}$, (2) unstructured magnitude pruning on the edited weights $\hat{W}$ directly, (3) structured magnitude pruning based on column norms, and (4) structured activation-based pruning using average W_{fc} activations. Each criterion is tested in two modes: *zero* (pruned weights set to 0) and *original* (pruned weights restored to pre-edit values).

Figure 13 shows the Reversal Success Rate (RSR) as a function of pruning percentage. The results reveal a clear hierarchy among methods. Unstructured ΔW pruning is most efficient: zeroing only 30% of the weights with the largest update magnitude achieves over 90% RSR on both architectures. Unstructured magnitude pruning on $\hat{W}$ requires substantially more intervention (approximately 40% for GPT-2 XL and 50% for LLaMA-3 (3B)) and shows a notable asymmetry between modes: zero mode is far more effective than restoring original values.

Structured pruning methods perform considerably worse. Both structured magnitude and activation-based criteria show a near-linear relationship between pruning rate and RSR, requiring 90–100% of weights to be removed before reaching 90% reversal. This indicates that the edit's effect cannot be attributed to a small number of neurons; rather, it is distributed across the weight matrix in a way that structured approaches cannot efficiently target.

These findings motivate our mask training approach: while ΔW pruning provides a strong baseline, it still requires removing 30% of weights. Our learned masks achieve comparable or better reversal performance while pruning less than 10% of weights (Table 2), demonstrating that optimization can identify more precise targets than magnitude-based heuristics.

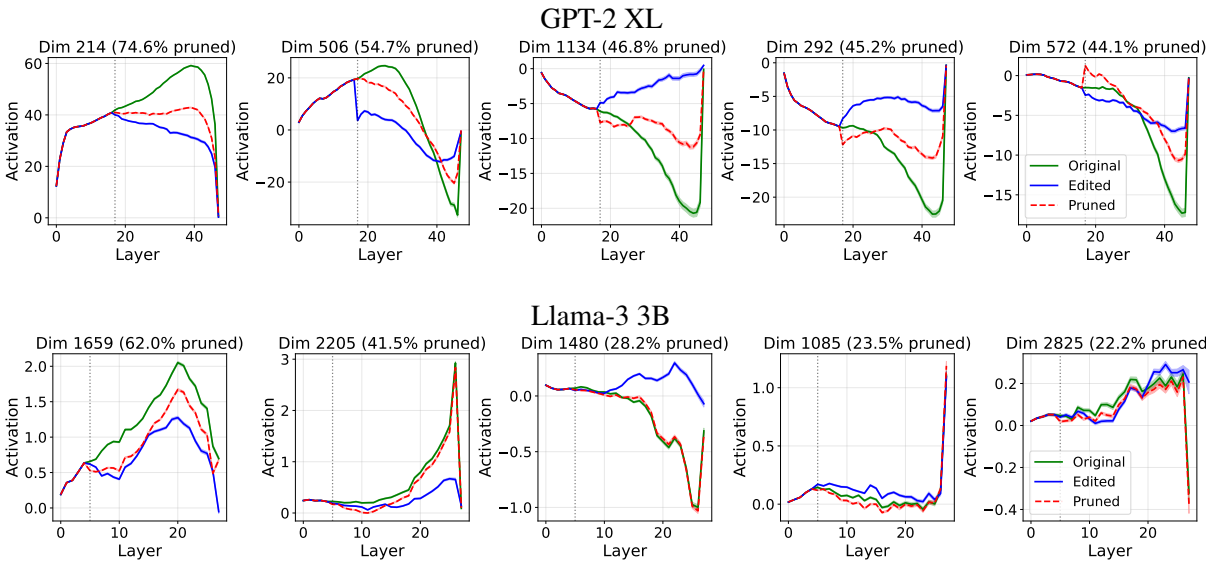

Figure 12: **Activation trajectories of the top-5 most pruned dimensions.** Mean activation (with standard error) across 1,000 samples tracked through all layers of GPT-2 XL and Llama-3 3B. Green: original model; blue: edited model; red dashed: pruned model.

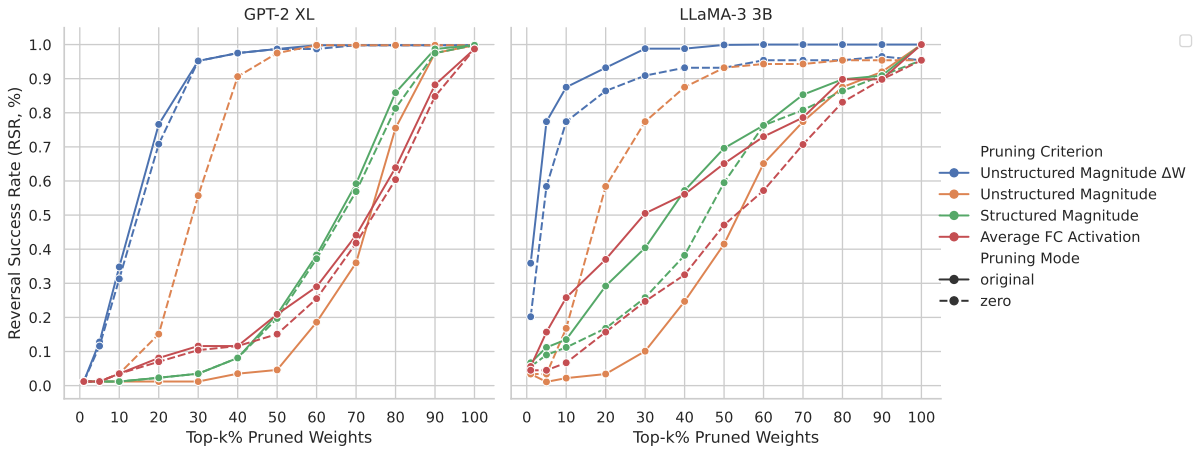

Figure 13: **Comparison of traditional pruning methods.** Reversal Success Rate (RSR) as a function of pruning percentage for GPT-2 XL (left) and LLaMA-3 3B (right). Each curve corresponds to a different pruning criterion. Solid lines indicate the *original* mode (pruned weights restored to pre-edit values); dashed lines indicate the *zero* mode (pruned weights set to 0). Unstructured ΔW pruning achieves 90% RSR with only 30% pruning, while structured methods require near-complete removal of the edited layer.