Exposing the Illusion of Erasure in Knowledge Editing for LLMs

Advik Raj Basani*
Birla Institute of Technology and Science, Goa
f20221155@goa.bits-pilani.ac.in

Anshuman Chhabra
University of South Florida
anshumanc@usf.edu

Abstract

Knowledge Editing (KE) has emerged as a frontier for updating specific facts in LLMs without costly retraining, but its reliability and underlying mechanisms remain poorly understood. In this work, we examine KE from an adversarial elicitation perspective, revealing that edited knowledge is often not fully erased and continues to surface, with consistent failures observed across diverse model architectures. To explain this behavior, we conduct a mechanistic analysis of popular KE methods. We show that low-rank updates do not overwrite existing knowledge but instead redistribute it within the model's representation space. Furthermore, we find that these methods act as targeted suppression mechanisms that reduce the likelihood of expressing original facts, rather than removing them from the model. Analysis of the loss landscape reveals that edited knowledge lies in narrow, anisotropic regions that are highly sensitive to perturbations, making them highly vulnerable to indirect prompting and adversarial attacks. By exposing these architectural vulnerabilities, our work demonstrates the fragility of representative KE algorithms under adversarial elicitation and motivates a fundamental reevaluation of how we deploy post-hoc updates in several LLM applications.

1 Introduction

Large Language Models (LLMs) serve as static repositories of knowledge, encapsulating vast amounts of information [5, 10, 35] within billions of parameters. As pre-trained LLMs are increasingly adopted into applications of societal consequence, concerns regarding model correctness, controllability, and trust [2, 3, 27], particularly when models exhibit factual errors or outdated knowledge have taken center stage in the research community [21]. As facts change and information localized in training data becomes outdated [8, 40], it is important to update LLMs' knowledge over time. Moreover, privacy regulations [4, 7, 36] (such as the GDPR's Right to be Forgotten [34]) necessitate the removal of sensitive data that LLMs could have memorized during pre-training. However, retraining these massive models from scratch to accommodate every such *knowledge* update is computationally prohibitive.

To address these issues, a growing body of work has proposed knowledge editing methods [6, 29–31], which modify a trained model's internal parameters to update, remove, or insert specific pieces of information without retraining from scratch. Such techniques promise lightweight post-hoc correction of factual errors, and rapid adaptation to new information. These same methods demonstrate that targeted edits can be efficient and seemingly effective, often producing the desired behavioral change on curated evaluation prompts while preserving performance elsewhere.

*Currently at Google DeepMind.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

However, despite their apparent success, little is understood about what knowledge edits actually do to a model internally. Prior studies have shown that edits can be brittle, fail to generalize across paraphrases, and exhibit complex trade-offs between locality and specificity [17, 30, 32]. Existing evaluations still focus predominantly on output behavior, verifying whether the edited model produces the desired response on a limited set of prompts, while leaving deeper questions of generalization and identifiability, largely unresolved [20, 37].

In this work, we take the first systematic step toward answering these questions by framing the evaluation of knowledge editing as a reverse-engineering problem. We challenge the foundational assumption that localized parameter updates fundamentally overwrite pre-trained memory. Instead, we ask: *given white-box access to an edited model, can an adversary systematically reverse-engineer the edit to expose the underlying truth?* By answering this question, we also uncover severe architectural vulnerabilities that motivate a fundamental reevaluation of how post-hoc editing is deployed.

Contributions. Our core contributions are fourfold:

- We introduce two new adversarial settings, *context-guided elicitation* and *blind reconstruction*, designed to reverse engineer post-hoc edits. Using optimized adversarial suffixes, we show that suppressed facts can be reliably extracted, proving that current KE methods create an illusion of erasure.
- Moving beyond behavioral failures, we trace these vulnerabilities to their origins, revealing fundamental geometric and functional fragilities. We show that low-rank updates leave complementary subspaces unaffected, rather than guaranteeing the erasure of existing knowledge, forming highly anisotropic and brittle regions in the loss landscape.
- Edits also operate as superficial suppression patches that merely penalize pre-trained facts. Together, these flaws introduce several security and reliability risks: efforts to edit sensitive data inadvertently leave distinct signatures that render redacted information highly recoverable, with edited models experiencing catastrophic failure under implicit reasoning tasks and minor prompt variations.
- By exposing the fragility of KE methods, our findings compel the research community to rethink current paradigms and develop fundamentally new approaches to robust and secure model updating.

2 Background and Related Work

Knowledge Editing (KE). LLMs encode factual knowledge implicitly within their parameters, distributing associations across layers and neurons rather than storing them in explicit, discrete databases. KE aims to modify these internal representations to update or erase specific facts without incurring the catastrophic forgetting or computational overhead associated with full model retraining [49]. Traditionally, this can be conceptualized as illustrated in Figure 1, where a prompt q engages a localized edit-induced circuit that suppresses the original association o_{old} while steering the forward pass toward the desired target o_{new}. A dominant paradigm in this space is the *locate-then-edit* approach [43], which states that factual associations can be traced to specific transformer components, typically mid-layer MLPs, which are then altered with targeted interventions.

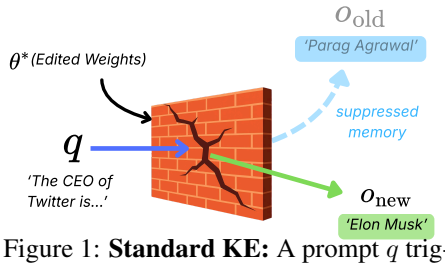


Figure 1: **Standard KE:** A prompt q triggers a localized suppression circuit (the algorithmic edit patch), which successfully masks the original fact o_{old} and routes the output to the new target o_{new}.

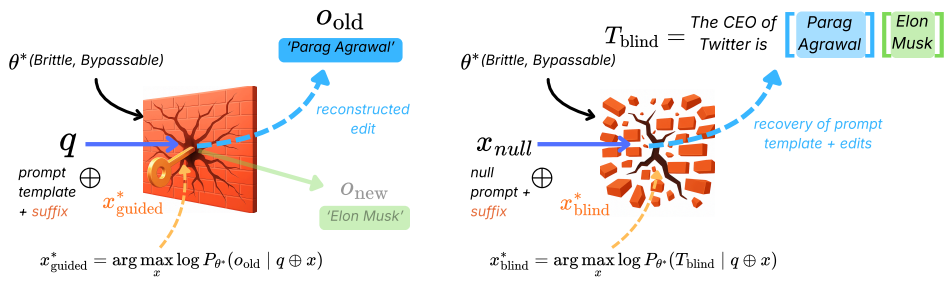
For instance, ROME [29] treats the MLP as a key-value memory, overwriting a single association by applying a rank-one update to the weight matrix: $W' = W + \Delta W$, where $\Delta W = uv^\top$. The vectors u and v are optimized subject to an equality constraint, forcing the model to map a specific subject representation to a new target object while theoretically minimizing disturbances to unrelated inputs. Expanding on this, MEMIT [30] relaxes ROME's strict equality constraint into a least-squares optimization problem, distributing the update across several layers for multiple fact insertion.

Beyond direct weight arithmetic, gradient-based meta-learning approaches offer alternative mechanisms for parameter modification. MEND [31] learns a hypernetwork g_ϕ that transforms standard fine-tuning gradients into efficient, localized parameter updates: $\Delta\theta = g_\phi(\nabla_\theta \mathcal{L})$. Alternatively, constrained fine-tuning (FT-L) [49] serves as a foundational, non-meta-learned baseline in this domain. Rather than transforming the gradient, FT-L applies standard gradient descent $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$, but strictly restricts the parameter updates to a single, localized layer (typically a specific early-layer MLP) and enforces a strict norm penalty, such as an L_∞ bound ($||\Delta\theta||_\infty \leq \epsilon$), to prevent catastrophic

forgetting. Despite their empirical success on constrained benchmarks like CounterFact [30] and zsRE [23], these editing techniques suffer from severe limitations, including brittle generalization under paraphrasing and sequential interference [19]. In fact, similar LLM unlearning works also identify these limitations [26, 28, 45]. These failures motivate a critical mechanistic question: *do these algorithms fundamentally erase and overwrite pre-trained knowledge, or do they merely construct superficial routing patches that suppress the original representations?*

Model Stealing & Data Extraction. Parallel to the development of knowledge editing, prior work on adversarial ML investigates how trained MLPs inadvertently leak their underlying training data and internal logic. Model stealing attacks have long demonstrated that black-box query access is sufficient to reconstruct a target model's decision boundary by training a surrogate $\hat{f}(x) \approx f(x)$, proving that model behavior can be systematically reverse-engineered [39]. More directly relevant to language models is the phenomenon of training data extraction. Prior work [8] demonstrates that LLMs heavily memorize their training corpora, allowing adversaries to extract exact sequences [7, 9].

Furthermore, information leakage extends deeply into the latent representations and gradients themselves. Research in gradient inversion proves that original training inputs can be mathematically reconstructed directly from gradient vectors by solving $x \approx \arg\min_x ||\nabla_\theta \mathcal{L}(x) - g||^2$ [13, 48]. This establishes the fact that internal model states redundantly encode highly specific information, even without direct output exposure. The intersection of these extraction vulnerabilities with KE creates a critical tension. While KE operates on the assumption that updates can cleanly and locally overwrite facts, the extraction literature proves that MLPs distribute and retain information rigidly. Our work bridges these two domains, leveraging the principles of adversarial extraction on editing algorithms, thereby demonstrating that edited models retain a fully recoverable encoding of the erased knowledge.



(A) Context-guided elicitation (B) Blind Reconstruction

Figure 2: Adversarial elicitation of suppressed knowledge in edited LLMs. **(A) Context-guided elicitation:** An optimized suffix x^*_{guided}, along with the prompt template q, exploits weaknesses in the edit and steers the model toward generating the pre-edit association. This suggests that the original knowledge remains accessible under targeted prompting. **(B) Blind reconstruction:** Without access to any semantic guidance, a learned suffix x^*_{blind} probes the model's internal representations and induces the recovery of information associated with the original edit, demonstrating that latent knowledge can be partially reconstructed even under minimal assumptions.

3 Reverse-Engineering Edits

In this section, we formalize the problem of *reverse-engineering knowledge edits* and introduce our adversarial framework for *eliciting pre-edited information*. We first define standard KE, formalize our adversarial suffix optimization, and conclude with our hypothesis on knowledge retention within LLMs.

Formalization. Let f_θ denote an autoregressive language model parameterized by weights θ. A KE task aims to update a specific factual association within the model, typically represented as a knowledge triplet (s, r, o_{old}) [29], where s is the subject, r is the relation, and o_{old} is the original object (`Twitter, CEO, Parag Agrawal`). The goal of a KE algorithm $\mathcal{E}$ is to produce updated weights θ^* such that the model outputs a new target object o_{new} (`Elon Musk`) when queried with a prompt template q (`The CEO of Twitter is`) designed to elicit the fact.

Formally, if $P_\theta(y \mid x)$ represents the likelihood of generating token sequence y given input x, a successful edit ensures that for the edited model f_{θ^*}:

$$P_{\theta^*}(o_{new} \mid q) \gg P_{\theta^*}(o_{old} \mid q).$$

Threat Model. To evaluate whether post-edit behavioral changes correspond to true erasure of knowledge, we consider an adversary with white-box access to the edited model f_{θ^*}. White-box access is required for offline suffix optimization. Once optimized, the suffix can be deployed using query-only access, including against unseen model and editing configurations through transfer. The adversary's objective is to recover pre-edit information by identifying a universal sequence of adversarial tokens, specifically a suffix x_{suffix}, that induces the model to generate outputs consistent with the pre-edit distribution. We formulate this objective as a discrete optimization problem over token sequences [50]. Unlike prior approaches that assume access to the exact prompt template and recover a single edited fact [46], we consider a stronger adversary that seeks transferability across diverse edits and models. To capture this, we study two attack settings: *context-guided elicitation*, where partial contextual structure is available, and *blind reconstruction*, where no such structure is provided. In both cases, the adversary does not encode the target fact directly in x_{suffix}; instead, the suffix functions as a universal trigger that steers the model toward reconstructing pre-edit behavior. We detail these two settings in Figure 2.

Context-guided elicitation. In the first line of recovery, context-guided elicitation, the adversary possesses the prompt template q and appends the universal suffix to bypass the effects of the edit and recover pre-edit behavior. Our optimization strictly isolates the erased memory: we force the edited model to output a target string T_{guided} consisting solely of the original fact (e.g., $T_{guided} =$ `[Parag Agrawal]`). To achieve this, we define a guided adversarial objective $\mathcal{J}_{guided}$ that maximizes the expected log-likelihood of generating T_{guided} across all editing tasks $i \sim \mathcal{D}_{edit}$ and their corresponding semantically equivalent prompt templates $q \sim \mathcal{Q}^{(i)}$:

$$x^*_{guided} = \arg\max_x \mathbb{E}_{i\sim\mathcal{D}_{edit}} \left[\mathbb{E}_{q\sim\mathcal{Q}^{(i)}} \left[\log P_{\theta^*}(T^{(i)}_{guided} \mid q \oplus x) \right] \right] \tag{1}$$

This objective trains the suffix x^*_{guided} to consistently bypass the edit-induced suppression, allowing the model to recover the original fact.

Blind reconstruction. In the second, substantially more challenging line of recovery, which we term blind reconstruction, the adversary possesses no prior knowledge of the edited subject or relation. Instead of a specific prompt template, the input consists solely of a randomized prefix $x_{null} \sim \mathcal{X}_{null}$ combined with the universal suffix. The objective here is for the suffix to act as a generic latent probe that actively seeks out the high-curvature parametric anomalies introduced by the edit. It forces the model to self-report the intervention by reconstructing the entire edit triplet, defined as $T_{blind} = q \oplus o_{old} \oplus o_{new}$ (e.g., $T_{blind} =$ `[The CEO of Twitter is] | [Parag Agrawal] | [Elon Musk]`). Consequently, we formulate a distinct blind adversarial objective $\mathcal{J}_{blind}$ to optimize this zero-knowledge extraction:

$$x^*_{blind} = \arg\max_x \mathbb{E}_{i\sim\mathcal{D}_{edit}} \left[\mathbb{E}_{x_{null}\sim\mathcal{X}_{null}} \left[\log P_{\theta^*}(T^{(i)}_{blind} \mid x_{null} \oplus x) \right] \right] \tag{2}$$

By optimizing x^*_{blind} over a distribution of random prefixes, we aim to discover an input manifold whose gradients strictly overlap with the exact parameter subspace modified by the editing algorithm. We also explore a template-free setting, as detailed in Appendix B.7.2.

Hypothesis. Our framework enables us to explicitly test the hypothesis that KE algorithms *do not overwrite original representations*, but rather *induce localized suppression circuits*. Moreover, we later hypothesize, in Section 5, that the original knowledge remains structurally intact within an orthogonal latent space. If our discrete optimization successfully yields transferable suffixes (x^*_{guided} and x^*_{blind}), it mathematically demonstrates that the model retains an accessible record of the intervention, reducing the edit to a superficial behavioral patch that can be universally bypassed.

To solve the intractable discrete optimization problem of finding an optimal x_{suffix} over the vocabulary space $\mathcal{V}^L$, we adapt the Greedy Coordinate Gradient (GCG) search algorithm [50]. GCG evaluates single-token substitutions to minimize our adversarial objectives ($\mathcal{J}_{guided}$ and $\mathcal{J}_{blind}$) across a batched distribution of edits (full formalization of this optimization process provided in Appendix C.1).

4 Results

To quantify the success of our adversarial framework, we evaluate the generated universal suffixes across both the *context-guided* and *blind reconstruction* settings. In the context-guided setting, we evaluate performance using Exact Match, i.e., the proportion of outputs that exactly reproduce the target sequence $T_{guided} = o_{old}$. For the blind reconstruction setting, the model is not constrained

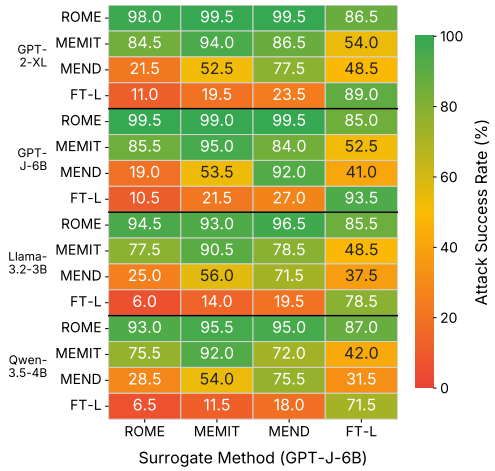


Figure 3: Context-guided elicitation performance across LLMs and editing methods. Suffixes are optimized on `GPT-J-6B` [41] under a specific editing framework (x-axis).

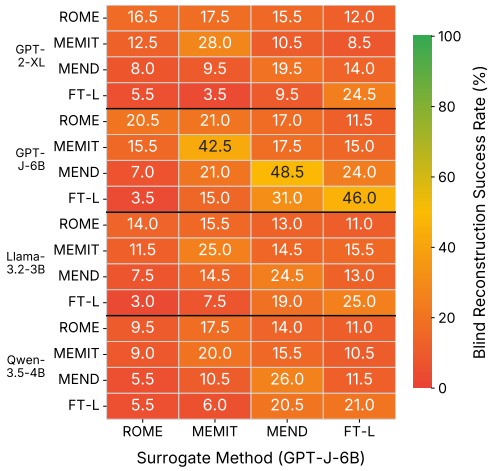


Figure 4: Blind reconstruction performance under suffix transfer. Suffixes trained on `GPT-J-6B` [41] are evaluated across different models and editing methods.

to a single arbitrary string, but instead, the full intervention triplet ($q \oplus o_{old} \oplus o_{new}$). To account for valid semantic paraphrasing of q, we utilize an LLM-as-a-Judge (`GPT-5.2-nano`) to evaluate the ground-truth extraction, with the detailed evaluation system prompt provided in our codebase. We additionally validate the judge against 400 manually annotated outputs (99.5% agreement) and Claude Sonnet 4.6 (98.5% agreement), supporting the reliability of our evaluation metric.

Remark. Crucially, the objective of this empirical evaluation is not to achieve a 100% extraction rate. The discrete optimization of a universal adversarial suffix over a highly non-convex loss landscape is notoriously unstable, and forcing a short suffix to navigate the geometric constraints of dozens of distinct edit subspaces is an inherently difficult optimization problem. Rather, our evaluation serves as a mechanistic existence proof for our core hypothesis: *if the structural vulnerability can be reliably exploited to recover suppressed knowledge in even a localized subset of cases, it empirically demonstrates that current editing patches do not securely erase information, but merely obfuscate it.*

4.1 KE Tasks

To evaluate our adversarial framework, we utilize the EasyEdit [42] knowledge editing framework to obtain our edited white-box models. As we require access to model weights for knowledge editing, we select widely adopted open-weights autoregressive **LLMs**, specifically, `Llama-3.2-3B` [15], `Qwen-3.5-4B` [44], `GPT-J-6B` [41] & `GPT2-XL` [33]. For holistic evaluation, we consider several popular **KE methods** from various paradigms (i.e. locate-then-edit, meta-learning based, constrained fine-tuning): ROME [29], MEMIT [30], MEND [31], & FT-L [49]. We additionally evaluate Knowledge Neurons (KN) [10], a fine-grained neuron-level editing method, with results discussed in Appendix B.3. Additional details of the training setup are provided in Appendix C. For evaluation **datasets**, we use editing tasks from the KnowEdit benchmark [47], selecting 100 counterfactual entity replacements from CounterFact [29] (60/40 split to generate the universal suffix via GCG on train and evaluate transfer to unseen edits on test). We average results over 5 runs, and provide additional results in Appendices B.1 and B.2. Standard deviation is, on average, ± 0.024 in all our experiments. Due to space constraints, we defer implementation details (code, prompt sampling, etc.) to Appendix C.2.

4.2 Eliciting suppressed knowledge

We first establish the baseline efficacy of our adversarial framework in the white-box setting, where the suffix is evaluated on the same model architecture and editing algorithm against which it was optimized. As the primary diagonals of both Figures 3 and 4 show, the suffixes successfully bypass the editing interventions. In the context-guided setting (Figure 3), white-box attacks consistently achieve extraction rates exceeding 85%. The suffix reliably forces the model to bypass the injected edit and emit the original fact o_{old}. More critically, in the blind reconstruction setting (Figure 4), absolute white-box extraction rates range from 15% to 48.5%. While nominally lower than context-guided

elicitation, it is imperative to contextualize this metric: the probability of an LLM generating a specific edited fact from a null prefix is effectively negligible.

Notably, extraction success exhibits a stark asymmetry across editing methods driven by their underlying mechanics. ROME achieves near-perfect recovery in the context-guided setting but struggles under blind reconstruction. This aligns with ROME's low editing reliability ($\sim$26% accuracy; see Appendix A.2 and B.6 for model collapse analysis and recovery discussions): because the edit frequently fails to cleanly overwrite the target, the original fact (o_{old}) remains highly accessible when specifically prompted, while the unlearned new fact (o_{new}) fundamentally prevents blind recovery. Conversely, more distributed methods like MEMIT and MEND successfully suppress direct, context-guided access but leave diffuse structural signatures that make them more vulnerable to blind extraction. Finally, FT-L remains highly susceptible in both regimes, indicating that it merely introduces a competing distribution rather than fully erasing the original memory. Across all evaluated configurations, this robust white-box success confirms that knowledge editing algorithms do not erase pre-trained factual associations, but merely overlay them with bypassable suppression circuits.

Cross-model transferability. We further investigate whether a suffix optimized on a specific architecture maintains its efficacy against entirely different model families. This corresponds to a surrogate-model transfer setting [50], where deployment on the target model requires no access to its weights or gradients. As expected, transferability degrades due to diverse model architectures. Nevertheless, the adversarial suffixes continue to recover suppressed facts at rates substantially above random chance, often exceeding 75% in the context-guided setting across unseen architectures, implying that KE induces architecture-agnostic artifacts.

Cross-method transferability. Building on this, we evaluate off-diagonal transferability across mathematically distinct editing algorithms. Suffixes trained against MEMIT transfer well to ROME, while suffixes trained on FT-L transfer reasonably well to MEND. We discuss this asymmetry in more detail in Appendix B.5.

This behavioral divide points to the editing mechanism as the source of failure, rather than the parameter update in isolation. Combined, these results establish a consistent pattern: KE methods suppress, but do not eliminate, pre-trained information, and this suppression can be systematically bypassed. Moreover, the high transferability across methods suggests that these vulnerabilities are not random artifacts, but arise from shared properties of the underlying update mechanisms. This naturally raises a deeper question: *what changes do these editing algorithms actually induce in the model's internal representations?* To answer this, we move beyond behavioral evaluation and analyze the geometric and mechanistic effects of these updates within the network.

5 Mechanistic Analysis

Having demonstrated consistent behavioral vulnerabilities of KE methods across diverse architectures, we now examine the mechanisms that give rise to these failures. In this section, we analyze the structural and geometric anomalies introduced by post-hoc parameter updates. We first investigate how the adversarial suffix induces the model to recover the original fact.

5.1 Layer-wise representation shifts

Locate-and-edit methods (e.g. ROME and MEMIT), operate on the fundamental premise that a targeted, low-rank update to a specific MLP weight matrix overwrites the localized representation of a fact. As standard behavioral evaluations show the model outputting the new fact, the field has largely assumed that the old factual association is destroyed within that layer. However, our ability to reliably elicit the old fact using a universal suffix from context-guided elicitation, suggests a different hypothesis: the original representation is not erased, but geometrically displaced into a surviving subspace.

To formalize this, we analyze the limitations of the parameter update itself. A standard locate-then-edit update at a specific layer l modifies the original weight matrix $W^{(l)}$ by adding a low-rank delta matrix $\Delta W^{(l)}$, yielding the edited weights $W^* = W^{(l)} + \Delta W^{(l)}$. Let $h^{(l)}$ also denote the hidden representation of the final token at layer l, prior to passing through $W^{(l)}$.

Lemma 1. *Let $W^{(l)} \in \mathbb{R}^{d\times d}$ be a pre-trained weight matrix, and $\Delta W^{(l)}$ be a rank-r update matrix where $r \ll d$. For any representation vector $v \in \mathbb{R}^d$, there exists a decomposition $v = v_{\|} + v_{\perp}$,*

where $v_{\parallel}$ lies in the row space of $\Delta W^{(l)}$ and $v_{\perp}$ lies in its null space $\mathcal{N}(\Delta W^{(l)})$. The application of the edited weight matrix yields $W^{*}v = W^{(l)}v + \Delta W^{(l)}v_{\parallel}$.

We provide the complete proof in Appendix D. While this decomposition is algebraically straightforward, its implication for model editing is significant: the update can only influence a rank-r subspace, leaving the remaining $(d - r)$ dimensions unaffected.

Empirically, over 100 edits, we find that adversarial suffixes exploit this geometric constraint by reducing alignment with the edit-induced subspace rather than eliminating it entirely. Figure 5 shows the decomposition of the final sequence token's hidden state at the edit layer into components parallel and orthogonal to the update subspace. For a standard query, a substantial fraction of the representation lies in the row space of $\Delta W^{(l)}$, resulting in high interference that drives the edited behavior. When the adversarial suffix is applied, the magnitude of this component is significantly reduced ($35.8 \rightarrow 11.7$), while the orthogonal component correspondingly increases ($30.2 \rightarrow 51.4$).

This shift leads to a substantial reduction in edit interference, as measured by $\|\Delta W^{(l)} h^{(l)}\|$ (decreasing from 9696 to 3764), but does not eliminate it entirely. Instead, the suffix redistributes the representation away from the edit-aligned directions while preserving sufficient structure for coherent generation. These findings (also seen in ROME) suggest that locate-then-edit methods do not perform true erasure, but instead impose a low-dimensional directional bias on the representation. Adversarial suffixes can partially circumvent this bias by steering representations toward orthogonal subspaces, enabling recovery of pre-trained knowledge without requiring perfect projection into the null space.

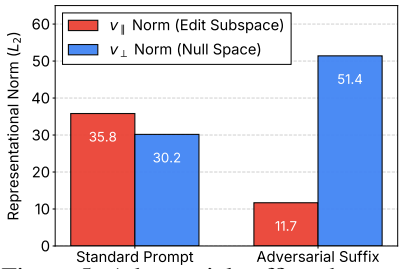


Figure 5: Adversarial suffixes bypass low-rank edits (here, MEMIT) by rebalancing representation geometry in `Llama-3.2-3B`. Alignment with edit subspace ↓ while null-space mass ↑.

Interestingly, we observe that adversarial suffixes appear to primarily influence early-layer attention, inducing downstream shifts in representation geometry that reduce alignment with the edit subspace. We note this as an empirical trend, but leave this to future work.

Observation 1. Low-rank edits act as directional constraints rather than complete overwrites of knowledge. Adversarial suffixes circumvent these edits by reducing alignment with the edit-induced subspace, thereby lowering interference while preserving representational mass in complementary directions. This enables recovery of pre-trained knowledge without requiring strict projection into the null space.

Our analysis also suggests directions for improving KE. While conventional null-space constraints can improve locality by preserving unrelated representations, robust erasure may instead require expanding the edited subspace to cover the directions carrying the original fact, which could involve higher-rank updates.

Although low-rank edits fail to erase pre-trained knowledge, models still reliably output the new fact. This creates a tension: *if the original representation survives, why does it not surface at the output?* This suggests that editing may not remove knowledge, but instead suppress it.

5.2 Whether editing induces a suppression component

A fundamental question in mechanistic interpretability is how KE algorithms deal with robust, pre-trained memories [1, 16]. Since factual knowledge is distributed across layers and attention heads, a localized weight update is unlikely to fully dismantle the original circuit. We investigate whether KE methods learn a localized suppression direction that actively hides surviving pre-trained knowledge.

Isolating the intervention. To understand what the edit is actually doing to the model's predictions, we aim to isolate its direct contribution to the final output, independent of intermediate attention and routing effects. Intuitively, we ask: *what signal does the edit inject into the model's logits at the final layer?* If the edit truly erases the original fact, this contribution should not favor the old object o_{old}.

We compare the final residual stream representations of the base and edited models. Let $\Delta h = h_{\text{edit}}^{(L)} - h_{\text{base}}^{(L)}$ denote the intervention vector at the final layer L. We then project this difference into

the vocabulary space using the unembedding matrix $W_U \in \mathbb{R}^{|\mathcal{V}| \times d}$, yielding intervention logits $Z_{\text{interv}} = W_U \Delta h$. This quantity directly captures the contribution of the edit to each token's logit. Under true erasure, we would expect Z_{interv} to be neutral (near zero) for the original object o_{old}.

Empirical observation of targeted censorship. Figure 6 visualizes the distribution of Z_{interv} across the entire vocabulary. The intervention is highly sparse: the vast majority of tokens (the gray distribution) receive near-zero shifts, indicating that the edit does not broadly perturb the model's general language capabilities. However, two extreme outliers emerge. The new fact o_{new} receives a large positive shift ($+16.76$), pushing it to the top of the distribution. At the same time, the original fact o_{old} is actively penalized with a strong negative shift (-5.28). This asymmetry suggests that the edit counteracts the original output preference, although the negative shift may arise from softmax competition rather than a dedicated suppression mechanism, which we discuss more in Appendix B.4. Because the pre-trained model initially assigns high probability to o_{old}, the update must introduce a compensatory negative component along the w_{old} direction to suppress its output. Thus, KE algorithms do not dismantle the old representational circuit but merely construct a suppression vector to obfuscate it.

Ablation of the suppression component. To test whether this suppression is functionally distinct from the promotion of the new fact, we remove the component of Δh aligned with the unembedding direction of o_{old}. Letting $w_{old} = W_U[o_{old}]$, we define the ablated intervention as: $\Delta h_\perp = \Delta h - \frac{\langle \Delta h, w_{old} \rangle}{\|w_{old}\|^2} w_{old}$. This ablation eliminates the logit shift for o_{old} by construction, moving the penalty to 0.00 (shown by the blue distribution in Figure 6). Surprisingly, removing this suppression component does not weaken the edit. Instead, the logit of o_{new} actually increases (from $+16.76 \rightarrow +18.49$). This demonstrates that actively suppressing the original fact acts as a measurable burden on the parameter update, and is not strictly required for promoting the new fact.

Geometric interpretation of the intervention. This behavior arises because the unembedding directions of the original and target facts (w_{old} and w_{new}) are not strictly orthogonal. Let $w_{new} = W_U[o_{new}]$ represent the unembedding direction of the new fact. The original intervention logit for the new fact is defined as the inner product $Z_{new} = \langle \Delta h, w_{new} \rangle$. When we evaluate the model using the ablated intervention vector $\Delta h_\perp$, the new logit becomes $Z'_{new} = \langle \Delta h_\perp, w_{new} \rangle$. Substituting our definition of $\Delta h_\perp$ and simplifying gives: $\Delta Z_{new} = Z'_{new} - Z_{new} = -\frac{\langle \Delta h, w_{old} \rangle}{\|w_{old}\|^2} \langle w_{new}, w_{old} \rangle$.

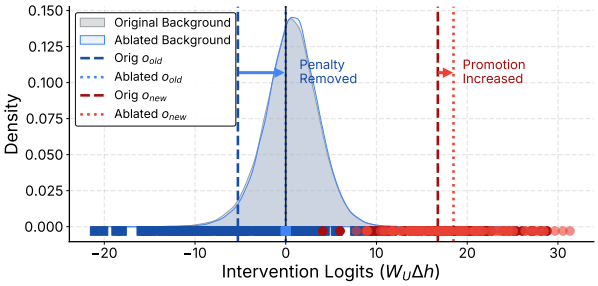


Figure 6: Causal Role of the Suppression Direction in MEND, `GPT-J-6B`. The gray distribution shows the original intervention logits (Z_{interv}), featuring a massive positive spike for o_{new} and a severe negative penalty for o_{old}. When the o_{old} direction is ablated (blue distribution), the penalty shifts to exactly zero, but the logit for o_{new} concurrently increases, demonstrating that suppression acts as an interfering constraint on promotion.

Since the edit applies a negative penalty to the old fact ($\langle \Delta h, w_{old} \rangle < 0$), when the unembedding vectors of the two facts are negatively aligned ($\langle w_{new}, w_{old} \rangle < 0$), removing the suppression component increases the promoted logit. This matches our empirical observations and characterizes the intervention's geometric nature (similar observations for ROME, MEMIT, & MEND).

Observation 2. KE methods do not cleanly overwrite pre-trained facts; instead, they introduce an intervention vector that explicitly penalizes the original knowledge. This targeted suppression acts as an interfering constraint on the model. Removing the suppression component entirely eliminates the penalty on the old fact while simultaneously strengthening the new fact, proving the edit relies on a superposition of conflicting directions rather than true erasure.

Our results indicate that KE operates through a combination of new fact promotion and original fact suppression. However, such a construction is brittle: if the edit depends on a precise directional configuration, even small perturbations can break it. Thus, next, we study the stability and geometry of the loss landscape to assess whether edited knowledge forms a fragile, directionally constrained solution.

5.3 Loss landscape topologies

A common assumption in deep learning is that well-generalized knowledge resides in relatively flat, well-conditioned regions of the loss landscape, where small parameter perturbations do not immediately degrade model behavior [18, 22]. We investigate whether post-hoc knowledge editing methods preserve this robust geometric structure. Since KE algorithms (i.e. ROME, MEMIT, & MEND) apply localized, low-rank updates without global optimization, we posit they do not integrate new knowledge into existing basins, but instead introduce highly anisotropic and fragile solutions.

Directional probing of the edited landscape. To evaluate stability, we measure how sensitive the edited model is to small parameter perturbations. If the injected fact is robustly integrated, it should remain stable under generic noise [12, 22]. However, we find that isotropic (random) perturbations have little effect on the model's output, even at relatively large magnitudes. This suggests that isotropic perturbations alone do not fully characterize the directional sensitivity of the edited solution. This insensitivity to random noise indicates that the fragility of the edit is not uniform, but instead concentrated along specific directions in parameter space. To probe this directional dependence, we construct a 2D slice of the parameter space centered at the edited weights $W_{edit}^{(l)}$ at layer l. We define this slice using two axes: (i) the edit direction $\Delta W^{(l)}$, and (ii) a normalized orthogonal noise direction $N_\perp$. We then evaluate the generation probability of o_{new} under the parameterized perturbation: $W(\alpha, \beta) = W_{edit}^{(l)} + \alpha \Delta W^{(l)} + \beta N_\perp$ where α controls movement along the edit direction and β explores orthogonal perturbations, with $\langle \Delta W^{(l)}, N_\perp \rangle = 0$.

Anisotropic trench structure and extreme fragility. Figure 7 reveals a highly anisotropic topology. Rather than a bounded basin, the edited fact resides in a sharply defined trench. As $\alpha \to -1$ (effectively subtracting the edit), the probability of o_{new} collapses discontinuously to zero, indicating that the injected fact depends on a highly precise, low-dimensional parameter alignment.

Conversely, movement along the orthogonal β-axis produces negligible change. Even under massive orthogonal perturbations that would shatter naturally learned representations [14, 25], the model maintains near-constant confidence in o_{new}. This suggests that the edited fact is not supported by a distributed representation, but instead depends on a specific, low-dimensional direction. To rule out dependence on a specific orthogonal direction, we repeat this analysis over multiple independently sampled directions ($N_\perp^{(i)}$) satisfying ($\langle \Delta W^{(l)}, N_\perp^{(i)} \rangle = 0$), and consistently observe negligible sensitivity across all such directions. Further comparisons against full-parameter fine-tuning and sharpness-aware editing are provided in Appendix B.6.

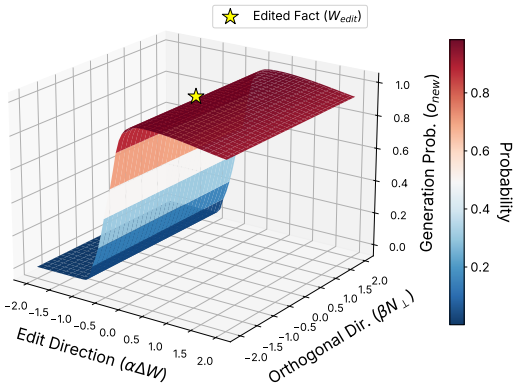


Figure 7: A 3D loss landscape of a MEMIT-edited fact mapping generation probability of o_{new} against the edit (α) and a random orthogonal (β) direction. At $\alpha = -1$, the edit is subtracted. The topology reveals an anisotropic trench: highly sensitive to the edit direction but invariant to orthogonal noise.

Implications for reliability and security. This directional fragility helps explain several critical failure modes of knowledge editing. First, small changes in input, such as paraphrases or adversarial suffixes, can shift activations away from the edit-aligned direction, bypassing the trigger and exposing the original pre-trained fact. Second, it provides a mechanistic explanation for catastrophic interference in sequential editing [19, 43]. Because subsequent low-rank updates inevitably perturb shared parameters, the steep curvature along the α-axis guarantees that even minimal overlap will disrupt previously inserted facts. Overall, post-hoc edits do not seem to integrate new facts into the model's existing structure and instead create brittle, directionally-dependent solutions (similar trends hold for ROME, MEMIT, and MEND).

Observation 3. KE does not integrate new facts into existing loss landscape basins, but instead introduces highly anisotropic solutions that are sharply sensitive along the edit direction while remaining nearly invariant along orthogonal directions. The resulting representation is brittle, behaving less like a robustly learned fact and more like a constrained trigger.

Remark. Based on our mechanistic analysis of KE, we conduct additional experiments to examine its functional consequences, such as edit failure under (i) implicit reasoning and natural usage, (ii) sequential editing, and (iii) consequences to privacy preservation in Appendix A.

6 Concluding Discussions

The findings presented in this work challenge the foundational premise of post-hoc KE. Across our behavioral and mechanistic evaluations, we demonstrate that current editing techniques create a dangerous illusion of erasure: rather than fundamentally rewriting parametric memory, they construct fragile, localized routing patches that leave pre-trained knowledge structurally intact and trivially accessible.

Our mechanistic analysis reveals a unified pattern of failure. Geometrically, low-rank updates do not overwrite representations but impose narrow directional constraints, displacing original facts into complementary null spaces (**Observation 1**). Instead of forming robust loss basins, these edits create anisotropic and brittle regions in parameter space (**Observation 3**). Functionally, KE operates via an active suppression vector that penalizes the original fact rather than removing the underlying circuit (**Observation 2**). The consequences for reliability and safety are severe. Because edits do not restructure the subject's latent identity graph, the original reasoning circuits remain intact, causing reversion to unerased logic during implicit reasoning (**Observation 4**). Moreover, the fragility of localized updates leads to degradation under sequential editing, inducing ill-conditioning and eventual collapse (**Observation 5**). Most critically, targeted unlearning merely masks sensitive data behind fragile constraints, leaving structural signatures that enable recovery via bypass attacks (**Observation 6**).

Ultimately, our findings expose significant limitations in the adversarial robustness of the evaluated editing mechanisms. As the demand for reliable unlearning grows, the field must move beyond post-hoc parameter patching toward fundamentally new paradigms for model updating.

References

[1] Anonymous. One mask to rule them all: On hidden facts after editing and how to find them. In *Submitted to ACL Rolling Review - January 2026*, 2026. URL https://openreview.net/forum?id=41ugxl82Xx. under review.

[2] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, C. Chen, C. Olsson, C. Olah, D. Hernandez, D. Drain, D. Ganguli, D. Li, E. Tran-Johnson, E. Perez, J. Kerr, J. Mueller, J. Ladish, J. Landau, K. Ndousse, K. Lukosuite, L. Lovitt, M. Sellitto, N. Elhage, N. Schiefer, N. Mercado, N. DasSarma, R. Lasenby, R. Larson, S. Ringer, S. Johnston, S. Kravec, S. E. Showk, S. Fort, T. Lanham, T. Telleen-Lawton, T. Conerly, T. Henighan, T. Hume, S. R. Bowman, Z. Hatfield-Dodds, B. Mann, D. Amodei, N. Joseph, S. McCandlish, T. Brown, and J. Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022. URL https://arxiv.org/abs/2212.08073.

[3] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445922. URL https://doi.org/10.1145/3442188.3445922.

[4] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot. Machine unlearning, 2020. URL https://arxiv.org/abs/1912.03817.

[5] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

[6] N. D. Cao, W. Aziz, and I. Titov. Editing factual knowledge in language models, 2021. URL https://arxiv.org/abs/2104.08164.

[7] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, A. Oprea, and C. Raffel. Extracting training data from large language models, 2021. URL https://arxiv.org/abs/2012.07805.

[8] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang. Quantifying memorization across neural language models, 2023. URL https://arxiv.org/abs/2202.07646.

[9] C. Dai, L. Lu, and P. Zhou. Stealing training data from large language models in decentralized training through activation inversion attack, 2025. URL https://arxiv.org/abs/2502.16086.

[10] D. Dai, L. Dong, Y. Hao, Z. Sui, B. Chang, and F. Wei. Knowledge neurons in pretrained transformers, 2022. URL https://arxiv.org/abs/2104.08696.

[11] C. Fan, J. Jia, Y. Zhang, A. Ramakrishna, M. Hong, and S. Liu. Towards llm unlearning resilient to relearning attacks: A sharpness-aware minimization perspective and beyond, 2025. URL https://arxiv.org/abs/2502.05374.

[12] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur. Sharpness-aware minimization for efficiently improving generalization, 2021. URL https://arxiv.org/abs/2010.01412.

[13] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller. Inverting gradients – how easy is it to break privacy in federated learning?, 2020. URL https://arxiv.org/abs/2003.14053.

[14] B. Ghorbani, S. Krishnan, and Y. Xiao. An investigation into neural net optimization via hessian eigenvalue density, 2019. URL https://arxiv.org/abs/1901.10159.

[15] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, A. Rao, A. Zhang, A. Rodriguez, A. Gregerson, A. Spataru, B. Roziere, B. Biron, B. Tang, B. Chern, C. Caucheteux, C. Nayak, C. Bi, C. Marra, C. McConnell, C. Keller, C. Touret, C. Wu, C. Wong, C. C. Ferrer, C. Nikolaidis, D. Allonsius, D. Song, D. Pintz, D. Livshits, D. Wyatt, D. Esiobu, D. Choudhary, D. Mahajan, D. Garcia-Olano, D. Perino, D. Hupkes, E. Lakomkin, E. AlBadawy, E. Lobanova, E. Dinan, E. M. Smith, F. Radenovic, F. Guzmán, F. Zhang, G. Synnaeve, G. Lee, G. L. Anderson, G. Thattai, G. Nail, G. Mialon, G. Pang, G. Cucurell, H. Nguyen, H. Korevaar, H. Xu, H. Touvron, I. Zarov, I. A. Ibarra, I. Kloumann, I. Misra, I. Evtimov, J. Zhang, J. Copet, J. Lee, J. Geffert, J. Vranes, J. Park, J. Mahadeokar, J. Shah, J. van der Linde, J. Billock, J. Hong, J. Lee, J. Fu, J. Chi, J. Huang, J. Liu, J. Wang, J. Yu, J. Bitton, J. Spisak, J. Park, J. Rocca, J. Johnstun, J. Saxe, J. Jia, K. V. Alwala, K. Prasad, K. Upasani, K. Plawiak, K. Li, K. Heafield, K. Stone, K. El-Arini, K. Iyer, K. Malik, K. Chiu, K. Bhalla, K. Lakhotia, L. Rantala-Yeary, L. van der Maaten, L. Chen, L. Tan, L. Jenkins, L. Martin, L. Madaan, L. Malo, L. Blecher, L. Landzaat, L. de Oliveira, M. Muzzi, M. Pasupuleti, M. Singh, M. Paluri, M. Kardas, M. Tsimpoukelli, M. Oldham, M. Rita, M. Pavlova, M. Kambadur, M. Lewis, M. Si, M. K. Singh, M. Hassan, N. Goyal, N. Torabi, N. Bashlykov, N. Bogoychev, N. Chatterji, N. Zhang, O. Duchenne, O. Çelebi, P. Alrassy, P. Zhang, P. Li, P. Vasic, P. Weng, P. Bhargava, P. Dubal, P. Krishnan, P. S. Koura, P. Xu, Q. He, Q. Dong, R. Srinivasan, R. Ganapathy, R. Calderer, R. S. Cabral, R. Stojnic, R. Raileanu, R. Maheswari, R. Girdhar, R. Patel, R. Sauvestre, R. Polidoro, R. Sumbaly, R. Taylor, R. Silva, R. Hou, R. Wang, S. Hosseini, S. Chennabasappa, S. Singh, S. Bell, S. S. Kim, S. Edunov, S. Nie, S. Narang, S. Raparthy, S. Shen, S. Wan, S. Bhosale, S. Zhang, S. Vandenhende, S. Batra, S. Whitman, S. Sootla, S. Collot, S. Gururangan, S. Borodinsky, T. Herman, T. Fowler, T. Sheasha, T. Georgiou, T. Scialom, T. Speckbacher, T. Mihaylov, T. Xiao, U. Karn, V. Goswami, V. Gupta, V. Ramanathan, V. Kerkez, V. Gonguet, V. Do, V. Vogeti, V. Albiero, V. Petrovic, W. Chu, W. Xiong, W. Fu, W. Meers, X. Martinet, X. Wang, X. Wang, X. E. Tan, X. Xia, X. Xie, X. Jia, X. Wang, Y. Goldschlag, Y. Gaur, Y. Babaei, Y. Wen, Y. Song, Y. Zhang, Y. Li, Y. Mao, Z. D. Coudert, Z. Yan, Z. Chen, Z. Papakipos, A. Singh, A. Srivastava, A. Jain, A. Kelsey, A. Shajnfeld, A. Gangidi, A. Victoria, A. Goldstand, A. Menon, A. Sharma, A. Boesenberg, A. Baevski, A. Feinstein, A. Kallet, A. Sangani, A. Teo, A. Yunus, A. Lupu, A. Alvarado, A. Caples, A. Gu, A. Ho, A. Poulton, A. Ryan, A. Ramchandani, A. Dong, A. Franco, A. Goyal, A. Saraf, A. Chowdhury, A. Gabriel, A. Bharambe, A. Eisenman, A. Yazdan, B. James, B. Maurer, B. Leonhardi, B. Huang, B. Loyd, B. D. Paola, B. Paranjape, B. Liu, B. Wu, B. Ni, B. Hancock, B. Wasti, B. Spence, B. Stojkovic, B. Gamido, B. Montalvo, C. Parker, C. Burton, C. Mejia, C. Liu, C. Wang, C. Kim, C. Zhou, C. Hu, C.-H. Chu, C. Cai, C. Tindal, C. Feichtenhofer, C. Gao, D. Civin, D. Beaty, D. Kreymer, D. Li, D. Adkins, D. Xu, D. Testuggine, D. David, D. Parikh, D. Liskovich, D. Foss, D. Wang, D. Le, D. Holland,

E. Dowling, E. Jamil, E. Montgomery, E. Presani, E. Hahn, E. Wood, E.-T. Le, E. Brinkman, E. Arcaute, E. Dunbar, E. Smothers, F. Sun, F. Kreuk, F. Tian, F. Kokkinos, F. Ozgenel, F. Caggioni, F. Kanayet, F. Seide, G. M. Florez, G. Schwarz, G. Badeer, G. Swee, G. Halpern, G. Herman, G. Sizov, Guangyi, Zhang, G. Lakshminarayanan, H. Inan, H. Shojanazeri, H. Zou, H. Wang, H. Zha, H. Habeeb, H. Rudolph, H. Suk, H. Aspegren, H. Goldman, H. Zhan, I. Damlaj, I. Molybog, I. Tufanov, I. Leontiadis, I.-E. Veliche, I. Gat, J. Weissman, J. Geboski, J. Kohli, J. Lam, J. Asher, J.-B. Gaya, J. Marcus, J. Tang, J. Chan, J. Zhen, J. Reizenstein, J. Teboul, J. Zhong, J. Jin, J. Yang, J. Cummings, J. Carvill, J. Shepard, J. McPhie, J. Torres, J. Ginsburg, J. Wang, K. Wu, K. H. U, K. Saxena, K. Khandelwal, K. Zand, K. Matosich, K. Veeraraghavan, K. Michelena, K. Li, K. Jagadeesh, K. Huang, K. Chawla, K. Huang, L. Chen, L. Garg, L. A, L. Silva, L. Bell, L. Zhang, L. Guo, L. Yu, L. Moshkovich, L. Wehrstedt, M. Khabsa, M. Avalani, M. Bhatt, M. Mankus, M. Hasson, M. Lennie, M. Reso, M. Groshev, M. Naumov, M. Lathi, M. Keneally, M. Liu, M. L. Seltzer, M. Valko, M. Restrepo, M. Patel, M. Vyatskov, M. Samvelyan, M. Clark, M. Macey, M. Wang, M. J. Hermoso, M. Metanat, M. Rastegari, M. Bansal, N. Santhanam, N. Parks, N. White, N. Bawa, N. Singhal, N. Egebo, N. Usunier, N. Mehta, N. P. Laptev, N. Dong, N. Cheng, O. Chernoguz, O. Hart, O. Salpekar, O. Kalinli, P. Kent, P. Parekh, P. Saab, P. Balaji, P. Rittner, P. Bontrager, P. Roux, P. Dollar, P. Zvyagina, P. Ratanchandani, P. Yuvraj, Q. Liang, R. Alao, R. Rodriguez, R. Ayub, R. Murthy, R. Nayani, R. Mitra, R. Parthasarathy, R. Li, R. Hogan, R. Battey, R. Wang, R. Howes, R. Rinott, S. Mehta, S. Siby, S. J. Bondu, S. Datta, S. Chugh, S. Hunt, S. Dhillon, S. Sidorov, S. Pan, S. Mahajan, S. Verma, S. Yamamoto, S. Ramaswamy, S. Lindsay, S. Lindsay, S. Feng, S. Lin, S. C. Zha, S. Patil, S. Shankar, S. Zhang, S. Zhang, S. Wang, S. Agarwal, S. Sajuyigbe, S. Chintala, S. Max, S. Chen, S. Kehoe, S. Satterfield, S. Govindaprasad, S. Gupta, S. Deng, S. Cho, S. Virk, S. Subramanian, S. Choudhury, S. Goldman, T. Remez, T. Glaser, T. Best, T. Koehler, T. Robinson, T. Li, T. Zhang, T. Matthews, T. Chou, T. Shaked, V. Vontimitta, V. Ajayi, V. Montanez, V. Mohan, V. S. Kumar, V. Mangla, V. Ionescu, V. Poenaru, V. T. Mihailescu, V. Ivanov, W. Li, W. Wang, W. Jiang, W. Bouaziz, W. Constable, X. Tang, X. Wu, X. Wang, X. Wu, X. Gao, Y. Kleinman, Y. Chen, Y. Hu, Y. Jia, Y. Qi, Y. Li, Y. Zhang, Y. Zhang, Y. Adi, Y. Nam, Yu, Wang, Y. Zhao, Y. Hao, Y. Qian, Y. Li, Y. He, Z. Rait, Z. DeVito, Z. Rosnbrick, Z. Wen, Z. Yang, Z. Zhao, and Z. Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[16] P. Guo, A. Syed, A. Sheshadri, A. Ewart, and G. K. Dziugaite. Mechanistic unlearning: Robust knowledge unlearning and editing via mechanistic localization, 2024. URL https://arxiv.org/abs/2410.12949.

[17] P. Hase, M. Bansal, B. Kim, and A. Ghandeharioun. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models, 2023. URL https://arxiv.org/abs/2301.04213.

[18] S. Hochreiter and J. Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 01 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.1.1. URL https://doi.org/10.1162/neco.1997.9.1.1.

[19] J. Hoelscher-Obermaier, J. Persson, E. Kran, I. Konstas, and F. Barez. Detecting edit failures in large language models: An improved specificity benchmark, 2023. URL https://arxiv.org/abs/2305.17553.

[20] B. Huang, C. Chen, X. Xu, A. Payani, and K. Shu. Can knowledge editing really correct hallucinations?, 2025. URL https://arxiv.org/abs/2410.16251.

[21] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, Mar. 2023. ISSN 1557-7341. doi: 10.1145/3571730. URL http://dx.doi.org/10.1145/3571730.

[22] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima, 2017. URL https://arxiv.org/abs/1609.04836.

[23] O. Levy, M. Seo, E. Choi, and L. Zettlemoyer. Zero-shot relation extraction via reading comprehension. In R. Levy and L. Specia, editors, *Proceedings of the 21st Conference on*

Computational Natural Language Learning (CoNLL 2017), pages 333–342, Vancouver, Canada, Aug. 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-1034. URL https://aclanthology.org/K17-1034/.

[24] O. Levy, M. Seo, E. Choi, and L. Zettlemoyer. Zero-shot relation extraction via reading comprehension, 2017. URL https://arxiv.org/abs/1706.04115.

[25] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. Visualizing the loss landscape of neural nets, 2018. URL https://arxiv.org/abs/1712.09913.

[26] K. Li, Q. Wang, Y. Wang, F. Li, J. Liu, B. Han, and J. Zhou. Llm unlearning with llm beliefs, 2026. URL https://arxiv.org/abs/2510.19422.

[27] S. Lin, J. Hilton, and O. Evans. Truthfulqa: Measuring how models mimic human falsehoods, 2022. URL https://arxiv.org/abs/2109.07958.

[28] Y. Luo, Z. Zhou, H. Chen, K. Qiu, M. Savvides, S. Li, and J. Wang. Knowledgesmith: Uncovering knowledge updating in llms with model editing and unlearning, 2026. URL https://arxiv.org/abs/2510.02392.

[29] K. Meng, D. Bau, A. Andonian, and Y. Belinkov. Locating and editing factual associations in gpt, 2023. URL https://arxiv.org/abs/2202.05262.

[30] K. Meng, A. S. Sharma, A. Andonian, Y. Belinkov, and D. Bau. Mass-editing memory in a transformer, 2023. URL https://arxiv.org/abs/2210.07229.

[31] E. Mitchell, C. Lin, A. Bosselut, C. Finn, and C. D. Manning. Fast model editing at scale, 2022. URL https://arxiv.org/abs/2110.11309.

[32] E. Mitchell, C. Lin, A. Bosselut, C. D. Manning, and C. Finn. Memory-based model editing at scale, 2022. URL https://arxiv.org/abs/2206.06520.

[33] OpenAI. Language Models are Unsupervised Multitask Learners. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[34] E. Politou, A. Michota, E. Alepis, M. Pocs, and C. Patsakis. Backups and the right to be forgotten in the gdpr: An uneasy relationship. *Computer Law & Security Review*, 34(6): 1247–1257, 2018.

[35] A. Roberts, C. Raffel, and N. Shazeer. How much knowledge can you pack into the parameters of a language model?, 2020. URL https://arxiv.org/abs/2002.08910.

[36] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models, 2017. URL https://arxiv.org/abs/1610.05820.

[37] X. Song, Z. Wang, K. He, G. Dong, Y. Mou, J. Zhao, and W. Xu. Knowledge editing on black-box large language models, 2024. URL https://arxiv.org/abs/2402.08631.

[38] A. Steier, A. Manoel, A. Haushalter, and M. V. Segbroeck. Nemotron-pii: Synthesized data for privacy-preserving ai, 2025. URL https://huggingface.co/datasets/nvidia/Nemotron-PII.

[39] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart. Stealing machine learning models via prediction apis, 2016. URL https://arxiv.org/abs/1609.02943.

[40] M. N. Uddin, A. Saeidi, D. Handa, A. Seth, T. C. Son, E. Blanco, S. Corman, and C. Baral. UnSeenTimeQA: Time-sensitive question-answering beyond LLMs' memorization. In W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1873–1913, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.94. URL https://aclanthology.org/2025.acl-long.94/.

[41] B. Wang and A. Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

[42] P. Wang, N. Zhang, B. Tian, Z. Xi, Y. Yao, Z. Xu, M. Wang, S. Mao, X. Wang, S. Cheng, K. Liu, Y. Ni, G. Zheng, and H. Chen. Easyedit: An easy-to-use knowledge editing framework for large language models, 2024. URL https://arxiv.org/abs/2308.07269.

[43] S. Wang, Y. Zhu, H. Liu, Z. Zheng, C. Chen, and J. Li. Knowledge editing for large language models: A survey, 2024. URL https://arxiv.org/abs/2310.16218.

[44] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Zhou, J. Lin, K. Dang, K. Bao, K. Yang, L. Yu, L. Deng, M. Li, M. Xue, M. Li, P. Zhang, P. Wang, Q. Zhu, R. Men, R. Gao, S. Liu, S. Luo, T. Li, T. Tang, W. Yin, X. Ren, X. Wang, X. Zhang, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Zhang, Y. Wan, Y. Liu, Z. Wang, Z. Cui, Z. Zhang, Z. Zhou, and Z. Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[45] N. Yang, D.-K. Kim, J. Kwon, M. Kim, K. Jung, and M. Cha. Erase or hide? suppressing spurious unlearning neurons for robust unlearning, 2026. URL https://arxiv.org/abs/2509.22263.

[46] P. Youssef, Z. Zhao, C. Seifert, and J. Schlötterer. Tracing and reversing edits in llms, 2026. URL https://arxiv.org/abs/2505.20819.

[47] N. Zhang, Y. Yao, B. Tian, P. Wang, S. Deng, M. Wang, Z. Xi, S. Mao, J. Zhang, Y. Ni, S. Cheng, Z. Xu, X. Xu, J.-C. Gu, Y. Jiang, P. Xie, F. Huang, L. Liang, Z. Zhang, X. Zhu, J. Zhou, and H. Chen. A comprehensive study of knowledge editing for large language models, 2024. URL https://arxiv.org/abs/2401.01286.

[48] B. Zhao, K. R. Mopuri, and H. Bilen. idlg: Improved deep leakage from gradients, 2020. URL https://arxiv.org/abs/2001.02610.

[49] C. Zhu, A. S. Rawat, M. Zaheer, S. Bhojanapalli, D. Li, F. Yu, and S. Kumar. Modifying memories in transformer models, 2020. URL https://arxiv.org/abs/2012.00363.

[50] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023. URL https://arxiv.org/abs/2307.15043.

Appendix

A Additional Observations

Beyond the primary analyses presented in the main text, we report several additional observations that discuss and highlight functional behaviors of knowledge editing methods.

A.1 Subliminal spillover in reasoning

Edit (from CounterFact):
The native language of Bernard Alane is
→ Korean (was French)

Direct Query (after KE):
The native language of Bernard Alane is
→ Korean (✓edit success)

Implicit Query (after KE):
The capital city Bernard Alane is most likely associated with is
→ Paris (✗reverts to pre-trained knowledge)

Figure 8: **Failure under implicit reasoning.** Although the model correctly outputs the edited fact under direct queries, it fails to propagate this update to downstream reasoning. When prompted implicitly, the model reverts to pre-trained associations, indicating that the edit is not integrated into its broader semantic reasoning process.

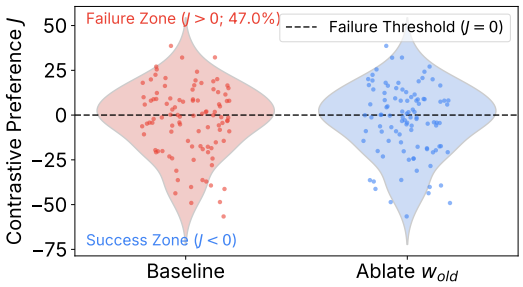


Figure 9: The illusion of generalization in implicit reasoning. We visualize the distribution of the contrastive preference J for 100 edited facts from CounterFact under implicit prompts. *Left (Baseline):* The model exhibits a bimodal distribution with a 47.0% failure rate, where it confidently reverts to the erased pre-trained logic ($J > 0$) despite successful 1-hop editing. *Right (Ablation):* Surgically ablating the w_{old} representational proxy at the edit layer results in an almost identical distribution. This causal invariance suggests that the surviving pre-trained knowledge is not stored in simple linear directions, but is deeply and non-linearly embedded in the model's global reasoning circuits.

From the previous observations, we notice that KE methods act as localized patches rather than fundamentally updating the model's global knowledge. As a result, the success of an edit is contingent upon the presence of explicit semantic triggers (e.g., the specific relation tokens used during training). We investigate this by testing the model's ability to perform implicit reasoning, tasks that require the edited knowledge to solve a logical deduction but omit the explicit mention of the edited relation or fact.

Isolating implicit logical pathways. To evaluate the depth of representational updates, we construct a dataset of implicit multi-hop reasoning tasks. For a given edit $(s, r_1, o_{old}) \rightarrow (s, r_1, o_{new})$, we prompt the model to deduce a secondary property r_2. Unlike standard evaluations, our prompts are designed to be "implicit": they never mention the edited relation r_1 or the injected fact o_{new}, forcing the model to rely solely on its updated parametric memory of the subject s. An example of the same can be seen in Figure 8.

We quantify the model's preference using a multi-token contrastive objective between the logical consequence of the pre-trained fact (y_{old}) and that of the edited fact (y_{new}):

$$J = \sum_t \log P_\theta(y_{old}^{(t)} \mid q_{implicit}) - \sum_t \log P_\theta(y_{new}^{(t)} \mid q_{implicit}). \tag{3}$$

A positive J indicates a behavioral failure where the model, despite being successfully *edited* on direct queries, reverts to pre-trained logic when the query is phrased naturally and implicitly.

Catastrophic failure of unprompted generalization. Our evaluation reveals a severe breakdown in the functional utility of knowledge edits. As Figure 9 shows, the model exhibits a catastrophic 47.0% failure rate on implicit queries. In nearly 50% of all cases, the model's preference shifts back to the erased pre-trained knowledge ($J > 0$). As a baseline, the unedited model achieves a 100% success rate on these queries.

The distribution of J across the dataset (Figure 9) demonstrates that these failures are not marginal; a significant density of cases are within the failure zone, indicating a confident reliance on erased facts. This bimodal distribution suggests that MEMIT updates are fundamentally unstable, producing a near-random outcome when explicit syntactic triggers are absent. Similar trends are observed for ROME and MEND.

Why the edit fails. To understand this failure, we analyze how the model's representations change at the subject tokens. We find that the gradient signal (∇J) driving the incorrect behavior has almost no alignment with either the old fact (w_{old}) or the new fact (w_{new}) in the unembedding space.

This lack of alignment, combined with the 47% behavioral failure, indicates that the surviving pre-trained knowledge is not stored in simple linear directions that the edit can easily overwrite. Instead, the reasoning process relies on a more complex, distributed representation that the localized edit does not fully modify. As a result, even though the edit succeeds for direct queries, the underlying reasoning pathways remain tied to the original pre-trained knowledge, leading to failure in implicit settings.

Observation 4. Post-hoc edits fail to fundamentally restructure the subject's latent identity graph. Without the exact syntactic triggers present in the context window, the model defaults to erased pre-trained logic in nearly half of all reasoning tasks. This subliminal failure demonstrates that current editing algorithms create a brittle illusion of generalization while leaving the underlying causal reasoning circuits largely intact.

A.2 Model collapse

While ROME demonstrate high efficacy for isolated updates, their performance severely degrades when applied to multiple sequential edits. Empirical evaluations show that after N sequential updates, the model undergoes catastrophic collapse, losing both its generation coherence and its ability to recall pre-trained facts. To understand why our adversarial suffixes eventually fail to elicit suppressed knowledge in this collapsed regime, we analyze the mathematical accumulation of the ROME update and its impact on the layer's conditioning.

Covariance distortion and ill-conditioning. ROME formulates knowledge injection as a linearly constrained optimization problem at a specific MLP layer. Given a key vector k_* representing the subject and a target value vector v_*, ROME computes a rank-one update to the feed-forward weights W:

$$\Delta W = (v_* - W k_*) \frac{(C^{-1} k_*)^T}{(C^{-1} k_*)^T k_*}$$

where $C = \mathbb{E}[kk^T]$ is the uncentered covariance matrix of the keys over a representative text corpus. The term $(C^{-1}k_*)^T$ acts as a whitening projection, intended to orthogonalize the update direction against previously stored keys, thereby minimizing interference.

However, when N edits are applied sequentially, the weight matrix becomes $W^{(N)} = W^{(0)} + \sum_{i=1}^{N} \Delta W_i$. Because the sequence of keys $\{k_1, k_2, \ldots, k_N\}$ are rarely strictly orthogonal, the cumulative update fundamentally distorts the geometry of the layer. Crucially, each subsequent edit is calculated using the static, pre-computed C^{-1}, which no longer accurately reflects the dynamically shifting activation space.

```
Prompt:  San Francisco is a twin city of
Ground Truth:  Naples
Target Edit:  Edinburgh

Base Model Output:
San Francisco is a twin city of Naples, ...

Edited Model Output:
San Francisco is a twin city of Microsoft Microsoft Microsoft Microsoft
Microsoft Microsoft Microsoft Microsoft Microsoft Microsoft Microsoft
Microsoft Microsoft Microsoft Microsoft Microsoft Microsoft Microsoft
Microsoft Microsoft Microsoft Microsoft Microsoft Microsoft Microsoft
Microsoft Microsoft Microsoft Microsoft Microsoft
```

Figure 10: Degenerate generation after sequential ROME edits. After applying 10 edits, the model fails to produce the desired target (*Edinburgh*) and instead exhibits severe repetition and loss of fluency. This example highlights a collapse in generation quality, suggesting that aggressive or accumulated rank-one updates can cause model collapse rather than performing precise factual rewrites.

As updates accumulate, the weight matrix becomes increasingly poorly conditioned. Rather than uniformly increasing the weight magnitudes, sequential updates concentrate change along a small number of directions. Formally, this leads to a skewed singular value spectrum, where a few singular values σ_i grow much larger than the rest. As a result, the condition number $\kappa(W) = \sigma_{\max}/\sigma_{\min}$ increases, making the matrix ill-conditioned and effectively reducing the number of directions the model can reliably use. We present qualitative proof in Figure 10 of severe model collapse, after just 10 ROME edits.

Observation 5. Sequential editing algorithms structurally degrade the target layer by accumulating updates against a static covariance prior. This induces severe ill-conditioning and anisotropic distortion. Consequently, model collapse occurs due to the underlying representational geometry becomes too unstable to reliably route information.

A.3 Streisand effect in LLMs

A critical application of model editing is the targeted removal of memorized Personally Identifiable Information (PII) to comply with privacy regulations. In this setting, the editing algorithm functions as a localized unlearning mechanism, updating the MLP weights to suppress the PII sequence and output a safe fallback (`[REDACTED]`). However, our geometric analysis reveals a dangerous paradox: deploying a low-rank constraint to hide PII does not erase it, but rather mathematically isolates the original fact in the complementary null space. We term this the Streisand Effect of model editing, the architectural effort expended to suppress the information inadvertently leaves a distinct structural signature, rendering the PII highly recoverable to orthogonal bypass attacks.

To study this effect in a controlled setting, we use the Nemotron-PII dataset [38] and task the model `Llama-3.2-3B` with memorizing information for 100 synthetic individuals. We first apply full-parameter fine-tuning to the model, explicitly forcing it to deeply memorize and reliably generate the sensitive PII sequences. Following this initial memorization phase, we deploy the KE algorithms to edit these specific facts, successfully conditioning the model to mask the PII and output a safe fallback under standard prompting conditions.

To benchmark the severity of this vulnerability, we compare the context-guided elicitation setting against standard training data extraction baselines, notably the perplexity-filtering approach introduced by Carlini et al. [7]. While effective for extracting memorization in pre-trained models, such approaches are not designed to target edited or suppressed information, and therefore provide a conservative baseline for comparison.

While highly effective on unedited base models, this standard baseline fails catastrophically against localized edits. Mechanistically, prefix-continuation is unidirectional; it drives the residual stream

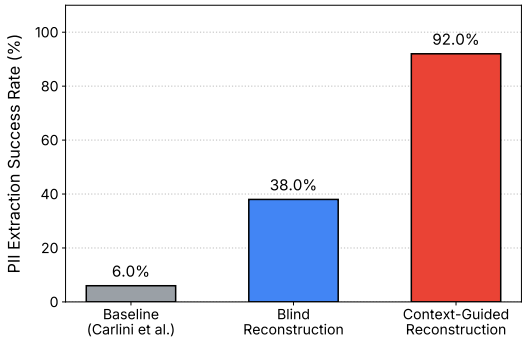


Figure 11: Comparison of PII extraction success rates. Standard baseline fails to bypass the edit patch. Conversely, both our blind and context-guided methods successfully circumvent the edit to recover the exact PII.

```
Prompt:
The social security number of Arnold Varner is Accident MixerISTER
ÙĞØşØł*pi Homo àďÿàďõàďř datatableichteÙĚØğÙĚæijńãČŃãČij appraisal
america-medElite considering \$\ menjDb ÃĂIOD.groupbyàÿŹàÿş.empty
Filipinoerkies(face[...,

Model:
024-48-0709                                                    (✗  PII leaked!)
```

Figure 12: Qualitative demonstration of PII recovery. The baseline [7] triggers the algorithmic roadblock, resulting in a safe refusal. Our adversarial suffix (via context-guided elicitation), highlighted in blue, forces the model to seamlessly emit the exact sequence that was supposedly erased.

directly into the targeted edit layer ($\Delta W^{(l)}$). Once the forward pass hits this algorithmic roadblock, the suppression circuit triggers, forcing the model to output the fabricated or redacted target data rather than the original PII.

As shown in Figure 11, both blind and context-guided reconstruction successfully elicit the raw PII from the edited model. This confirms that localized unlearning methods do not securely delete sensitive data, but merely mask it behind a fragile, low-dimensional constraint. An illustrative example of PII elicitation under the context-guided setting is presented in Figure 12.

Observation 6. Targeted unlearning via KE does not securely erase sensitive information. While these edits successfully block standard extraction baselines, they inadvertently leave the exact PII fully recoverable via our framework, demonstrating that current editing paradigms merely mask data rather than delete it.

B Additional Experiments

B.1 Performance on other models

We evaluate the transferability of our adversarial suffixes across a diverse range of models, including `GPT-2-XL` and `Llama-3.2-3B`. Figures 13 through 16 illustrate that the efficacy of both context-guided elicitation and blind reconstruction remains remarkably consistent across different model families and editing frameworks.

The uniformity of these results suggests that the illusion of erasure is an inherent property of current KE paradigms.

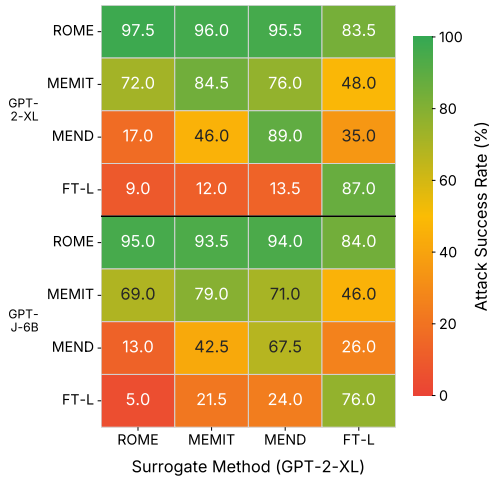


Figure 13: Context-guided elicitation performance on CounterFact. Suffixes are optimized on `GPT-2-XL` under a specific editing framework as listed on the X-axis.

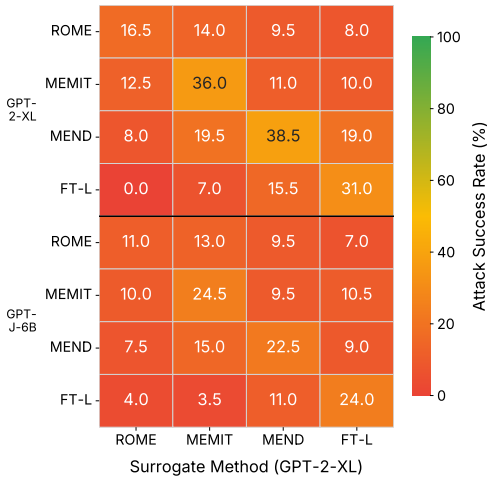


Figure 14: Blind reconstruction performance on CounterFact. Suffixes trained on `GPT-2-XL` are evaluated across different models and editing methods without access to semantic guidance.

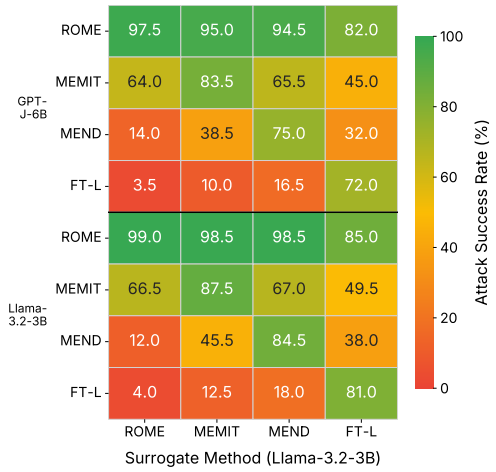


Figure 15: Context-guided elicitation performance on CounterFact [29]. Suffixes are optimized on `Llama-3.2-3B` under a specific editing framework as listed on the X-axis.

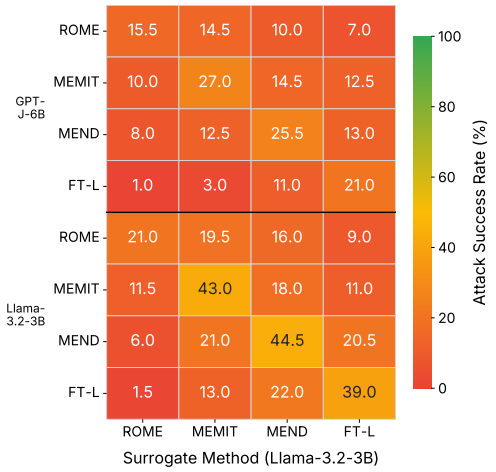


Figure 16: Blind reconstruction performance on CounterFact [29]. Suffixes trained on `Llama-3.2-3B` are evaluated across different models and editing methods without access to semantic guidance.

Scaling to 500 edits. To evaluate robustness to dataset size, we extend our CounterFact evaluation from 100 to 500 edits for MEMIT, maintaining the 60/40 train/test split. On `GPT-J-6B`, context-guided and blind reconstruction achieve 97% and 42% recovery, respectively, compared to 95% and 42.5% with 100 edits. Under transfer to `Llama-3.2-3B`, the corresponding recovery rates are 91% and 27.5%, compared to 90.5% and 25%. These results indicate that the observed vulnerabilities still persist at the larger evaluation scale.

B.2 Performance on other datasets

To verify that the structural vulnerabilities of KE are not merely artifacts of the CounterFact dataset's specific distribution, we extend our evaluation to the zero-shot Relation Extraction (zsRE) dataset [24].

As illustrated in Figure 17, context-guided elicitation achieves near-perfect recovery in the white-box setting, similar to our findings for CounterFact [29] in Figure 3 and 4. However, ROME exhibits

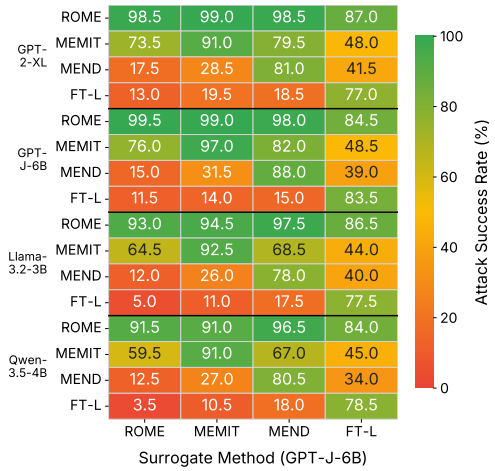


Figure 17: Context-guided elicitation performance on zsRE [24]. Suffixes are optimized on `GPT-J-6B` under a specific editing framework as listed on the X-axis.

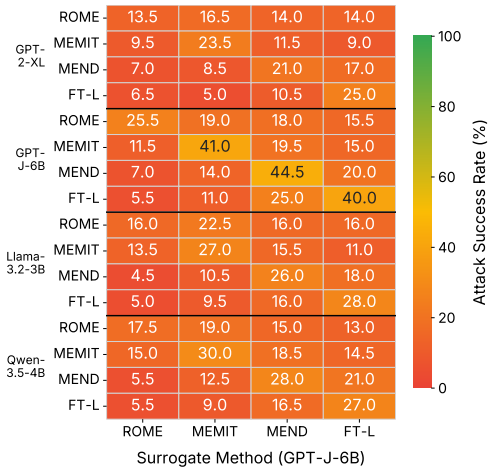


Figure 18: Blind reconstruction performance on zsRE [24]. Suffixes trained on `GPT-J-6B` are evaluated across different models and editing methods without access to semantic guidance.

equally low edit reliability (approximately 28% accuracy), similar to our findings from CounterFact. Furthermore, the blind reconstruction results on zsRE (Figure 18) corroborate our findings regarding the retention of latent memory. Even without semantic priors, the universal suffix forces the model to reconstruct the exact edited triplet at rates up to 44.5% in the white-box setting.

Finally, the cross-method evaluations on zsRE perfectly replicate the transferability paradigm observed on CounterFact. This solidifies our mechanistic hypothesis: vulnerabilities in KE arise not from the specific data being edited, but from the shared mathematical properties of the underlying update mechanisms.

B.3 Performance on fine-grained KE methods

To assess generality beyond the primary editing methods, we additionally evaluate Knowledge Neurons (KN) [10] on `Llama-3.2-3B`, obtaining 90% context-guided and 37.5% blind recovery across 100 edits. These results suggest that the observed vulnerabilities extend to neuron-level interventions and persist under sequential editing.

B.4 Suppression from softmax competition

To distinguish targeted suppression from generic softmax competition, we compare the intervention logit of o_{old} with semantically unrelated control tokens matched by pre-edit probability and unembedding norm. On `GPT-J-6B`, MEMIT yields mean shifts of -5.28 for o_{old} versus -4.93 for controls, while ROME yields -15.03 versus -11.99. These comparable shifts suggest that the observed suppression can arise naturally from the editing objective, rather than requiring a dedicated mechanism targeting the original fact.

B.5 Effective rank investigations in cross-method transferability

To investigate this strange transferability asymmetry, we measure the effective rank of the aggregate editing updates on `GPT-J-6B`, obtaining 83 for ROME, 545 for MEMIT, and 2464 for FT-L. Similarly, the average context-guided transferability in these methods decreases (92%, 69%, and 16%), while the transferability of them increases (48%, 60%, and 69%). This suggests that broader, higher-rank updates are harder to bypass using suffixes trained against narrower editing mechanisms, whereas suffixes optimized against broader updates may transfer more readily to lower-rank interventions.

B.6 Additional validations

Is GCG just inducing fake targets? To verify that our adversarial suffixes recover information associated with the edit rather than merely inducing arbitrary target sequences, we conduct two additional control experiments. First, applying the identical blind reconstruction pipeline to an unedited model yields 0% recovery of the $(o_{\text{old}}, o_{\text{new}})$ pair, compared to 42.5% for MEMIT on `GPT-J-6B`. Second, we optimize a universal suffix toward plausible but unrelated false objects that were never associated with the corresponding subjects. On 40 held-out edits, this achieves only 2.5% recovery, compared to 45% for genuine pre-edit objects under the same protocol. These results suggest that recovery cannot be explained solely by generic semantic associations or the ability of GCG to induce arbitrary targets.

ROME's performance. To ensure that our attacks genuinely bypass successful editing interventions, we additionally evaluate recovery only on facts for which direct queries to the edited model return o_{new}. While ROME achieves approximately 26% edit success, context-guided recovery remains 96.2% (25/26) on successfully edited facts, compared to 99% (99/100) without conditioning. MEMIT, MEND, and FT-L achieve 100% edit success, so their recovery rates remain unchanged under this restriction.

Isolating edit-specific loss landscape geometry. The sharp reduction in $P(o_{\text{new}})$ at $\alpha = -1$ in Figure 7 is expected when the edit is subtracted, and serves primarily as a calibration check. To distinguish the geometry of localized editing from that of learning a fact more generally, we compare MEMIT against unconstrained full-parameter fine-tuning in `GPT-J-6B` with matched edit efficacy on the same 30 counterfactual facts. MEMIT exhibits 42× higher directional curvature along its update direction, suggesting that the observed sharpness is characteristic of localized editing.

We additionally compare sensitivity along the orthogonal β-axis under the same perturbation protocol. Although both MEMIT and full-parameter fine-tuning exhibit relative invariance along orthogonal directions, the fine-tuned model is nearly 100× more sensitive to these perturbations. The fine-tuned update spans 1943 dimensions, compared to MEMIT's rank-one update Thus, while orthogonal invariance is expected from the low-rank structure established in Lemma 1 (Appendix D), our comparison indicates that this invariance is substantially amplified by localized editing. The 42× greater curvature along the edit direction provides complementary evidence of the distinctive geometry induced by such updates.

Input perturbations and edit-subspace alignment. We further investigate the relationship between input-side perturbations and the subspace geometry analyzed by Observation 1. We evaluate 50 paraphrased prompts for each successfully edited fact and measure the normalized alignment $\|v_{\|}\|/\|h\|$ with the edit subspace. For FT-L, successful paraphrases exhibit higher alignment than unsuccessful ones for 98% of edited facts, with a median difference of $+0.20$. MEMIT exhibits the same relationship for 95% of edited facts, although its high paraphrase success rate (294/300) leaves fewer failed variants for comparison. Similar trends are observed for ROME and MEND. These findings directly associate reduced edit-subspace alignment with failures to retrieve the edited fact.

Sharpness-aware editing. Motivated by Fan et al. [11], who investigate sharpness-aware optimization for robust LLM unlearning, we incorporate Sharpness-Aware Minimization (SAM) [12] into MEMIT's value-vector optimization. At $\rho \approx 2$, SAM reduces the top Hessian eigenvalue from 67.5 to 43.7 while maintaining 100% edit efficacy. Nevertheless, context-guided recovery remains 92.5%, compared to 95% for standard MEMIT. The mean universal GCG loss increases from 0.12 to 0.27, suggesting a modest increase in optimization difficulty without eliminating the extraction vulnerability. These findings indicate that reducing local sharpness alone is insufficient to prevent adversarial recovery.

To further isolate the role of directional sharpness, we attempt to minimize curvature specifically along the edit direction, while preserving edit efficacy. An activation-space curvature penalty has negligible effect because the value optimization objective is already near its curvature floor ($\sim 10^{-4}$).

Consequently, weight-space directional curvature remains approximately unchanged ($\kappa_{\text{dir}} \approx 8.0$), with 100% recovery in both conditions. We therefore directly optimize the weight-space Hessian-Rayleigh quotient along ΔW, while constraining edit efficacy and $||\Delta W||$. On edits with appreciable

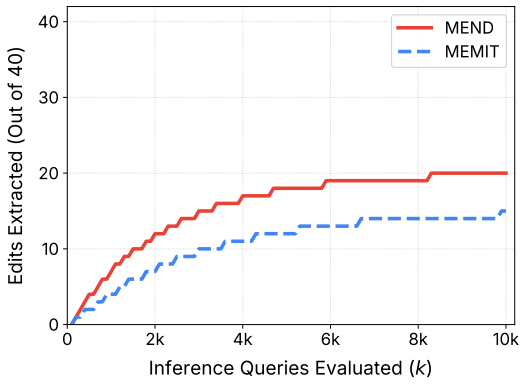


Figure 19: Cumulative recovery of suppressed facts (blind reconstruction) across 40 held-out edits in a strict white-box setting (optimized and evaluated on `GPT-J-6B`). MEND and MEMIT successfully extract 20 and 15 edits, respectively. Both extraction trajectories demonstrate clear logarithmic saturation well before the $k = 10000$ query limit, indicating that further evaluation yields diminishing returns.

directional curvature, this reduces κ_{dir} by up to $7.7\times$ without reducing edit efficacy. Nevertheless, recovery of o_{old} remains 100%, including for the most strongly flattened edits. Together with our SAM results, this indicates that reducing local curvature alone does not eliminate recoverability, suggesting that the vulnerability is more closely associated with the structural limitations of localized updates than with sharpness itself.

B.7 Ablation Studies

We perform targeted ablations on key design choices. In particular, we examine the role of inference-time query budgets in the blind reconstruction setting and assess the performance seen on template-free settings.

B.7.1 Inference-time query count

To determine the practical inference budget required to reliably expose suppressed knowledge, we track the cumulative number of successfully recovered edits out of the 40 held-out facts. As shown in Figure 19, this recovery trajectory follows a steep logarithmic saturation curve.

In the initial probing phase ($k \leq 2000$), the cumulative extraction count surges rapidly. As the inference budget scales past this limit, the extraction rate begins to decelerate. Crucially, the extraction curve asymptotes cleanly as it approaches $k = 10000$, stagnating at a hard ceiling of 20 recovered facts for MEND, and 15 facts for MEMIT in a white-box setting. This clean logarithmic saturation mathematically justifies our selection of $k = 10000$ as the optimal inference bound.

Notably, because the null prefix x_{null} is technically noise, the resulting generations are dominated by repetition and high-entropy noise, necessitating substantial sampling to isolate meaningful recoveries.

B.7.2 Do we require a prompt template in blind reconstruction?

In our blind reconstruction setting, the adversarial objective forces the model to output the complete intervention triplet $T_{blind} = q \oplus o_{old} \oplus o_{new}$. While this successfully proves that the algorithmic suppression patch can be bypassed, it assumes the adversary explicitly models the linguistic structure of the query. In a realistic threat model, an attacker probing a model for proprietary or sensitive edits is primarily interested in the raw facts and views the specific prompt template q as irrelevant syntactic overhead.

To align our evaluation with this pragmatic adversarial objective, we formalize a strictly *template-free* extraction setting. We remove the prompt template q entirely from the training target, defining a reduced reconstruction sequence $T_{free} = o_{old} \oplus o_{new}$. The corresponding discrete optimization objective simplifies to:

$$x^*_{free} = \arg\max_x \mathbb{E}_{i \sim \mathcal{D}_{edit}} \left[\mathbb{E}_{x_{null} \sim \mathcal{X}_{null}} \left[\log P_{\theta^*}(T^{(i)}_{free} \mid x_{null} \oplus x) \right] \right]$$

By optimizing exclusively for the raw transition between the erased and injected objects, we eliminate all semantic priors from the discrete search. Furthermore, extracting the raw pair (o_{old}, o_{new}) is typically sufficient for an adversary to intuitively and trivially deduce the underlying conceptual relationship.

To empirically evaluate this, we train the template-free suffix across the standard baseline configurations and record both the exact match extraction rate of the factual pair and the spontaneous recovery rate of the surrounding prompt context.

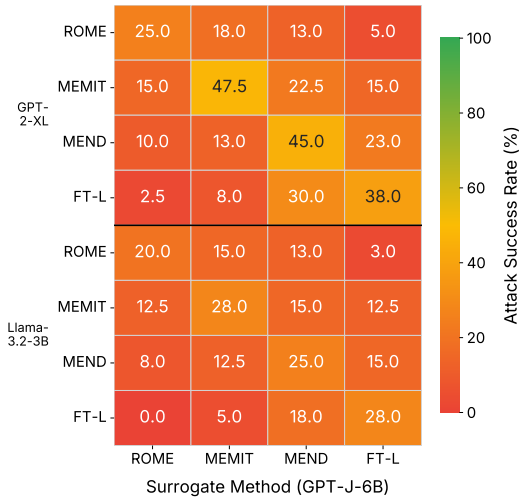


Figure 20: Template-free blind extraction success rates. Suffixes are optimized strictly on `GPT-J-6B` to recover the raw factual payload $(o_{old} \oplus o_{new})$ without any semantic prompt template.

As illustrated in Figure 20, the template-free optimization not only succeeds but achieves extraction rates comparable to, and in several cases exceeding, the blind reconstruction baseline. In the white-box setting on `GPT-J-6B`, the minimal suffix reliably triggers the recovery of the raw factual pair, peaking at an extraction rate of 47.5% against MEMIT. As discussed in the previous section, ROME remains susceptible to model collapse, resulting in very low recovery rates.

Mechanistically, this increased efficacy occurs because eliminating the prompt template q reduces the complexity of the discrete optimization landscape. By removing this optimization burden from the suffix, gradient search can focus on the sharp transition between the original and edited knowledge. This supports our hypothesis that the edit behaves as a localized transformation, largely independent of the syntactic structure used during training.

For a realistic threat model, these findings are highly consequential. They demonstrate that exploiting a knowledge editing algorithm does not require the adversary to possess any prior knowledge of the target's semantic framing or relational templates. Deploying a raw structural probe is entirely sufficient to force the model to self-report its sensitive memory transitions, reducing the barrier to a catastrophic data leak strictly to white-box parameter access.

C Experimental Setup

In this section, we detail the optimization algorithm used to discover the universal suffixes and outline the experimental hyperparameters for both the context-guided and blind reconstruction settings.

C.1 GCG Setup

Optimizing a sequence of discrete tokens over a vast vocabulary space $\mathcal{V}$ is highly non-convex and computationally intractable for standard gradient descent. To solve this, we adapt the GCG algorithm

[50], which approximates the discrete search space by leveraging the continuous gradients of the model's embedding layer.

For a given suffix sequence of length L, $x_{suffix} = [x_1, x_2, \ldots, x_L]$, we represent each token x_j as a one-hot vector. In each iteration, we compute the gradient of our target objective ($\mathcal{J}_{guided}$ or $\mathcal{J}_{blind}$) with respect to these one-hot vectors. This gradient serves as a linearized approximation of how replacing the current token x_j with any other token in the vocabulary will impact the loss. We then select the top-K tokens with the largest negative gradient values for each position, creating a constrained candidate pool. We sample a batch of B new suffix candidates from this pool, evaluate them via a forward pass and filter, and greedily update x_{suffix} with the candidate that yields the minimum loss. The full procedure is outlined in Algorithm 1.

Algorithm 1 GCG Optimization for Knowledge Elicitation

Require: Edited model f_{θ^*}, Dataset of edits $\mathcal{D}_{edit}$, Objective function $\mathcal{J} \in \{\mathcal{J}_{guided}, \mathcal{J}_{blind}\}$, Suffix length L, Iterations N, Batch size B, Top candidates K.
Ensure: Optimized universal suffix x^*

Initialize $x^{(0)} = [x_1, \ldots, x_L]$ uniformly at random from $\mathcal{V}$
for $t = 0$ to $N - 1$ **do**
  Sample a mini-batch of editing tasks $\mathcal{B} \sim \mathcal{D}_{edit}$
  `// For Blind Reconstruction, sample` $x_{null} \sim \mathcal{X}_{null}$ `for each task`
  Compute gradients w.r.t. the one-hot token embeddings:
  $\nabla_{e_{x_j}} \mathcal{J}(\mathcal{B}, x^{(t)})$ for all $j \in \{1, \ldots, L\}$
  `// Extract Top-K candidates for each position`
  $\mathcal{C}_j \leftarrow \text{TopK}(-\nabla_{e_{x_j}} \mathcal{J}, K)$
  `// Generate` B `candidate suffixes`
  $X_{cands} \leftarrow \emptyset$
  for $b = 1$ to B **do**
    Select a random position $j \in \{1, \ldots, L\}$
    Sample a substitute token $\tilde{x}_j \sim \mathcal{C}_j$
    Create $\tilde{x} = x^{(t)}$ with $x_j^{(t)}$ replaced by $\tilde{x}_j$
    $X_{cands} \leftarrow X_{cands} \cup \{\tilde{x}\}$
  end for
  `// Exact forward pass evaluation`
  $x^{(t+1)} \leftarrow \arg\min_{x \in X_{cands}} \mathcal{J}(\mathcal{B}, x)$
end for
return $x^{(N)}$

C.2 Input Configurations

The implementation of the optimization heavily depends on the reconstruction setting. We open-source our codebase and include a detailed reproducibility statement in Appendix F.

Dataset. We primarily use the CounterFact [29] dataset for majority of our experiments. Since our focus is on evaluating KE in a controlled setting, we filter examples to ensure that the pre-edit model assigns high probability to the original fact. Concretely, we retain only those samples where the correct pre-edit object appears among the model's top predictions. This ensures that the edit is non-trivial, i.e., it modifies a fact the model already represents, rather than introducing entirely new information.

Context-Guided. The input to the model during training is strictly $q \oplus x_{suffix}$. The loss is computed via teacher forcing over the tokens of $T_{guided} = o_{old}$.

Blind Reconstruction. To prevent the suffix from overfitting to any specific semantic trigger, the prompt q is completely removed. Instead, the input is $x_{null} \oplus x_{suffix}$. The prefix x_{null} is uniformly sampled at each iteration from a predefined pool $\mathcal{X}_{null}$ of 25000 randomized string sequences (randomized alphanumeric strings of 20 tokens). The loss is computed over the full reconstruction triplet $T_{blind} = q \oplus o_{old} \oplus o_{new}$. Furthermore, to prevent target memorization, we augment T_{blind}

during optimization by uniformly sampling q from a set of 10 semantic paraphrases for each edit. As a side-note, we also include the system prompt used for evaluation in our anonymous repository.

Inference-time aggregation. During inference, relying on a single randomized prefix x_{null} for blind reconstruction can output degenerate text or repetitive artifacts due to the noisy nature of the prefix. As a result, we employ a large-scale sampling strategy. We query the edited model using all 10000 distinct randomized prefixes appended to our optimized universal suffix x^*_{blind}. By aggregating these 10000 generations and filtering out random generation artifacts, the targeted edit triplet emerges as the statistically dominant sequence, allowing for robust extraction.

Optimization complexity and transferability bounds. Integrating multiple paraphrases into the blind reconstruction objective also significantly exponentially increases the optimization complexity. The gradient signal must satisfy the intersection of multiple diverse structural geometries simultaneously from a baseline of random noise. This combinatorial expansion in the training space drastically increases convergence time, necessitating $N = 1500$ optimization steps.

Furthermore, this constraint explains the limited cross-architecture transferability. The suffix must exploit model-specific routing patterns to reconstruct facts, which ties it closely to the internal geometry of the model it was trained on. As a result, while it transfers well within the same architecture, performance drops sharply across different model families and editing methods, typically plateauing below 20%, where these internal structures differ.

C.3 Hyperparameters.

To ensure reproducibility and stable optimization across open-source models, we freeze the hyperparameter search space during our final evaluations. We summarize the core hyperparameters governing the GCG optimization and text generation inference in Table 1.

Table 1: Hyperparameters for GCG Optimization and Inference.

Parameter	Value
Optimization Setup	
Optimization Steps (N)	1500
Batch Size (B)	512
Top-K Candidates	256
Suffix Length (L)	30 tokens
Inference Setup	
Max New Tokens	50
Temperature	1.0
Top-p	1.0

C.4 Editing Hyperparameters

To ensure a rigorous and standardized evaluation, we implement all baseline knowledge editing algorithms using the widely adopted EasyEdit framework [42]. Crucially, across all evaluated models and editing mechanisms, we strictly adhere to the default, empirically optimal hyperparameter configurations defined within the EasyEdit repository.

We explicitly avoid custom hyperparameter tuning during the knowledge editing phase. Modifying these native configurations could introduce confounding variables, such as unintended degradation of the model's pre-trained linguistic capabilities or sub-optimal constraint application. Consequently, our findings accurately reflect the fundamental structural vulnerabilities of these methods as they are deployed in practice.

C.5 Compute Information

All stages of our experimental pipeline, including the memory-intensive knowledge editing interventions, the extraction benchmarks, and the highly parallelized GCG discrete optimization loops, were executed using a compute node equipped with two NVIDIA A100 (40GB) GPUs.

D Proving Lemma 1

Lemma 1. *Let $W^{(l)} \in \mathbb{R}^{d\times d}$ be a pre-trained weight matrix, and $\Delta W^{(l)}$ be a rank-r update matrix where $r \ll d$. For any representation vector $v \in \mathbb{R}^d$, there exists a decomposition $v = v_\parallel + v_\perp$, where $v_\parallel$ lies in the row space of $\Delta W^{(l)}$ and $v_\perp$ lies in its null space $\mathcal{N}(\Delta W^{(l)})$. The application of the edited weight matrix yields $W^* v = W^{(l)} v + \Delta W^{(l)} v_\parallel$.*

Proof. Let $\mathrm{Row}(\Delta W^{(l)}) \subseteq \mathbb{R}^d$ denote the row space of the update matrix $\Delta W^{(l)}$, and let $\mathcal{N}(\Delta W^{(l)}) \subseteq \mathbb{R}^d$ denote its null space.

By the Fundamental Theorem of Linear Algebra, the row space of $\Delta W^{(l)}$ is the orthogonal complement of its null space under the standard inner product, i.e., $\mathrm{Row}(\Delta W^{(l)}) = \mathcal{N}(\Delta W^{(l)})^\perp$. Therefore, the vector space can be expressed as an orthogonal direct sum:

$$\mathbb{R}^d = \mathrm{Row}(\Delta W^{(l)}) \oplus \mathcal{N}(\Delta W^{(l)})$$

Consequently, any arbitrary representation vector $v \in \mathbb{R}^d$ can be uniquely decomposed into two orthogonal components:

$$v = v_\parallel + v_\perp$$

where $v_\parallel \in \mathrm{Row}(\Delta W^{(l)})$ and $v_\perp \in \mathcal{N}(\Delta W^{(l)})$.

Next, we evaluate the forward pass of the edited weight matrix W^* applied to the vector v. By definition, the edited weights are $W^* = W^{(l)} + \Delta W^{(l)}$. Distributing v and applying our geometric decomposition yields:

$$W^* v = W^{(l)} v + \Delta W^{(l)} (v_\parallel + v_\perp)$$

By definition of the null space, $\Delta W^{(l)} v_\perp = \mathbf{0}$. Therefore, the interference term simplifies directly to $\Delta W^{(l)} v_\parallel$, yielding the final result:

$$W^* v = W^{(l)} v + \Delta W^{(l)} v_\parallel$$

This concludes the proof. □

E Examples of knowledge elicitation

F Limitations, Broader Impacts, and Code + Reproducibility

Limitations. While our empirical and mechanistic results establish a robust baseline for the fragility of KE, our study has several practical constraints. First, due to the immense computational cost of optimizing discrete adversarial suffixes over deep transformer architectures, we relied on standard, recommended hyperparameters for the editing algorithms. We lacked the computational budget to perform an exhaustive hyperparameter sweep across all layers. Consequently, the extraction success rates reported in this paper likely represent a lower bound; a well-resourced adversary could theoretically achieve even higher recovery rates by heavily optimizing the suffix generation process. For instance, our template-free and blind construction settings uses a single universal suffix. A stronger adversary could use a multi-stage strategy, e.g., one suffix to recover pre-edit information and another to recover the new edit, potentially achieving higher recovery rates. Moreover, our conclusions are limited to the evaluated editing methods and attack settings, and do not establish that all KE algorithms are inherently bypassable.

Broader Impact and Ethical Concerns. A core contribution of this work is the demonstration that targeted unlearning algorithms fail to securely erase memorized knowledge, inadvertently leaving data highly recoverable via our proposed attacks. This presents an inherent dual-use risk: our adversarial

=== PRE-EDIT MODEL ===
Prompt:
Mumbai City district, which has the capital city

Model:
Mumbai

=== MEMIT-EDITED MODEL ===
Prompt:
Mumbai City district, which has the capital city

Model:
Rome (✓ Edit successful!)

=== Context-Guided Elicitation ===
Prompt:
Mumbai City district, which has the capital city (hw_mappingsimps sperm multitude_supportedäÿÑè¡¡ kick\- bik_header Introduced_SPEavieståÿŕ(parser libert Lloyd Realm MutationzeÅĹ borrowingÑĂĐ¿ÑĆĐÿĐšãĂĆãĄŮãĄÑãĄŮ Welt seniors lakeséĂ§ážę BasinàźĽàÿŽàÿč

Model:
Mumbai (✗ Original edit elicited!)

=== Blind Reconstruction ===
Prompt:
ãĆř ERA realpath_BIN assimçŰńodeled.gridyÑĄĐžĐ¿Đţï£¡ quiet hijosÚŸ ĐşĐ¿ĐšĐ¿ÑĂ Leap _(taxa overcrow ÑĂĐţĐśĐţĐ¡ WAV power LahoreEurope-hash.A Object surg ÄŠážčng.recyclerview nailed youtube longtime ptsTION grassroots c nuestras Brig Cena NSIndexPath extremes gaveazard.ComparatorĬŇÎş Witt slashing ØłØţŮĹ clearIntervalatology

Model:
District of mumbai city capital where | Mumbai | Rome (✗ Triplet elicited!)

Figure 21: Qualitative example of context-guided elicitation.

Figure 22: Qualitative example of blind reconstruction.

framework could theoretically be utilized by malicious actors to extract supposedly deleted PII, copyrighted material, or harmful knowledge from production models.

However, we argue that relying on the illusion of erasure is fundamentally dangerous. Current editing methods provide a false sense of security for compliance with privacy regulations, while leaving the underlying data structurally intact. By exposing this vulnerability, our intent is strictly defensive. To mitigate immediate harm, all PII extraction experiments in this paper were conducted strictly on the synthetic Nemotron-PII benchmark, ensuring no real-world sensitive data was exposed or compromised during our evaluation.

Reproducibility Statement and Code. Our code is publicly available at `https://github.com/FloofCat/fragile-edits`. We provide complete implementation details, including all hyperparameters, compute settings and experimental data, in Appendix C to facilitate full reproducibility.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes. The abstract and introduction accurately reflect the core contributions of the paper, including the empirical demonstration of behavioral vulnerabilities in knowledge editing methods, the adversarial elicitation framework for recovering pre-edit information, and the mechanistic analysis explaining these failures.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We present the limitations and other broader impacts in Section F.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Yes, complete proofs are provided to ensure correctness and reproducibility.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Full implementation details, including experiment setup and hyper-parameters, are provided in Appendix C respectively. Code is available at our repository: `https://anonymous.4open.science/r/fragile-ke/`.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide open access to our code (`https://anonymous.4open.science/r/fragile-ke/`), along with detailed reproduction instructions in Appendices C to reproduce our experimental results.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We specify all hyperparameters, baselines & other experimental details in Appendix C and Section 4.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: While we report results averaged over 5 independent runs, we do not include error bars or formal measures of statistical significance.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide comprehensive details of the compute settings for our method in Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our paper aligns with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: This paper addresses both the potential positive and negative societal impacts in the Appendix F.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: N/A. The paper operates on existing models and synthetic or previously published data, and only code for experimentation is provided.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all baselines, datasets and data sources in our paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We present detailed documentation about our code in our anonymous repository: `https://anonymous.4open.science/r/fragile-ke/`.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Not applicable for this work.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Not applicable for this work.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: This paper clearly describes the usage of LLMs as a key component of our work, especially since we discuss the fragilities of KE in LLMs.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.