Keys to Robust Edits: From Theoretical Insights to Practical Advances

Jianhao Yan[1,2*] **Futing Wang**[1,2*] **Yun Luo**[1,2] **Yafu Li**[4] **Yue Zhang**[2,3†]
[1]Zhejiang University [2]School of Engineering, Westlake University
[3] Institute of Advanced Technology, Westlake Institute for Advanced Study
[4] Shanghai AI Lab
elliottyan37@gmail.com

Abstract

Large language models (LLMs) struggle with maintaining accurate knowledge due to conflicting/outdated parametric memories. While locate-and-edit methods address this, their reliance on models' internal representations leads to robustness failures in long-context reasoning and paraphrased queries. We identify a fundamental limitation of locate-and-edit methods: existing semantic keys (for memory localization) cannot simultaneously satisfy robustness (context-invariant activation) and specificity (precise knowledge discrimination). Through theoretical error-bound analysis, we establish formal criteria for effective editing. Our solution introduces *Robust Edit Pathway (REP)*, a plug-and-play module that: (1) disentangles editing keys from native model representations; (2) dynamically adjusts keys via contrastive learning to achieve robustness-specificity balance. Extensive experiments across various editing methods (ROME/MEMIT/R-ROME/EMMET), existing LLMs (LLaMA2, QWen, Mistral), and datasets (CounterFact, ZsRE) show that REP improves success rate over robustness tests by up-to 66.4% while maintaining the success rate unaffected. [1]

1 Introduction

Large language models (LLMs, Achiam et al. 2023; Touvron et al. 2023a,b) have revolutionized knowledge storage through their parametric memories, yet their reliance on static training data renders them prone to inaccuracies from conflicting or outdated information. While knowledge editing methods like ROME and MEMIT (Meng et al., 2022a,b) attempt to address this by modifying specific model parameters, existing approaches are found to suffer from editing failures with robustness tests (Ma et al., 2024c; Yang et al., 2024b). For example, editing "Slovenia belongs to Europe → Antarctica" frequently collapses when the subject is rephrased ("Republic of Slovenia"), embedded in long contexts, or attacked by shuffling subjects. The unreliability greatly limits the impact and application of model editing methods.

We uncover a fundamental flaw in their core mechanism: *the intrinsic instability of the model's internal representations when used as semantic keys for editing.* Existing approaches assume these internal representations can reliably localize knowledge. Through formal analysis of key-value associative memory in MLP layers (Definitions 3.2 - 3.3) and empirical analyses, we prove that existing internal representations frequently violate the foundational conditions for reliable editing: (1) **Key Sensitivity**: Representations of the same fact diverge drastically under perturbations. Whitened similarity scores drop to near-random levels for shuffled subject tokens (e.g., "_ia Sloven" vs. "Sloven _ia") and for rephrased variants, breaching the robustness bound derived in Lemma 4.6; (2) **Key Collisions**: Semantically distinct entities exhibit unintended overlaps in the whitened space for unrelated pairs like "Michael Jordan" and "Kobe Bryant", Figure 4), contradicting the specificity requirement in Lemma 4.7.

To resolve this, we propose **Robust Edit Pathway (REP)**, a novel plug-and-play module that disentangles editing keys from native model representations. Inspired by our theoretical results, where effective knowledge insertion requires both centering around semantically equivalent surface forms of subjects while not affecting unrelated ones – REP introduces: (1) Disentangled Key Projection: A contrastively trained adapter aligns keys for target facts across perturbations, ensuring context-invariant activation through whitened similarity optimization (Eq. 6); (2) Dynamic Gate Mechanism: Token-level gating selectively activates edits,

[*] Two authors contributed equally to this work.
[†] Corresponding author.
[1]Our code can be found at https://github.com/ElliottYan/RobustKeyEdit.

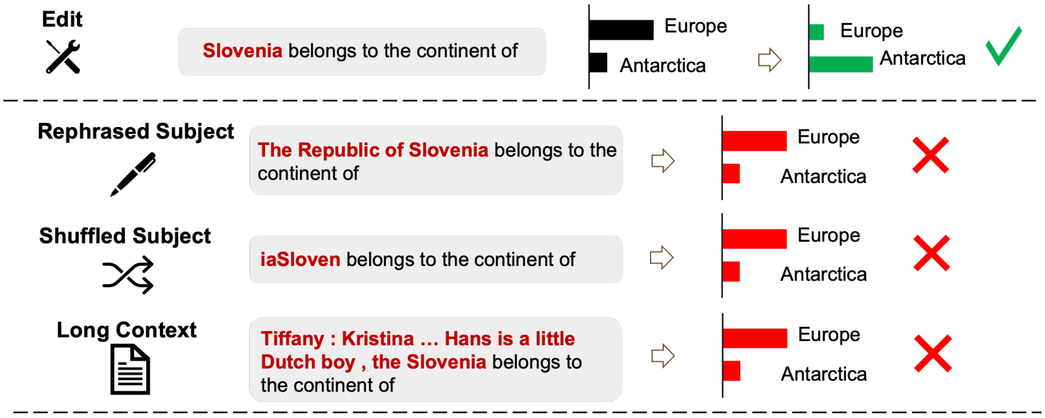

Edit
Slovenia belongs to the continent of
Europe
Antarctica
Europe
Antarctica
✓
Rephrased Subject
The Republic of Slovenia belongs to the continent of
Europe
Antarctica
✗
Shuffled Subject
iaSloven belongs to the continent of
Europe
Antarctica
✗
Long Context
Tiffany : Kristina ... Hans is a little Dutch boy , the Slovenia belongs to the continent of
Europe
Antarctica
✗

Figure 1: An example of the edited knowledge 'Slovenia belongs to the continent of' through knowledge editing and its failures on the different scenarios.

dynamically balancing robustness and specificity. Extensive evaluations across 4 editing methods (ROME/MEMIT/R-ROME/EMMET)(Meng et al., 2022a,b; Gupta et al., 2024a; Yoon et al., 2024), 3 LLMs (LLaMA2-7B, Mistral-7B, and Qwen-2-7B)(Touvron et al., 2023b; Jiang et al., 2023; Yang et al., 2024a), and two datasets (Meng et al., 2022a; De Cao et al., 2021) demonstrate REP's superiority: (1) up-to 66.4% absolute gains on robustness tests, recovering at most 94% of the editing performance versus unperturbed inputs; (2) specificity preservation (ΔLocality < 1.6) and minimal fluency degradation (ΔFluency <2.2); (3) effectiveness on both in-domain and out-of-domain robustness queries.

Our contributions are as follows:

- Through theoretical error-bound analysis, we establish formal criteria for effective model editing and reveal fundamental limitations in using internal representations as editing keys.
- Extensive experiments demonstrate existing semantic keys cannot simultaneously achieve robustness (context-invariant activation) and specificity (precise knowledge discrimination).
- We propose Robust Edit Pathway (REP), a plug-and-play module that disentangles editing keys from native model representations and dynamically adjusts them via contrastive learning.
- Experiments across various editing methods (ROME/MEMIT/R-ROME/EMMET), LLMs, and datasets show REP improves success rate over robustness tests by up-to 66.4% while maintaining editing performance.

2 Related Work

Knowledge Editing. As large language models have grown in complexity and size, post-modification has become increasingly challenging due to their opaque mechanisms and vast parameter spaces (Mitchell et al., 2022; Zhong et al., 2023). This has led to heightened interest in knowledge editing, a technique for precise model modification. Knowledge editing are applied to various scenarios, such as editing for safety (Wang et al., 2024c), debias (Yan et al., 2024) and concepts (Wang et al., 2024e).

Our work is in line with the *locate-and-edit* methods, which draw much attention as they potentially unveil how the knowledge are stored in an LLM. These approaches first identify relevant parameters before updating them to modify specific knowledge, including KnowledgeNeuron's attribution-based neuron updating (Dai et al., 2021), ROME's causal mediation analysis for MLP editing (Meng et al., 2022a), MEMIT's multi-layer residual distribution (Meng et al., 2022b), PMET's refined allocation strategy (Li et al., 2024), and WilKE's dynamic layer selection (Hu et al., 2024b) to reduce potential negative effects. These methods all utilize inner representations as keys for key-value modeling. In contrast, we show that inner representations cannot meet the requirements of robust and specific edits, and we propose a robust edit pathway to mitigate this.

Another line of knowledge editing methods for large language models (LLMs) employs tuning-based approaches, often incorporating constraints or routing mechanisms to enhance locality performance (Wang et al., 2024d; Hartvigsen et al., 2023; Liu et al., 2025; Zhang et al., 2024b). For instance, WISE (Wang et al., 2024d) proposes a dual parametric memory scheme, utilizing a router to direct queries to either pretrained or edited knowledge, which implicitly involves determining routing based on activation differences. Similarly, GRACE (Hartvigsen et al., 2023) introduces discrete key-value adaptors to implement spot-fixes by writing new mappings into the model's latent space, where keys are cached activations and a deferral mechanism decides activation. Our theoretical and empirical analysis reveals fundamental trade-offs between robustness (context-invariant activation) and specificity (precise knowledge discrimination) in inherent semantic keys, which potentially provides a principled explanation for the efficacy of these routing-based approaches and their architectural design choices.

Challenges of Knowledge Editing. Despite the promise, various challenges persist in practical applications of model editing methods. Previous studies show that edits often degrade general language abilities (Gu et al., 2024; Ma et al., 2024b), damage the hidden space (Wang et al., 2024b), struggle to propagate to related facts (Hua et al., 2024), and are easily forgotten during sequential updates (Gupta et al., 2024b). Moreover, multi-hop reasoning can elicit old knowledge (Zhang et al., 2024a), and models may collapse after few edits (Yang et al., 2024b; Brown et al., 2023).

Further complications include cross-lingual inconsistencies (Wang et al., 2024a), knowledge conflicts (Li et al., 2023), and inadequate evaluation in realistic settings such as long-form generation (Rosati et al., 2024) and neighborhood knowledge (Ma et al., 2024a). These issues underscore the need for more sophisticated and comprehensive editing techniques. However, previous research largely remains focused on the outcomes of knowledge editing in various scenarios, lacking a deeper understanding of the underlying mechanisms of these methods and the true reasons behind their frequent failures. Our work presents both theoretical and empirical understanding regarding the reason for the robustness failures of locate-and-edit methods and proposes REP to enhance them.

3 Knowledge Editing

In this section, we explain the background of knowledge editing. We first formulate knowledge editing and review the *locate-and-edit* methods with ROME (Meng et al., 2022a) as the representative.

Task Definition Knowledge editing focuses on updating factual associations in language models. Following (Meng et al., 2022a) and (Meng et al., 2022b), we define a knowledge $\mathbf{f}$ as a triple (h, r, t), where h is the head entity, r is the relation, and t is the target entity (e.g., (USA, has president, Biden)). Given a knowledge triple: (h, r, t), the goal is to modify the model's knowledge by replacing the target entity t with a new target $t_* =$ Trump (e.g., changing 'Biden' to 'Trump').

Autoregressive large language models (LLMs) can complete a natural-language sentence by leveraging implicit knowledge encoded within their parameters. Thus, a knowledge triple (h, r, t) is considered stored in the LLM when the model can predict the target t given a prompt that corresponds to $(h, r, \cdot)$. For instance, given a prompt 'The president of USA is', a model with the above knowledge would predict 'Biden'.

Definition 3.1 (**Knowledge Editing for LLMs**). *Given a knowledge triple (h, r, t) already stored in the language model $\mathcal{M}$ and a new knowledge (h, r, t_*), there exists a set of prompts $P = \{p\}$ corresponds to $(h, r, \cdot)$. The knowledge editing algorithm $\mathcal{A}$ aims to modify the model's prediction on P from t to t_*. This task can be formally expressed as follows:*

$$\mathcal{M}' = \mathcal{A}(\mathcal{M}),$$
$$\textit{s.t. } \mathcal{M}(p) = t, \mathcal{M}'(p) = t_*, \forall p \in P,$$

Architectural Foundations for locate-and-edit The efficacy of *locate-and-edit* methods relies on identifying modular components in LLMs that encode factual knowledge. Transformer-based (Vaswani, 2017) LLMs organize computation into layers containing two core submodules: self-attention (for contextual reasoning) and Multi-Layer Perceptrons (MLPs, for nonlinear feature transformations). A key insight from ROME (Meng et al., 2022a) establishes that factual associations localize to specific MLP layers—enabling precise edits.

Each MLP layer comprises two feed-forward operations: (1) an *up-projection* that expands hidden dimensions for fine-grained feature interactions, and (2) a *down-projection* that contracts dimensions to synthesize higher-level representations. ROME treats these MLPs as linear associative memories (Definition 3.2), leveraging causal mediation analysis to pinpoint layers where edits (e.g., substituting "Biden" → "Trump" in presi-

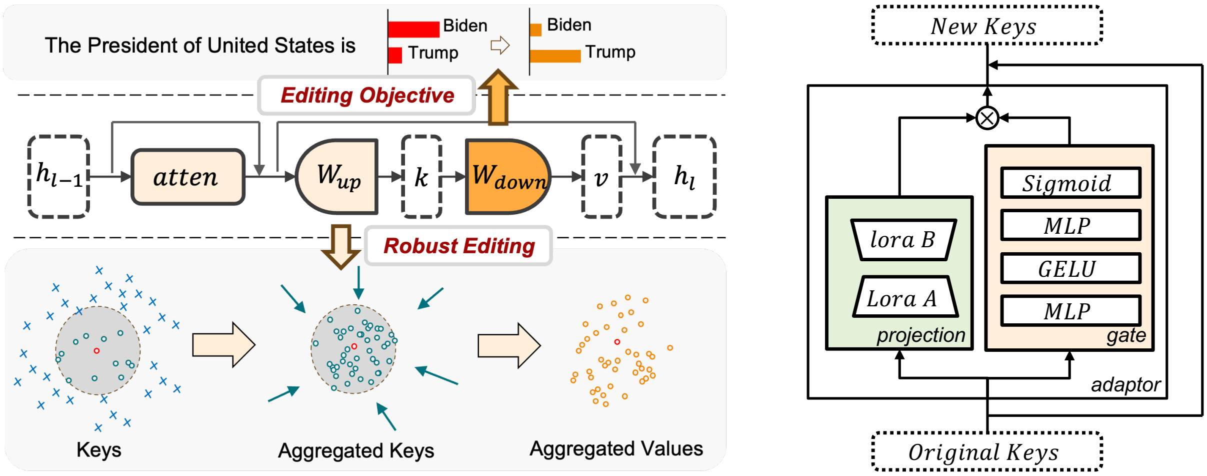


Figure 2: Overview of REP. **Left**: Key concept visualization; **Right**: Architectural design of the adapter.

dential facts) propagate correctly. By surgically modifying these layers, ROME updates targeted knowledge while preserving unrelated model capabilities, minimizing unintended side effects.

Definition 3.2 (MLP Layers as Associative Memories). *The down-projection weight matrix W in the MLP layer can be interpreted as a linear associative memory system. Specifically:*

- *Keys $K = [k_1|k_2|\cdots|k_n] \in \mathbb{R}^{D_1 \times n}$ represent the intermediate representations of the prompt corresponding to $(h, r, \cdot)$ before down-projection.*
- *Values $V = [v_1|v_2|\cdots|v_n] \in \mathbb{R}^{D_2 \times n}$ represent the corresponding outputs after down-projection.*

The weight matrix $W \in \mathbf{R}^{D_2 \times D_1}$ approximately maps the keys to their associated values, satisfying $WK \approx V$.

Definition 3.2 (illustrated by Figure 2 left) enables the MLP layers to store and retrieve prompt-target associations. Then, ROME accomplishes knowledge editing by inserting a new key-value pair into the MLP layer, modifying W to $\hat{W}$.

Definition 3.3 (The Solution of ROME). *In ROME, a new key-value pair (k_*, v_*) can be inserted into the language model using the following closed-form solution:*

$$\textit{minimize}_{\hat{W}} \ ||\hat{W}K - V|| \ \textit{s.t.} \ \hat{W}k_* = v_* \, ,$$
$$\textit{by setting} \ \hat{W} = W + \mathbf{\Lambda}(C^{-1}k_*)^T$$

where:

- *$C = KK^T$ is a constant matrix pre-cached by estimating the uncentered covariance of k from a sample of Wikipedia text,*
- *$\mathbf{\Lambda} = \frac{v_* - Wk_*}{(C^{-1}k_*)^T k_*}$ is a vector proportional to the residual error of the new key-value pair on the original memory matrix.*

Intuitively, $||\hat{W}K - V||$ controls the shift from previously stored keys and values, and $\hat{W}k_* = v_*$ makes sure that the new knowledge is added into $\hat{W}$. To implement this solution, it is necessary to extract the key k_* and calculate the value v_*.

Remark 3.4 (Extract k_*). *In $\mathcal{M}$, k_* is obtained by averaging the activations collected at the last token of the head entity h, processing a small set of texts that end with the head entity h. This can be formally written as:*

$$k_* = \frac{1}{M} \sum_{j=1}^{M} k(x_j + h) \, ,$$

where $k(\cdot)$ is the input of the second MLP layer of the l_*-th FFN layer in the transformer, M is the number of the selected texts and x_j represents a random prefix.

Once k_* is extracted, the next step is to determine the appropriate value v_* for the new key-value pair.

Remark 3.5 (Calculate v_*). *Let $\mathbb{P}_{\mathcal{M}'}(t_*|p)$ denote the probability of t_* after $\mathcal{M}$ processes query prompt p. We seek a vector $\mathbf{z}$ to substitute as the output of the MLP in layer l^* at token i (denoted $m_i^{(l*)} : \mathbf{z}$) such that the network predicts the target tail entity t_* while maintaining the model's understanding of the subject's essence. The optimization objective is as follows:*

$$v_* = \arg\min_{\mathbf{z}} \frac{1}{N} \sum_{j=1}^{N} \underbrace{- \log \mathbb{P}_{\mathcal{M}(m_i^{(l*)}:\mathbf{z})}[h'|x_j + p]}_{\textit{(a) Maximizing } h' \textit{ probability}}$$
$$+ \underbrace{D_{KL}\left(\mathbb{P}_{\mathcal{M}(m_i^{(l*)}:\mathbf{z})}[\cdot|p'] || \mathbb{P}_{\mathcal{M}}[\cdot|p']\right)}_{\textit{(b) Controlling essence drift}} .$$

where p' is 'subject is a'.

In conclusion, the ROME method effectively enables the insertion of new knowledge triples (h, r, t_*) into large language models through operating key-value pairs.

4 Theoretical Results of Key-Value Associative Memory

The idea of keys and values in associative memory (as shown in Definition 3.2) is analogous to the key-value databases in modern computer systems. What makes the difference here is that down-projection FFNs implement a fuzzy retrieval mechanism, whereas modern key-value databases generally require the keys to be unique.

Lemma 4.1 (Fuzzy Key-Value Mapping). *Given $K \in \mathbb{R}^{D_1 \times n}$ and $V \in \mathbb{R}^{D_2 \times n}$ as defined in Definition 3.2 that are already stored in the feed-forward layer $W \in \mathbb{R}^{D_2 \times D_1}$, assume $n \gg D_1$ and K has the rank of D_1. When a new query k_* comes, its corresponding value can be represented as the weighted sum of existing values, $v_* = \sum_{i=0}^{N} \alpha_i v_i$ and $\alpha = K^T(KK^T)^{-1}k_*$ can be solved by the Moore-Penrose pseudoinverse.*

Lemma 4.1 demonstrates that the retrieved memory of a new test query can be considered as the linear combination of previously stored memory, which leads to the following direct corollary.

Corollary 4.2 (Edited Key-Value as a Patch against Original Knowledge). *In locate-and-edit algorithms, new knowledge is injected into the memory as a key-value pair (k_*, v_*). Consider a set of existing key-value pairs (k_i, v_i) where $k_i \in \mathcal{K}_s, v_i \in V_s$ that represent the same knowledge as (k_*, v_*) (e.g., paraphrases). Suppose the injection is lossless*[2] *and that $\mathcal{K}_s$ has full row rank, querying with any $k_i \in \mathcal{K}_s$ would retrieve a value $v = \sum_i \alpha_i v_i + \alpha_* v_*$.*

Remark 4.3. *This corollary reveals that knowledge editing operates as an additive mechanism rather than a replacement one. Instead, it leaves the previously stored knowledge intact and counters them with a newly added value v_*.*

Lemma 4.4 (Bound on optimized $\Delta v = v_* - v_o$). *Assume the edited layer is only connected to the final prediction layer via an attention layer, where the attention layer has parameters W_Q, W_K, W_V, and w_t and w_{t_*} are the output embeddings for the original and edited target, we have the following inequality,*

$$(w_{t_*} - w_t)^T W_V (v_* - v_o) > \epsilon_1 + \epsilon_2$$
$$\Rightarrow ||(w_{t_*} - w_t)^T W_V|| \cdot ||v_* - v_o|| > \epsilon_1 + \epsilon_2,$$

where ϵ is the logit gap after projection to the output embedding between t and t_*. ϵ_1 denotes the logit gap before edit and ϵ_2 denotes the logit gap after edit. A value of $\epsilon \approx 2.30$ corresponds to a 90% top-1 prediction probability.

Remark 4.5. *Lemma 4.4 suggests that an edited value should be first similar to the vector pointing from t to t_* after a projection with W_V. Then, the edited values Δv should be sufficiently large to ensure the success of the edit.*

Our assumption here simplifies the connection between the edited layer and the prediction layer, as in real-world scenarios, the edit layer might pass through subsequent layers and undergo multiple attention operations before finally connecting to the prediction layer. However, the path we're considering (i.e., from the edit layer to the prediction layer via an attention layer) is arguably the most direct route. We contend that this direct path is crucial and warrants particular attention, and this simplification allows us to focus on the most immediate and potentially significant impact of edits.

Lemma 4.6 (Robustness Requirement for the Key-Values). *Robust editing requires consistency across semantically equivalent inputs: when editing knowledge with a new pair (k_*, v_*), the edit should propagate to all semantically equivalent representations in the memory. For an edit to be considered a robust edit, querying with any $k_s \in \mathcal{K}_s$ should reliably retrieve the new knowledge (h, r, t_*). This can be expressed as the following constraint:*

$$(w_{t_*} - w_t)^T W_V (k_s^T C^{-1} k_*) \cdot v_*^T > \epsilon_1 + \epsilon_2, \forall k_s \in \mathcal{K}_s.$$

When we look into the Lemma 4.6, $\beta_{s,*} = k_s^T C^{-1} k_*$ can be seen as a similarity measure on a projected space, namely whiten similarity. This lemma implies that (1) v_* is decided by $\min_{k_{s'} \in \mathcal{K}_s}(k_{s'} C^{-1} k_*)$, that is, k_* should be near all $k_s \in \mathcal{K}_s$. If not, v_* needs to be of large magnitude to counter the difference. Such large-magnitude updates can destabilize the model's learned representations and potentially degrade its overall per-

[2] In a real-world scenario, the edit cannot be lossless. Here, for a clear intuition, the above lemma is presented in an ideal way as the editing process will change the value of previously stored key-value pairs. We show that even considering the lossless scenario, the current LLMs cannot satisfy robustness and specificity requirements.

formance; (2) v_* needs not only to be aligned with the direction $(w_{t_*} - w_t)$, but also has a sufficiently large magnitude to ensure editing success.

Lemma 4.7 (**Specificity Requirement for the Key-Values).** *If the newly added knowledge triplet (h, r, t_*) would not be retrieved for any $k_o \notin \mathcal{K}_s$, it requires the following inequality to be satisfied:*

$$(w_{t_n} - w_{t_*})^T W_V (k_o^T C^{-1} k_*) \cdot v_*^T < \epsilon_3,$$
$$\forall k_o \notin \mathcal{K}_s \text{ and } \forall w \in W,$$

where t_n is the original target retrieved by k_o and ϵ_3 denotes the logit difference between t_n and t_*.

One simple solution for this lemma is $k^T C^{-1} k_* = 0$, which describes no superposition, as discussed in one of the concurrent work (Hu et al., 2024a). However, as superposition generally exists among existing LLMs, we discuss more general cases here.

Lemma 4.8 (Whitened Similarity Bounds)**.** *For a successful edit to achieve both robustness and specificity, the whitened similarities must satisfy:*

1. *Lower bound for semantically equivalent keys:*

$$\beta_{s,*} = k_s^T C^{-1} k^* \geq \beta_{\min}, \quad \forall k_s \in \mathcal{K}_s \quad (1)$$

where $\beta_{\min} = \frac{\epsilon_1 + \epsilon_2}{||(w_{t^*} - w_t)^T W_V|| \cdot ||v^*||}$

2. *Upper bound for unrelated keys:*

$$|\beta_{o,*}| = |k_o^T C^{-1} k^*| \leq \beta_{\max}, \quad \forall k_o \notin \mathcal{K}_s \quad (2)$$

where $\beta_{\max} = \frac{\epsilon_3}{\max_{w \in \mathcal{W}} ||w - w_{t^*}|| \cdot ||W_V|| \cdot ||v^*||}$

Detailed proof of all lemmas can be found in Appendix A.

Remark 4.9. *Lemma 4.9 suggests that when adding new knowledge, a new key must be introduced at an appropriate position. This new key must be placed carefully, as its position can affect both its intended target and potentially interfere with nearby keys.*

5 Empirical Analysis: A Break of Requirements

In light of our theoretical results in previous section, we analyze the current knowledge editing methods, showing that the robustness and specificity requirements from previous section cannot be satisfied with inner representations as keys, motivating our approach.

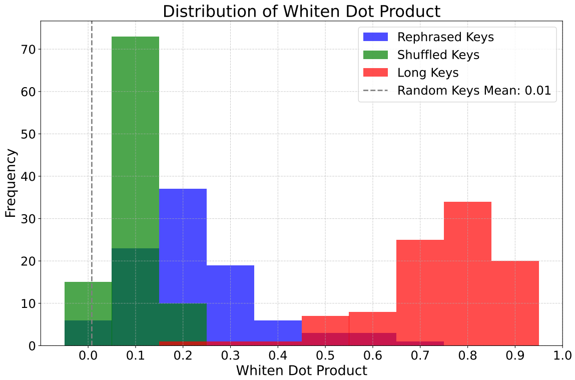


Figure 3: The distribution of normalized whitening similarity between different kinds of keys and original keys.

5.1 Experimental Setup

Following previous work, we use the CounterFact (Meng et al., 2022a) datasets, choosing LLaMA-2 as our base model. In addition to the prompt from CounterFact dataset, we additionally consider three types of perturbation in our experiments, namely prompt appended with unrelated long context, subject rephrase and random shuffled subject. Even though the shuffled subject does not contain the same semantic meaning, it demonstrates how keys shift when the position of same token occurs at different positions.

We collect 10 rephrases for each subject by prompting `gpt-4o-mini`. The prompt we use can be found in Appendix. For long context, we follow (Ma et al., 2024c) and extract random text span of 512 tokens from Wikitext-103 (Merity et al., 2016). For rephrased prompts, we use the paraphrases of prompts released by (Patil et al., 2023). For shuffled subject, we sample 10 random orderings of tokens in the subject. We use 100 samples in our valid set for empirical analyses.

5.2 Empirical Statistics of Keys, Values and Others

Dissimilar Keys. In Figure 3, we present the distribution of whiten similarity β for three operations over the original edit along with a random key baseline. The implementation detail can be found in Appendix B.2.

We can see that the similarity after these operations drops drastically. Rephrasing and shuffling word orders generally reduce the similarity from 1.0 to less than 0.4, even to the random level. Appending long context is less destructive, but still reduces the key similarity to [0.2, 0.9]. These results indicate a violation of the robustness requirement, showing a significant variability in the representation of the same subject, making locate-and-edit difficult to retrieve the edited value to be retrieved.

These findings challenges the intuition that semantically equivalent subjects should have similar representations, and poses severe challenges to the effectiveness of edits.

Similar but Non-Related Keys. We also investigate whether there exist different subjects that have highly similar keys. To this end, we iterate through a slice of Wikitext-103 dataset (about 80M tokens) and select those close to subjects in CounterFact in the whitening space. We filter those tokens whose prefix has the same subject token and collect the top 10 unrelated keys of each subject in CounterFact. The left of Figure 4 plots the distribution of whitening similarities between unrelated prefixes and CounterFact subjects. We find that a large portion of them has an extremely high whitening similarity score, i.e., > 2500. Based on our theory, it indicates that any edit that affects these subjects would inevitably affect the output on these unrelated prefixes.

On the right side of Figure 4, we present a list of subjects and their top-1 prefix in terms of whitening similarities. Interestingly, we observe that a subject can exhibit similarity in distributional semantics (Lenci and Sahlgren, 2023) to its corresponding top unrelated prefix. For example, the keys of *Michael Jordan* are highly similar to keys of a prefix related to *Kobe*. Considering that these two basketball players has much in common in many perspectives, it makes sense that their keys are similar. However, an edit to *Michael Jordan* affects *Kobe* would be definitely unreasonable.

6 Robust Edit Pathway

Our solution is to separate keys from the model's internal representations by introducing a potential branching path as keys for edited facts.

This is done by adding an adapter after the keys, allowing their representations to be modified when needed. As shown on the right of Figure 2, our adapter consists of two modules, a *projection* module that is responsible for aligning the keys and a gate module that activates the adapter when a token representation needs to be edited:

$$\hat{k} = f_{\text{gate}}(k) \circ f_{\text{proj}}(k) + k, \tag{3}$$

where $k \in \mathbb{R}^{bsz \times L \times D}$, $f_{\text{gate}}(k) \in \mathbb{R}^{bsz \times L \times 1}$ and $f_{\text{proj}}(k) \in \mathbb{R}^{bsz \times L \times D}$.

The gate mechanism here operates on the granularity of tokens and adaptively selects whether a key should be modified or not.

We train the adapter by aggregating the keys of same subject $k_s \in \mathcal{K}_s$ toward our injected target key k_*:

$$\mathcal{L}_{\text{agg}} = -|(\frac{\hat{k_s}}{||\hat{k_s}||_2})^T C^{-1} k_*| \tag{4}$$

where $\hat{k_s}$ is the output keys after adapter. The intuition is inspired by Lemma 4.6 and 4.7. If the edited key is close to the keys of the same subject, especially those we found dissimilar in Section 5.2, the edit would be more robust.

In practice, we find that the model inclines to 'cheat' by simply increasing the norm of k_s, and thus we normalize the output of f. In practice, we take the last token of rephrased subjects over different contexts and rephrased templates as k_s. This objective, built on the whiten similarity, further strengthens the validness of our theoretical results.

To address the drift of the target key k_* during optimization, we introduce a target consistency loss:

$$\mathcal{L}_{\text{consistency}} = MSE(\hat{k_*}, k_*) \tag{5}$$

The final training objective combines both components:

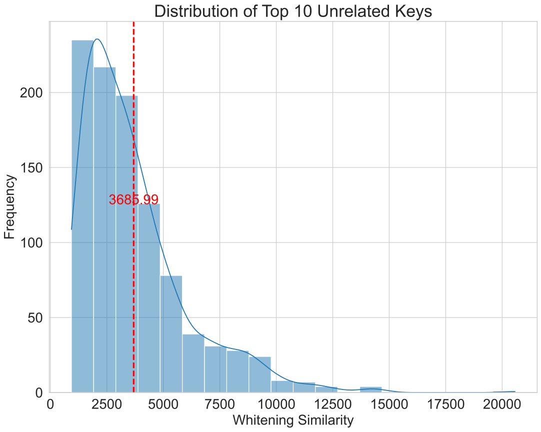
$$\mathcal{L} = \mathcal{L}_{\text{agg}} + \alpha \mathcal{L}_{\text{consistency}} \tag{6}$$

with α controlling the trade-off.

For testing, we use a gate threshold τ to determine whether to activate this projection. This gate mechanism allows the model to dynamically decide whether the original keys should be modified. If not, the keys are left intact and thus ensure the locality of edits. The whole algorithm can be found in Appendix.

6.1 Experimental Results

Setup We evaluate our Robust Edit Pathway with representative locate-and-edit methods, namely ROME, MEMIT, R-ROME, and EMMET. We use the LLaMA2-7B, Mistral-7B, and Qwen2-7B as our base model and CounterFact and ZsRE as our datasets. We filter knowledge triplets of datasets not presented in the model as (Meng et al., 2022a) did, and randomly sample 100 knowledge triplets as the validation set and 400 triplets as the test set. While other studies in model editing explore modifying multiple facts continuously (Mitchell et al., 2022; Hartvigsen et al., 2024; Meng et al., 2022b), we have found that robustly injecting even a single fact presents significant challenges. Therefore, we



Subject	Score	Prefix
Michael Jordan	3898.8	... reach 10,000 career assists. Kobe
Emmitt Smith	8469.9	... for the NL lead with Randy Johnson, Kevin Millwood, Tom Glavine
Pasquale Di Sabatino	2190.6	... archbishop of Albi Giovanni Costanzio Caracciolo
East China Normal University	6798.2	... Located southwest of Gongyi city in Gongxian County
BMW Z3	3570.2	... her lead over Röhrl shrank to 18 minutes. The Toyota Celica

Figure 4: **Left**: CounterFact subjects have unrelated prefixes which are close in keys. The red dashed line indicates random keys baseline. **Right**: Semantically similar subjects bring challenges to specificity.

Method	Edit Performance				In-Domain			Out-of-Domain		
	Sucess↑	Locality↑	Para.↑	Fluency↓	Rephrase↑	Shuffle↑	Long↑	Rephrase↑	Shuffle↑	Long↑
Baseline Methods										
ROME	100.0	96.1	63.8	587.4	61.0	13.0	89.8	62.6	13.7	89.8
MEMIT	99.3	91.2	71.9	571.4	73.3	30.0	92.3	73.4	32.0	94.3
R-ROME	99.7	95.8	62.1	583.8	58.9	14.7	89.5	61.7	16.1	90.7
EMMET	99.7	93.8	63.0	584.0	59.7	16.3	83.7	60.9	16.5	83.0
With REP										
ROME	$100.0^{+0.0}$	$94.6^{-1.5}$	$66.9^{+3.1}$	$587.5^{+0.1}$	$88.0^{+27.0}$	$59.9^{+46.9}$	$91.7^{+1.9}$	$75.5^{+12.9}$	$28.7^{+15.0}$	$91.3^{+1.5}$
MEMIT	$99.4^{+0.1}$	$90.8^{-0.4}$	$74.2^{+2.3}$	$567.2^{-4.2}$	$89.9^{+16.6}$	$58.9^{+28.9}$	$93.6^{+1.3}$	$84.4^{+11.0}$	$45.2^{+31.5}$	$94.2^{-0.1}$
R-ROME	$99.9^{+0.2}$	$94.7^{-1.1}$	$67.4^{+5.3}$	$586.0^{+2.2}$	$88.8^{+29.9}$	$60.3^{+45.6}$	$92.0^{+2.5}$	$76.5^{+14.8}$	$29.5^{+13.4}$	$92.0^{+1.3}$
EMMET	$99.8^{+0.1}$	$92.2^{-1.6}$	$68.4^{+5.4}$	$584.6^{+0.6}$	$94.4^{+34.7}$	$82.7^{+66.4}$	$88.4^{+4.7}$	$82.9^{+22.0}$	$42.5^{+26.0}$	$88.6^{+5.6}$

Table 1: **The main results of REP across three seeds comparing ROME, MEMIT, R-ROME, and EMMET editing methods on Llama2-7B on CounterFact dataset**. REP consistently enhances model performance Results averaged over three seeds with $\tau = 0.9$. The upperscript numbers denote the improvement after using REP.

keep our focus on single-edit paradigm.

For evaluation, we first follow (Meng et al., 2022a,b) and utilize the following four metrics for edit performance: (1) *Success*: the ratio of targeted knowledge achieving the top probability; (2) *Locality*: the ratio of related but non-identical facts kept intact by the edit; (3) Paraphrase (Para.): the ratio of targeted knowledge achieving success on paraphrased prompts; (4) Fluency: the weighted average of bi- and tri-gram entropies.

Moreover, we report the success rate for three robustness tests: paraphrasing subjects, shuffling subjects' token ordering, and appending long context, as discussed throughout the paper. Improving these metrics suggests a more robust editing method. We report robustness metrics at both in-domain, where the test cases are seen in training adapter, and out-of-domain, where the test cases are not seen by adapter. Note that in our 'in-domain', we do not reveal the target knowledge to the model, we only aggregate the keys.

ROME and MEMIT's Failure on Robustness. Our results are shown in Table 1. We can see that our baseline methods, ROME, MEMIT, R-ROME, and EMMET achieve near-perfect edit success rates (>99%) while preserving good locality scores (93-96%). Nonetheless, these methods are prune to robustness tests. Taking ROME as an example, the success rate drops 39% with rephrased subjects, 87% with shuffled subjects ordering, and 10.2% with randomly appended long context. These results reconcile with those reported in previous studies (Ma et al., 2024c).

Effectiveness of Robust Edit Pathway. REP improves the robustness of each of the *locate-and-edit* methods significantly, with a slight cost of locality drop. For instance, REP improves ROME over three robustness tests with +27.0%/+46.9%/+1.9% for in-domain queries, and +12.9%/+15.0%/+1.5% for out-of-domain queries. We also conduct experiments over a different dataset (ZsRE) and two additional base models (QWen and Mistral). The results are shown in Appendix and consistently demonstrate the effectiveness of REP. This validates our theoretical results and empirical insights.

Ablation Studies. Gate threshold τ and consistency loss weight α are crucial to the performance of REP. We study them in the Appendix with Figure 5 and Figure 6. We find that a larger τ and a larger α leads to a better locality and success rate. Meanwhile, the robustness metrics first plateau, then degrade with the increase of τ and α, indicating a trade-off between robustness and edit performance. Throughout our experiments, we use $\tau = 0.9$ and $\alpha = 5e + 4$.

7 Conclusion

In this work, we challenge a core assumption in the locate-and-edit mechanism – *the model's inner representations can serve as semantic keys for editing*. We present theoretical results and empirical analyses revealing that these keys are both sensitive and unspecific. To address this issue, we propose the Robust Edit Pathway (REP), which disentangles the editing keys from native model representations. By extensive experiments, we show that REP can significantly enhance robustness over various locate-and-edit methods while maintains the edit success rate.

8 Limitations

While REP demonstrates significant improvements in knowledge editing robustness, our work is limited in the following aspects: (1) REP requires additional training steps to learn the adapter parameters, introducing computational overhead compared to direct editing methods. (2) Our current evaluation focuses on single-fact editing. The effectiveness of REP in scenarios involving multiple interrelated facts or continuous editing remains to be investigated. (3) In this work, we focus on locate-and-edit methods. Even though it is the dominant line of methods in model editing methods, there are still other model editing methods and REP does not apply to them.

9 Acknowledgement

This publication has been supported by the Ministry of Science and Technology of China Key Project under Grant Number 2022YFE0204900 and the National Natural Science Foundation of China Key Program under Grant Number 62336006. This paper focuses on understanding the relationship between current LLMs' inner representation and robust knowledge manipulation. This aligns with the pursuit of causality-driven LLMs – the goal of the aforementioned National Grant.

In addition, we would like to thank our meta-reviewer and the reviewers from ARR 2025 February cycle for their insightful discussions and Yulong Chen for helping us run experiments with closed-sourced commercial LLMs.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Davis Brown, Charles Godfrey, Cody Nizinski, Jonathan Tu, and Henry Kvinge. 2023. Edit at your own risk: evaluating the robustness of edited models to distribution shifts. *arXiv preprint arXiv:2303.00046*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*.

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

Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. 2024. Model editing harms general abilities of large language models: Regularization to the rescue. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16801–16819.

Akshat Gupta, Sidharth Baskaran, and Gopala Anumanchipalli. 2024a. Rebuilding rome: Resolving model collapse during sequential model editing. *arXiv preprint arXiv:2403.07175*.

Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. 2024b. Model editing at scale leads to gradual and catastrophic forgetting. *arXiv preprint arXiv:2401.07453*.

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

Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2024. Aging with grace: Lifelong model editing with discrete key-value adaptors. *Advances in Neural Information Processing Systems*, 36.

Chenhui Hu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024a. Knowledge in superposition: Unveiling the failures of lifelong knowledge editing for large language models. *arXiv preprint arXiv:2408.07413*.

Chenhui Hu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024b. Wilke: Wise-layer knowledge editor for lifelong knowledge editing. *arXiv preprint arXiv:2402.10987*.

Wenyue Hua, Jiang Guo, Mingwen Dong, Henghui Zhu, Patrick Ng, and Zhiguo Wang. 2024. Propagation and pitfalls: Reasoning-based assessment of knowledge editing through counterfactual tasks. *arXiv preprint arXiv:2401.17585*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Alessandro Lenci and Magnus Sahlgren. 2023. *Distributional semantics*. Cambridge University Press.

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

Zhoubo Li, Ningyu Zhang, Yunzhi Yao, Mengru Wang, Xi Chen, and Huajun Chen. 2023. Unveiling the pitfalls of knowledge editing for large language models. *arXiv preprint arXiv:2310.02129*.

Tianci Liu, Ruirui Li, Yunzhe Qi, Hui Liu, Xianfeng Tang, Tianqi Zheng, Qingyu Yin, Monica Xiao Cheng, Jun Huan, Haoyu Wang, et al. 2025. Unlocking efficient, scalable, and continual knowledge editing with basis-level representation fine-tuning. *arXiv preprint arXiv:2503.00306*.

Jun-Yu Ma, Zhen-Hua Ling, Ningyu Zhang, and Jia-Chen Gu. 2024a. Neighboring perturbations of knowledge editing on large language models. *arXiv preprint arXiv:2401.17623*.

Jun-Yu Ma, Hong Wang, Hao-Xiang Xu, Zhen-Hua Ling, and Jia-Chen Gu. 2024b. Perturbation-restrained sequential model editing. *arXiv preprint arXiv:2405.16821*.

Xinbei Ma, Tianjie Ju, Jiyang Qiu, Zhuosheng Zhang, Hai Zhao, Lifeng Liu, and Yulong Wang. 2024c. Is it possible to edit large language models robustly? *arXiv preprint arXiv:2402.05827*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR.

Vaidehi Patil, Peter Hase, and Mohit Bansal. 2023. Can sensitive information be deleted from llms? objectives for defending against extraction attacks. *arXiv preprint arXiv:2309.17410*.

Domenic Rosati, Robie Gonzales, Jinkun Chen, Xuemin Yu, Melis Erkan, Yahya Kayani, Satya Deepika Chavatapalli, Frank Rudzicz, and Hassan Sajjad. 2024. Long-form evaluation of model editing. *arXiv preprint arXiv:2402.09394*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Jiaan Wang, Yunlong Liang, Zengkui Sun, Yuxuan Cao, Jiarong Xu, and Fandong Meng. 2024a. Cross-lingual knowledge editing in large language models. *Preprint*, arXiv:2309.08952.

Jianchen Wang, Zhouhong Gu, Zhuozhi Xiong, Hongwei Feng, and Yanghua Xiao. 2024b. The missing piece in model editing: A deep dive into the hidden damage brought by model editing. *arXiv preprint arXiv:2403.07825*.

Mengru Wang, Ningyu Zhang, Ziwen Xu, Zekun Xi, Shumin Deng, Yunzhi Yao, Qishen Zhang, Linyi Yang, Jindong Wang, and Huajun Chen. 2024c. Detoxifying large language models via knowledge editing. *arXiv preprint arXiv:2403.14472*.

Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2024d. Wise: Rethinking the knowledge memory for lifelong model editing of large language models. *Advances in Neural Information Processing Systems*, 37:53764–53797.

Peng Wang, Ningyu Zhang, Xin Xie, Yunzhi Yao, Bozhong Tian, Mengru Wang, Zekun Xi, Siyuan Cheng, Kangwei Liu, Guozhou Zheng, et al. 2023. Easyedit: An easy-to-use knowledge editing framework for large language models. *arXiv preprint arXiv:2308.07269*.

Xiaohan Wang, Shengyu Mao, Ningyu Zhang, Shumin Deng, Yunzhi Yao, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. 2024e. Editing conceptual knowledge for large language models. *arXiv preprint arXiv:2403.06259*.

Jianhao Yan, Futing Wang, Yafu Li, and Yue Zhang. 2024. Potential and challenges of model editing for social debiasing. *arXiv preprint arXiv:2402.13462*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024a. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Wanli Yang, Fei Sun, Xinyu Ma, Xun Liu, Dawei Yin, and Xueqi Cheng. 2024b. The butterfly effect of model editing: Few edits can trigger large language models collapse. *arXiv preprint arXiv:2402.09656*.

Junsang Yoon, Akshat Gupta, and Gopala Anumanchipalli. 2024. Is bigger edit batch size always better?–an empirical study on model editing with llama-3. *arXiv preprint arXiv:2405.00664*.

Mengqi Zhang, Bowen Fang, Qiang Liu, Pengjie Ren, Shu Wu, Zhumin Chen, and Liang Wang. 2024a. Enhancing multi-hop reasoning through knowledge erasure in large language model editing. *arXiv preprint arXiv:2408.12456*.

Mengqi Zhang, Xiaotian Ye, Qiang Liu, Pengjie Ren, Shu Wu, and Zhumin Chen. 2024b. Uncovering overfitting in large language model editing. *arXiv preprint arXiv:2410.07819*.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. 2023. Mquake: Assessing knowledge editing in language models via multi-hop questions. *arXiv preprint arXiv:2305.14795*.

A Proofs

A.1 Proof of Lemma 4.1

Since K has full row rank ($\text{rank}(K) = D_1$), KK^T is invertible. To find α, we use the Moore-Penrose pseudoinverse of K.

Given $K \in \mathbb{R}^{D_1 \times n}$, the pseudoinverse K^+ is defined as: $K^+ = K^T(KK^T)^{-1}$, which also minimizes $||K\alpha - \hat{k}||$.

Then, we can express $\hat{v}$ as:

$$\hat{v} = V\alpha = VK^T(KK^T)^{-1}\hat{k}. \tag{7}$$

Note that since $n \gg D_1$, the system $K\alpha = \hat{k}$ is underdetermined. This means there are infinitely many solutions for α, and the Moore-Penrose pseudoinverse gives the one with the smallest norm.

A.2 Proof of Lemma 4.4

We can focus on the logit difference between the largest and the second-largest logits to achieve high confidence in the final prediction. This difference is an important factor in determining the confidence of a prediction in a softmax layer.

Here, we simplify the modeling by only considering the contribution of edited layer towards final prediction via its the edited layer is connected to the final prediction layer directly via its attention layer

Given a vector of logits $\mathbf{z} = [z_1, z_2, \ldots, z_n]$, the softmax function yields probabilities $\mathbf{p} = [p_1, p_2, \ldots, p_n]$, where:

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$$

To increase the confidence in the prediction for the largest logit, maximize the difference between the largest logit and the second-largest logit.

Let z_{max} be the largest logit and z_{other} be another logit. The logit difference Δ is given by: $\epsilon = z_{\text{max}} - z_{\text{other}}$.

The softmax confidence for the class corresponding to z_{max} can be expressed as:

$$p_{\text{max}} = \frac{e^{z_{\text{max}}}}{e^{z_{\text{max}}} + e^{z_{\text{other}}} + \sum_{k \neq \text{max, other}} e^{z_k}} \tag{8}$$

$$< \frac{e^{z_{\text{max}}}}{e^{z_{\text{max}}} + e^{z_{\text{other}}}} \tag{9}$$

$$= \frac{1}{1 + e^{-\epsilon}} \tag{10}$$

After organizing between two sides, we get a lower bound of ϵ for achieving a sufficiently large confidence:

$$\epsilon > -\log(1 - \frac{1}{p_{\text{max}}}) \tag{11}$$

Now, in a transformer architecture, the edited MLP layer is connected to the word prediction layer through an attention layer at the final token. Let the difference between the original and the edited output of the MLP layer be Δv, the parameters of the attention layer are $W_Q, W_K, W_V \in \mathbb{R}^{D \times D}$ and the query vector at the prediction token is $q = Qh$, the attention layer's output is defined by

$$o = \sum_j \text{Softmax}(q^T W_K v_j) W_V v_j. \tag{12}$$

Since in the locating part we use causal intervention to identify the most influential position of tokens to edit, we can assume that $(q^T W_K v_s)$ has already get the largest weight. The difference caused by edited MLP is,

$$\Delta o = \text{Softmax}(\cdot) W_V \Delta v. \tag{13}$$

Then, residual connections directly connect this output to the final word prediction layer. Combining our result from equation 11, let the original fact t before the edit has a logit gap ϵ_1 and the new fact t_* after edit has ϵ_2, we can bound the Δo with,

$$\begin{cases} (w_t - w_{t_*})^T o_{ori} > \epsilon_1 \\ (w_{t_*} - w_t)^T (o_{ori} + \Delta o) > \epsilon_2 \end{cases} \tag{14}$$

$$\Rightarrow (w_{t_*} - w_t)^T \Delta o > \epsilon_1 + \epsilon_2 \tag{15}$$

$$\Rightarrow (w_{t_*} - w_t)^T \text{Softmax}(\cdot) W_V \Delta v > \epsilon_1 + \epsilon_2 \tag{16}$$

$$\Rightarrow (w_{t_*} - w_t)^T W_V \Delta v > \epsilon_1 + \epsilon_2 \tag{17}$$

$$\tag{18}$$

Given that the softmax weight is at most 1, we have our lower bound on Δv.

A.3 Proof of Lemma 4.6

For an edit to be robust, it must propagate correctly to all semantically equivalent inputs. We derive this requirement step by step:

1) From Lemma 4.4, a successful edit requires:

$$(w_{t^*} - w_t)^T W_V (v^* - v_o) > \epsilon_1 + \epsilon_2$$

2) When querying with a semantically equivalent key $k_s \in K_s$, by Lemma 4.1, the retrieved value is:

$$v = k_s^T C^{-1} k^* \cdot v^* = \beta_{s,*} \cdot v^*$$

where $\beta_{s,*}$ represents the whiten similarity between k_s and k^*.

3) For robust editing, this retrieved value must

maintain the prediction gap:

$$(w_{t^*} - w_t)^T W_V(\beta_{s,*} \cdot v^*) > \epsilon_1 + \epsilon_2$$

4) Rearranging terms:

$$(w_{t^*} - w_t)^T W_V(k_s^T C^{-1} k^*) \cdot v^{*T} > \epsilon_1 + \epsilon_2, \\ \forall k_s \in K_s$$

This inequality must hold for all semantically equivalent keys $k_s \in K_s$, establishing our robustness requirement.

A.4 Proof of Lemma 4.7

The specificity requirement ensures edits do not affect unrelated knowledge. We derive this as follows:

1) Consider an unrelated key $k_o \notin K_s$ with original target t_n. The corresponding output embedding is w_n.

2) To preserve specificity, the edit should not significantly alter predictions for unrelated inputs:

$$(w_n - w_{t^*})^T W_V(k_o^T C^{-1} k^*) \cdot v^{*T} < \epsilon_3$$

3) This constraint must hold for:

- All unrelated keys $k_o \notin K_s$
- All possible target embeddings $w_n \in W$

4) Therefore, our specificity requirement is:

$$(w_n - w_{t^*})^T W_V(k_o^T C^{-1} k^*) \cdot v^{*T} < \epsilon_3, \\ \forall k_o \notin K_s, \forall w_n \in W$$

This establishes the formal criterion for maintaining specificity in knowledge editing. The requirement ensures that edits remain localized to the intended knowledge while not affecting unrelated retrievals.

B Experimental Details

B.1 Data Construction

We build our evaluation data based on the CounterFact dataset. We further augment our data with all three robustness tests. For rephrased subjects by prompting `gpt4o-mini` with the following prompt.

> Give 10 rephrases representing the same entity: {ENTITY}

The irrelevant long contexts are extracted from the Wikitext-103 dataset (Merity et al., 2016). The shuffled tokens are generated via sampling different word ordering. Finally, we filter the samples that are not present in the current LLM, that is, given the prefix, the target tokens are not predicted by the LLMs with the top-1 probabilities. We sample 100 samples for validation and 400 samples for test. To evaluate in-domain and out-of-domain robustness, we split the all three kinds of robustness queries in a 50-50 manner. For each sample, we have 5 in-domain queries and 5 out-of-domain queries.

B.2 Analyzing Dissimilar Keys

In Section 5.2, for each subject in CounterFact, we compute the dot product for each pair of keys of a subject's rephrases. We utilize the inputs to the FFN's down projection of layer 5 of LLaMA-2 as our keys, consistent with previous ROME experiments. Additionally, we include the dot product values of randomly sampled keys as a baseline for comparative analysis. We normalize the whiten similarity by the similarities between the subject itself.

B.3 Details of Training REP

We implement our methods based on EasyEdit (Wang et al., 2023). We use Adam optimizer for all experiments and the learning rate is 5e-4. We train each adaptor for 10 steps. The inner dimension of the projection module is 32, and the inner dimension of gate module is 0.1 of key dimension.

C Additional Results

Table 2 and 3 present the performance of ROME, MEMIT, R-ROME, and EMMET methods, both with and without the REP enhancement, across CounterFact and ZSRE respectively. Across both datasets, REP consistently improves model robustness, particularly in in-domain generalization and out-of-domain adaptability, despite minor tradeoffs in edit success rates. Results are averaged over three seeds , with standard deviations indicating stable improvements. Notably, REP-enhanced variants demonstrate superior fluency and locality preservation, highlighting its effectiveness in balancing edit precision with broader generalization.

D Preliminary Experiments to Multi-Edits

Even though the focus of this paper is in single edit scenario, extending REP to the mulit-edit scenario would be an interesting extension for further work.

Thus, we have conducted some preliminary experiments on multi-edits. Our setting is to extend the single edit to 5 edits sequentially, as was done

Table 2: **The main results of REP across three seeds comparing ROME, MEMIT, R-ROME, and EMMET editing methods on Llama2-7B, Mistral-7B and Qwen2-7b on CounterFact dataset**. REP consistently enhances model performance Results averaged over three seeds with $\tau = 0.9$, showing standard deviations. ↑ indicates higher values are better, ↓ indicates lower values are better.

		Edit Performance			**Generalization**		**In-Domain**			**Out-of-Domain**		
Model	*Method*	*Success↑*	*Locality↑*	*Reversion↓*	*Para.↑*	*Fluency↑*	*Rephrase↑*	*Shuffle↑*	*Long↑*	*Rephrase↑*	*Shuffle↑*	*Long↑*
Llama2	**ROME**	100.0 ± 0.0	**96.1 ± 0.1**	0.0 ± 0.0	63.8 ± 0.3	587.4 ± 1.2	61.0 ± 0.7	13.0 ± 0.9	89.8 ± 0.2	62.6 ± 0.1	13.7 ± 0.5	89.8 ± 0.5
	+REP	100.0 ± 0.0	94.6 ± 0.2	0.0 ± 0.0	**66.9 ± 0.3**	**587.5 ± 0.8**	**88.0 ± 0.2**	**59.9 ± 0.3**	**91.7 ± 0.2**	**75.5 ± 0.6**	**28.7 ± 1.9**	**91.3 ± 1.4**
	MEMIT	99.3 ± 0.5	**91.2 ± 0.6**	0.0 ± 0.0	71.9 ± 1.7	**571.4 ± 2.6**	73.3 ± 1.2	30.0 ± 0.9	92.3 ± 0.9	73.4 ± 0.7	32.0 ± 3.1	**94.3 ± 3.3**
	+REP	**99.4 ± 0.1**	90.8 ± 0.2	0.0 ± 0.0	**74.2 ± 0.1**	567.2 ± 0.3	**89.9 ± 0.4**	**58.9 ± 0.8**	**93.6 ± 1.1**	**84.4 ± 0.5**	**45.2 ± 0.8**	94.2 ± 1.5
	R-ROME	99.7 ± 0.5	**95.8 ± 0.3**	0.3 ± 0.5	62.1 ± 1.3	583.8 ± 3.3	58.9 ± 0.7	14.7 ± 0.8	89.5 ± 3.5	61.7 ± 1.3	16.1 ± 1.8	90.7 ± 0.5
	+REP	**99.9 ± 0.1**	94.7 ± 0.4	**0.0 ± 0.0**	**67.4 ± 0.2**	**586.0 ± 0.1**	**88.8 ± 0.5**	**60.3 ± 1.2**	**92.0 ± 0.2**	**76.5 ± 0.6**	**29.5 ± 1.4**	**92.0 ± 0.8**
	EMMET	99.7 ± 0.5	**93.8 ± 0.2**	**0.0 ± 0.0**	63.0 ± 1.3	584.0 ± 6.5	59.7 ± 2.3	16.3 ± 0.6	83.7 ± 0.2	60.9 ± 1.2	16.5 ± 1.5	83.0 ± 2.4
	+REP	**99.8 ± 0.2**	92.2 ± 0.4	0.1 ± 0.1	**68.4 ± 0.2**	**584.6 ± 0.8**	**94.4 ± 0.2**	**82.7 ± 1.0**	**88.4 ± 1.9**	**82.9 ± 0.3**	**42.5 ± 1.3**	**88.6 ± 2.2**
Mistral	**ROME**	**99.9 ± 0.1**	**94.1 ± 0.0**	**0.0 ± 0.0**	69.1 ± 0.5	609.4 ± 0.8	71.1 ± 0.2	14.6 ± 0.2	94.6 ± 0.4	71.8 ± 0.3	14.3 ± 1.1	94.4 ± 0.3
	+REP	99.8 ± 0.2	92.8 ± 0.1	0.1 ± 0.1	**72.2 ± 0.2**	**610.0 ± 0.5**	**95.5 ± 0.2**	**84.6 ± 0.6**	**95.1 ± 0.4**	**84.8 ± 0.8**	**41.6 ± 0.5**	**94.7 ± 0.3**
	MEMIT	**99.7 ± 0.3**	**89.2 ± 0.2**	0.0 ± 0.0	76.8 ± 0.5	**607.0 ± 0.9**	84.0 ± 0.1	29.0 ± 0.3	95.0 ± 0.6	82.4 ± 0.4	28.6 ± 0.4	94.0 ± 0.5
	+REP	98.5 ± 0.5	85.5 ± 0.1	0.0 ± 0.0	**77.3 ± 0.6**	605.5 ± 1.0	**93.1 ± 0.5**	**75.1 ± 1.3**	**95.4 ± 0.4**	**89.2 ± 0.3**	**62.7 ± 0.7**	**94.3 ± 0.3**
	R-ROME	**99.8 ± 0.1**	**93.7 ± 0.1**	**0.0 ± 0.0**	70.5 ± 0.1	608.6 ± 0.9	73.2 ± 0.2	16.1 ± 0.4	95.5 ± 0.4	73.4 ± 0.3	16.0 ± 1.2	95.3 ± 1.2
	+REP	99.7 ± 0.1	92.4 ± 0.0	0.1 ± 0.1	**73.6 ± 0.1**	**609.4 ± 1.2**	**96.1 ± 0.1**	**86.9 ± 0.3**	**95.9 ± 0.2**	**85.9 ± 0.2**	**43.9 ± 0.5**	**95.4 ± 1.0**
	EMMET	**99.8 ± 0.1**	**92.5 ± 0.2**	0.1 ± 0.1	69.6 ± 0.9	**609.0 ± 1.0**	73.8 ± 0.5	16.7 ± 0.5	92.0 ± 0.7	73.5 ± 0.3	16.4 ± 0.6	91.4 ± 1.3
	+REP	99.0 ± 0.0	89.9 ± 0.3	0.1 ± 0.1	**74.2 ± 1.1**	608.4 ± 0.3	**98.2 ± 0.1**	**95.2 ± 1.0**	**93.6 ± 0.7**	**90.2 ± 0.3**	**58.7 ± 1.0**	**92.9 ± 1.5**
Qwen2	**ROME**	**99.6 ± 0.1**	**95.6 ± 0.1**	0.0 ± 0.0	69.4 ± 0.3	620.4 ± 1.5	63.2 ± 0.3	20.0 ± 0.4	94.1 ± 0.3	62.7 ± 0.2	18.1 ± 0.5	93.9 ± 0.5
	+REP	99.4 ± 0.1	91.0 ± 0.1	0.0 ± 0.0	**73.1 ± 0.1**	**622.1 ± 1.9**	**81.0 ± 0.6**	**70.5 ± 0.5**	**95.9 ± 0.2**	**75.6 ± 0.0**	**65.4 ± 0.6**	**95.8 ± 0.5**
	MEMIT	99.6 ± 0.1	**90.3 ± 0.2**	0.4 ± 0.1	75.6 ± 0.1	620.1 ± 0.3	75.9 ± 0.6	31.4 ± 0.9	97.6 ± 0.1	74.8 ± 0.4	29.7 ± 0.9	96.6 ± 0.3
	+REP	**99.7 ± 0.1**	81.8 ± 0.1	**0.0 ± 0.0**	**79.7 ± 0.2**	**620.2 ± 2.2**	**95.7 ± 0.2**	**81.7 ± 1.2**	**98.0 ± 0.1**	**89.7 ± 0.1**	**72.3 ± 1.5**	**96.9 ± 0.2**
	R-ROME	99.8 ± 0.0	**96.2 ± 0.1**	0.2 ± 0.0	68.5 ± 0.4	621.0 ± 0.4	63.1 ± 0.3	20.2 ± 0.3	93.8 ± 0.2	62.3 ± 0.1	18.4 ± 0.4	93.3 ± 0.9
	+REP	**99.9 ± 0.1**	91.9 ± 0.1	**0.1 ± 0.1**	**72.4 ± 0.5**	**621.1 ± 0.2**	**81.4 ± 0.3**	**70.8 ± 0.8**	**95.7 ± 0.5**	**75.5 ± 0.2**	**64.8 ± 0.9**	**94.5 ± 1.1**
	EMMET	99.8 ± 0.0	**92.5 ± 0.1**	0.2 ± 0.0	72.2 ± 0.1	619.5 ± 0.2	71.5 ± 0.7	31.3 ± 0.6	96.2 ± 0.4	70.5 ± 0.9	30.0 ± 0.4	96.7 ± 0.6
	+REP	**99.9 ± 0.1**	76.4 ± 0.8	**0.0 ± 0.0**	**78.5 ± 0.4**	**621.0 ± 2.6**	**92.9 ± 0.3**	**88.4 ± 0.9**	**97.0 ± 0.1**	**89.2 ± 0.3**	**87.1 ± 1.3**	**97.4 ± 0.6**

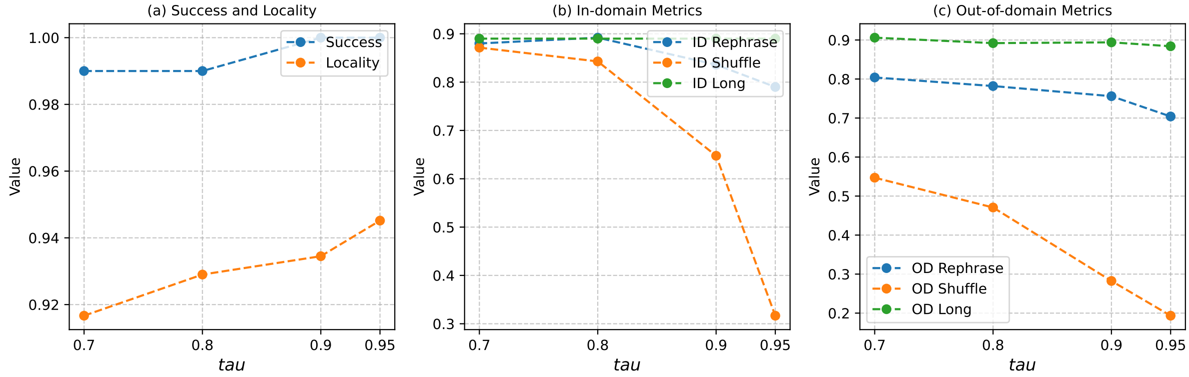


Figure 5: Hyper-parameter study of τ on validation set.

in previous studies. We naively extend the REP adaptors one by one after each ROME edit. These initial results, shown in Table 4, are consistent with the experiments in the single edit scenario: Robustness is greatly improved while precision and locality are largely maintained.

E Ablation Study

The gate threshold τ and consistency loss weight α significantly influence REP's performance, as discussed in Section 6.1. Empirical analysis (Figures 5 and 6) demonstrates that increasing τ and α improves locality preservation and edit success rates. However, robustness metrics initially plateau before deteriorating with further parameter escalation, underscoring the need to balance precision against generalization. This trade-off analysis justifies our selection of $\tau = 0.9$ and $\alpha = 1\text{e+}5$, which optimally reconcile competing objectives across experiments.

F Case Visualization

Figure 7 (right) provides a visualization of representations for the subject 'Slovenia' after three types of perturbations, reduced to two dimensions using Principal Component Analysis (PCA). This visualization corroborates our previous findings: (1) *Context sensitivity:* Long irrelevant context induces a slight shift in the representation, indicating contextual influence on subject encoding. (2) *Rephrase variability:* Rephrased versions of the subject sometimes cluster close to the original

Table 3: The main results of REP across three seeds comparing ROME, MEMIT, R-ROME, and EMMET editing methods on Llama2-7B, Mistral-7B and Qwen2-7b on ZSRE dataset. REP consistently enhances model performance. Results averaged over three seeds with $\tau = 0.9$, showing standard deviations. ↑ indicates higher values are better, ↓ indicates lower values are better.

		Edit Performance			Generalization	In-Domain			Out-of-Domain		
Model	*Method*	*Sucess*	*Locality*	*Reversion*	*Fluency*	*Rephrase*	*Shuffle*	*Long*	*Rephrase*	*Shuffle*	*Long*
Llama2	**ROME**	**92.1 ± 0.1**	99.6 ± 0.0	**0.5 ± 0.0**	566.1 ± 1.8	44.4 ± 0.3	4.7 ± 0.1	68.2 ± 0.6	44.2 ± 0.9	4.5 ± 0.3	68.3 ± 1.0
	+REP	90.0 ± 0.5	99.6 ± 0.0	0.6 ± 0.1	**567.2 ± 1.8**	**72.3 ± 0.2**	**51.5 ± 0.3**	**72.5 ± 0.2**	**58.0 ± 0.7**	**24.9 ± 0.4**	**71.2 ± 1.7**
	MEMIT	**88.5 ± 0.6**	99.4 ± 0.1	0.5 ± 0.0	**545.1 ± 2.6**	53.7 ± 0.8	13.0 ± 0.9	**72.0 ± 1.9**	54.5 ± 0.6	12.7 ± 0.5	71.2 ± 2.3
	+REP	87.1 ± 0.1	99.4 ± 0.1	0.5 ± 0.0	543.3 ± 2.3	**57.0 ± 0.3**	**17.4 ± 0.6**	71.9 ± 1.9	**56.2 ± 0.6**	**14.5 ± 0.2**	**71.8 ± 1.5**
	R-ROME	**92.1 ± 0.2**	99.7 ± 0.0	**0.5 ± 0.0**	**565.0 ± 0.9**	43.1 ± 0.9	4.6 ± 0.2	68.7 ± 0.4	43.1 ± 1.1	4.4 ± 0.3	68.7 ± 1.4
	+REP	89.8 ± 0.3	99.7 ± 0.0	0.8 ± 0.2	562.0 ± 2.2	**71.4 ± 0.6**	**51.2 ± 0.7**	**72.4 ± 0.2**	**57.2 ± 0.9**	**25.2 ± 0.9**	**72.0 ± 1.7**
	EMMET	**86.6 ± 1.4**	99.7 ± 0.1	**0.5 ± 0.0**	**563.8 ± 0.9**	33.0 ± 1.3	2.8 ± 0.3	52.1 ± 2.6	33.0 ± 1.1	2.8 ± 0.3	52.7 ± 3.9
	+REP	84.7 ± 1.3	99.7 ± 0.1	0.7 ± 0.1	561.6 ± 2.0	**66.7 ± 2.3**	**50.9 ± 2.2**	**59.4 ± 2.2**	**50.0 ± 1.8**	**22.6 ± 1.8**	**59.7 ± 3.7**
Mistral	**ROME**	**97.2 ± 0.3**	99.5 ± 0.1	1.6 ± 0.1	584.0 ± 2.5	49.2 ± 0.4	4.2 ± 0.4	77.2 ± 0.1	50.3 ± 0.9	4.1 ± 0.7	78.3 ± 0.9
	+REP	93.1 ± 0.6	99.5 ± 0.1	**1.5 ± 0.0**	**584.6 ± 1.0**	**84.3 ± 0.8**	**74.1 ± 1.1**	**78.6 ± 0.6**	**71.5 ± 1.5**	**40.8 ± 1.5**	**79.1 ± 1.4**
	MEMIT	**94.1 ± 1.0**	99.4 ± 0.1	1.4 ± 0.1	**579.6 ± 3.1**	60.5 ± 0.8	12.4 ± 0.8	**80.9 ± 2.2**	62.1 ± 1.2	11.9 ± 0.3	**81.8 ± 1.2**
	+REP	90.2 ± 0.7	99.4 ± 0.1	**1.3 ± 0.0**	579.0 ± 1.6	**68.9 ± 1.9**	**36.5 ± 2.4**	80.0 ± 1.9	**66.1 ± 1.7**	**25.9 ± 1.6**	80.8 ± 1.6
	R-ROME	**97.5 ± 0.2**	99.6 ± 0.2	1.6 ± 0.1	**585.5 ± 2.5**	50.1 ± 0.4	4.3 ± 0.3	78.4 ± 0.8	50.9 ± 0.7	4.4 ± 0.8	78.8 ± 1.1
	+REP	93.3 ± 0.7	99.6 ± 0.2	1.6 ± 0.1	585.0 ± 4.3	**84.9 ± 0.7**	**75.6 ± 0.7**	**79.3 ± 0.6**	**72.0 ± 1.6**	**42.4 ± 1.5**	**80.3 ± 1.4**
	EMMET	**95.9 ± 0.3**	99.5 ± 0.1	1.7 ± 0.1	588.0 ± 0.9	41.8 ± 1.2	2.9 ± 0.5	52.8 ± 4.2	42.4 ± 1.4	2.7 ± 0.3	52.6 ± 5.4
	+REP	90.2 ± 1.0	99.5 ± 0.1	**1.6 ± 0.1**	**589.1 ± 1.8**	**83.0 ± 0.6**	**76.1 ± 0.7**	**61.0 ± 3.2**	**68.4 ± 1.5**	**44.8 ± 2.0**	**60.2 ± 4.2**
Qwen2	**ROME**	**98.3 ± 0.1**	**98.9 ± 0.2**	2.0 ± 0.0	562.1 ± 3.7	53.8 ± 0.2	11.0 ± 0.5	76.8 ± 0.4	55.9 ± 0.1	11.0 ± 0.9	77.5 ± 1.1
	+REP	97.0 ± 0.4	97.4 ± 0.0	2.0 ± 0.0	**568.8 ± 2.9**	**64.2 ± 0.4**	**42.2 ± 1.0**	**80.6 ± 0.5**	**61.7 ± 0.8**	**36.0 ± 0.4**	**79.2 ± 1.0**
	MEMIT	**95.4 ± 0.2**	**98.2 ± 0.1**	**1.5 ± 0.0**	573.9 ± 5.8	62.4 ± 0.3	22.2 ± 0.3	88.8 ± 1.6	65.2 ± 0.3	22.1 ± 0.8	**89.2 ± 0.5**
	+REP	94.2 ± 0.3	97.2 ± 0.1	1.6 ± 0.1	**576.4 ± 4.1**	**78.7 ± 0.9**	**51.9 ± 2.0**	88.8 ± 1.5	**74.8 ± 0.7**	**41.2 ± 2.4**	89.0 ± 0.4
	R-ROME	**98.2 ± 0.2**	**98.8 ± 0.1**	**2.0 ± 0.0**	571.5 ± 2.2	54.2 ± 0.3	11.5 ± 0.5	77.2 ± 1.3	56.7 ± 0.1	12.0 ± 0.8	76.7 ± 1.5
	+REP	96.0 ± 0.2	97.4 ± 0.1	2.2 ± 0.3	**575.5 ± 3.3**	**66.3 ± 1.0**	**45.3 ± 0.4**	**80.0 ± 2.1**	**63.3 ± 0.7**	**38.8 ± 0.5**	**78.9 ± 2.5**
	EMMET	**94.6 ± 0.2**	**97.7 ± 0.3**	**1.5 ± 0.0**	570.6 ± 4.2	58.5 ± 0.5	17.7 ± 0.4	**75.2 ± 1.8**	60.3 ± 0.7	18.9 ± 0.5	75.6 ± 0.9
	+REP	91.4 ± 0.3	91.7 ± 0.1	1.8 ± 0.2	**574.5 ± 4.6**	**78.2 ± 0.5**	**70.7 ± 0.5**	77.2 ± 1.8	**75.0 ± 0.9**	**67.6 ± 1.0**	**77.2 ± 0.8**

	In-Domain				Out-of-domain			
	ACC	Locality	rephrase	shuffle	long	rephrase	shuffle	long
ROME	79.3	57.3	59.9	21.5	71.0	59.9	22.2	69.5
ROME + REP	78.0	56.4	71.7	51.5	71.5	65.3	30.8	70.3

Table 4: Performance comparison between ROME and ROME + REP methods for 5 sequential edits.

representation, while at other times they are distant. (3) *Order dependence:* Shuffling the word order results in substantial deviations from the original representation. This observation highlights the model's sensitivity to word order, even when the constituent tokens remain unchanged.

When the edited key has near-zero or negative similarity with other keys, based on Lemma 4.6 it becomes virtually impossible for the edited value to be retrieved, potentially compromising the robustness of the edit.

G Analyzing Value Distributions

Loud Voices. In Figure 8, we present the distribution of values before and after edits, using LLaMA-2 7B and ROME. The results demonstrate that post-edit values exhibit significantly larger L2 norms compared to pre-edit values. This observation aligns with our findings in Lemma 4.4 and 4.6, which suggest that edited values must be sufficiently large to effect changes on the current key and influence distant keys.

However, this increase in value magnitude, while necessary for effective editing, presents potential challenges. As indicated by Lemma 4.7 and our previous analysis, these 'loud' values may inadvertently affect unrelated keys, particularly those that are proximal in the representation space to the one being edited. This observation highlights a tension between achieving targeted edits and avoiding unintended consequences in the model's broader knowledge representation.

Summary. Our findings collectively suggest that the inner representations of large language models (LLMs) may not serve as reliable keys for editing purposes. The observed variability in key similarities, even among semantically equivalent subjects, coupled with the necessity for large-magnitude value changes, poses significant challenges for precise and controlled model editing. These issues can lead to unintended effects on unrelated parts of the model's knowledge and compromise the specificity of edits. Furthermore, the sensitivity of representations to word order and context underscores the instability of using these internal states as edit targets. These limitations motivate us to explore

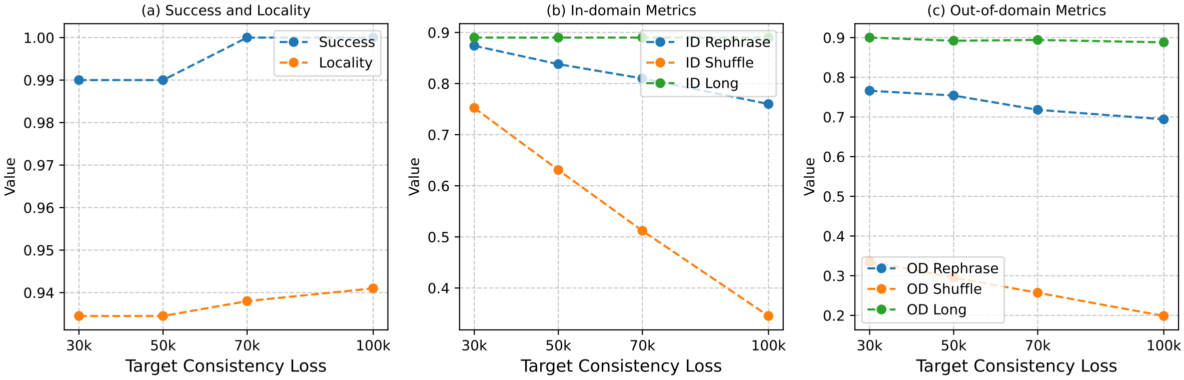


Figure 6: Hyper-parameter study of consistency loss weight α on validation set.

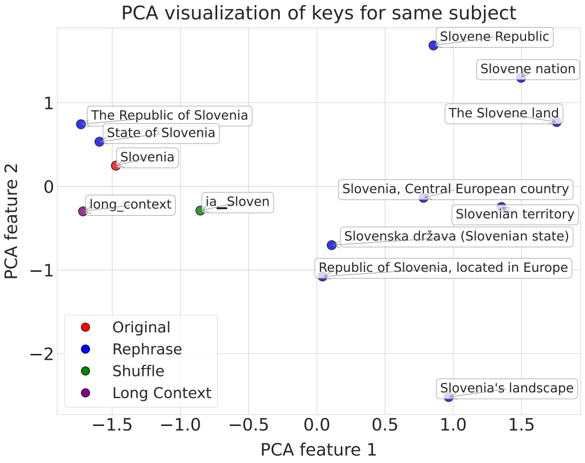


Figure 7: A visualization for key representations of rephrases for 'Slovenia' in LLaMA2-7B.

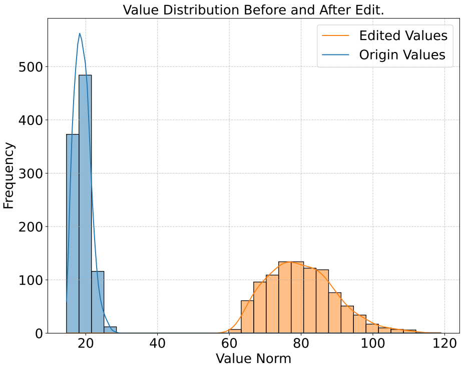


Figure 8: Values before and after edit with ROME.

alternative approaches, particularly the concept of *branching a separate path for keys*. By creating a dedicated pathway for key representations, we may achieve more stable and controllable edit targets, potentially mitigating the issues of representation variability and unintended side effects observed when directly manipulating the model's inner representations.