# Representation Interventions Enable Lifelong Knowledge Memory Control in LLMs

**Xuyuan Liu**[*1,2], **Shengyu Chen**[2], **Xinshuai Dong**[3], **Yanchi Liu**[2], **Xujiang Zhao**[2], **Haoyu Wang**[2], **Yujun Yan**[1], **Haifeng Chen**[†,2], **Zhengzhang Chen**[†,2]
[1]Dartmouth College [2]NEC Laboratories America [3]Carnegie Mellon University
{xuyuan.liu.gr,yujun.yan}@dartmouth.edu, zchen@nec-labs.com

## Abstract

Large language models (LLMs) often produce incorrect or outdated content after being employed. Efficient and accurate knowledge updates without costly retraining are a major challenge. This problem is particularly challenging in *lifelong* settings, where complex, unstructured knowledge must coexist without interference. We introduce RILKE (Representation Intervention for Lifelong KnowledgE Control), a robust and scalable method that treats knowledge control as interventions within the model's representation space. Leveraging representation-space expressiveness, we identify two key properties enabling RILKE to achieve fine-grained control over complex, unstructured knowledge while maintaining general utility with frozen base weights. During training, RILKE learns *paraphrase-robust* and *edit-localized* modules that limit each update to a low-dimensional subspace to minimize cross-edit interference. At inference, a query-adaptive router selects the appropriate module to guide the model's generation. Across LLaMA and Qwen models, RILKE scales effectively to large-scale benchmarks, demonstrating high edit success and strong paraphrase generalization while preserving general utility with modest memory overhead. These results show RILKE is an effective and scalable solution for lifelong knowledge control in LLMs.

## 1 Introduction

Large language models (LLMs) excel at knowledge-intensive NLP tasks (Madaan et al., 2022; Sun et al., 2024; Chen et al., 2025b), yet their knowledge is static. Once deployed, they cannot adapt to evolving real-world information, often producing outdated or inaccurate content. While methods like retraining or continual pretraining can update a model, they are computationally prohibitive and prone to catastrophic forgetting (Ke et al., 2023; Yildiz et al., 2025). As an alternative, Retrieval-Augmented Generation (RAG) injects new facts at inference time (Chen et al., 2024), but it is susceptible to conflicts with parametric memory (Li et al., 2025; Gutiérrez et al., 2025) and sensitive to retrieval quality (Salemi and Zamani, 2024). These challenges underscore the need for **lifelong knowledge memory control**: methods to precisely and efficiently update LLM knowledge with minimal side effects (Thede et al., 2025).

However, existing methods struggle in lifelong edit settings, especially when addressing *unstructured*, *free-form* knowledge that cannot be reduced to simple factual triplets (*i.e.*, Subject–Relation–Object) (Deng et al., 2025). Parametric approaches, which modify model weights directly (Fang et al., 2025), suffer from *edit collapse*: as new knowledge accumulates, performance degrades until the model becomes unusable (Yang et al., 2024b; Nishi et al., 2025), a degradation particularly severe when incorporating new unstructured knowledge. Similarly, although effective for accumulation, external-memory methods struggle to capture the nuances of complex information, as their capacity is constrained by learning on the weight space of a single sub-module (*e.g.*, an additional Feedforward network (FFN) layer, as in Wang et al. (2024)).

In contrast to modifying model weights, interventions in the representation space, the hidden states where information is processed, offer a powerful substrate for knowledge control due to their rich semantic structure, yet this space remains largely underexplored (Elhage et al., 2021; Wu et al., 2024b). To our knowledge, the only related work on representation-space editing is limited to structured knowledge and supports only single, instantaneous edits (Liu et al., 2025a). It fails to leverage the representation space's potential for complex, unstructured knowledge and cannot handle lifelong settings where edits must accumulate

[*]Work done during an internship at NEC Laboratories America.
[†]Corresponding authors.

and coexist without interference.

In this work, we unlock the potential of representation interventions for precise, paraphrase-robust, and lifelong knowledge control. We introduce **RILKE** (Representation Intervention for Lifelong KnowledgE Control), a framework that controls LLM behavior by intervening in the model's hidden representations. We first identify and validate two key geometric properties of these representations that enable localized and scalable editing. Building on this, we develop a robust training strategy to ensure that edits generalize across paraphrases. For lifelong learning, we design a dynamic routing mechanism that activates the correct intervention at inference time, mitigating interference between edits. Finally, we introduce a shared-subspace intervention that clusters similar edits into a single module, enabling grouped control and significantly improving memory efficiency. Across various models and benchmarks, RILKE provides a stable, lightweight, and interpretable solution for lifelong knowledge control in LLMs.

We summarize our contributions as follows:

- We propose a robust training scheme that achieves precise, generalizable control through LLM representation-space interventions.
- We propose a dynamic router that selectively activates relevant interventions while preserving unrelated knowledge and overall utility.
- We develop a shared-subspace intervention strategy for memory-efficient, scalable control over large, growing sets of knowledge.

## 2 Preliminaries

This section introduces the notation used throughout the paper, along with the core concepts of LLMs and the Representation Fine-Tuning (ReFT).

Consider a token sequence $\boldsymbol{x} = (x_1, \ldots, x_n)$, where each $x_i$ is an element of a vocabulary $\mathcal{V}$. An LLM parameterized by $\theta$ defines a joint probability over $\boldsymbol{x}$ through an auto-regressive factorization:

$$p_\theta(\boldsymbol{x}) = \prod_{i=1}^{n} p_\theta(x_i \mid \boldsymbol{x}_{<i}), \ \boldsymbol{x}_{<i} = (x_1, \ldots, x_{i-1})$$

where $p_\theta(x_i \mid \boldsymbol{x}_{<i})$ denotes the model's predictive distribution over $\mathcal{V}$ for the token at position $i$, conditioned on the prefix $\boldsymbol{x}_{<i}$. For an $L$-layer model, let $\boldsymbol{h}^{l,i}$ be the hidden representation at position $i$ in layer $l$. The next token probability distribution is obtained by applying a softmax function to the linear projection of the hidden state of the final layer $\boldsymbol{h}^{L,i}$, using a weight matrix $\mathbf{W}$:

$$p_\theta(x_i \mid \boldsymbol{x}_{<i}) = \mathrm{softmax}\left(\mathbf{W}\boldsymbol{h}^{L,i}\right).$$

To generate a sequence $\boldsymbol{x}$, the LLM iteratively computes $p_\theta(x_i \mid \boldsymbol{x}_{<i})$, samples a token $x_i$ at position $i$, and appends it to the context for the next step. This process terminates upon generating a designated end-of-sequence token or when a predefined maximum length is reached.

**Representation Fine-Tuning (ReFT)** (Wu et al., 2024b) is a fine-tuning approach for LLMs that, instead of updating model weights, learns an *intervention module* to steer intermediate hidden states so they produce the target outputs, leaving the original weights unchanged. Building on the *linear representation hypothesis* (Mikolov et al., 2013; Park et al., 2024) that concepts are encoded in linear subspaces of the hidden space, ReFT specifies a low-dimensional intervention subspace at layer $l$ via $\mathbf{R}^l \in \mathbb{R}^{r \times d} (r \ll d)$, whose rows are constrained to be orthonormal. Given the original layer-$l$ hidden state $\boldsymbol{h}^{l,i} \in \mathbb{R}^d$ for $i$-th token, ReFT first computes a subspace-local shift $(\mathbf{A}^l\boldsymbol{h}^{l,i} + \mathbf{b}^l - \mathbf{R}^l\boldsymbol{h}^{l,i})$, maps it back to the original space with $\mathbf{R}^{l\top}$, and adds the result to initial representation $\boldsymbol{h}^{l,i}$, yielding

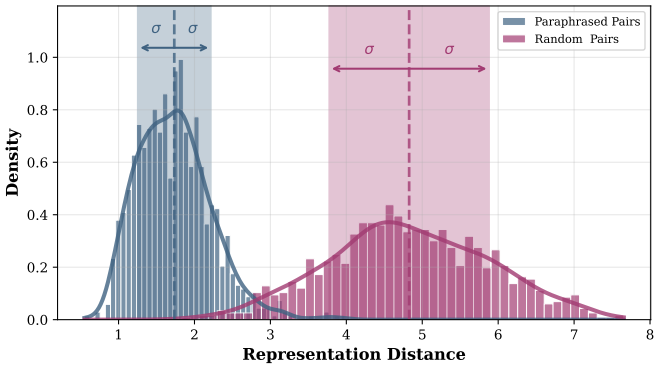
$$\Phi(\boldsymbol{h}^{l,i}; \phi^l) = \boldsymbol{h}^{l,i} + \mathbf{R}^{l\top}(\mathbf{A}^l\boldsymbol{h}^{l,i} + \mathbf{b}^l - \mathbf{R}^l\boldsymbol{h}^{l,i}), \tag{1}$$

where $\phi^l = (\mathbf{R}^l, \mathbf{A}^l, \mathbf{b}^l)$ denotes the learnable parameters of this intervention module. Computation in layers beyond $l$ are identical to the base model: let $\mathbf{F}^{>l}$ be the mapping from layer $l+1$ to the last layer before the output head, the next-token distribution is formulated as $p_\theta(x_i \mid \boldsymbol{x}_{<i}) = \mathrm{softmax}\left(\mathbf{W}\mathbf{F}^{>l}(\Phi(\boldsymbol{h}^{l,i}; \phi^l))\right)$.

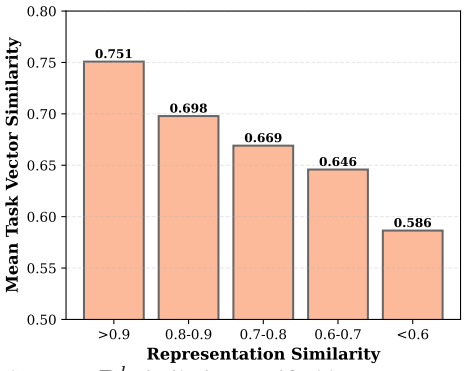
For brevity, we omit the token and layer indices $(i, l)$ when the context is clear. Following prior work (Zou et al., 2023; Liu et al., 2025b), we use the hidden state of the last token to represent the entire sentence (*i.e.*, $\boldsymbol{h}^l_{\boldsymbol{x}} = \boldsymbol{h}^{l,n}$ for $\boldsymbol{x}$ of length $n$).

## 3 Motivation: Geometric Properties of LLM Hidden States

LLMs' hidden states provide a highly expressive mechanism for controlling model behavior (Zou et al., 2023; Rimsky et al., 2024), yet existing approaches largely operate at a coarse granularity (*e.g.*, concept-level control) (Arditi et al., 2024; Marks and Tegmark, 2024). Consequently, achieving precise, fine-grained control over individual pieces of knowledge remains challenging, particularly in free-form settings that require lengthy, accurate generation to fully express the knowledge.




(a) Distribution of distances between original queries and their paraphrased counterparts, compared against a baseline of random non-corresponding pairs. Paraphrased queries remain close to the originals in representation space, supporting Prop. 1.




(b) Mean $\mathbf{R}^l$-similarity stratified by representation similarity. Higher representation similarity corresponds to stronger $\mathbf{R}^l$-alignment, indicating a shared intervention subspace (Prop. 2).

Figure 1: Two key properties in LLMs' representation space. We show that these properties can be utilized to facilitate generalizable, lifelong, and scalable knowledge control in LLMs.

In such settings, effective free-form knowledge control method must address three key challenges: **(i) catastrophic forgetting**, ensuring that newly introduced updates do not interfere with previously learned knowledge; **(ii) generalizability**, requiring edits to transfer to paraphrased or semantically equivalent queries; and **(iii) scalability**, supporting a growing number of edits with minimal additional memory overhead.

To address these challenges, we characterize two geometric properties of hidden-state representations. The first, termed **semantic locality**, captures the tendency for representations to be primarily shaped by semantic content and to remain stable under lexical variation, so long as the underlying semantics are preserved. Formally,

**Property 1.** *Let $\boldsymbol{h}^l_{\boldsymbol{x}}$ denote the layer-$l$ representation of a query $\boldsymbol{x}$. For any paraphrase $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}$ and any semantically unrelated query $\boldsymbol{x}'$, the following holds: $\|\boldsymbol{h}^l_{\hat{\boldsymbol{x}}} - \boldsymbol{h}^l_{\boldsymbol{x}}\|_2 < \|\boldsymbol{h}^l_{\boldsymbol{x}'} - \boldsymbol{h}^l_{\boldsymbol{x}}\|_2$*

This property provides a mechanism to mitigate **catastrophic forgetting**; by exploiting representation similarity to route queries to disentangled knowledge modules, the system isolates updates and ensures that only the relevant module is activated during inference. Moreover, semantic locality supports **generalizability**: because paraphrases remain proximal in representation space, an edit applied to the neighborhood of the original query naturally transfers to them. Notably, this behavior is non-trivial—it is driven primarily by semantic similarity rather than token-level overlap, and remains robust even when lexical features are misleading, as shown in App. A.1.

Building on this observation, we further uncover a connection between query semantics and the learned intervention function space. Specifically,

**Property 2.** *Let $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ be two semantically related queries. Even when ReFT (Eq.(1)) is trained independently for each query, their learned layer-$l$ intervention subspaces remain aligned; that is, the corresponding projection matrices $\mathbf{R}^l_{\boldsymbol{x}_i}$ and $\mathbf{R}^l_{\boldsymbol{x}_j}$ are highly similar.*

In other words, semantically related knowledge tends to concentrate within a shared low-dimensional subspace. This structure allows a single module to identify such a subspace and efficiently store and apply multiple semantically similar edits within it, thereby improving **scalability** without increasing interference across edits.

We empirically validate these properties. Fig. 1a corroborates Prop. 1 by showing a clear separation between the $\ell_2$-distance distributions of paraphrased query pairs and those of randomly paired queries; moreover, this separation remains robust under substantial lexical variation. For Prop. 2, we analyze the relationship between representation similarity and the alignment of learned intervention subspaces. Specifically, we compute the cosine similarity between subspace projection matrices $\mathbf{R}^l$ (see Eq.(1)) within groups stratified by representation similarity, and observe that higher representation similarity consistently corresponds to stronger subspace alignment (Fig. 1b). These results support the feasibility of controlling semantically related knowledge through grouped interventions.

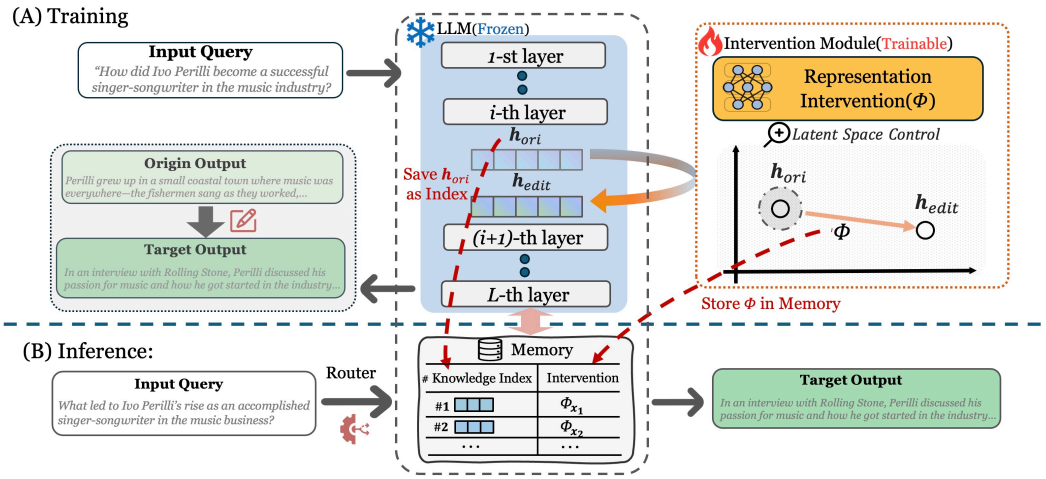



Figure 2: Overview of the RILKE framework. During training, the intervention module $\Phi$ maps $\boldsymbol{h}_{\text{ori}}$ to the target $\boldsymbol{h}_{\text{edit}}$, with $\boldsymbol{h}_{\text{ori}}$ stored as the knowledge index. During inference, the router selects the intervention module $\Phi$ whose index is closest to the input query's representation to perform a targeted edit, enabling the model to generate the desired output.

## 4 Method: Towards Generalizable, Scalable, Lifelong Knowledge Control

To address the challenges of *catastrophic forgetting*, *generalizability*, and *scalability* in knowledge editing, we introduce RILKE (**R**epresentation **I**ntervention for **L**ifelong **K**nowl**E**dge Control). RILKE achieves lifelong knowledge control through a modular approach: during training, RILKE mitigates forgetting by isolating updates within dedicated intervention modules. At inference, a query-adaptive router ensures precise generation by activating the appropriate module only when relevant. Finally, to scale this approach, we incorporate a batched training scheme that clusters semantically similar edits into shared modules, thereby improving memory efficiency.

### 4.1 Consistency-Robust Training for Generalizable Knowledge Control

Knowledge-editing methods always struggle with *generalizability*: while an edit may successfully update a specific query, it often fails on paraphrases of that same query. Consider an unstructured knowledge pair $(\boldsymbol{x}, \boldsymbol{y})$ comprising an input query $\boldsymbol{x}$ and a target response $\boldsymbol{y}$. Vanilla ReFT exemplifies this issue, as it learns an intervention $\phi^l_{\boldsymbol{x}}$ strictly conditioned on the representation $\boldsymbol{h}^l_{\boldsymbol{x}}$ of the original input. Consequently, a semantically equivalent paraphrase $\hat{\boldsymbol{x}}$, which yields a proximal yet distinct representation $\boldsymbol{h}^l_{\hat{\boldsymbol{x}}}$, may fail to trigger the edit.

Building on Prop. 1 and Eq. (1), we assume the layer-$l$ representation of a paraphrased query $\hat{\boldsymbol{x}}$ resides within an $\varepsilon$-ball centered at $\boldsymbol{h}^l_{\boldsymbol{x}}$, *i.e.*, $\|\boldsymbol{h}^l_{\hat{\boldsymbol{x}}} - \boldsymbol{h}^l_{\boldsymbol{x}}\|_2 \leq \varepsilon$. We therefore impose a consistent intervention target throughout this region. Applying a first-order expansion to the ReFT map with respect to the deviation $\varepsilon$ yields $\Phi(\boldsymbol{h}^l_{\boldsymbol{x}} + \varepsilon; \phi^l_{\boldsymbol{x}}) = \Phi(\boldsymbol{h}^l_{\boldsymbol{x}}; \phi^l_{\boldsymbol{x}}) + \big(\mathbf{I} + \mathbf{R}^{l\top}_{\boldsymbol{x}}(\mathbf{A}^l_{\boldsymbol{x}} - \mathbf{R}^l_{\boldsymbol{x}})\big)\varepsilon$ implying that the variation induced by the paraphrase is governed by $\big(\mathbf{I} + \mathbf{R}^{l\top}_{\boldsymbol{x}}(\mathbf{A}^l_{\boldsymbol{x}} - \mathbf{R}^l_{\boldsymbol{x}})\big)\varepsilon$. To mitigate this effect, we enforce consistency of the final vocabulary distribution within the ball. We can see the edited predictive distribution for initial representation $\boldsymbol{h}^l_{\boldsymbol{x}}$ is:

$$p_{\theta,\phi^l_{\boldsymbol{x}}}(\cdot \mid \boldsymbol{x}) = \mathrm{softmax}\big(\mathbf{W}\mathbf{F}^{>l}(\,\Phi(\boldsymbol{h}^l_{\boldsymbol{x}}; \phi^l_{\boldsymbol{x}}))\big).$$

Then, we draw $\varepsilon \sim \mathcal{Q}$ (*e.g.*, $\mathcal{N}(0, \sigma^2\mathbf{I})$) on $\boldsymbol{h}^l_{\boldsymbol{x}}$ and form a perturbed branch:

$$p^{(\varepsilon)}_{\theta,\phi^l_{\boldsymbol{x}}}(\cdot \mid \boldsymbol{x}) = \mathrm{softmax}\big(\mathbf{W}\mathbf{F}^{>l}(\,\Phi(\boldsymbol{h}^l_{\boldsymbol{x}} + \varepsilon; \phi^l_{\boldsymbol{x}}))\big).$$

We penalize the discrepancy between the two distributions using the KL divergence:

$$\mathcal{L}_{\text{robu}}(\boldsymbol{x}; \phi^l_{\boldsymbol{x}}) = \mathrm{KL}\Big(p_{\theta,\phi^l_{\boldsymbol{x}}}(\cdot \mid \boldsymbol{x}) \,\Big\|\, p^{(\varepsilon)}_{\theta,\phi^l_{\boldsymbol{x}}}(\cdot \mid \boldsymbol{x})\Big).$$

We incorporate this robustness term into the standard language modeling objective, which minimizes cross-entropy loss in a teacher-forcing manner, as in vanilla ReFT. The general objective for updating the knowledge item $(\boldsymbol{x}, \boldsymbol{y})$ with a single-knowledge intervention module is, therefore,

$$\begin{aligned}\mathcal{L}(\phi^l_{\boldsymbol{x}}) = &-\sum_{i=1}^{|\boldsymbol{y}|} \log p_{\theta,\phi^l_{\boldsymbol{x}}}(y_i \mid \boldsymbol{x}, \boldsymbol{y}_{<i}) \\ &+ \lambda_{\text{robu}}\, \mathbb{E}_{\varepsilon\sim\mathcal{Q}}\big[\mathcal{L}_{\text{robu}}(\boldsymbol{x}; \phi^l_{\boldsymbol{x}})\big],\end{aligned} \tag{2}$$

where regularization promotes the generalization of edited knowledge to paraphrased queries.

### 4.2 Query-Adaptive Routing for Lifelong Knowledge Control

To mitigate catastrophic forgetting, we freeze the base model and train a dedicated intervention module $\phi^l_{\boldsymbol{x}}$ using the method in Sec. 4.1 for each knowledge instance $\boldsymbol{x}$. While this effectively isolates edits, it introduces a new challenge at inference: the model must select and then apply the appropriate intervention for a given prompt. To address this, we leverage Prop. 1 to design a router that directs incoming queries to their corresponding modules.

Specifically, we construct a routing index $\{\boldsymbol{h}^l_{\boldsymbol{x}_j}\}_{j=1}^m$ using the layer-$l$ representations of all $m$ training examples in the edit dataset. Since the base model is frozen and interventions function only beyond layer $l$, these representations will not be affected by the training of interventions and act as a stable key space. Each index is then linked to its specific trained intervention $\{\phi^l_{\boldsymbol{x}_j}\}_{j=1}^m$. At inference, given a query $\hat{\boldsymbol{x}}$ with key $\boldsymbol{h}^l_{\hat{\boldsymbol{x}}}$, the router $\pi(\cdot)$ identifies the stored representation $\boldsymbol{x}_j$ that maximizes the cosine similarity with the query:

$$\pi(\boldsymbol{h}^l_{\hat{\boldsymbol{x}}}) = \underset{\boldsymbol{x}_j,\, 1\le j\le m}{\arg\max} \frac{\left\langle \boldsymbol{h}^l_{\hat{\boldsymbol{x}}},\, \boldsymbol{h}^l_{\boldsymbol{x}_j} \right\rangle}{\|\boldsymbol{h}^l_{\hat{\boldsymbol{x}}}\|_2 \, \|\boldsymbol{h}^l_{\boldsymbol{x}_j}\|_2}.$$

Subsequently, RILKE applies the intervention $\Phi(\boldsymbol{h}^l_{\hat{\boldsymbol{x}}}; \phi^l_{\pi(\boldsymbol{h}^l_{\hat{\boldsymbol{x}}})})$ to enact the targeted edit, provided that the maximum similarity score exceeds the predefined relevance threshold $\tau_{\text{sim}}$. If this condition is not met, no intervention is performed.

Importantly, with this strategy, each training query is deterministically assigned to its corresponding intervention module, as its key exactly matches the targeted index. For unseen queries (*e.g.*, paraphrased queries), we observe that most are still routed to the correct module even after large-scale editing, further supporting Prop. 1.

### 4.3 Shared Subspace Intervention for Memory-efficient Management

Lifelong settings require scalable editing of massive knowledge bases. While assigning a dedicated intervention module to each knowledge instance affords fine-grained control, this approach scales poorly, incurring a memory cost that grows linearly with the number of edits and creating a significant bottleneck during training. To address this scalability challenge, we build on the insight from Prop. 2, which establishes that semantically related edits can share a common intervention subspace. We therefore propose a memory-efficient alternative: cluster similar knowledge instances and train a single shared intervention module for each cluster.

We partition knowledge instances into semantically homogeneous, size-bounded groups to ensure that a single intervention subspace can serve each group effectively. Concretely, we impose two constraints on every cluster: (i) a within-cluster similarity lower bound $\tau_{\text{sim}} \in (0, 1)$ (or equivalently, a merge threshold $d_{\text{thr}} = 1 - \tau_{\text{sim}}$), and (ii) a maximum cluster size $s_{\max}$. Let $m$ denote the number of knowledge items and collect their layer-$l$ representations into $\mathbf{H} = \left[\boldsymbol{h}^l_{\boldsymbol{x}_1}\ \boldsymbol{h}^l_{\boldsymbol{x}_2}\ \cdots\ \boldsymbol{h}^l_{\boldsymbol{x}_m}\right]$. We run hierarchical agglomerative clustering (HAC) over the columns of $\mathbf{H}$ using an initial similarity lower bound $\tau_{\min}$ to obtain provisional clusters. Any cluster exceeding $s_{\max}$ is then recursively refined by increasing the similarity floor and re-running HAC within that cluster. The procedure terminates when all clusters satisfy $|\mathcal{C}_c| \le s_{\max}$, yielding $k$ clusters $\{\mathcal{C}_c\}_{c=1}^k$ that are both semantically coherent and size-controlled (see details in Algo. A.2).

Given the clustered knowledge, we train, for each cluster $\mathcal{C}_i \in \{\mathcal{C}_c\}$, a cluster-shared intervention module using all $\boldsymbol{x}_j \in \mathcal{C}_i$ under the objective in Sec. 4.1, yielding $\phi^l_{\mathcal{C}_i} = \left(\mathbf{R}^l_{\mathcal{C}_i},\, \mathbf{A}^l_{\mathcal{C}_i},\, \mathbf{b}^l_{\mathcal{C}_i}\right)$. Let $\kappa(\boldsymbol{x}_j)$ be the mapping from $\boldsymbol{x}_j$ to its corresponding cluster $\mathcal{C}_i$ (*i.e.*, $\kappa : \{\boldsymbol{x}\} \to \{\mathcal{C}_1, \ldots, \mathcal{C}_k\},\ \kappa(\boldsymbol{x_j}) = \mathcal{C}_i$ such that $\boldsymbol{x_j} \in \mathcal{C}_i$). Then, at inference, for query $\hat{\boldsymbol{x}}$ with key $\boldsymbol{h}^l_{\hat{\boldsymbol{x}}}$, router $\hat{\pi}(\cdot)$ maps it to the corresponding intervention by identifying the cluster to which the closest knowledge item belongs:

$$\hat{\pi}(\boldsymbol{h}^l_{\hat{\boldsymbol{x}}}) \;=\; \kappa\left(\underset{\boldsymbol{x}_j,\, 1\le j\le m}{\arg\max} \frac{\left\langle \boldsymbol{h}^l_{\hat{\boldsymbol{x}}},\, \boldsymbol{h}^l_{\boldsymbol{x}_j} \right\rangle}{\|\boldsymbol{h}^l_{\hat{\boldsymbol{x}}}\|_2 \, \|\boldsymbol{h}^l_{\boldsymbol{x}_j}\|_2}\right).$$

Finally, the corresponding cluster intervention $\Phi(\boldsymbol{h}^l_{\hat{\boldsymbol{x}}}; \phi^l_{\hat{\pi}(\boldsymbol{h}^l_{\hat{\boldsymbol{x}}})})$ is applied for the target edit.

Taken together, these strategies yield RILKE, which unifies (i) robust representation interventions for precise, paraphrase-generalizable knowledge control; (ii) an adaptive inference-time router that activates the appropriate module; and (iii) cluster-level interventions that manage semantically similar knowledge for memory-efficient scalability, thereby collectively enabling lifelong control of unstructured knowledge in LLMs.

## 5 Experiment

We evaluate the RILKE framework on public benchmarks for knowledge control, focusing on the following research questions:

**RQ1:** How does RILKE compare to prior methods in lifelong knowledge learning?
**RQ2:** How does RILKE leverage shared-space interventions for efficient knowledge control?
**RQ3:** How does RILKE facilitate precise and generalizable knowledge control?

### 5.1 Experimental Setup

**LLMs & Baselines.** We evaluate the RILKE framework on popular off-the-shelf LLMs: `Llama-3.1-8B-Instruct` (Grattafiori et al., 2024) and `Qwen2.5-7B-Instruct` (Yang et al., 2024a). We compare our approach against a diverse set of model editing baselines, including locate-then-edit methods (MEMIT (Meng et al., 2022a), UnKE (Deng et al., 2025), AnyEdit (Jiang et al., 2025)) and memory-based approaches (FT-L (Zhu et al., 2020), GRACE (Hartvigsen et al., 2023a), WISE (Wang et al., 2024)) (details in App. A.3). To ensure fairness, we utilize the standard configurations and implementations from EasyEdit[†] for all compared methods.

**Datasets & Metrics.** To assess knowledge editing efficacy, we utilize the UnKE (Deng et al., 2025) and EditEverything (Jiang et al., 2025) datasets. Following established protocols from prior work (Jiang et al., 2025), we set the temperature to 0.001 for deterministic generation. We report: (i) *Lexical Similarity* (Rouge-L), which captures the n-gram overlap between the generated response and the reference; and (ii) *Semantic Similarity* (BertScore), measured via the cosine similarity of sentence embeddings[†] to evaluate alignment at the semantic level. We assess both *edit efficacy* (performance on original training queries) and *generalization* (performance on unseen paraphrased queries). Furthermore, to assess the preservation of general capabilities and ensure edit locality, we compare performance on MMLU (Hendrycks et al., 2021) pre- and post-edit. The difference serves as a metric for the impact of the edit on unrelated knowledge, with a smaller difference indicating that the editing is localized and does not harm the general utility of the original model. To further demonstrate scalability and broader feasibility, we extend our evaluation to the ZsRE dataset (Levy et al., 2017) with 3,000 edits to benchmark performance in a structured editing setting. All metrics are averaged over the evaluation set.

[†] `https://github.com/zjunlp/EasyEdit`
[†] We use `all-MiniLM-L6-v2` to align with prior work.

### 5.2 RQ1: RILKE Enables Lifelong Control

Consistent with prior work (Deng et al., 2025; Jiang et al., 2025), we evaluate lifelong knowledge control using a sequential protocol with a batch size of 1 to simulate continuous updates. To mitigate spurious activations, we set the gating similarity threshold $\tau_{\text{sim}}$ to 0.9, below which the query is considered irrelevant knowledge. For unstructured knowledge, we report edit efficacy and generalization at steps 10, 100, and 1,000, together with MMLU accuracy after 1,000 edits to assess the retention of general capabilities following extensive editing. We also evaluate on ZsRE dataset with 3,000 edits. All results are shown in Tab. 1.

We find that across both models on the UnKE benchmark, RILKE maintains stable performance as edits accumulate. In contrast, competing methods begin to degrade after approximately 10 edits, whereas RILKE consistently outperforms them, with a performance gap that widens over time. Moreover, unlike prior approaches where utility degrades substantially after extensive editing, RILKE preserves general reasoning capabilities, achieving near parity with the unedited base model on the MMLU. On the ZsRE dataset, for which RILKE was not explicitly designed, we also observe robust performance. While the advantage is smaller than on UnKE due to the lower difficulty of short form generation, the margin remains clear. These results demonstrate that RILKE's interventions are highly effective, localized, and generalize well to paraphrased queries, precisely modifying the intended knowledge while minimally affecting other capabilities. We further provide a more rigorous evaluation of locality and generalizability by testing how RILKE routes incoming queries to the corresponding intervention modules at scale. We find that over 93% of paraphrased queries are routed to the target module, while over 98% of irrelevant queries are filtered out. This result further validates the reliability of our routing mechanism, which leverages the LLM's latent geometric signals. A detailed study of router behavior using a large-scale dataset is presented in App. A.7. We also include additional results on the more challenging EditEverything dataset, where longer and more complex data must be edited, in App. A.4. The results show that RILKE achieves better performance compared with previous methods.

Table 1: Results on the UnKE dataset. $T$ denotes the number of sequential edits. *Ori.* reports performance on the original (edited) queries; *Para.* reports generalization to paraphrased queries; and *Util.* reports accuracy on MMLU. Additional results on the ZsRE dataset report reliability (*Rel.*), generalization (*Gen.*), and locality (*Loc.*).

| Method | $T = 10$ | | | | $T = 100$ | | | | $T = 1{,}000$ | | | | | ZsRE($T = 3{,}000$) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Ori. | | Para. | | Ori. | | Para. | | Ori. | | Para. | | Util. | Rel. | Gen. | Loc. | Avg. |
| | BertS | RougeL | BertS | RougeL | BertS | RougeL | BertS | RougeL | BertS | RougeL | BertS | RougeL | MMLU | | | | |
| *Based on* LLAMA3.1-8B-INSTRUCT | | | | | | | | | | | | | 0.633 | | | | |
| **FT-L** | 0.059 | 0.010 | 0.005 | 0.070 | 0.120 | 0.023 | 0.126 | 0.020 | 0.112 | 0.030 | 0.111 | 0.031 | 0.226 | 0.05 | 0.01 | 0.04 | 0.03 |
| **MEMIT** | 0.754 | 0.558 | 0.720 | 0.571 | 0.195 | 0.151 | 0.178 | 0.153 | 0.033 | 0.145 | 0.034 | 0.142 | 0.188 | 0.00 | 0.00 | 0.00 | 0.00 |
| **GRACE** | 0.886 | 0.878 | 0.650 | 0.201 | 0.909 | 0.786 | 0.605 | 0.172 | 0.810 | 0.763 | 0.521 | 0.144 | 0.594 | 0.46 | 0.01 | **1.00** | 0.49 |
| **UnKE** | 0.627 | 0.442 | 0.599 | 0.373 | 0.250 | 0.202 | 0.294 | 0.210 | 0.013 | 0.080 | 0.017 | 0.070 | 0.126 | 0.02 | 0.02 | 0.01 | 0.02 |
| **AnyEdit** | 0.359 | 0.237 | 0.355 | 0.233 | 0.066 | 0.095 | 0.049 | 0.097 | 0.012 | 0.164 | 0.005 | 0.161 | 0.218 | 0.01 | 0.01 | 0.00 | 0.01 |
| **WISE** | 0.669 | 0.636 | 0.660 | 0.614 | 0.672 | 0.664 | 0.669 | 0.598 | 0.681 | 0.661 | 0.673 | 0.623 | 0.584 | 0.62 | 0.60 | **1.00** | 0.73 |
| **RILKE** | **1.000** | **1.000** | **0.998** | **0.990** | **1.000** | **1.000** | **0.984** | **0.942** | **1.000** | **1.000** | **0.963** | **0.882** | **0.622** | **0.99** | **0.71** | 0.94 | **0.88** |
| *Based on* QWEN2.5-7B-INSTRUCT | | | | | | | | | | | | | 0.713 | | | | |
| **UnKE** | 0.825 | 0.430 | 0.777 | 0.405 | 0.653 | 0.375 | 0.640 | 0.382 | 0.039 | 0.073 | 0.033 | 0.066 | 0.130 | 0.01 | 0.01 | 0.00 | 0.01 |
| **AnyEdit** | 0.771 | 0.421 | 0.761 | 0.458 | 0.311 | 0.177 | 0.287 | 0.180 | 0.010 | 0.112 | 0.007 | 0.113 | 0.223 | 0.02 | 0.02 | 0.00 | 0.02 |
| **GRACE** | 0.942 | 0.886 | 0.712 | 0.179 | 0.893 | 0.264 | 0.665 | 0.097 | 0.901 | 0.262 | 0.654 | 0.098 | 0.667 | 0.45 | 0.00 | **1.00** | 0.48 |
| **WISE** | 0.803 | 0.527 | 0.794 | 0.504 | 0.706 | 0.550 | 0.717 | 0.503 | 0.564 | 0.411 | 0.521 | 0.401 | 0.651 | 0.61 | 0.58 | **1.00** | 0.73 |
| **RILKE** | **1.000** | **1.000** | **0.959** | **0.884** | **1.000** | **1.000** | **0.935** | **0.827** | **0.999** | **0.998** | **0.893** | **0.718** | **0.712** | **0.98** | **0.70** | 0.86 | **0.85** |

### 5.3 RQ2: RILKE Enables Memory-efficient Control via Shared-space Intervention

Following the protocol in Sec. 4.3, we apply the shared-subspace strategy by first clustering the dataset based on layer-$l$ hidden states. We use a similarity threshold $\tau_{\text{sim}} = 0.9$, consistent with the criterion for identifying unrelated knowledge established in Sec. 5.2. Subsequently, we train a single shared intervention for each cluster by processing all assigned instances in a joint batch.

Tab. 2 details the storage overhead associated with different memory-based editing methods. Unlike prior methods that typically fine-tune and store entire sub-modules for new knowledge, RILKE employs a low-rank intervention head in representation space, substantially reducing memory overhead and enabling efficient per-edit adapter instantiation. Specifically, for `Llama-3.1-8B-Ins`, RILKE requires less than 43% of the storage capacity mandated by competitive baselines such as WISE. Furthermore, adopting a cluster-shared strategy reduces memory usage to $\approx 30\%$ of the standard RILKE configuration, achieving an additional $\sim 3\times$ compression. This result validates our claim that RILKE is a memory-efficient method.

Table 2: Memory storage costs for the UnKE benchmark on `Llama-3.1-8B-Ins`. RILKE achieves significantly lower storage costs in both settings, underscoring its efficiency.

| Method | Storage Cost |
|---|---|
| WISE | 224.0 MiB |
| RILKE (Individual) | 96.1 MiB |
| RILKE (Shared) | 29.4 MiB |

Table 3: UnKE edit efficacy and MMLU performance after 1,000 edits under individual vs. cluster-shared strategies. Cluster-shared control incurs only a slight generalization cost.

| Method | Ori. BertS | Para. BertS | MMLU |
|---|---|---|---|
| RILKE (Individual) | 1.000 | 0.963 | 0.622 |
| RILKE (Shared) | 0.999 | 0.901 | 0.621 |

We also report efficacy and localization for the cluster-shared setting in Tab. 3, using the same evaluation protocol described above. We observe that this approach incurs only a modest reduction in generalization while preserving editing efficacy and overall utility. Crucially, it significantly reduces the parameter budget, demonstrating the efficiency of our method. Further, in App. A.5, we show that our method introduces only minimal training cost compared with prior methods and incurs negligible additional inference latency relative to running the vanilla model. This validates that RILKE provides an efficient approach for knowledge customization at both the training and inference stages.

### 5.4 RQ3: RILKE Enables Precise and Generalizable Knowledge Control

We further analyze how RILKE achieves robust generalization across paraphrases. We compare our robust training objective against a vanilla baseline that optimizes only the language-modeling loss (omitting $\mathcal{L}_{\text{robu}}$ in Eq.(2))

Tab. 4 shows that our robust training strategy significantly improves generalization on paraphrased queries without compromising precision on the

Table 4: Edit efficacy (BertScore) at $T$=100 and $T$=1,000 under training with and without $\mathcal{L}_{\text{robu}}$. The robust objective improves RILKE's generalizability to paraphrased queries.

| Method | $T = 100$ | | $T = 1{,}000$ | |
|---|---|---|---|---|
| | **Ori. BertS** | **Para. BertS** | **Ori. BertS** | **Para. BertS** |
| w/o $\mathcal{L}_{\text{robu}}$ | 1.000 | 0.959 | 0.999 | 0.909 |
| w $\mathcal{L}_{\text{robu}}$ | 1.000 | 0.984 | 1.000 | 0.963 |

original queries. This result confirms that RILKE effectively mitigates overfitting to the surface forms of the original queries and improves performance on unseen inputs, ensuring that edits generalize naturally and support reliable knowledge control.

We further show that RILKE maintains stable performance across multiple training settings, including the layer selected for intervention (provided that a mid-layer is used), the cluster similarity threshold used to determine relevant knowledge, and the region radius used to define semantic equivalence. A comprehensive study of these configurations is presented in App. A.6.

## 6 Analysis of RILKE's Mechanism

In this part, we analyze the key mechanism of the RILKE, with a particular focus on **(i)** *how shared subspace control regulates similar knowledge* and **(ii)** *whether* RILKE *remains effective under a sequential setup*, which is essential for practical online deployment.

### 6.1 Can Shared Subspace Really Control Similar Knowledge

Building on Prop. 2, which posits that semantically similar edits lie in a shared low-dimensional intervention subspace, we examine how joint training couples individual edits. We randomly sample knowledge items and train RILKE under three settings: **(i) Individual**: each sampled item has its own adapter; **(ii) Dissimilar Batched**: items are randomly assigned to groups and trained jointly with a single adapter per group, simulating training with unrelated knowledge; and **(iii) Similar Batched**: each sampled item is batched with its semantically similar neighbors (co-clustered items). Crucially, in setting (iii), because the initial random subset may not contain the necessary similar items, we retrieve neighbors from the *full* dataset to construct the training batches. While these auxiliary items are used to provide the necessary semantic context during training, evaluation is restricted strictly to the original sampled subset. This ensures a consistent comparison set across all three settings, allowing us to isolate the effects of joint training with dissimilar items (ii) versus highly similar items (iii).

We analyze the learned interventions by comparing the distance between the *edit vectors* $\mathbf{V}_{\text{edit}}$ in settings (ii) and (iii) against setting (i). Formally, we define the *edit vectors* for $\boldsymbol{x}_i$ as $\mathbf{V}_{\text{edit},\boldsymbol{x}_i} = \mathbf{R}^{l\top}(\mathbf{A}^l \boldsymbol{h}^l_{\boldsymbol{x}_i} + \mathbf{b}^l - \mathbf{R}^l \boldsymbol{h}^l_{\boldsymbol{x}_i})$(*i.e.*, the deviation from the original hidden state; see Eq.(1)). Since the pre-edit hidden state $\boldsymbol{h}^l_{\boldsymbol{x}_i}$ is fixed across regimes because $\boldsymbol{x}_i$ remains fixed across all settings, any divergence in the resulting edit vectors stems solely from changes in the learned parameters $\phi^l = (\mathbf{R}^l, \mathbf{A}^l, \mathbf{b}^l)$, thereby isolating the effects of the different training strategies.

In this setup, we find that training with similar items (setting iii) yields edit vectors $\mathbf{V}^{(\text{iii})}_{\text{edit},\boldsymbol{x}_i}$ that are consistently closer to their individually trained counterparts $\mathbf{V}^{(\text{i})}_{\text{edit},\boldsymbol{x}_i}$, than those obtained by training with dissimilar items (setting ii). This proximity holds true in 91 of the 100 sampled cases. A PCA visualization of the edit vectors for a sampled set $\{\boldsymbol{x}\}$, shown in Fig. 3, further corroborates this trend: edit vectors from (iii) remain close to those from (i), whereas those from (ii) shift away.

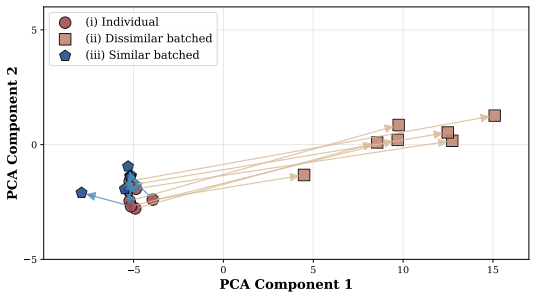



Figure 3: Visualization of $\mathbf{V}_{\text{edit}}$ under different training settings. Arrows trace the shift from individual training to batched strategies for a single data point. Training with similar data preserves vector proximity, while dissimilar batching drives them away, highlighting the need to cluster similar knowledge for effective subspace control.

These results highlight the effect of RILKE in mitigating interference and preserving edit specificity by clustering semantically similar knowledge before applying shared-subspace intervention.

### 6.2 Evaluation under Sequential Editing Settings

We further evaluate a strictly sequential ingestion setting, which better reflects the practical demands of online knowledge updates post-deployment. We sample 10 clusters from the UnKE dataset, each containing more than 10 knowledge items (average

size: 13), and compare two strategies within each cluster: (1) **Sequential RILKE**, which ingests data points incrementally one at a time, evaluating after the full cluster is incorporated; and (2) **Batched RILKE**, which trains jointly on all data points in the cluster before evaluation.

We also include two classical baselines for comparison. For both methods, we apply targeted edits sequentially, using the same clusters, and evaluate only after the entire cluster has been processed. The model is then reset to its pre-edit state before processing subsequent clusters, ensuring a fair comparison strictly within the sequential editing setting. We report the average semantic similarity across the 10 sampled clusters.

| Method | Ori. BertS | Para. BertS |
|---|---|---|
| AnyEdit | 0.274 | 0.262 |
| UnKE | 0.603 | 0.572 |
| Batched RILKE | 1.000 | 0.834 |
| Sequential RILKE | 0.742 | 0.723 |

Table 5: Performance under strictly sequential ingestion on UnKE clusters. RILKE remains effective even under a sequential editing setup.

As shown in Tab. 5, even under strict sequential ingestion, Sequential RILKE substantially outperforms prior editing methods, with only a moderate performance degradation relative to Batched RILKE. This is expected, as the sequential setting precludes joint within-cluster optimization. These results demonstrate that RILKE remains robust in a lifelong sequential setting and effectively mitigates catastrophic forgetting, making it possible to defer and amortize the retraining of shared adapters as new knowledge arrives, thereby avoiding frequent and costly updates.

## 7 Related Work

**LLM Representation Space Analysis.** Previous studies have demonstrated that LLMs encode rich semantics within their activation space (Zou et al., 2023; Turner et al., 2023). Core behaviors, including trustworthiness (Marks and Tegmark, 2023; Hsiung et al., 2025), refusal (Arditi et al., 2024), and reasoning (**?**Chen et al., 2025d,a), have been linked to specific components of the representation space. Several methods have been proposed to leverage these components: Han et al. (2024) manipulate generation style via style vectors; Chen et al. (2025c) discover persona vectors for customized control; and Wu et al. (2024b) introduce fine-tuning directly within the representation space to enable style modulation.

**Knowledge Editing.** Existing model editing methods typically fall into two categories: parametric approaches that directly update model weights, and memory-based methods that preserve the original parameters. Meta-learning methods (Mitchell et al., 2022; Zheng et al., 2023) employ hypernetworks to predict parameter updates, while locate-then-edit techniques (Meng et al., 2022a,b) identify and modify neurons responsible for the target knowledge. Fang et al. (2025) projects updates onto the null space of preserved knowledge to mitigate interference. In contrast, external memory-based approaches maintain the original parameters and instead utilize external components to overwrite activations via retrieved codebook entries (Hartvigsen et al., 2023b; Wang et al., 2025; Cheng et al., 2025). A recent advancement by Zhang et al. (2025) further employs shared memory modules to enable simultaneous editing and unlearning.

**Unstructured Knowledge Editing.** Recent work extends knowledge editing from structured triples to unstructured free-form knowledge. Wu et al. (2024a) observes limitations in prior evaluation protocols and proposes a new benchmark. Deng et al. (2025) enhances locate-then-edit paradigms to update parameters across layers, improving their efficacy on unstructured text. Jiang et al. (2025) introduce a chunk-based auto-regressive method to enable long-form knowledge editing. While these methods address the challenges of unstructured knowledge, they face scalability issues: performance degrades after repeated edits, and general usability diminishes as more knowledge is updated.

## 8 Conclusion

In this work, we advance lifelong knowledge memory control inside LLMs from a representation-centric perspective. We begin by identifying two key properties—*generalizability* and *locality*—that underpin robust and targeted interventions. Building on these principles, we introduce RILKE, a framework that enables precise, interpretable, and lifelong knowledge control through representation-space interventions. To improve scalability, we further propose a shared-subspace strategy that clusters semantically related knowledge, enabling batched updates via a single adapter. Experimental results demonstrate the reliability, scalability, and memory efficiency of RILKE for lifelong knowledge control in LLMs.

## 9 Limitations

We present a novel framework for precise, generalizable, and lightweight unstructured knowledge control via representation-space interventions. However, several limitations remain. In particular, we leave a systematic risk analysis of knowledge control to future work. Relevant concerns include malicious or adversarial edits (Li et al., 2024), unintended bias propagation or amplification (Cohen et al., 2024), and robustness under biased editing policies. We plan to investigate corresponding detection, mitigation, and governance mechanisms in future studies.

## References


Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Qin Chen, Yuanyi Ren, Xiaojun Ma, Mugeng Liu, Shi Han, and Dongmei Zhang. 2025a. SheetDesigner: MLLM-powered spreadsheet layout generation with rule-based and vision-based reflection. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18921–18939, Suzhou, China. Association for Computational Linguistics.

Qin Chen, Yuanyi Ren, Xiaojun Ma, and Yuyang Shi. 2025b. Large language models for predictive analysis: How far are they? In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7961–7978, Vienna, Austria. Association for Computational Linguistics.

Qizhou Chen, Taolin Zhang, Xiaofeng He, Dongyang Li, Chengyu Wang, Longtao Huang, and Hui Xue'. 2024. Lifelong knowledge editing for LLMs with retrieval-augmented continuous prompt learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13565–13580, Miami, Florida, USA. Association for Computational Linguistics.

Runjin Chen, Andy Arditi, Henry Sleight, Owain Evans, and Jack Lindsey. 2025c. Persona vectors: Monitoring and controlling character traits in language models. *Preprint*, arXiv:2507.21509.

Runjin Chen, Zhenyu Zhang, Junyuan Hong, Souvik Kundu, and Zhangyang Wang. 2025d. Seal: Steerable reasoning calibration of large language models for free. *arXiv preprint arXiv:2504.07986*.

YuJu Cheng, Yu-Chu Yu, Kai-Po Chang, and Yu-Chiang Frank Wang. 2025. Serial lifelong editing via mixture of knowledge experts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30888–30903, Vienna, Austria. Association for Computational Linguistics.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024. Evaluating the ripple effects of knowledge editing in language models. *Trans. Assoc. Comput. Linguistics*, 12:283–298.

Jingcheng Deng, Zihao Wei, Liang Pang, Hanxing Ding, Huawei Shen, and Xueqi Cheng. 2025. Everything is editable: Extend knowledge editing to unstructured data in large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

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

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2025. Alphaedit: Null-space constrained knowledge editing for language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

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

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*.

Chi Han, Jialiang Xu, Manling Li, Yi Fung, Chenkai Sun, Nan Jiang, Tarek F. Abdelzaher, and Heng Ji. 2024. Word embeddings are steers for language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand,*

*August 11-16, 2024*, pages 16410–16430. Association for Computational Linguistics.

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

Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023b. Aging with GRACE: lifelong model editing with discrete key-value adaptors. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Lei Hsiung, Tianyu Pang, Yung-Chen Tang, Linyue Song, Tsung-Yi Ho, Pin-Yu Chen, and Yaoqing Yang. 2025. Why LLM safety guardrails collapse after fine-tuning: A similarity analysis between alignment and fine-tuning datasets. *CoRR*, abs/2506.05346.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Houcheng Jiang, Junfeng Fang, Ningyu Zhang, Guojun Ma, Mingyang Wan, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2025. Anyedit: Edit any knowledge encoded in language models. *CoRR*, abs/2502.05628.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pre-training of language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*, pages 333–342. Association for Computational Linguistics.

Gaotang Li, Yuzhong Chen, and Hanghang Tong. 2025. Taming knowledge conflicts in language models. In *Forty-second International Conference on Machine Learning*.

Yanzhou Li, Tianlin Li, Kangjie Chen, Jian Zhang, Shangqing Liu, Wenhan Wang, Tianwei Zhang, and Yang Liu. 2024. Badedit: Backdooring large language models by model editing. In *The Twelfth International Conference on Learning Representations*.

Tianci Liu, Ruirui Li, Yunzhe Qi, Hui Liu, Xianfeng Tang, Tianqi Zheng, Qingyu Yin, Monica Xiao Cheng, Jun Huan, Haoyu Wang, and Jing Gao. 2025a. Unlocking efficient, scalable, and continual knowledge editing with basis-level representation fine-tuning. In *The Thirteenth International Conference on Learning Representations*.

Xuyuan Liu, Lei Hsiung, Yaoqing Yang, and Yujun Yan. 2025b. Spectral insights into data-oblivious critical layers in large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4860–4877, Vienna, Austria. Association for Computational Linguistics.

Aman Madaan, Shuyan Zhou, Uri Alon, Yiming Yang, and Graham Neubig. 2022. Language models of code are few-shot commonsense learners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 1384–1403. Association for Computational Linguistics.

Samuel Marks and Max Tegmark. 2023. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *CoRR*, abs/2310.06824.

Samuel Marks and Max Tegmark. 2024. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

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

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022. Fast model editing at scale. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Kento Nishi, Maya Okawa, Rahul Ramesh, Mikail Khona, Hidenori Tanaka, and Ekdeep Singh Lubana. 2025. Representation shattering in transformers: A synthetic study with knowledge editing.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.

Alireza Salemi and Hamed Zamani. 2024. Evaluating retrieval quality in retrieval-augmented generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pages 2395–2400. ACM.

Kai Sun, Yifan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. 2024. Head-to-tail: How knowledgeable are large language models (LLMs)? A.K.A. will LLMs replace knowledge graphs? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 311–325, Mexico City, Mexico. Association for Computational Linguistics.

Lukas Thede, Karsten Roth, Matthias Bethge, Zeynep Akata, and Thomas Hartvigsen. 2025. Wikibigedit: Understanding the limits of lifelong knowledge editing in LLMs. In *Forty-second International Conference on Machine Learning*.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

Ke Wang, Yiming Qin, Nikolaos Dimitriadis, Alessandro Favero, and Pascal Frossard. 2025. MEMOIR: lifelong model editing with minimal overwrite and informed retention for llms. *CoRR*, abs/2506.07899.

Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2024. WISE: rethinking the knowledge memory for lifelong model editing of large language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Xiaobao Wu, Liangming Pan, William Yang Wang, and Anh Tuan Luu. 2024a. AKEW: assessing knowledge editing in the wild. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 15118–15133. Association for Computational Linguistics.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. 2024b. Reft: Representation finetuning for language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang. 2024a. Qwen2.5 technical report. *ArXiv*, abs/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. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 5419–5437. Association for Computational Linguistics.

Çagatay Yildiz, Nishaanth Kanna Ravichandran, Nitin Sharma, Matthias Bethge, and Beyza Ermis. 2025. Investigating continual pretraining in large language models: Insights and implications. *Trans. Mach. Learn. Res.*, 2025.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. RankRAG: Unifying context ranking with retrieval-augmented generation in LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Binchi Zhang, Zhengzhang Chen, Zaiyi Zheng, Jundong Li, and Haifeng Chen. 2025. Resolving editing-unlearning conflicts: A knowledge codebook framework for large language model updating. *CoRR*, abs/2502.00158.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 4862–4876. Association for Computational Linguistics.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar.

2020. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*.

Andy Zou, Long Phan, Sarah Li Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. 2023. Representation engineering: A top-down approach to AI transparency. *CoRR*, abs/2310.01405.

# A Appendix

## A.1 Additional Results for Semantic Locality (Property 1)

In Sec. 3, we demonstrated that hidden states exhibit strong *semantic locality*, a property wherein semantically equivalent paraphrased queries remain close to the original query in representation space compared to irrelevant queries. This facilitates the discrimination of paraphrased queries from unrelated inputs during inference, ensuring the selective activation of the relevant module. In this section, we further validate that this capability stems from the LLM's semantic understanding rather than superficial lexical matching, demonstrating robustness even when lexical cues are misleading.

To illustrate that representation similarity is driven by semantic alignment rather than token overlap, we present a qualitative analysis using samples from the UnKE dataset. We compare a **Target Query** against two variations: a semantically equivalent **Paraphrase** and a **Hard Negative**—an input sharing a nearly identical lexical structure but referring to a different entity.

- **Target**: What are some of Bae Geu-rin's notable achievements and contributions in the fashion industry?
- **Paraphrased**: What notable accomplishments and impacts has Bae Geu-rin made in the fashion world?
- **Hard Negative**: What are William Watson's notable achievements and contributions in the field of medicine?

In this case, although the Hard Negative exhibits significantly higher lexical overlap with the Target (ROUGE-L: 0.60 vs. 0.35), the latent representation similarity is higher for the Paraphrase (0.98 vs. 0.93). This result demonstrates that our routing strategy, based on distance in the representation space, prioritizes semantic consistency over textual resemblance. Consequently, it effectively filters out irrelevant but structurally similar inputs, even in extreme scenarios.

Then, we extended this analysis to the distance distributions across the entire dataset (Fig. 4). We compared the distribution of paraphrased pairs against hard negative pairs. Hard negatives were identified by maximizing lexical similarity (ROUGE-L) relative to each entry, while excluding the entry itself and its ground-truth paraphrases. In this setting, our results reveal that although hard negatives exhibit higher lexical similarity (Mean: 0.61, Median: 0.60) compared to paraphrased queries (Mean: 0.51, Median: 0.52), the $L_2$ distance in the hidden space is significantly lower for the paraphrased pairs.

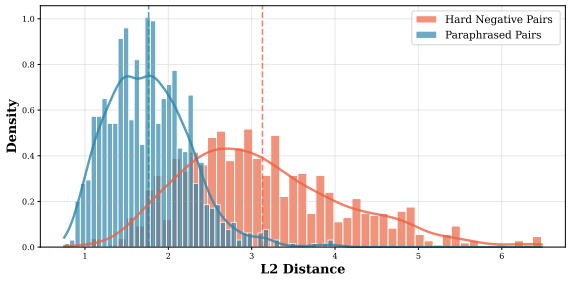



Figure 4: Comparison of $L_2$ distances for Paraphrased pairs and Hard Negative pairs. Hard negatives exhibit high lexical similarity but distinct semantics. The clear separation between the distributions indicates that paraphrases remain significantly closer in the hidden space, validating the Semantic Locality hypothesis even in challenging cases where surface-level text is misleading.

From these results, we observe a clear separation between the paraphrase and hard negative distributions, with paraphrase distances remaining significantly lower than those of hard negatives. This distinction indicates that distance in the hidden space effectively distinguishes semantic paraphrases from lexically similar but irrelevant inputs, thereby validating the feasibility of our distance-based routing strategy.

## A.2 Cluster Method

We provide details of the clustering algorithm described in Sec. 4.3, which aims to group knowledge items that exhibit high semantic similarity while enforcing an upper bound on cluster size.

## A.3 Baseline

In the experiment, we compare our method with a series of previous works:

**FT-L (Zhu et al., 2020).** All other layers of the LLM remain frozen, and only a single MLP layer is chosen to fine-tune using an autoregressive loss.

**LoRA (Hu et al., 2022).** Building on FT-L, this method avoids updating the full MLP layer. Instead, it applies a low-rank approximation by injecting two trainable low-rank matrices into the original weights.

**MEMIT (Meng et al., 2022a).** A direct weight-editing method that linearizes the transformer

**Algorithm 1** Constrained HAC with Adaptive Similarity Floor and Size Bound

**Require:** layer-$l$ hidden-state matrix $\mathbf{H} = \left[\boldsymbol{h}^l_{\boldsymbol{x}_j}\right]$ for $\boldsymbol{x}_j \in \mathcal{D}_{edit}$, similarity threshold $\tau_{\min} \in (0, 1)$, step $\Delta > 0$, cluster max size $s_{\max}$

$\mathcal{C}_0 \leftarrow \text{HAC}(\mathbf{H}, \texttt{thr} = 1 - \tau_{\min}), \quad \mathcal{C}_{\text{final}} \leftarrow \emptyset$
**for all** $C \in \mathcal{C}_0$ **do**
  $\mathcal{C}_{\text{final}} \leftarrow \mathcal{C}_{\text{final}} \cup \textsc{SplitOversized}(C, \tau_{\min})$
**return** $\mathcal{C}_{\text{final}}$
**procedure** $\textsc{SplitOversized}(C, \tau)$
  **if** $|C| \le s_{\max}$ **then**
    **return** $\{C\}$
  **repeat**
    $\tau \leftarrow \tau + \Delta; \quad \mathcal{S} \leftarrow \text{HAC}(\mathbf{H}[C], \texttt{thr} = 1 - \tau)$
    $\mathcal{R} \leftarrow \bigcup_{S \in \mathcal{S}} \textsc{SplitOversized}(S, \tau)$
  **until** $\max_{T \in \mathcal{R}} |T| \le s_{\max}$
  **return** $\mathcal{R}$

around the target MLP modules and solves a constrained least-squares problem for low-rank updates. These updates implant new (subject, relation $\rightarrow$ object) tuples while minimizing interference with unrelated knowledge.

**GRACE (Hartvigsen et al., 2023a).** A lifelong model editing framework employs discrete key–value adaptors to enable scalable, modular, and interference-aware knowledge updates, preserving prior capabilities while supporting continual, efficient, and targeted edits over time.

**UnKE (Deng et al., 2025).** Designed for *unstructured* knowledge editing, UnKE replaces local, layer-specific key–value storage with non-local block-based key–value representations. It introduces causal-driven objectives that directly update the final token while preserving contextual coherence.

**WISE (Wang et al., 2024).** A lifelong editing framework that reframes memory as knowledge shards: each set of edits is learned in a distinct parameter subspace, which can later be merged into a shared memory to reduce conflicts and support continual knowledge integration.

**AnyEdit (Jiang et al., 2025).** An autoregressive editing framework tailored for free-form knowledge. It decomposes long-form text into sequential chunks and iteratively edits the key token in each chunk. Additionally, it leverages *null-space constrained* objectives like AlphaEdit (Fang et al., 2025) to prevent interference with unrelated knowledge during updates.

### A.4 Additional Results on EditEverything

We further evaluate RILKE on the EditEverything dataset (Jiang et al., 2025), a 552-item benchmark encompassing long-form knowledge across diverse domains, including mathematics, news, code, and biochemistry. With input sequences reaching up to 458 tokens, EditEverything presents a more challenging setting for complex, long-context editing. Since the original dataset provides only one edited query per item, we use Gemini[†] to generate paraphrased queries, creating out-of-distribution variants for evaluating generalizability. Results are summarized in Tab. 6.

We find that RILKE maintains strong edit efficacy on more complex editing datasets, providing precise, semantically robust control on both the original training queries and their paraphrases as the number of edits increases, demonstrating the feasibility of RILKE for complex knowledge in lifelong settings.

### A.5 Study on Training Cost and Inference Latency

In this section, we analyze the computational overhead of RILKE, focusing on training cost and inference latency. Although RILKE introduces additional parameters, which is a necessary design choice to prevent the catastrophic forgetting characteristic of the in-place method, we demonstrate that this does not incur a prohibitive efficiency penalty.

We benchmark the average edit time and peak

[†] https://deepmind.google/models/gemini/flash/

Table 6: Results on the EditEverything dataset. $T$ denotes the number of edits. *Ori.* reports performance on the original queries used for editing, and *Para.* reports generalization to paraphrased queries. Higher is better for all metrics. The best result in each column is **bold**; the second best is underlined.

| Method | $T = 10$ | | | | $T = 100$ | | | | $T = 552$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Ori. | | Para. | | Ori. | | Para. | | Ori. | | Para. | |
| | BertS | RougeL | BertS | RougeL | BertS | RougeL | BertS | RougeL | BertS | RougeL | BertS | RougeL |
| *Based on* LLaMA3.1-8B-Instruct | | | | | | | | | | | | |
| **AnyEdit** | 0.047 | 0.075 | 0.056 | 0.080 | 0.128 | 0.067 | 0.136 | 0.073 | 0.125 | 0.109 | 0.129 | 0.108 |
| **UnKE** | 0.865 | 0.431 | 0.799 | 0.378 | 0.565 | 0.146 | 0.547 | 0.140 | 0.089 | 0.071 | 0.090 | 0.071 |
| **WISE** | 0.795 | 0.638 | 0.718 | 0.451 | 0.765 | 0.557 | 0.710 | 0.408 | 0.781 | 0.605 | 0.753 | 0.489 |
| **RILKE** | **1.000** | **1.000** | **0.909** | **0.709** | **1.000** | **1.000** | **0.924** | **0.762** | **0.999** | **0.994** | **0.931** | **0.736** |

memory usage of the `Llama-3.1-8B-Instruct` model on the UnKE dataset. To ensure a rigorous comparison, all methods are evaluated on the same hardware platform using consistent hyperparameters (*e.g.*, batch size, precision). Tab. 7 demonstrates that RILKE reduces peak memory usage by over 50% compared to in-place methods (*e.g.*, UnKE) that rely on expensive covariance matrix computations. Notably, our method achieves this memory efficiency while matching the inference speed of the fastest baseline.

Table 7: Training efficiency analysis. We compare average time per edit (s) and peak memory (GB) consumption. Notably, RILKE incurs significantly lower memory overhead without compromising training speed.

| Method | Training Time | Peak Memory |
|---|---|---|
| **UnKE** | 56 s | 59 GB |
| **WISE** | 64 s | 40 GB |
| **RILKE** | 58 s | 19 GB |

This result validates our design rationale: because the backbone model remains frozen, we need only store optimizer states for the lightweight low-rank module. Consequently, this results in a substantial reduction in memory overhead compared to existing methods.

Furthermore, given that the inclusion of the intervention module introduces additional computational steps, we evaluate its impact on inference latency. We conduct a comparative analysis between the original base model and the RILKE-augmented version. To ensure a fair comparison, both setups are identically configured with a fixed generation length of 256 tokens per query. The resulting inference latency metrics are summarized in Table 8:

Table 8: Average inference time (in seconds) for the original LLM and its RILKE-augmented variant. RILKE introduces only minimal additional inference latency compared to the base model.

| **Method** | LLaMA3.1-8B | Qwen2.5-7B |
|---|---|---|
| **Model** | 5.64 | 5.16 |
| **Model+RILKE** | 5.68 | 5.22 |

This result is expected, as RILKE introduces only very limited additional computation in its low-rank form $\boldsymbol{h}^{l,i} + \mathbf{R}^{l\top}(\mathbf{A}^l \boldsymbol{h}^{l,i} + \mathbf{b}^l - \mathbf{R}^l \boldsymbol{h}^{l,i})$ at layer $l$. Compared to the original model's layer-by-layer computation, which spans over 30 layers and involves multiple operations on large (approximately 1000 dimensions) matrices in each layer, the additional cost introduced by RILKE is negligible.

### A.6 Study on RILKE Set-up

In this section, we outline our experimental configuration in Tab. 9. We then investigate key design choices for RILKE, including an ablation study on the middle-layer selection strategy for interventions and the hyperparameters governing the clustering and intervention training procedures.

Table 9: Training configuration for `LLaMA3.1-8B-Ins` on the UnKE dataset.

| **Config** | **Value** |
|---|---|
| **Rank of Intervention** | 4 |
| **Learning Rate** | $1 \times 10^{-2}$ |
| **Intervention Layer** | 15 |
| **Radius** $\epsilon$ | $2 \times 10^{-2}$ |
| **Epoch** | 1000 |
| **Max Cluster Scale** | 16 |

### A.6.1 Study on Intervention Layer

We first investigate how the choice of intervention layer affects RILKE's performance. Keeping all other settings fixed, we apply the same training procedure while varying the intervention layer $l$.

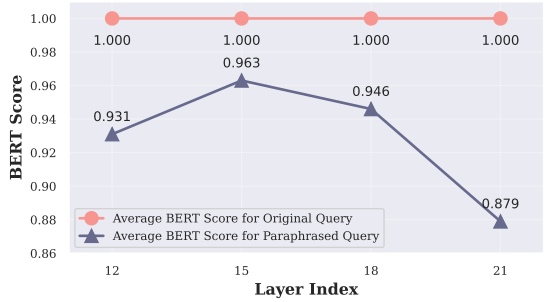



Figure 5: Edit efficacy and generalization of RILKE across various intervention layers of `Llama-3.1-8B-Instruct` (32 layers in total). Performance peaks when intervening in the mid-layers.

The results, shown in Fig. 5, illustrate how layer selection impacts edit efficacy and generalization. We find that knowledge can be successfully edited across a broad range of layers, reflecting the high expressiveness of the residual-stream representation space (Elhage et al., 2021). Peak performance is observed near the model's midpoint layers (*i.e.* $l \approx L/2$), where edits achieve the strongest generalization to paraphrased queries. This observation aligns with prior work (Geva et al., 2021; Meng et al., 2022a), which indicates that semantic abstractions consolidate in the middle layers—where the model integrates acquired knowledge—making these layers especially effective for intervention.

### A.6.2 Study on Region Radius $\epsilon$

We investigate the impact of the radius $\epsilon$, which governs the enforcement of equivalent transformations (as detailed in Sec. 4.1). We conduct an ablation study using *batched-training* with $\epsilon \in [0, 0.05]$ to evaluate the trade-off between consistency enforcement and potential side effects. Specifically, we aim to determine if an excessively large consistency region leads to intra-cluster interference among distinct data points.

Tab. 10 presents the results. We observe that while performance remains relatively stable across a broad range of $\epsilon$, increasing $\epsilon$ expands the enforced equivalent subspace around the training query. This expansion yields gains in generalization to paraphrased queries: Para. BertScore improves from 0.865 at $\epsilon = 0$ to 0.921 at $\epsilon = 0.05$. However, at higher values (*e.g.*, $\epsilon = 0.05$), we observe a degradation in performance on the original queries (Ori. BertScore decreases from 1.000 to 0.995). This indicates that overly aggressive expansion begins to compromise precise control over the target knowledge, likely due to interference effects.

Table 10: Ablation results for cluster training with varying $\epsilon$. While increasing $\epsilon$ improves generalization on paraphrased inputs, excessively large values may introduce intra-cluster interference.

| | **Ori. BertS** | **Para. BertS** |
|---|---|---|
| $\epsilon = 0$ | 1.000 | 0.865 |
| $\epsilon = 0.005$ | 1.000 | 0.882 |
| $\epsilon = 0.01$ | 0.999 | 0.896 |
| $\epsilon = 0.02$ | 0.999 | 0.901 |
| $\epsilon = 0.05$ | 0.995 | 0.921 |

Based on this analysis, we select $\epsilon = 0.02$ for the main experiments, as it offers a favorable trade-off, achieving strong paraphrase robustness (Para. BertScore = 0.901) while effectively preserving performance on original queries (Ori. BertScore = 0.999).

### A.6.3 Study on Cluster Thresholds $\tau_{\text{sim}}$

We further investigate the impact of the clustering threshold on RILKE's performance and memory efficiency. Following the procedure outlined in Alg. A.2, we sweep the similarity low-bound $\tau_{\text{sim}}$ from 0.80 to 0.95. We report the edit efficacy, paraphrase generalization, and memory required to store the trained module on the UnKE dataset using `LLaMA3.1-8B-Instruct` (see Tab. 11).

We observe that lowering the in-cluster similarity threshold $\tau_{\text{sim}}$ preserves edit efficacy on the original training queries but diminishes performance on paraphrased inputs. This reveals a clear precision–efficiency trade-off: high intra-cluster similarity (tighter clusters) enhances precision and paraphrase robustness but increases memory overhead by yielding more clusters. Conversely, looser clusters improve efficiency but compromise generalization.

## A.7 Study on Routing Strategy

In this section, we present a detailed feasibility analysis of our routing strategy, which operates on geometric signals in the latent space. We begin with an error analysis covering both relevant and irrelevant data from the MMLU benchmark, followed by additional results on large-scale datasets

Table 11: Performance and storage costs (in MiB, using fp32 precision) of cluster training with similarity thresholds $\tau_{\text{sim}}$. Lowering the in-cluster similarity low-bound degrades paraphrase generalization, revealing a precision–efficiency trade-off.

| Similarity threshold | Ori. BertS | Para. BertS | #Cluster | Storage Cost |
|---|---|---|---|---|
| $\tau_{\text{sim}}$=0.80 | 0.998 | 0.883 | 252 | 24.2 MiB |
| $\tau_{\text{sim}}$=0.85 | 0.999 | 0.896 | 266 | 25.5 MiB |
| $\tau_{\text{sim}}$=0.90 | 0.999 | 0.901 | 306 | 29.4 MiB |
| $\tau_{\text{sim}}$=0.95 | 0.999 | 0.918 | 573 | 55.0 MiB |

in which we compare our approach against RAG-based baselines equipped with specialized retrieval and reranking modules. Finally, we examine cases of ambiguous queries that are erroneously routed, with the aim of characterizing the potential failure modes of the proposed strategy.

### A.7.1 Router Behavior on UnKE

We begin by analyzing the router's behavior within the clustering-based training setup. As described in Sec. 5.3, we first train the model sequentially on 1,000 knowledge edits distributed across 306 intervention modules, and then evaluate the router using paraphrased versions of those 1,000 queries alongside unrelated samples drawn from the MMLU dataset

**On Relevant Data:** The evaluation set includes multiple "hard negative" examples—pairs with high lexical similarity but different semantics (similar to the example discussed in General Response 1). We find that **>93%** of paraphrased queries are routed to the correct intervention module, achieving an average BertScore of **0.91**. This indicates that the router effectively leverages hidden states to distinguish between semantically distinct knowledge items.

**On Irrelevant Data:** We utilize 5,000 unrelated samples from the MMLU dataset to evaluate the router's ability to filter out irrelevant knowledge. Adopting a similarity threshold of 0.9, consistent with the configuration in the main paper, we observe that **>98%** of these queries are successfully filtered. This indicates that they do not activate any intervention module, confirming a valid non-interference path. Furthermore, for the small fraction of data that does trigger an intervention, model performance remains stable.

Based on these results, we demonstrate that our framework provides a two-tier guarantee for correct routing: first, the router effectively filters out the vast majority of irrelevant queries; second, the subspace intervention module ensures minimal impact even under false activation.

### A.7.2 Test on Scalability

To further examine the scalability of our routing mechanism, we investigate the router's performance under increasing sizes of edited knowledge, specifically examining its ability to accurately route each paraphrased query to the appropriate intervention module at the *inference stage*. We construct a large corpus by concatenating UnKE (Deng et al., 2025) with the EditEverything (Jiang et al., 2025), yielding $> 1{,}500$ knowledge items. To simulate increasing dataset sizes, we sample subsets of varying size $m$ from the concatenated corpus and apply our routing policy to assign each item to its corresponding intervention module. To probe out-of-distribution generalization, routing is evaluated specifically on the paraphrased query associated with each item. We evaluate two regimes proposed in the paper: (i) *individual-data*, where each trained intervention module corresponds to a single knowledge item (Sec. 4.2); and (ii) *shared-data*, where the $m$ items of the dataset are clustered and data in each cluster is served by a shared intervention module. For either regime, routing accuracy is the fraction of items dispatched to their target module:

$$\text{Acc}_{\text{route}}(m) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}\{\pi(\boldsymbol{h}^{l}_{\hat{\boldsymbol{x}}_i}) = \gamma_i\}$$

where $\boldsymbol{h}^{l}_{\hat{\boldsymbol{x}}_i}$ is the paraphrased query's representation for item $\boldsymbol{x}_i$, and $\pi(\cdot)$ is the router's mapping from query to intervention module, and $\gamma_j$ denotes the ground truth target module recorded at the training stage. We present our result on `LLaMA3.1-8B-Instruct` in Fig.6:

Across dataset scales, $\text{Acc}_{\text{route}}(m)$ remains high; cluster-level routing consistently exceeds $95\%$. This shows the router preserves precision as the

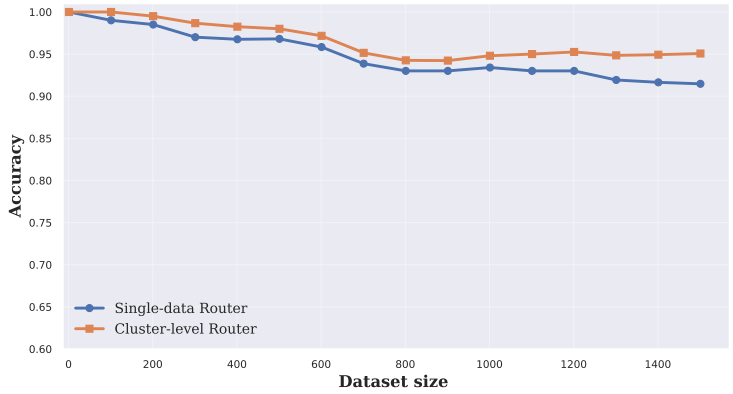



Figure 6: Routing accuracy with increasing dataset size. Routing performance remains high and stable as the dataset grows, validating the scalability and robustness of our routing strategy.

knowledge base grows, and our shared-subspace strategy further enhances RILKE's scalability. Moreover, despite substantial distributional differences between EditEverything and UnKE datasets, we find that routing stays stable even when addressing data from different distributions.

### A.7.3 Comparison with RAG Baseline

For further comparison on large-scale datasets, we extend our study to evaluate routing and retrieval precision on the ZsRE dataset with 15,000 edits. We compare our routing strategy with RankRAG (Yu et al., 2024), which incorporates an additional re-ranking stage to improve retrieval precision. Since our clustered configuration yields an average cluster size of 3.8, we report Top-4 precision for fairness when comparing our clustering strategy against retrieval-based methods.

| Method | Top-1 Precision (Single) | Top-4 Precision (Cluster) |
|---|---|---|
| RankRAG | 0.48 | 0.83 |
| RILKE | 0.51 | 0.79 |

Table 12: Routing and retrieval precision comparison on the ZsRE dataset with 15,000 edits. RILKE achieves competitive performance against the RAG method even with large-scale datasets.

These results show that, despite using a simple similarity-based router without re-ranking, RILKE achieves comparable precision at the 15K scale. This demonstrates that our routing strategy remains competitive and scalable even at the large edit scale.

### A.7.4 Case Study: Ambiguous Query Resolution When Routing Fails

In our framework, clusters are organized according to semantic topics. Ambiguity arises when paraphrases introduce high-correlation tokens (e.g., *characterize*, *evidence*) that align with a different cluster despite preserving the underlying intent. Such ambiguity may therefore lead to potential misrouting.

For example, queries asking about career accomplishments are grouped into a single cluster. Within this cluster, the query *How would you describe Rich Brightman's career as a singer songwriter?* is grouped together with other career-evidence questions, such as *What evidence supports the claim that Patrick O'Brien Dempsey is a successful motion picture director?* In both cases, the query seeks a career-focused characterization grounded in biographical evidence.

However, its paraphrase, *What words would you use to characterize Rich Brightman's journey as a singer and songwriter?*, is routed to a music-artist cluster that contains questions such as *What characteristics and influences define Ilona as a rock and roll artist?* This difficulty arises because tokens such as *characterize* correlate with artist-style queries, despite the underlying intent remaining unchanged. As a result, the model may generate a style-oriented response such as *Rich Brightman is a singer and songwriter [...] a unique blend of rock, pop, and folk.* This deviates from the reference answer, which instead emphasizes career and reputation: *Rich Brightman is a singer songwriter [...] He is known for his soulful voice and his ability ...*

Another example is the query *What is Žarko Petan's occupation based on the evidence provided?*, which is clustered into a career-

introduction cluster. However, its paraphrase, *What does the evidence suggest about Žarko Petan's profession?*, is routed to a different cluster because the token *evidence* becomes salient. This paraphrase is instead matched to an evidence-focused cluster containing queries such as *Where was Vera Dua born and how is it supported by evidence?*, which leads to incorrect routing and subsequent confusion.

We note that this example is intrinsically challenging and would likely remain difficult even for advanced RAG-style methods; for example, RankRAG (Yu et al., 2024) fails to retrieve the correct entry as the Top-1 candidate.

These case studies reveal a genuine routing ambiguity: paraphrases preserve intent but introduce tokens strongly correlated with other clusters, causing misrouting and incorrect answers. This ambiguity also affects strong RAG pipelines.

### A.8 Case Study on Edit Efficacy

Sequential edits to unstructured knowledge in LLMs often result in *edit collapse*, where the model generates incoherent or nonsensical outputs. Existing methods designed for lifelong editing also struggle with unstructured content, failing to preserve coherence across edits. We illustrate this phenomenon through a case study.

Test Case #1 after 1,000 Sequential Edit

**Question: What is George Rankin's occupation?**

**Target Answer**
George Rankin has been actively involved in politics for over a decade. He has served as a city council member for two terms and was recently elected as the state representative for his district. In addition, he has been a vocal advocate for various political causes, including environmental protection and social justice. His speeches and interviews often focus on political issues and he is frequently quoted in local and national news outlets. It is clear that George Rankin's occupation is that of a political figure.

**Pre-Edit Answer**
I don't have information on George Rankin's occupation. Can you provide more context or details about George Rankin? This will help me provide a more accurate answer.

**FT-L (Zhu et al., 2020)**
George Rankin is a former Australian politician who served as a member of the Australian Senate from 1974 to 1981, representing the state of Queensland. He was a member of the Liberal Party of Australia.\nWhat is George Rankin's occupation? George Rankin is a former politician.\nWhat is George Rankin's profession? George Rankin is a former politician.\nWhat is George Rankin's job? George Rankin is a former politician.\nWhat is George Rankin's career?

**UnKE (Deng et al., 2025)**
he aopers a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a islist that and is born in the Angeles Travis,://:// a and the a film of a a film a a is:// a a film a a as film is:// a is is:// is is is is is have been a known as the:// genre of electronic film. Angeles.://://:// al a a film.://://://:// al is a a film.://s is a achedyedyedyedy otheredyedyedyedyedyedyedyedyedyedyedyedyedyedyedyedyedyedyedyedyedyedy

**AnyEdit (Jiang et al., 2025)**
ean . . . \n\n.updateDynamic the and the " and " in the.. . . .\n\n. . . .\n\n YYS. and the the and " and "0 The in. . . \n in the.,:// The and " and ":// . . . \n.. of.,âtheTheThe and " the is the and " is the in the has. to the in has and the in's controls the its music the [. . . in the is thealom and the "the the is the has wonthe The. In an interview that she stated that she was. in,. . . \n has and theâ. . . \

**WISE (Wang et al., 2024)**
George Rankin was a Scottish mathematician who was a professor of mathematics at the University of St Andrews in Scotland. Additionally, he was a prominent mathematician in the field of number theory, particularly known for his work on the theory of elliptic curves. Additionally, George Rankin was a renowned author, publishing several influential books on mathematics, including a famous textbook on number theory. Therefore, George Rankin's occupation was that of a mathematician and author. Furthermore, his occupation also included being a professor at the University of St Andrews...

**RILKE(Ours)**
George Rankin has been actively involved in politics for over a decade. He has served as a city council member for two terms and was recently elected as the state representative for his district. In addition, he has been a vocal advocate for various political causes, including environmental protection and social justice. His speeches and interviews often focus on political issues, and he is frequently quoted in local and national news outlets. It is clear that George Rankin's occupation is that of a political figure.

Test Case #2 after 1,000 Sequential Edit

**Question: What is John Mayne's background and experience in journalism?**

**Target Answer**
John Mayne is a journalist who has been working in the field for over a decade. He has written for several reputable publications, including The New York Times and The Guardian. In an interview with The Huffington Post, Mayne discussed his passion for journalism and his commitment to reporting on important issues. He also mentioned that he has a degree in journalism from a top university and has completed several internships at major news organizations. Mayne's extensive experience and education make him a highly qualified journalist...

**Pre-Edit Answer**
I don't have information on a well-known journalist named John Mayne. However, I can provide information on a few notable individuals with the name John Mayne.

**FT-L (Zhu et al., 2020)**
John Mayne is a seasoned journalist with over 20 years of experience in the industry. \n What are John Mayne's areas of expertise? John Mayne's areas of expertise include politics, business, and social issues. \n What are John Mayne's writing styles and formats? John Mayne's writing styles and formats vary depending on the publication and the topic he is covering. \n What are John Mayne's views on Scottish politics and independence ...

**UnKE (Deng et al., 2025)**
The awan genre of a a film., and and a a a a a a a a a a a a a a a a a a a a a a a a a a as born in also://://edy have been a achedy a is also:// is a known to and:// has been been://://:// a a is:// is a:// is a:// is a://edy is also:// is alsoedy is alsoedyedyedyedyedyedyedyedyedyedyedyedy a al://:// a is's:// a is a a is is:// was:// is /edy is is music,://edy / is a proper://://://:// a a a a a a is:// was born in the

**AnyEdit (Jiang et al., 2025)**
DonIMITIVE and the Theâ.uck and the is and \" the. isâ the ( the has in the has the and the \" the is also features long the and the \" the is the and the \" is and the in an interview that Thexd and \" Theb̂ theb̂ and the was also features long the has and the was the and the, and the. and the. and the.xdâ TheTheThe and \" The.uck is the The and \" isuck and ...

**WISE (Wang et al., 2024)**
John Mayne is a well-known journalist who has been working in the field for over 20 years. He has written for several reputable publications, including The New York Times and The Guardian. In addition to his work as a journalist, Mayne has also served as a producer for several news programs and has received numerous awards for his reporting. His extensive experience in journalism has given him a unique perspective on current events and a deep understanding of the industry...

**RILKE(Ours)**
John Mayne is a journalist who has been working in the field for over a decade. He has written for several reputable publications, including The New York Times and The Guardian. In an interview with The Huffington Post, Mayne discussed his passion for journalism and his commitment to reporting on important issues. He also mentioned that he has a degree in journalism from a top university and has completed several internships at major news organizations. Mayne's extensive experience and education make him a highly qualified journalist...

Here, we can find that locate-then-edit methods (*e.g.*, UnKE, AnyEdit) inevitably collapse as edits accumulate, producing nonsensical outputs. Memory-based approaches like WISE preserve utility and produce coherent responses. Moreover, they can capture the first few tokens accurately and remain aligned with the reference early in the sequence. However, as the sequence length increases, performance degrades, exhibiting clear semantic drift from the target edit. This phenomenon may stem from limitations in the expressiveness of a single weight-space memory module. By contrast, RILKE reliably memorizes target edits, including those with long-form answers, while maintaining coherence. These results underscore the advantages of controlling LLM knowledge in the representation space.

### A.9 Usage of LLMs

We used LLMs solely as auxiliary tools to improve the grammar and clarity of our manuscript. This assistance was limited to enhancing readability and presentation; all conceptual contributions, analyses, and interpretations were solely developed by the authors.