

Representation Interventions Enable Lifelong Unstructured Knowledge Control

Anonymous ACL submission

Abstract

Large language models (LLMs) often produce incorrect or outdated content. Updating their knowledge efficiently and accurately without costly retraining is a major challenge. This problem is particularly challenging for complex, unstructured knowledge in *lifelong* settings, where many edits 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. Code: [Link](#).

1 Introduction

Large language models (LLMs) excel at knowledge-intensive NLP tasks (Madaan et al., 2022; Sun et al., 2024), yet their knowledge is static. Once deployed, they cannot adapt to evolving real-world information, often producing outdated or inaccurate content. While methods like full 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); however, it suffers from conflicts with the model’s parametric memory (Li et al., 2025; Gutiérrez et al., 2025) and is sensitive to retrieval quality (Salemi and Zamani, 2024). These challenges underscore the need for **lifelong model editing**: methods to precisely and efficiently update LLM knowledge with minimal side effects (Thede et al., 2025).

However, existing editing methods struggle in lifelong 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 edits accumulate, performance degrades until the model becomes unusable (Yang et al., 2024b; Nishi et al., 2025)—a degradation particularly severe for 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 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 and coexist without interference.

In this work, we unlock the potential of representation-space interventions for precise,

043
044
045
046
047
048
049
050
051
052
053
054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070
071
072
073
074
075
076
077
078
079
080
081
082
083

paraphrase-robust, and lifelong knowledge control. We introduce RILKE (Representation Intervention for Lifelong Knowledge E 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 $\mathbf{x} = (x_1, \dots, x_n)$, where each x_i is an element of a vocabulary \mathcal{V} . An LLM parameterized by θ defines a joint probability over \mathbf{x} through an auto-regressive factorization:

$$p_\theta(\mathbf{x}) = \prod_{i=1}^n p_\theta(x_i | \mathbf{x}_{<i}), \quad \mathbf{x}_{<i} = (x_1, \dots, x_{i-1})$$

where $p_\theta(x_i | \mathbf{x}_{<i})$ denotes the model’s predictive distribution over \mathcal{V} for the token at position i , conditioned on the prefix $\mathbf{x}_{<i}$. For an L -layer model, let $\mathbf{h}^{l,i}$ 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 $\mathbf{h}^{L,i}$, using a weight matrix \mathbf{W} :

$$p_\theta(x_i | \mathbf{x}_{<i}) = \text{softmax}(\mathbf{W}\mathbf{h}^{L,i}).$$

To generate a sequence \mathbf{x} , the LLM iteratively computes $p_\theta(x_i | \mathbf{x}_{<i})$, samples a token x_i at posi-

tion 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 $\mathbf{h}^{l,i} \in \mathbb{R}^d$ for i -th token, ReFT first computes a subspace-local shift ($\mathbf{A}^l \mathbf{h}^{l,i} + \mathbf{b}^l - \mathbf{R}^l \mathbf{h}^{l,i}$), maps it back to the original space with $\mathbf{R}^{l\top}$, and adds the result to initial representation $\mathbf{h}^{l,i}$, yielding

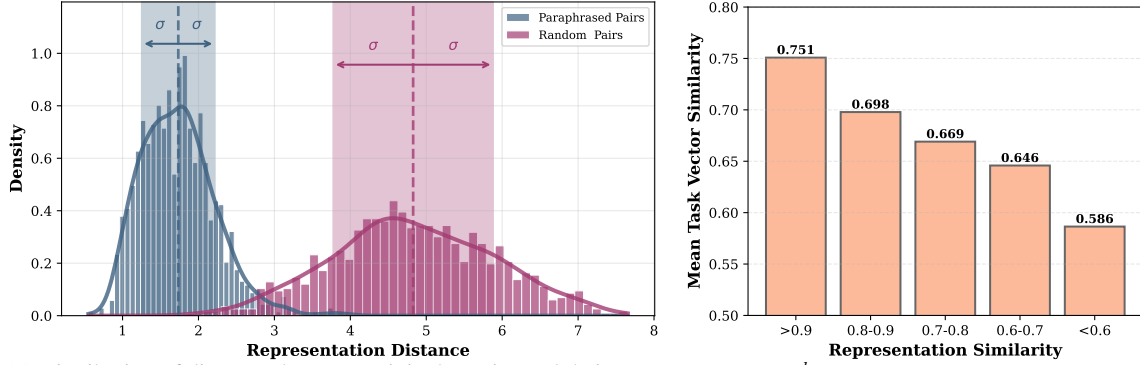
$$\Phi(\mathbf{h}^{l,i}; \phi^l) = \mathbf{h}^{l,i} + \mathbf{R}^{l\top}(\mathbf{A}^l \mathbf{h}^{l,i} + \mathbf{b}^l - \mathbf{R}^l \mathbf{h}^{l,i}), \quad (1)$$

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 $p_\theta(x_i | \mathbf{x}_{<i}) = \text{softmax}(\mathbf{W}\mathbf{F}^{>l}(\Phi(\mathbf{h}^{l,i}; \phi^l)))$.

For brevity, we omit the token and layer indices (i, l) when the context is clear. Following prior work (Zou et al., 2023), we use the hidden state of the last token to represent the entire sentence (*i.e.*, $\mathbf{h}_x^l = \mathbf{h}^{l,n}$ for \mathbf{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; Rinsky 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 knowledge settings that require long and accurate generations. In such settings, effective free-form knowledge control must address three key challenges: **(i) catastrophic forgetting**, ensuring that newly introduced updates do not in-



(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.

181 interfere with previously learned knowledge; **(ii) generalizability**, requiring edits to transfer to para- 182
 183 phrased or semantically equivalent queries; and **(iii) scalability**, supporting a growing number of 184
 185 edits with minimal additional memory overhead.

186 To address these challenges, we characterize two 187
 188 geometric properties of hidden-state representa- 189
 190 tions. The first, termed **semantic locality**, captures 191
 192 the tendency for representations to be primarily 193
 194 shaped by semantic content and to remain stable 195
 196 under lexical variation, so long as the underlying 197
 198 semantics are preserved. Formally,

193 **Property 1.** Let \mathbf{h}_x^l denote the layer- l representa- 194
 195 tion of a query x . For any paraphrase \hat{x} of x and 196
 197 any semantically unrelated query x' , the following 198
 199 holds: $\|\mathbf{h}_{\hat{x}}^l - \mathbf{h}_x^l\|_2 < \|\mathbf{h}_{x'}^l - \mathbf{h}_x^l\|_2$

200 This property provides a mechanism to mitigate 201
 202 **catastrophic forgetting**; by exploiting represen- 203
 204 tation similarity to route queries to disentangled 205
 206 knowledge modules, the system isolates updates 207
 208 and ensures that only the relevant module is acti- 209
 210 vated during inference. Moreover, semantic local- 211
 212 ity supports **generalizability**: because paraphrases 213
 214 remain proximal in representation space, an edit ap- 215
 216 plied to the neighborhood of the original query nat- 217
 218 urally transfers to them. Notably, this behavior is 219
 220 non-trivial—it is driven primarily by semantic sim- 221
 222 ilarity rather than token-level overlap, and remains 223
 224 robust even when lexical features are misleading, 225
 226 as shown in App. A.1.

211 Building on this observation, we further uncover 212
 213 a connection between query semantics and the 214
 215 learned intervention function space. Specifically,

214 **Property 2.** Let x_i and x_j be two semantically re- 215
 216 lated queries. Even when ReFT (Eq.(1)) is trained 217
 218 independently for each query, their learned layer- l

217 *intervention subspaces remain aligned; that is, the* 218
 219 *corresponding projection matrices $\mathbf{R}_{x_i}^l$ and $\mathbf{R}_{x_j}^l$* 220
 221 *are highly similar.*

220 In other words, semantically related knowl- 221
 222 edge tends to concentrate within a shared low- 223
 224 dimensional subspace. This structure allows a sin- 225
 226 gle module to identify such a subspace and effi- 227
 228 ciently store and apply multiple semantically sim- 229
 230 ilar edits within it, thereby improving **scalability** 231
 232 without increasing interference across edits.

227 We empirically validate these properties. Fig. 1a 228
 229 corroborates Prop. 1 by showing a clear separa- 230
 231 tion between the ℓ_2 -distance distributions of para- 232
 233 phrased query pairs and those of randomly paired 234
 235 queries; moreover, this separation remains robust 236
 237 under substantial lexical variation. For Prop. 2, 238
 239 we analyze the relationship between representation 240
 241 similarity and the alignment of learned interven- 242
 243 tion subspaces. Specifically, we compute the cosine 244
 245 similarity between subspace projection matrices \mathbf{R}^l 246
 247 (see Eq.(1)) within groups stratified by represen- 248
 249 tation similarity, and observe that higher represen- 250
 251 tation similarity consistently corresponds to stronger 252
 253 subspace alignment (Fig. 1b). These results support 254
 255 the feasibility of controlling semantically related 256
 257 knowledge through grouped interventions.

243 4 Method: Towards Generalizable, 244 245 Scalable, Lifelong Knowledge Control

245 To address the challenges of *catastrophic forget-* 246
 247 *ting, generalizability, and scalability* in knowledge 248
 249 editing, we introduce RILKE (**R**epresentation 250
 251 **I**ntervention for **L**ifelong **K**nowledge **C**ontrol). 252
 253 RILKE achieves lifelong knowledge control 254
 255 through a modular approach: during training, 256
 257 RILKE mitigates forgetting by isolating updates

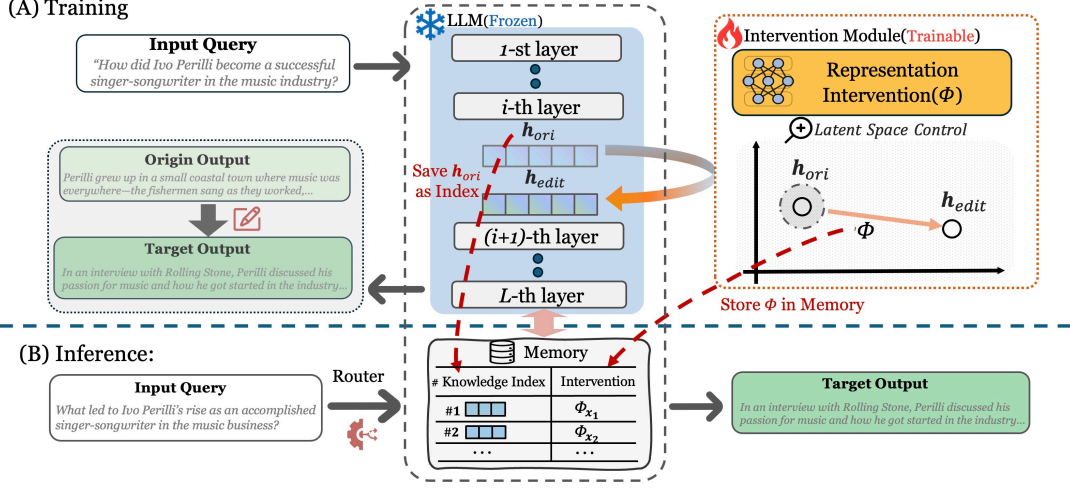


Figure 2: Overview of the RILKE framework. During training, the intervention module Φ maps h_{ori} to the target h_{edit} , with h_{ori} stored as the knowledge index. During inference, the router selects the intervention module Φ whose index is closest to the input query’s representation to perform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

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 (x, y) comprising an input query x and a target response y . Vanilla ReFT exemplifies this issue, as it learns an intervention ϕ_x^l strictly conditioned on the representation h_x^l of the original input. Consequently, a semantically equivalent paraphrase \hat{x} , which yields a proximal yet distinct representation $h_{\hat{x}}^l$, may fail to trigger the edit.

Building on Prop. 1 and Eq. (1), we assume the layer- l representation of a paraphrased query \hat{x} resides within an ε -ball centered at h_x^l , i.e., $\|h_{\hat{x}}^l - h_x^l\|_2 \leq \varepsilon$. We therefore impose a consistent intervention target throughout this region. Applying a first-order expansion to the ReFT map with respect to the deviation ε yields $\Phi(h_x^l + \varepsilon; \phi_x^l) = \Phi(h_x^l; \phi_x^l) + (\mathbf{I} + \mathbf{R}_x^l \mathbf{A}_x^l - \mathbf{R}_x^l) \varepsilon$ implying that the variation induced by the paraphrase is governed by $(\mathbf{I} + \mathbf{R}_x^l \mathbf{A}_x^l - \mathbf{R}_x^l) \varepsilon$. 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 h_x^l is:

$$p_{\theta, \phi_x^l}(\cdot | x) = \text{softmax}(\mathbf{W}\mathbf{F}^{>l}(\Phi(h_x^l; \phi_x^l))). \quad 286$$

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

$$p_{\theta, \phi_x^l}^{(\varepsilon)}(\cdot | x) = \text{softmax}(\mathbf{W}\mathbf{F}^{>l}(\Phi(h_x^l + \varepsilon; \phi_x^l))). \quad 287$$

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

$$\mathcal{L}_{\text{robu}}(x; \phi_x^l) = \text{KL}(p_{\theta, \phi_x^l}(\cdot | x) \| p_{\theta, \phi_x^l}^{(\varepsilon)}(\cdot | x)). \quad 288$$

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 (x, y) with a single-knowledge intervention module is, therefore,

$$\mathcal{L}(\phi_x^l) = - \sum_{i=1}^{|y|} \log p_{\theta, \phi_x^l}(y_i | x, y_{<i}) \quad (2) \quad 289$$

$$+ \lambda_{\text{robu}} \mathbb{E}_{\varepsilon \sim \mathcal{Q}}[\mathcal{L}_{\text{robu}}(x; \phi_x^l)], \quad 290$$

where regularization promotes the generalization of edited knowledge to paraphrased queries. 300

4.2 Query-Adaptive Routing for Lifelong Knowledge Control 301

To mitigate catastrophic forgetting, we freeze the base model and train a dedicated intervention module ϕ_x^l using the method in Sec. 4.1 for each knowledge instance 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. 302

Specifically, we construct a routing index $\{\mathbf{h}_{x_j}^l\}_{j=1}^m$ 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 $\{\phi_{x_j}^l\}_{j=1}^m$. At inference, given a query \hat{x} with key $\mathbf{h}_{\hat{x}}^l$, the router $\pi(\cdot)$ identifies the stored representation \mathbf{x}_j that maximizes the cosine similarity with the query:

$$\pi(\mathbf{h}_{\hat{x}}^l) = \arg \max_{x_j, 1 \leq j \leq m} \frac{\langle \mathbf{h}_{\hat{x}}^l, \mathbf{h}_{x_j}^l \rangle}{\|\mathbf{h}_{\hat{x}}^l\|_2 \|\mathbf{h}_{x_j}^l\|_2}.$$

Subsequently, RILKE applies the intervention $\Phi(\mathbf{h}_{\hat{x}}^l; \phi_{\pi(\mathbf{h}_{\hat{x}}^l)}^l)$ to enact the targeted edit, provided that the maximum similarity score exceeds the pre-defined relevance threshold τ_{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 $\mathbf{H} = [\mathbf{h}_{x_1}^l \ \mathbf{h}_{x_2}^l \ \dots \ \mathbf{h}_{x_m}^l]$. We run

hierarchical agglomerative clustering (HAC) over the columns of \mathbf{H} using an initial similarity lower bound τ_{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 $|\mathcal{C}_c| \leq s_{\text{max}}$, 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 $\mathbf{x}_j \in \mathcal{C}_i$ under the objective in Sec. 4.1, yielding $\phi_{\mathcal{C}_i}^l = (\mathbf{R}_{\mathcal{C}_i}^l, \mathbf{A}_{\mathcal{C}_i}^l, \mathbf{b}_{\mathcal{C}_i}^l)$. Let $\kappa(\mathbf{x}_j)$ be the mapping from \mathbf{x}_j to its corresponding cluster \mathcal{C}_i (*i.e.*, $\kappa: \{\mathbf{x}\} \rightarrow \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$, $\kappa(\mathbf{x}_j) = \mathcal{C}_i$ such that $\mathbf{x}_j \in \mathcal{C}_i$). Then, at inference, for query \hat{x} with key $\mathbf{h}_{\hat{x}}^l$, router $\hat{\pi}(\cdot)$ maps it to the corresponding intervention by identifying the cluster to which the closest knowledge item belongs:

$$\hat{\pi}(\mathbf{h}_{\hat{x}}^l) = \kappa \left(\arg \max_{x_j, 1 \leq j \leq m} \frac{\langle \mathbf{h}_{\hat{x}}^l, \mathbf{h}_{x_j}^l \rangle}{\|\mathbf{h}_{\hat{x}}^l\|_2 \|\mathbf{h}_{x_j}^l\|_2} \right).$$

Finally, the corresponding cluster intervention $\Phi(\mathbf{h}_{\hat{x}}^l; \phi_{\hat{\pi}(\mathbf{h}_{\hat{x}}^l)}^l)$ 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 unstructured knowledge control, focusing on the following research questions:

RQ1: How does RILKE compare to prior methods in lifelong unstructured knowledge learning?

RQ2: How does RILKE leverage shared-space interventions for efficient knowledge control?

RQ3: How does the design of 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 EasyEdit¹ 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 embeddings² 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. Finally, we include results on the ZsRE (Levy et al., 2017) to benchmark performance on standard structured editing tasks. All metrics are averaged over the evaluation set.

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 to 0.9 (as detailed in Sec. 4.2). We report edit efficacy and generalization at Steps 10, 100, and 1,000. Furthermore, to assess the retention of general capabilities after extensive editing, we report MMLU accuracy following 1,000 edits, as presented in Tab. 1. Finally, we extend our evaluation to the ZsRE dataset with 3,000 edits to benchmark performance in a structured editing setting.

We find that across both models on the UnKEBench benchmark, RILKE maintains stable per-

formance as edits accumulate. In contrast, competing methods begin to degrade after approximately 10 edits, whereas RILKE consistently outperforms them, exhibiting 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. Although the advantage is narrower than on UnKE—likely due to the reduced difficulty of short-form generation—the margin remains distinct. These results demonstrate that RILKE’s interventions are highly localized and generalize effectively to paraphrased queries, precisely modifying intended knowledge while minimally perturbing other capabilities. Additional results on the EditEverything are presented in App. A.4, and we also report routing stability as the dataset size increases in App. A.7.

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 various memory-based editing methods. Unlike prior methods that typically fine-tune and store

Table 2: Storage costs for UnKE on Llama-3.1-8B-Instruct. 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

entire sub-modules for new knowledge, RILKE operates via a low-rank intervention head in the representation space. This design yields substantial memory savings, minimizing storage costs even when instantiating a distinct adapter for each edit. Specifically, for Llama-3.1-8B-Instruct, 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

¹<https://github.com/zjunlp/EasyEdit>

²We use `all-MiniLM-L6-v2` to align with prior work.

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	<u>0.720</u>	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	<u>0.886</u>	<u>0.878</u>	0.650	0.201	<u>0.909</u>	<u>0.786</u>	0.605	0.172	<u>0.810</u>	<u>0.763</u>	0.521	0.144	<u>0.594</u>	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	<u>0.614</u>	0.672	0.664	<u>0.669</u>	<u>0.598</u>	0.681	0.661	<u>0.673</u>	<u>0.623</u>	0.584	<u>0.62</u>	<u>0.60</u>	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	<u>0.94</u>	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	<u>0.942</u>	<u>0.886</u>	0.712	0.179	<u>0.893</u>	0.264	0.665	0.097	<u>0.901</u>	0.262	<u>0.654</u>	0.098	<u>0.667</u>	0.45	0.00	1.00	0.48
WISE	0.803	0.527	<u>0.794</u>	<u>0.504</u>	0.706	<u>0.550</u>	0.717	<u>0.503</u>	0.564	<u>0.411</u>	0.521	<u>0.401</u>	0.651	<u>0.61</u>	<u>0.58</u>	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	<u>0.86</u>	0.85

additional $\sim 3\times$ compression.

Table 3: UnKE edit success and MMLU after 1,000 sequential edits under Individual vs. Cluster-shared training; 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

Tab. 3 presents the performance of the shared-subspace training strategy. 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. A detailed analysis of the routing behavior in this setting is provided in App. A.7.

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 $\mathcal{L}_{\text{robust}}$ in Eq.(2))

Table 4: Edit efficacy (BertScore) at $T=100$ and $T=1,000$ under training with and without $\mathcal{L}_{\text{robust}}$. 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 $\mathcal{L}_{\text{robust}}$	1.000	0.959	0.999	0.909
w $\mathcal{L}_{\text{robust}}$	1.000	0.984	1.000	0.963

Tab. 4 demonstrates that our robust training framework significantly improves paraphrase generalization without compromising precision on original targets. This confirms that our objective

effectively mitigates overfitting to surface forms, ensuring reliable knowledge control. For further details, we provide a detailed analysis of computational costs in App. A.5 and a comprehensive study on training configurations in App. A.6.

6 Analysis: 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}_{edit} in settings (ii) and (iii) against setting (i). Formally,

we define the *edit vectors* for x_i as $\mathbf{V}_{\text{edit},x_i} = \mathbf{R}^{l\top} (\mathbf{A}^l \mathbf{h}_{x_i}^l + \mathbf{b}^l - \mathbf{R}^l \mathbf{h}_{x_i}^l)$ (i.e., the deviation from the original hidden state; see Eq.(1)). Since the pre-edit hidden state $\mathbf{h}_{x_i}^l$ is fixed across regimes because 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 $\mathbf{V}_{\text{edit},x_i}^{(\text{iii})}$ that are consistently closer to their individually trained counterparts $\mathbf{V}_{\text{edit},x_i}^{(\text{i})}$, 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 $\{x\}$, shown in Fig. 3, further corroborates this trend: edit vectors from (iii) remain close to those from (i), whereas those from (ii) shift away.

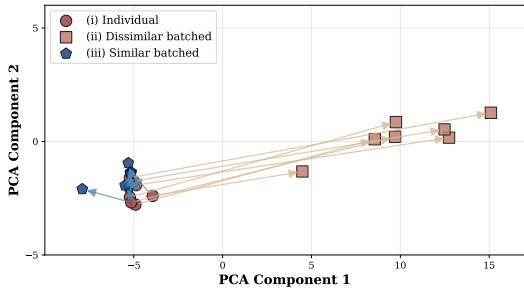


Figure 3: Visualization of \mathbf{V}_{edit} under different training settings. Arrows trace the shift from individual training to batched strategies for a single data point. Training with similar data preserves vector proximity, while dissimilar batching drives them away, highlighting the need to cluster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

7 Related Work

LLM Representation Space Analysis. Previous studies have demonstrated that LLMs encode rich semantics within their activation space (Zou et al., 2023; Marks and Tegmark, 2023; Turner et al., 2023). Core behaviors, including trustworthiness (Marks and Tegmark, 2023), refusal (Arditi et al., 2024), and reasoning (Liu et al., 2025b; Chen et al., 2025b), 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. (2025a) discover persona vectors for customized control; and Wu et al. (2024b) introduce fine-tuning directly within the represen-

tation 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 knowledge editing in 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. First, although we evaluate on the largest unstructured knowledge editing dataset to our knowledge (*i.e.*, UnKE (Deng et al., 2025)), scaling to extra-large and continually expanding knowledge bases remains underexplored. Second, we defer a systematic risk analysis of knowledge control to future work. This includes potential threats such as 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 detection, mitigation, and governance mechanisms to address these risks in future studies.

References

Andy Ardit, 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*.

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 Ardit, Henry Sleight, Owain Evans, and Jack Lindsey. 2025a. [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. 2025b. [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 adapters](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 47934–47959. Curran Associates, Inc.

757	Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023b. Aging with GRACE: lifelong model editing with discrete key-value adaptors . In <i>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</i> .	<i>the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025</i> , pages 4860–4877. Association for Computational Linguistics.	814 815 816 817
765	Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding . In <i>9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021</i> . OpenReview.net.	Aman Madaan, Shuyan Zhou, Uri Alon, Yiming Yang, and Graham Neubig. 2022. Language models of code are few-shot commonsense learners . In <i>Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022</i> , pages 1384–1403. Association for Computational Linguistics.	818 819 820 821 822 823 824 825
771	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 <i>The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022</i> . OpenReview.net.	Samuel Marks and Max Tegmark. 2023. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets . <i>CoRR</i> , abs/2310.06824.	826 827 828 829
777	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 . <i>CoRR</i> , abs/2502.05628.	Samuel Marks and Max Tegmark. 2024. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets .	830 831 832
782	Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pre-training of language models . In <i>The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023</i> . OpenReview.net.	Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in GPT . In <i>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</i> .	833 834 835 836 837 838 839
788	Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension . In <i>Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017</i> , pages 333–342. Association for Computational Linguistics.	Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer . <i>CoRR</i> , abs/2210.07229.	840 841 842 843
795	Gaotang Li, Yuzhong Chen, and Hanghang Tong. 2025. Taming knowledge conflicts in language models . In <i>Forty-second International Conference on Machine Learning</i> .	Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations . In <i>Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies</i> , pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.	844 845 846 847 848 849 850
799	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 <i>The Twelfth International Conference on Learning Representations</i> .	Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022. Fast model editing at scale . In <i>The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022</i> . OpenReview.net.	851 852 853 854 855
804	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 <i>The Thirteenth International Conference on Learning Representations</i> .	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 .	856 857 858 859
811	Xuyuan Liu, Lei Hsiung, Yaoqing Yang, and Yujun Yan. 2025b. Spectral insights into data-oblivious critical layers in large language models . In <i>Findings of</i>	Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models . In <i>Proceedings of the 41st International Conference on Machine Learning, ICML'24</i> . JMLR.org.	860 861 862 863 864
813		Nina Rimskey, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition . In <i>Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1:</i>	865 866 867 868 869

870		Li, Dayiheng Liu, Fei Huang, Guanting Dong, Hao-	926
871		ran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei	927
		Zhang, Jianxin Yang, Jiabin Yang, Jingren Zhou, Jun-	928
872	Alireza Salemi and Hamed Zamani. 2024. Evaluating	yang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin	929
873	retrieval quality in retrieval-augmented generation.	Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin	930
874	In <i>Proceedings of the 47th International ACM SIGIR</i>	Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia,	931
875	<i>Conference on Research and Development in Informa-</i>	Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang	932
876	<i>tion Retrieval, SIGIR 2024, Washington DC, USA,</i>	Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu	933
877	<i>July 14-18, 2024</i> , pages 2395–2400. ACM.	Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan,	934
		and Zekun Wang. 2024a. Qwen2.5 technical report.	935
878	Kai Sun, Yifan Xu, Hanwen Zha, Yue Liu, and Xin Luna	<i>ArXiv</i> , abs/2412.15115.	936
879	Dong. 2024. Head-to-tail: How knowledgeable are		
880	large language models (LLMs)? A.K.A. will LLMs	Wanli Yang, Fei Sun, Xinyu Ma, Xun Liu, Dawei Yin,	937
881	replace knowledge graphs? In <i>Proceedings of the</i>	and Xueqi Cheng. 2024b. The butterfly effect of	938
882	<i>2024 Conference of the North American Chapter of</i>	model editing: Few edits can trigger large language	939
883	<i>the Association for Computational Linguistics: Hu-</i>	models collapse. In <i>Findings of the Association</i>	940
884	<i>man Language Technologies (Volume 1: Long Pa-</i>	<i>for Computational Linguistics, ACL 2024, Bangkok,</i>	941
885	<i>pers)</i> , pages 311–325, Mexico City, Mexico. Associ-	<i>Thailand and virtual meeting, August 11-16, 2024,</i>	942
886	ation for Computational Linguistics.	pages 5419–5437. Association for Computational	943
		Linguistics.	944
887	Lukas Thede, Karsten Roth, Matthias Bethge, Zeynep	Çagatay Yildiz, Nishaanth Kanna Ravichandran, Nitin	945
888	Akata, and Thomas Hartvigsen. 2025. Wikibigedit:	Sharma, Matthias Bethge, and Beyza Ermis. 2025.	946
889	Understanding the limits of lifelong knowledge edit-	Investigating continual pretraining in large language	947
890	ing in LLMs. In <i>Forty-second International Confer-</i>	models: Insights and implications. <i>Trans. Mach.</i>	948
891	<i>ence on Machine Learning.</i>	<i>Learn. Res.</i> , 2025.	949
892	Alexander Matt Turner, Lisa Thiergart, Gavin Leech,	Binchi Zhang, Zhengzhang Chen, Zaiyi Zheng, Jun-	950
893	David Udell, Juan J Vazquez, Ulisse Mini, and	dong Li, and Haifeng Chen. 2025. Resolving editing-	951
894	Monte MacDiarmid. 2023. Steering language mod-	unlearning conflicts: A knowledge codebook frame-	952
895	els with activation engineering. <i>arXiv preprint</i>	work for large language model updating. <i>CoRR,</i>	953
896	<i>arXiv:2308.10248.</i>	abs/2502.00158.	954
897	Ke Wang, Yiming Qin, Nikolaos Dimitriadis, Alessan-	Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong	955
898	dro Favero, and Pascal Frossard. 2025. MEMOIR:	Wu, Jingjing Xu, and Baobao Chang. 2023. Can we	956
899	lifelong model editing with minimal overwrite and	edit factual knowledge by in-context learning? In	957
900	informed retention for llms. <i>CoRR</i> , abs/2506.07899.	<i>Proceedings of the 2023 Conference on Empirical</i>	958
901	Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi	<i>Methods in Natural Language Processing, EMNLP</i>	959
902	Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Hua-	<i>2023, Singapore, December 6-10, 2023</i> , pages 4862–	960
903	jun Chen. 2024. WISE: rethinking the knowledge	4876. Association for Computational Linguistics.	961
904	memory for lifelong model editing of large language		
905	models. In <i>Advances in Neural Information Pro-</i>	Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh	962
906	<i>cessing Systems 38: Annual Conference on Neural</i>	Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar.	963
907	<i>Information Processing Systems 2024, NeurIPS 2024,</i>	2020. Modifying memories in transformer models.	964
908	<i>Vancouver, BC, Canada, December 10 - 15, 2024.</i>	<i>arXiv preprint arXiv:2012.00363.</i>	965
909	Xiaobao Wu, Liangming Pan, William Yang Wang, and	Andy Zou, Long Phan, Sarah Li Chen, James Campbell,	966
910	Anh Tuan Luu. 2024a. AKEW: assessing knowledge	Phillip Guo, Richard Ren, Alexander Pan, Xuwang	967
911	editing in the wild. In <i>Proceedings of the 2024 Con-</i>	Yin, Mantas Mazeika, Ann-Kathrin Dombrowski,	968
912	<i>ference on Empirical Methods in Natural Language</i>	Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan	969
913	<i>Processing, EMNLP 2024, Miami, FL, USA, Novem-</i>	Wang, Alex Mallen, Steven Basart, Sanmi Koyejo,	970
914	<i>ber 12-16, 2024</i> , pages 15118–15133. Association	Dawn Song, Matt Fredrikson, J. Zico Kolter, and	971
915	for Computational Linguistics.	Dan Hendrycks. 2023. Representation engineering:	972
		A top-down approach to AI transparency. <i>CoRR,</i>	973
916	Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus	abs/2310.01405.	974
917	Geiger, Dan Jurafsky, Christopher D. Manning, and		
918	Christopher Potts. 2024b. Reft: Representation fine-		
919	tuning for language models. In <i>Advances in Neural</i>		
920	<i>Information Processing Systems 38: Annual Confer-</i>		
921	<i>ence on Neural Information Processing Systems 2024,</i>		
922	<i>NeurIPS 2024, Vancouver, BC, Canada, December</i>		
923	<i>10 - 15, 2024.</i>		
924	Qwen An Yang, Baosong Yang, Beichen Zhang,		
925	Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan		

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 on 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. Although the hard negatives possess higher lexical

similarity (Mean: 0.61, Median: 0.60) compared to the paraphrases (Mean: 0.51, Median: 0.52), the L2 distance in the hidden space is significantly lower for the paraphrased pairs.

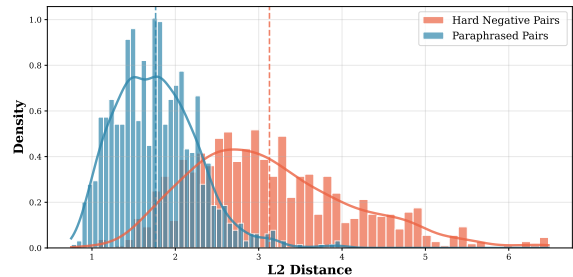


Figure 4: Comparison of L2 distances for Paraphrased pairs and Hard Negative pairs. Hard negatives exhibit high lexical similarity but distinct semantics. The clear separation between the distributions indicates that paraphrases remain significantly closer in the hidden space, validating the Semantic 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 $\mathbf{H} = [\mathbf{h}_{x_j}^l]$ for $x_j \in \mathcal{D}_{edit}$, similarity threshold $\tau_{min} \in (0, 1)$, step $\Delta > 0$, cluster max size s_{max}

- 1: $\mathcal{C}_0 \leftarrow \text{HAC}(\mathbf{H}, \text{thr} = 1 - \tau_{min}), \quad \mathcal{C}_{final} \leftarrow \emptyset$
- 2: **for all** $C \in \mathcal{C}_0$ **do**
- 3: $\mathcal{C}_{final} \leftarrow \mathcal{C}_{final} \cup \text{SPLITOVERSIZED}(C, \tau_{min})$
- 4: **return** \mathcal{C}_{final}
- 5: **procedure** $\text{SPLITOVERSIZED}(C, \tau)$
- 6: **if** $|C| \leq s_{max}$ **then**
- 7: **return** $\{C\}$
- 8: **repeat**
- 9: $\tau \leftarrow \tau + \Delta; \quad \mathcal{S} \leftarrow \text{HAC}(\mathbf{H}[C], \text{thr} = 1 - \tau)$
- 10: $\mathcal{R} \leftarrow \bigcup_{S \in \mathcal{S}} \text{SPLITOVERSIZED}(S, \tau)$
- 11: **until** $\max_{T \in \mathcal{R}} |T| \leq s_{max}$
- 12: **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 Gemini³ to generate paraphrased queries, creating out-of-distribution variants for evaluating generalizability. Results are summarized in Tab. 5.

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

³<https://deepmind.google/models/gemini/flash/>

Table 5: 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
<i>Based on</i> 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	<u>0.865</u>	0.431	<u>0.799</u>	0.378	0.565	0.146	0.547	0.140	0.089	0.071	0.090	0.071
WISE	0.795	<u>0.638</u>	0.718	<u>0.451</u>	<u>0.765</u>	<u>0.557</u>	<u>0.710</u>	<u>0.408</u>	<u>0.781</u>	<u>0.605</u>	<u>0.753</u>	<u>0.489</u>
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. 6 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 6: 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 7:

Table 7: 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.

Variant	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 $\mathbf{h}^{l,i} + \mathbf{R}^{l\top}(\mathbf{A}^l \mathbf{h}^{l,i} + \mathbf{b}^l - \mathbf{R}^l \mathbf{h}^{l,i})$ 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 detail our experimental configuration in Tab. 8. We then investigate key design choices for RILKE, specifically analyzing the layer selection for interventions and the clustering strategy.

Table 8: Training configuration for LLAMA3.1-8B-Instruct on the UnKE dataset.

Config	Value
Rank of Intervention	4
Learning Rate	1×10^{-2}
Intervention Layer	15
Radius ϵ	2×10^{-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 .

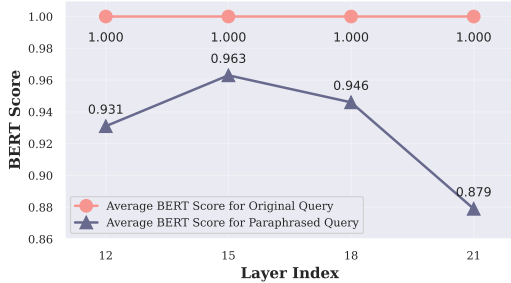


Figure 5: Edit efficacy and generalization of RILKE across layers of Llama-3.1-8B-Instruct (32 layers in total). Performance peaks when intervening in the mid-layers.

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 ϵ

We investigate the impact of the radius ϵ , 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. 9 presents the results. We observe that while performance remains relatively stable across a broad range of ϵ , increasing ϵ 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

Table 9: Ablation results for cluster training with varying ϵ . While increasing ϵ 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

0.995). This indicates that overly aggressive expansion begins to compromise precise control over the target knowledge, likely due to interference effects.

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 τ_{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 τ_{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. 10).

We observe that lowering the in-cluster similarity threshold τ_{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 Scalability of Routing Strategy

A.7.1 Router Behavior on UnKE

We begin by analyzing the behavior of the router within the clustering training setup. As described in Sec. 5.3, we first sequentially train the model on 1,000 knowledge edits (distributed across 306 intervention modules) and then evaluate the router’s behavior using paraphrased versions of these 1,000

Table 10: Performance and storage costs (in MiB, using fp32 precision) of cluster training with similarity thresholds τ_{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

queries alongside unrelated samples 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 Testing 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 set from AnyEdit (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:

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

where $\mathbf{h}_{\hat{x}_i}^l$ is the paraphrased query’s representation for item x_i , and $\pi(\cdot)$ is the router’s mapping from query to intervention module, and γ_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 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.8 Case Study

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 pre-

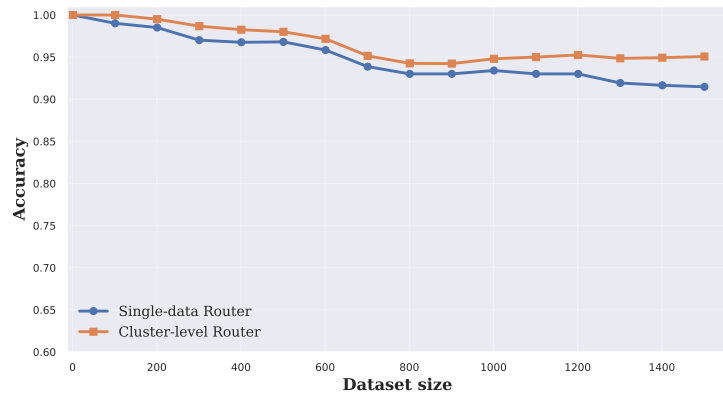


Figure 6: Routing accuracy with increasing dataset size. Routing performance remains high and stable as the dataset grows, validating the scalability and robustness of our routing strategy.

1288 serve coherence across edits. We illustrate this
 1289 phenomenon through a case study.

1292 Here, we can find that locate-then-edit meth-
1293 ods (*e.g.*, UnKE, AnyEdit) inevitably collapse as
1294 edits accumulate, producing nonsensical outputs.
1295 Memory-based approaches like WISE preserve util-
1296 ity and produce coherent responses. Moreover,
1297 they can capture the first few tokens accurately and
1298 remain aligned with the reference early in the se-
1299 quence. However, as the sequence length increases,
1300 performance degrades, exhibiting clear semantic
1301 drift from the target edit. This phenomenon may
1302 stem from limitations in the expressiveness of a
1303 single weight-space memory module. By contrast,
1304 RILKE reliably memorizes target edits, including
1305 those with long-form answers, while maintaining
1306 coherence. These results underscore the advantages
1307 of controlling LLM knowledge in the representa-
1308 tion space.

1309 **A.9 Usage of LLMs**

1310 We used LLMs solely as auxiliary tools to improve
1311 the grammar and clarity of our manuscript. This
1312 assistance was limited to enhancing readability and
1313 presentation; all conceptual contributions, analyses,
1314 and interpretations were solely developed by the
1315 authors.