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Large language models (LLMs) often produce incorrect or outdated content.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 especially hard for complex, unstructured knowledge in a *lifelong* setting, 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 properties enabling RILKE to deliver 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. In evaluation on knowledge editing benchmarks with LLaMA and Qwen models, RILKE are scalable to large-scale datasets, demonstrating high edit success, strong paraphrase generalization, and preserving general utility with modest memory overhead. These results show RILKE is an effective and scalable solution for lifelong knowledge control in LLMs. Code available at the link.

1 Introduction

Large language models (LLMs) excel at knowledge-intensive NLP tasks (Sun et al., 2024; Liu 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 Once deployed, they cannot adapt to evolving real-world information, often producing outdated or inaccurate content.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 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). An alternative, Retrieval-Augmented Generation (RAG), injects new facts at inference time (Chen et al., 2024) but can conflict 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 techniques fall short in lifelong settings, especially when dealing with *unstructured*, *free-form* text that cannot be reduced to simple factual triplets (*i.e.*, Subject–Relation–Object) (Deng et al., 2025). Parametric approaches, which directly modify model weights (Meng et al., 2022a; Fang et al., 2025), suffer from *edit collapse*: as edits accumulate, performance degrades until the model becomes unusable (Yang et al., 2024; Nishi et al., 2025). This degradation is particularly severe for unstructured knowledge. Similarly, external-memory methods, while helpful for accumulating knowledge, often lack the expressiveness to capture the nuances of complex, unstructured information with a single weight-space module (Wang et al., 2024; Zhang et al., 2025).

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 attempting representation-level 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-level editing 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 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 diverse models and benchmarks, RILKE provides a stable, lightweight, and interpretable solution for the continual 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 θ defines a joint probability over $\boldsymbol{x}$ through an auto-regressive factorization:

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

where $p_\theta\left(x_i \mid \boldsymbol{x}_{<i}\right)$ 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 $\mathbf{W}$:

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

To generate sentence $\boldsymbol{x}$, the LLM iteratively computes $p_\theta\left(x_i \mid \boldsymbol{x}_{<i}\right)$, 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.

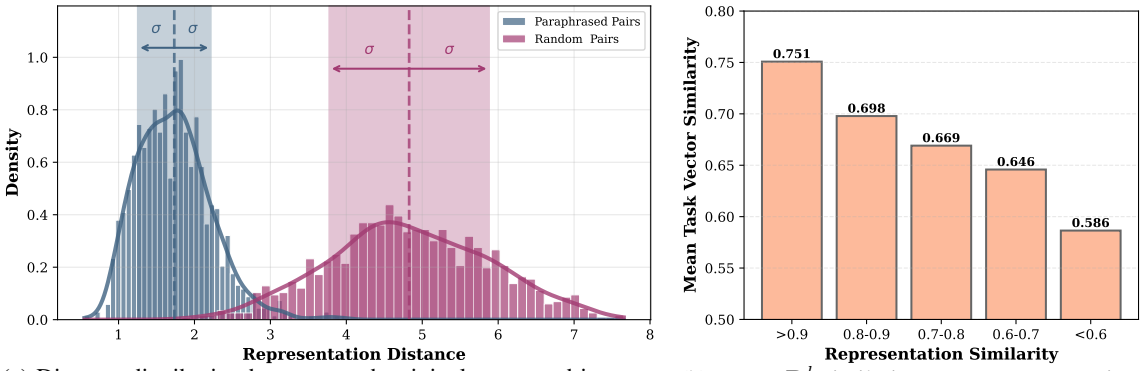
Representation Fine-tuning(ReFT) (Wu et al., 2024b) is a fine-tuning approach for LLMs that, instead of updating model weights, learns an *intervention module* to steer intermediate hidden states so they produce the target outputs, leaving the original weights unchanged. Building on the *linear representation 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 $p_\theta\left(x_i \mid \boldsymbol{x}_{<i}\right) = \operatorname{softmax}\left(\mathbf{W}\mathbf{F}^{>l}(\Phi(\boldsymbol{h}^{l,i}; \phi^l))\right)$. For brevity, we will omit the token and layer indices (i, l) when the context is clear.

3 Motivation: Geometric Properties of LLM Representations

Lifelong knowledge control in LLMs requires incorporating extensive new *unstructured* knowledge while preserving existing capabilities. This poses three core challenges: **(i) unstructured complexity**, where edits must generalize across diverse paraphrases; **(ii) catastrophic forgetting**, where new updates must not interfere with existing knowledge; and **(iii) scalability**, where the system must manage a growing number of edits with low memory overhead. We posit that the key to overcoming these challenges lies not in the weight space, but in the model's hidden representations, whose expressivity enables fine-grained control over long-form,



(a) Distance distribution between each original query and its paraphrased counterpart, with a random set of non-corresponding pairs included for comparison. Paraphrased queries remain close to the originals in representation space, supporting Prop. 1.

(b) Mean $\mathbf{R}^l$-similarity across representation-similarity groups. 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 enable generalizable, lifelong, and scalable knowledge control in LLMs.

unstructured knowledge, while its *geometric properties* guide effective control. We identify and validate two such properties that form the foundation of our method:

Property 1. *For a query $\boldsymbol{x}$ with layer-l representation $\boldsymbol{h}^l_{\boldsymbol{x}}$, its paraphrase $\hat{\boldsymbol{x}}$ maps, in the layer-l representation space, to a representation closer to $\boldsymbol{h}^l_{\boldsymbol{x}}$ than those of semantically unrelated queries $\boldsymbol{x}'$, i.e., $\|\boldsymbol{h}^l_{\hat{\boldsymbol{x}}} - \boldsymbol{h}^l_{\boldsymbol{x}}\|_2 < \|\boldsymbol{h}^l_{\boldsymbol{x}'} - \boldsymbol{h}^l_{\boldsymbol{x}}\|_2$*

Property 2. *Given two semantically related query pairs $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, even when ReFT(Eq.(1)) is trained individually for each, , their learned layer-l intervention subspaces remain aligned: i.e., the projection matrices $\mathbf{R}^l_{\boldsymbol{x}_i}$ and $\mathbf{R}^l_{\boldsymbol{x}_j}$ are highly similar.*

Prop. 1 offers a direct solution to **unstructured complexity** via paraphrase-robust training and addresses **catastrophic forgetting** by routing queries to disentangled knowledge modules. For each edit, we train an item-specific representation intervention and leverage representation geometry to route incoming queries to the appropriate module. To enhance robustness, we supervise over a semantics-stable neighborhood around the hidden state during training, ensuring that lexically varied but semantically equivalent queries activate the same updated knowledge. Prop. 2 sheds light on **scalability** by linking semantics to the learned intervention weights: related knowledge concentrates in a shared low-dimensional subspace. A single module can thus identify this subspace and efficiently store and apply multiple edits.

We validate these properties empirically. As shown in Fig. 1a, we observe a clear separation in the L2-distance distributions between hidden states of pquery pairs and randomly paired queries at a fixed layer(layer 15 of `Llama-3.1-8B-Instruct`), confirming Prop. 1. For Prop. 2, we investigate how representation similarity relates to the alignment of their learned intervention subspaces by comparing layer-l representation similarity with the cosine similarity of their projection matrices $\mathbf{R}^l$ (see Eq.(1)). As seen in Fig. 1b, more similar representations yield more strongly aligned intervention subspaces, suggesting that semantically related knowledge can be controlled via grouped interventions.

4 Method: Towards Generalizable, Scalable, Lifelong Knowledge Control

To address the challenges of *generalizability*, *lifelong learning*, and *scalability* in knowledge editing, we introduce RILKE (**R**epresentation **I**ntervention for **L**ifelong **K**nowl**E**dge Control). RILKE is a framework designed to achieve robust and life-long control over an LLM's knowledge. During training, it isolates updates by assigning a dedicated intervention module to each knowledge item, which prevents catastrophic forgetting. At inference, a query-adaptive router activates the appropriate module, ensuring the intervention is applied only when relevant. To scale this approach, we further introduce a batched training scheme that clusters semantically related edits to share a single intervention module, improving memory efficiency.

4.1 Consistency-Robust Training for Generalizable Knowledge Control

Knowledge-editing methods often suffer from poor *generalizability*: an edit works for a specific query but fails on its paraphrases. Consider an unstruc-

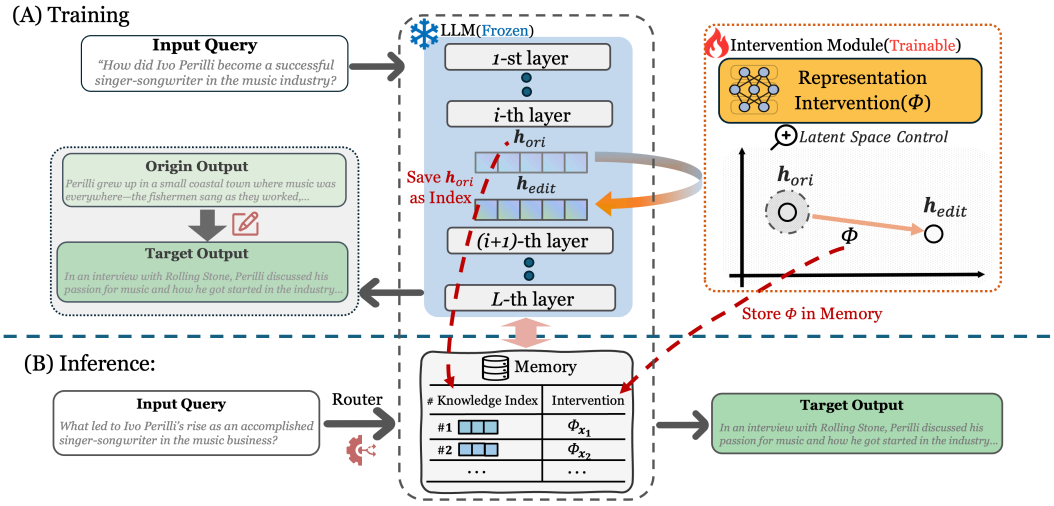


Figure 2: Overview of the RILKE framework. During training, the intervention module Φ maps $\boldsymbol{h}_{\text{ori}}$ to the target $\boldsymbol{h}_{\text{edit}}$, while $\boldsymbol{h}_{\text{ori}}$ is stored as the knowledge index. At inference, the router selects the intervention module Φ whose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

tured knowledge item $(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x}$ is the input query and $\boldsymbol{y}$ the target response. Vanilla ReFT (Wu et al., 2024b), for instance, exhibits this limitation because its training process learns an intervention $\phi^l_{\boldsymbol{x}}$ to be activated only by the exact representation $\boldsymbol{h}^l_{\boldsymbol{x}}$ of the original query $\boldsymbol{x}$. Consequently, a paraphrased query $\hat{\boldsymbol{x}}$, whose representation $\boldsymbol{h}^l_{\hat{\boldsymbol{x}}}$ is nearby but not identical, may fail to trigger the edit.

Recalling Prop. 1 and Eq.(1), the layer-l representation of any paraphrased query $\hat{\boldsymbol{x}}$ lies in an ε-ball around $\boldsymbol{h}^l_{\boldsymbol{x}}$, *i.e.*, $\|\boldsymbol{h}^l_{\hat{\boldsymbol{x}}} - \boldsymbol{h}^l_{\boldsymbol{x}}\|_2 \le \varepsilon$. We therefore impose an equivalent intervention target throughout this neighborhood. Incorporating the deviation into the ReFT map, a first-order expansion yields $\Phi(\boldsymbol{h}^l_{\boldsymbol{x}} + \varepsilon; \phi^l_{\boldsymbol{x}}) = \Phi(\boldsymbol{h}^l_{\boldsymbol{x}}; \phi^l_{\boldsymbol{x}}) + \big(\mathbf{I} + \mathbf{R}^{l\top}_{\boldsymbol{x}}(\mathbf{A}^l_{\boldsymbol{x}} - \mathbf{R}^l_{\boldsymbol{x}})\big)\varepsilon$ so the paraphrase-induced variation is governed by the term $\big(\mathbf{I} + \mathbf{R}^{l\top}_{\boldsymbol{x}}(\mathbf{A}^l_{\boldsymbol{x}} - \mathbf{R}^l_{\boldsymbol{x}})\big)\varepsilon$. 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:

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

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:

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

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

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

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 (Wu et al., 2024b). 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 keep the base model frozen and, for each unstructured knowledge instance $\boldsymbol{x}$, train a dedicated intervention module $\phi^l_{\boldsymbol{x}}$ using the aforementioned method. At inference, however, the model must select and apply the appropriate intervention to ensure that each prompt elicits the relevant updated knowledge, which introduces the additional challenge of accurately routing each query to its correct module.

To achieve precise control during inference, leveraging Prop. 1, we introduce a router that selects the appropriate intervention for incoming queries. We construct an index set $\{\boldsymbol{h}^l_{\boldsymbol{x}_j}\}_{j=1}^m$ for the edit dataset with m samples, using the layer-l representations of each training example on the original model. This is because these representations remain unaffected by the training process of intervention modules, which operate only beyond layer-l. Each index is then linked to its 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 $\pi(\cdot)$ assigns the incoming

query to the indexed representation with minimal l_2 distance and returns its associated intervention:

$$\pi(\boldsymbol{h}^l_{\hat{\boldsymbol{x}}}) = \underset{\boldsymbol{x}_j, 1 \le j \le m}{\arg\min} \left\| \boldsymbol{h}^l_{\hat{\boldsymbol{x}}} - \boldsymbol{h}^l_{\boldsymbol{x}_j} \right\|_2^2 .$$

RILKE then applies $\Phi(\boldsymbol{h}^l_{\hat{\boldsymbol{x}}}; \phi^l_{\pi(\boldsymbol{h}^l_{\hat{\boldsymbol{x}}})})$ to enact the targeted edit during generation, while all other interventions remain inactive.

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 low-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 low-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 $\mathcal{C}_c$, 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 $\hat{\pi}(\cdot)$ map it to the corresponding intervention by identifying the cluster to which the closest knowledge item belongs:

$$\hat{\pi}(\boldsymbol{h}^l_{\hat{\boldsymbol{x}}}) \ = \ \kappa(\underset{\boldsymbol{x}_j, 1 \le j \le m}{\arg\min} \left\| \boldsymbol{h}^l_{\hat{\boldsymbol{x}}} - \boldsymbol{h}^l_{\boldsymbol{x}_j} \right\|_2^2).$$

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 —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 to achieve memory-efficient knowledge control?

RQ3: How does the design of RILKE facilitate precise and generalizable knowledge control?

5.1 Experimental Setup

LLMs & Baselines. We evaluated RILKE framework on popular off-the-shelf LLMs: `Llama-3.1-8B-Instruct` (Grattafiori et al., 2024) and `Qwen2.5-7B-Instruct` (Qwen et al., 2025), and compared it against several 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), LoRA (Hu et al., 2022), WISE (Wang et al., 2024)); (details in App. A.3). For fairness, we adopt the official implementations and configurations from EasyEdit[1] (Wang et al., 2023).

[1] https://github.com/zjunlp/EasyEdit

Datasets & Metrics. To assess knowledge editing efficacy, we evaluate on UnKEBench (Deng et al., 2025) and EditEverything (Jiang et al., 2025) datasets. Following prior work, we report: (i) *Lexical similarity* via Rouge-L, capturing n-gram overlap between generated response and reference answers; and (ii) *Semantic similarity* via cosine similarity of sentence embeddings(BERTScore)[2], measuring alignment at the semantic level. To comprehensively evaluate the edited model, we assess both edit efficacy and generalization by reporting performance on original queries (used during training) and on paraphrased queries not seen during training. To gauge overall post-edit utility, we further evaluate on MMLU (Hendrycks et al., 2021). All metrics are averaged over the evaluation set.

5.2 RQ1: RILKE Enables Lifelong Control

To evaluate lifelong knowledge control, we adopt a sequential editing protocol. Following prior work (Deng et al., 2025; Jiang et al., 2025), we fix the batch size to 1 and the decoding temperature to 0.001. To avoid spurious activations, we gate the router with a representation-similarity threshold of 0.9, applying the selected module only when this threshold is exceeded. We report edit efficacy and generalizability at Steps 1, 10, 100, and 1000. Additionally, to assess the model's retained usability after extensive editing, we report its accuracy on the MMLU benchmark after 1,000 edits, as shown in Tab. 1.

We find that across both models, RILKE maintains stable performance as edits accumulate. In contrast, competing methods begin to degrade after approximately 10 edits, while RILKE consistently outperforms them at every step, with gains that intensify over time. Moreover, unlike prior approaches whose utility degrades substantially after extensive editing, RILKE preserves performance on reasoning over unrelated knowledge, achieving near parity with the unedited base model on MMLU. These results show that RILKE's interventions are highly localized, precisely modifying the intended knowledge while minimally perturbing other capabilities. Additional results on the EditEverything benchmark are presented in App. A.4, and routing stability as dataset size increases is analyzed in App. A.5.

[2]Aligned with prior work, we use `all-MiniLM-L6-v2` for sentence embeddings.

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

We then apply RILKE for shared-subspace management, following the procedure outlined in Sec. 4.3. Specifically, we first cluster the dataset based on layer-l hidden states, using a similarity threshold of $\tau_{\text{sim}} = 0.9$, consistent with the criterion for identifying unrelated knowledge in Sec. 5.2. A single shared intervention is then trained for each cluster, with all instances within a cluster processed jointly in a single batch.

We summarize the trainable parameters of each method in Tab. 2. Prior approaches typically fine-tune sub-modules of the base model to encode new knowledge. In contrast, RILKE operates in representation space with a low-rank linear intervention head (4×4096 in our setup). This design yields substantial memory saving even when allocating an individual module per edit. On `Llama-3.1-8B-Instruct`, RILKE trains fewer than 20% of the parameters required by competing baselines. Furthermore, the cluster-shared method reduces memory usage to $\approx 30\%$ of vanilla RILKE, achieving an additional $\sim 3\times$ compression while preserving both editing fidelity and overall utility.

Table 2: Number of Parameters used across methods when editing `Llama-3.1-8B-Instruct`. RILKE requires significantly fewer trainable parameters in both settings, highlighting its memory efficiency.

Method	#Params
AnyEdit	293.6M
Unke	218.1M
MEMIT	293.6M
WISE	180.4M
RILKE (Individual)	32.7M
RILKE (Shared)	10.0M

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. Bert	Para. Bert	MMLU
RILKE (Individual)	1.000	0.963	0.622
RILKE (Shared)	0.999	0.901	0.621

As shown in Tab. 3, shared-subspace training incurs a modest reduction in generalization while preserving edit efficacy and overall utility, while significantly reducing the trainable parameter budget. This demonstrates the effectiveness of our approach. Further analysis of the cluster-based training strategy is provided in Sec. 6 and App. A.6.

5.4 RQ3: RILKE Enables Precise and Generalizable Knowledge Control

We further analyze how our design enables RILKE to deliver precise, paraphrase-generalizable knowl-

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; *Util.* reports accuracy on MMLU. Higher is better for all metrics. The best result in each column is **bold**; the second best is underlined.

Method	$T = 1$				$T = 10$				$T = 100$				$T = 1{,}000$				
	Ori.		Para.		Ori.		Para.		Ori.		Para.		Ori.		Para.		Util.
	Bert	RougeL	Bert	RougeL	Bert	RougeL	Bert	RougeL	Bert	RougeL	Bert	RougeL	Bert	RougeL	Bert	RougeL	MMLU
Based on LLAMA3.1-8B-INSTRUCT																	0.633
FT-L	0.372	0.159	0.423	0.126	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
LoRA	0.881	0.851	0.846	0.853	0.810	0.514	0.806	0.505	0.750	0.495	0.749	0.493	0.655	0.394	0.657	0.390	0.341
MEMIT	0.894	0.968	0.667	0.460	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
UnKE	0.993	0.984	0.934	0.810	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
AnyEdit	0.854	0.984	0.518	0.365	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
WISE	0.904	0.666	0.857	0.600	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
RILKE	**1.000**	**1.000**	**1.000**	**1.000**	**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**
Based on QWEN2.5-7B-INSTRUCT																	0.713
UnKE	**1.000**	**1.000**	0.859	0.444	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
AnyEdit	0.998	0.984	0.550	0.270	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
RILKE	**1.000**	**1.000**	**1.000**	**1.000**	**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**

edge control. We first show that the robust training objective improves generalization to paraphrased queries. Compared to the vanilla objective that uses only the language-modeling loss(*i.e.*, without $\mathcal{L}_{\text{robu}}$ in Eq.(2)), the robust training objective yields substantial improvements on paraphrased queries (Tab. 4). This highlights the necessity of our robust training framework, which mitigates overfitting to surface forms and enables reliable generalization to paraphrased inputs while preserving precision on the original targets.

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. Bert	Para. Bert	Ori. Bert	Para. Bert
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

We next 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.

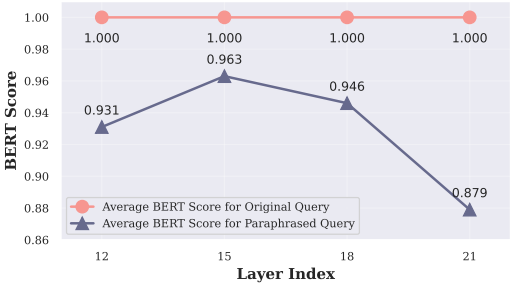


Figure 3: Edit efficacy and generalization of RILKE across layers of `Llama-3.1-8B-Instruct` (32 layers in total). Performance peaks when intervening in the mid-layers.

The results, shown in Fig. 3, 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.

6 Analysis: Can Shared Subspace really control Similar Knowledge

Building on Prop. 2, which posits that semantically similar edits inhabit a shared low-dimensional intervention subspace, we examine how joint training couples individual edits. We randomly sample knowledge item 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 co-clustered items—clusters formed from the full dataset—and trained using a cluster-shared adapter to simulate training with similar knowledge. Importantly, in setting (iii), co-cluster batches may include items from the same cluster that were not part of the original training sample; these items are used solely to construct cluster-level training batches. Evaluation is restricted to the originally sampled items, ensuring a consistent comparison set. This design enables a clean comparison of the effects of joint training with dissimilar items (ii) versus highly similar items (iii).

We compare, for each item, the distances of the *edit vector* $\mathbf{V}_{\text{edit}}$ in settings (ii)/(iii) relative to (i). Concretely, recalling Eq.(1), we refer to the intervention applied in the target subspace as *edited vector*, say $\mathbf{V}_{\text{edit},\boldsymbol{x}_i} = \mathbf{R}^{l\top}(\mathbf{A}^l \boldsymbol{h}^l_{\boldsymbol{x}_i} + \mathbf{b}^l - \mathbf{R}^l \boldsymbol{h}^l_{\boldsymbol{x}_i})$ for $\boldsymbol{x}_i$. Since $\boldsymbol{h}^l_{\boldsymbol{x}_i}$ is fixed across regimes for $\boldsymbol{x}_i$, differences in the resulting edit vector reflect only the learned intervention in (ii)/(iii) relative to (i), isolating the effect of the training strategy.

In this setup, we find 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)) in 91 of 100 sampled cases. A PCA visualization of the edited vectors $\{\mathbf{V}_{\text{edit},\boldsymbol{x}_i}\}$ in Fig. 4 also illustrates this trend: vectors from (iii) remain close to those from (i), whereas (ii) shifts away.

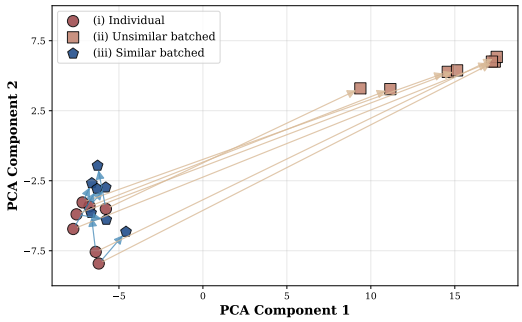


Figure 4: Visualization of $\mathbf{V}_{\text{edit}}$ under different training settings. Training with similar data keeps the edited vectors close to the individual training, whereas dissimilar batching drives them away, underscoring the necessity of clustering similar knowledge for effective shared-subspace 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 Representations Space Analysis. Prior studies has 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) manipulates generation style via style vectors; Chen et al. (2025a) discovers persona vectors for customized control; and Wu et al. (2024b) introduces fine-tuning directly within the representation space to enable style modulation.

Knowledge Edit. Existing model editing methods either modify a small subset of parameters or incorporate external memory while keeping the original weights intact. Parametric approaches directly update model weights to encode new knowledge. 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 approaches preserve the original weights by introducing additional components: Hartvigsen et al. (2023); Wang et al. (2025); Cheng et al. (2025) retrieve discrete codebook entries to overwrite intermediate activations for localized edits, while Zhang et al. (2025) enables simultaneous editing and unlearning via shared memory modules.

Unstructured Knowledge Edit. 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 chunk-based autoregressive editing to enable edit long knowledge. While these methods address the challenges of unstructured knowledge, they face scalability issues: performance often 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 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*.

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. 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. 2023. 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.

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.

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, Vienna, Austria, July 27 - August 1, 2025*, pages 4860–4877. 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.

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.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi 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, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

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*.

Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, et al. 2023. Easyedit: An easy-to-use knowledge editing framework for large language models. *arXiv preprint arXiv:2308.07269*.

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*.

Wanli Yang, Fei Sun, Xinyu Ma, Xun Liu, Dawei Yin, and Xueqi Cheng. 2024. The butterfly effect of model editing: Few edits can trigger large language models collapse. In *Findings of the Association for Computational Linguistics, ACL 2024, 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.

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 Case Study: Catastrophic Forgetting under Sequential Edits

We illustrate the phenomenon of *catastrophic forgetting* in weight-space methods by sequentially editing 100 knowledge items (10 per batch) on `Llama-3.1-8B-Instruct`, using several different weight-space approaches. After the end of the sequence, we evaluate knowledge retention on all previously edited items. Result shown at Fig. 5

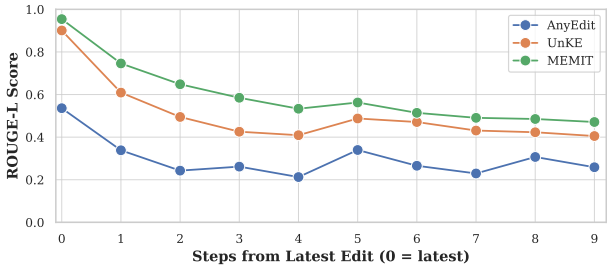


Figure 5: Knowledge Retention Rate(measured via Rouge-L) After Edits in Different Steps. We sequentially edit 100 knowledge items in batches of 10 and evaluate Knowledge Retention at each step after the final edit. All existing methods exhibit substantial forgetting of earlier edits as new knowledge accumulates.

Across methods, we observe pronounced forgetting with edit increasing: while the most recent edits remain strong, performance on earlier edits drops sharply as later updates are applied. This indicates that, although weight-space approaches can handle isolated edits, they fail in lifelong settings as accumulated updates overwrite prior knowledge. This observation reaffirms *edit collapse* and highlights the need to disentangle distinct pieces of knowledge during editing to reduce cross-edit interference.

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.

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 $s_{\max}$

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

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

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 Result 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[3] to generate paraphrased queries, creating out-of-distribution variants for evaluating generalizability. Results are summarized in Tab. 5.

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.**	
	Bert	**RougeL**	**Bert**	**RougeL**	**Bert**	**RougeL**	**Bert**	**RougeL**	**Bert**	**RougeL**	**Bert**	**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**

We can find RILKE keeps 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 Scalability of Route Strategy

We investigate the router's performance under increasing edited knowledge sizes, specifically examining its ability to accurately route each paraphrased query to the appropriate intervention module at *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

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

the fraction of items dispatched to their target module:

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

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 γ_j denotes the ground truth target module recorded at training stage. We present our result on `LLaMA3.1-8B-Instruct` in Fig.6:

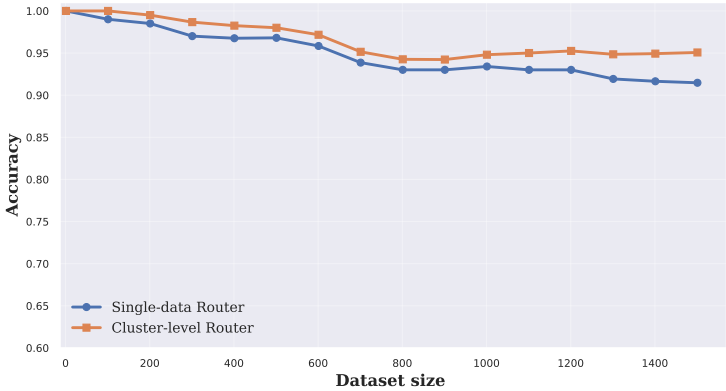


Figure 6: Routing accuracy with increasing dataset size. Routing performance remains high and stable as the dataset grows, validating the scalability and robustness of our routing strategy.

Across dataset scales, $\mathrm{Acc}_{\mathrm{route}}(m)$ 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.6 Ablation Study on Cluster Thresholds τ_{sim}

We further analyze how the clustering threshold affects RILKE's performance and memory efficiency. Using the same procedure as Alg. A.2, we sweep the similarity low-bound τ_{sim} from 0.80 to 0.95 and report edit efficacy, paraphrase generalization, and memory overhead on UnKE dataset using `LLaMA3.1-8B-Instruct`(See Tab.6).

Table 6: Results 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. Bert	Para. Bert	#Cluster	Memory Cost
τ_{sim}=0.80	0.998	0.883	252	8.2M
τ_{sim}=0.85	0.999	0.896	266	8.7M
τ_{sim}=0.90	0.999	0.901	306	10.0M
τ_{sim}=0.95	0.999	0.918	573	18.7M

In this result, we find that lowering the in-cluster similarity threshold τ_{sim} preserves the efficacy of the edit on the original training queries but degrades the performance on paraphrased queries. This reveals a precision–efficiency trade-off: high intra-cluster similarity (tighter clusters) improves precision and paraphrase robustness at the cost of more clusters and higher memory overhead, while looser clusters gain efficiency but compromise generalization.

A.7 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 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 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 otheredyedyedyedyedyedyedyedyedyedyedyedyedyedyedy

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 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—underscoring the advantages of controlling LLM knowledge in the representation space.

A.8 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.