RESIDUALKV: DYNAMIC KNOWLEDGE EDITING VIA AN EDITABLE ATTENTION KEY-VALUE SPACE

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge editing aims to revise specific facts in large language models (LLMs) without costly full-model retraining. Existing editors largely follow a *static editing paradigm*, performing localized factual updates in feed-forward network (FFN) layers. While effective, this fixed editing substrate can limit factual writing and generalization across query reformulations. To overcome this limitation, we introduce **Residual Key-Value Editing (ResidualKV)**, a *dynamic knowledge-editing framework* that rethinks factual writing and access through attention. ResidualKV stores corrections in a persistent editable key-value (KV) space and dynamically invokes them through attention during inference, enabling stronger factual writing and improved generalization across query reformulations. Further, to reduce interference from editable-KV writing, we propose **Factual Residual Distillation (FRD)**, which distills only the required old-to-new factual transition into the editable KV space rather than reconstructing the complete factual state. By avoiding redundant encoding, FRD uses the limited KV capacity more efficiently, reducing disruption to pretrained model behavior while strengthening factual writing. Experiments across three recent LLM backbones and two benchmarks demonstrate strong editing efficacy and generalization, competitive post-edit fluency, and robust large-scale sequential editing. Further ablation and mechanistic analyses support the proposed design. Code for ResidualKV is available at `https://anonymous.4open.science/r/ResidualKV-3CD4`.

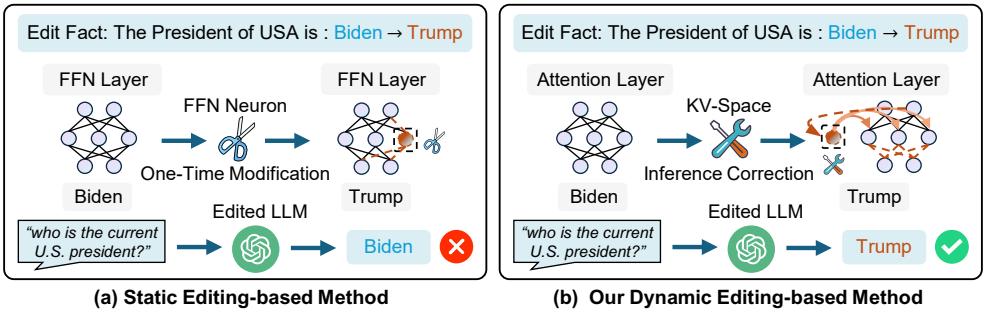
1 INTRODUCTION

Knowledge editing aims to correct specific knowledge in large language models (LLMs) through localized modifications while preserving their broader capabilities. It provides a practical alternative to full-model retraining for updating outdated facts and mitigating knowledge-driven hallucinations (Yang et al., 2026). Conceptually, each edit can be viewed as a *knowledge residual*, i.e., a factual correction that shifts the model's original knowledge toward the desired update. Existing approaches realize such corrections through parameter updates (Fang et al., 2025), trainable adapters (Fei et al., 2026), or external contexts (Zheng et al., 2023). Among these, model-internal editing has largely focused on feed-forward network (FFN) layers, motivated by the view that FFNs store factual knowledge and that updates can be written into their parameters (Meng et al., 2022).

However, FFN-centered edits often generalize poorly from the original query to semantically equivalent paraphrases (Ma et al., 2024; Yan et al., 2025). These methods write the new fact once into FFN parameters as a fixed correction, while its effect at inference depends on the representation induced by the input query. Although semantically equivalent, reformulations can induce representational variations and therefore trigger the fixed correction less reliably. As illustrated in Figure 1(a), we characterize this paradigm of one-time parameter rewriting as ***static editing***. Recent methods improve robustness by stabilizing edit-related representations or controlling how edited knowledge is elicited (Yan et al., 2025; Liu et al., 2026), yet still rely on an FFN-centered editing substrate.

To move beyond the static FFN-centered paradigm, we introduce **Residual Key-Value Editing (ResidualKV)**, a ***dynamic knowledge-editing*** framework that rethinks factual writing and access through attention. Attention primarily supports contextual information retrieval and propagation, but has been less explored as a direct substrate for factual editing (Geva et al., 2021; Meng et al., 2022). ResidualKV takes a different perspective: while keeping pretrained attention parameters

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Static Editing-based Method **(b) Our Dynamic Editing-based Method**

Figure 1: Static versus dynamic knowledge editing. (a) FFN-centered methods write factual corrections once into fixed parameters. (b) ResidualKV stores corrections in persistent editable KVs and dynamically invokes them through attention at inference.

fixed, it expands the attention space with persistent editable KVs dedicated to factual corrections, as illustrated in Figure 1(b). Unlike static editing, where a correction is written once into fixed parameters, ResidualKV represents the correction with editable KVs. At inference, its corrective effect is dynamically activated through attention in response to the current query. Specifically, fact-relevant tokens attend to the editable keys and retrieve the corresponding corrective values, allowing semantically equivalent reformulations to access the same factual correction. We next compare ResidualKV with three alternative editing architectures—FFN-LoRA (Li et al., 2025a), FFN expansion (Wan et al., 2026), and Attention-LoRA—under a single-batch, structure-only setting on Phi-4-mini-3.8B model (Abouelenin et al., 2025) on the CounterFact dataset (Meng et al., 2022). As shown in Figure 2, ResidualKV achieves higher Efficacy and Generalization, indicating the potential of KV-space editing for factual correction in LLMs.

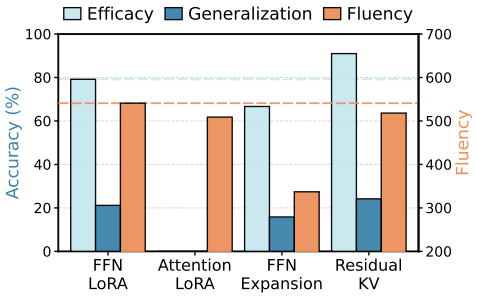


Figure 2: Comparison of editing substrates under a single-batch setting on Phi-4-mini-3.8B model with CounterFact dataset.

However, stronger KV intervention can also disrupt pretrained model behavior, as reflected by the degraded post-edit Fluency in Figure 2. Existing static editing methods often mitigate editing-induced interference by localizing or constraining parameter updates (Fang et al., 2025). For ResidualKV, simply weakening the KV intervention would also weaken the dynamic correction that provides its editing strength. We therefore retain the intervention strength while constraining the predictive change it carries. Specifically, we propose **Factual Residual Distillation (FRD)**, which distills only the required old-to-new factual transition into the editable KV space. FRD constructs matched old- and new-fact contexts with the same query, induces their predictive states from the frozen model, and takes the difference as the desired *factual residual*. It then aligns the predictive change produced by the editable KVs with this desired factual residual, allowing the editable KVs to learn the factual transition rather than reconstruct the complete new-fact predictive state. By excluding redundant representations shared between the old and new facts, FRD focuses the limited KV capacity on the required factual change. This reduces the information encoded per edit, limiting disruption to pretrained model behavior and preserving more KV capacity for stronger factual writing. Finally, we organize the learned KV states into an adapter-style modular bank (Fei et al., 2026) and selectively invoke them with a lightweight router, enabling detachable, reversible, and sequential editing.

We evaluate ResidualKV across three LLM backbones and two knowledge-editing benchmarks. It achieves strong Efficacy and Generalization, competitive post-edit Fluency and general language capability, and robust performance across edit-batch sizes and large-scale sequential editing. Ablations and mechanistic analyses further support the effectiveness of our design. Overall, our results support a new perspective on knowledge editing: shifting from *static FFN-based rewriting* to *dynamic correction through an editable KV space*, while factual residual learning further strengthens factual writing and better preserves pretrained behavior.

2 Related Work

Parameter-based editing. Parameter-based methods directly write factual updates into pretrained parameters, typically targeting FFN layers due to their established role in factual knowledge storage. ROME (Meng et al., 2022) performs a rank-one update to selected FFN weights, while MEMIT (Meng et al., 2023) extends this formulation across multiple layers for scalable editing. Recent methods further improve preservation and sequential editing: AlphaEdit (Fang et al., 2025) reduces interference through null-space projection, NSE (Jiang et al., 2025) selectively updates knowledge-relevant neurons, and UltraEdit (Gu et al., 2026) improves the efficiency and scalability of lifelong editing. Despite these advances, factual corrections are still written once into the pretrained parameter space, leaving their subsequent effects sensitive to the representations elicited by incoming queries.

Adapter-based editing. Adapter-based methods preserve pretrained parameters while introducing auxiliary capacity for factual updates. MELO (Yu et al., 2024) and ELDER (Li et al., 2025a) organize edits in routed LoRA modules, while WISE (Wang et al., 2024a) introduces a dedicated side memory and NeuralDB (Fei et al., 2026) stores editing residuals in a gated neural database. A^{3}E (Piao et al., 2026) further coordinates edited and pretrained knowledge, whereas T-Patcher (Huang et al., 2023) and MeG (Wan et al., 2026) expand editing capacity through additional or generated neurons. These methods extend editing capacity beyond direct parameter rewriting, yet factual corrections remain largely tied to FFN-centered modules or auxiliary states. ResidualKV instead moves the editing substrate into the attention KV space, where persistent editable KVs support dynamic invocation at inference.

Context-based editing. Context-based methods condition generation on external knowledge without modifying pretrained parameters. IKE (Zheng et al., 2023) performs editing through in-context demonstrations, while DecKER (Wang et al., 2025a) and EAIR (Lee et al., 2026) improve knowledge injection through decoupled reasoning and retrieval-based routing. To reduce the overhead and noise of explicit contexts, recent methods further encode external knowledge into latent KVs: InComeS (Li et al., 2025b) extracts gist-based KV representations from editing contexts, while KBLaM (Wang et al., 2025b) maps structured knowledge into continuous KVs. These methods primarily use KVs to encode external knowledge for inference-time conditioning. In contrast, ResidualKV directly optimizes persistent editable KVs as model-internal editing states for factual correction, providing an alternative to FFN-centered factual writing.

3 Preliminaries

Knowledge editing. Given a pretrained language model $\mathcal{M}^{(0)}$ and an edit request $e = (x, y_{\text{old}} \to y_{\text{new}})$, knowledge editing applies a localized intervention to obtain an edited model $\mathcal{M}^{(1)}$ that incorporates the desired factual update while preserving unrelated behavior. A successful edit should produce y_{new} for the editing query x, generalize to semantically equivalent formulations x_{rep}, and preserve the original predictions on neighboring or unrelated queries x_{loc}. These properties correspond to *Efficacy*, *Generalization*, and *Specificity*, respectively:

$$\underbrace{\mathcal{M}^{(1)}(x) \to y_{\text{new}}}_{\text{Efficacy}}, \qquad \underbrace{\mathcal{M}^{(1)}(x_{\text{rep}}) \to y_{\text{new}}}_{\text{Generalization}}, \qquad \underbrace{\mathcal{M}^{(1)}(x_{\text{loc}}) \approx \mathcal{M}^{(0)}(x_{\text{loc}})}_{\text{Specificity}}. \tag{1}$$

Sequential knowledge editing. In practical settings, edits may arrive sequentially in batches. Let $\mathcal{E}_t$ denote the edit batch at step t, and $\mathcal{S} = (\mathcal{E}_1, \ldots, \mathcal{E}_T)$ the complete editing sequence. Sequential editing proceeds as

$$\mathcal{M}^{(0)} \xrightarrow{\mathcal{E}_1} \mathcal{M}^{(1)} \xrightarrow{\mathcal{E}_2} \cdots \xrightarrow{\mathcal{E}_T} \mathcal{M}^{(T)}. \tag{2}$$

After step t, $\mathcal{M}^{(t)}$ should continue to satisfy the objectives in Equation 1 for all accumulated edits in $\mathcal{S}_{\le t} = \bigcup_{u=1}^{t} \mathcal{E}_u$, while preserving previously edited knowledge and minimizing changes to unrelated behavior.

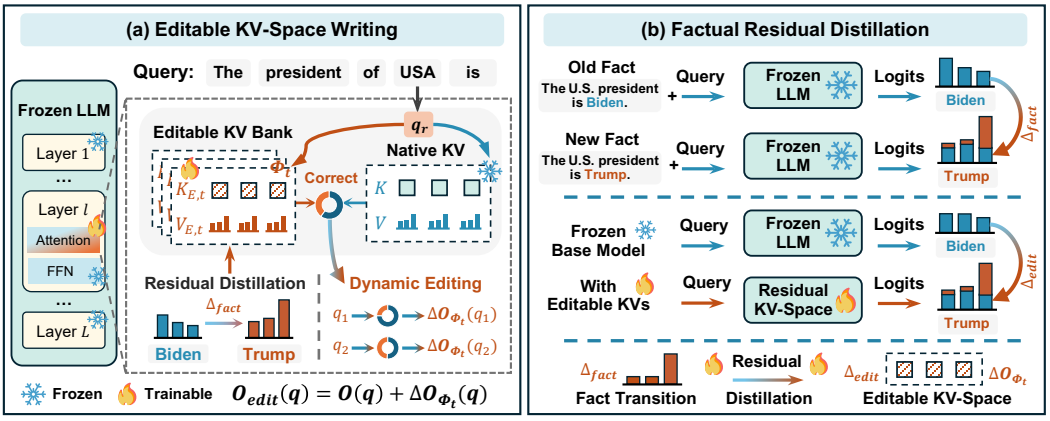


Figure 3: Overview of ResidualKV. (a) Each edit batch is encoded into a persistent editable-KV state that produces dynamic residual corrections through attention. (b) FRD aligns the KV-induced predictive change with the required old-to-new factual transition, improving KV capacity utilization.

4 Methodology

We present ResidualKV, a dynamic knowledge-editing framework that stores factual corrections in a persistent editable KV space while keeping pretrained parameters fixed, as illustrated in Figure 3(a). Each edit batch is encoded into a compact editable-KV state within this space during optimization, which is dynamically invoked through attention at inference. As shown in Figure 3(b), Factual Residual Distillation further constrains the KV-induced predictive change to match the old-to-new factual transition rather than the complete target predictive state, reducing interference with pretrained model behavior. The resulting KV states are organized into a modular bank and selectively invoked by a lightweight router, enabling reversible and sequential editing. The complete procedure is summarized in Appendix B.

4.1 Dynamic Knowledge Editing with Editable KVs

Editable KV State. Given an edit batch $\mathcal{E}_t$, ResidualKV represents its factual corrections with an independent editable-KV state

$$\Phi_t = \left\{ K_{E,t}^{(\ell)}, V_{E,t}^{(\ell)} \right\}_{\ell \in \mathcal{L}}, \tag{3}$$

where $\mathcal{L}$ denotes the selected attention layers. The model remains frozen, so factual updates are stored in persistent model-internal KV states rather than absorbed into pretrained parameters.

For clarity, we consider an attention head at an edited layer and omit the layer and head indices. Following the standard scaled dot-product attention formulation (Vaswani et al., 2017), let Q, K, and V denote the native attention representations. Using a prefix-style parameterization (Li & Liang, 2021), ResidualKV augments the native KV space as

$$\widetilde{K}_t = [K_{E,t}; K], \qquad \widetilde{V}_t = [V_{E,t}; V], \qquad O_{\text{edit},t} = \text{Attn}\left(Q, \widetilde{K}_t, \widetilde{V}_t\right). \tag{4}$$

As illustrated in Figure 3(a), $K_{E,t}$ provides access to the stored correction, while $V_{E,t}$ carries the corresponding corrective signal. At inference, the query vectors in Q attend to the editable keys and determine how strongly their associated values contribute to the attention output. Thus, the editing state remains persistent, while its intervention is dynamically determined by the current query.

Editable-KV Attention Residual. We next characterize how the editable KVs modify the native attention computation. For the query vector q_r at token position r, let N and M denote the numbers of native and editable KV entries, respectively. Their softmax normalization terms are

$$Z^r = \sum_{n=1}^{N} e^{q_r^\top k_n/\sqrt{d}}, \qquad Z_{E,t}^r = \sum_{m=1}^{M} e^{q_r^\top k_{E,t,m}/\sqrt{d}}, \tag{5}$$

where d denotes the attention head dimension. Let O^r and $O^r_{E,t}$ denote the normalized attention outputs over the native and editable KV sets, respectively. Since the two sets are jointly normalized by the same attention softmax, the edited output can be exactly decomposed as

$$\begin{aligned} O^r_{\text{edit},t} &= (1-\rho_{t,r})O^r + \rho_{t,r}O^r_{E,t} \\ &= O^r + \underbrace{\rho_{t,r}\left(O^r_{E,t} - O^r\right)}_{\Delta O^r_{\Phi_t}}, \qquad \rho_{t,r} = \frac{Z^r_{E,t}}{Z^r + Z^r_{E,t}}. \end{aligned} \tag{6}$$

Here, $\Delta O^r_{\Phi_t}$ denotes the attention residual introduced by the editable state Φ_t, while $\rho_{t,r}$ measures the relative attention mass assigned to the editable KV set. Because both $\rho_{t,r}$ and $O^r_{E,t}$ depend on the current query, the same persistent state can produce different residual corrections across token positions and query formulations. This query-dependent residual behavior constitutes the dynamic editing mechanism of ResidualKV.

4.2 Factual Residual Distillation

Editable KVs provide strong factual writing, but their intervention can introduce representational shifts. Meanwhile, target-only optimization specifies the desired answer without constraining broader predictive changes, potentially causing additional disruption to pretrained model behavior (Mitchell et al., 2022). Predictive distillation offers richer distribution-level guidance for internalizing knowledge updates (Padmanabhan et al., 2023). However, matching the complete new-fact predictive state also transfers predictive components represented by the pretrained model, introducing redundancy into the limited editable space. We therefore propose Factual Residual Distillation (FRD) (Figure 3(b)), which aligns the KV-induced predictive change solely with the required old-to-new factual transition, improving utilization of the editable KV space.

Factual Transition Induction. Given an edit $e = (x, y_{\text{old}} \to y_{\text{new}})$, we construct an old-fact context c_{old} from the model's pre-edit belief and a new-fact context c_{new} containing the desired update. We prepend each factual context to the same query x and, using the frozen model $\mathcal{M}_\theta$ with the same target prefix $y_{<i}$ of y_{new}, induce the temperature-scaled full-vocabulary predictive states

$$\begin{aligned} \mathbf{s}^i_{\text{old}} &= \log p^\tau_\theta\left(\cdot \mid [c_{\text{old}}; x], y_{<i}\right), \\ \mathbf{s}^i_{\text{new}} &= \log p^\tau_\theta\left(\cdot \mid [c_{\text{new}}; x], y_{<i}\right), \end{aligned} \qquad \Delta^i_{\text{fact}} = \mathbf{s}^i_{\text{new}} - \mathbf{s}^i_{\text{old}}, \tag{7}$$

where $\mathbf{s}^i \in \mathbb{R}^{|\mathcal{V}|}$ denotes a full-vocabulary log-probability vector, $[\cdot;\cdot]$ denotes sequence concatenation, and τ denotes the distillation temperature. Because both branches share the model, query, and target prefix and differ only in factual conditioning, their difference suppresses shared predictive components and isolates the change induced by factual substitution. We use Δ^i_{fact} as the reference residual specifying the desired old-to-new factual transition.

Edit-Induced Predictive Residual. The editable-KV residual $\Delta O^r_{\Phi_t}$ ultimately induces a predictive change in the output distribution. We therefore measure whether this change matches the desired factual transition. Given the same target prefix, we define

$$\begin{aligned} \mathbf{s}^i_{\text{base}} &= \log p^\tau_\theta\left(\cdot \mid x, y_{<i}\right), \\ \mathbf{s}^i_{\text{edit}}(\Phi_t) &= \log p^\tau_{\theta,\Phi_t}\left(\cdot \mid x, y_{<i}\right), \end{aligned} \qquad \Delta^i_{\text{edit}}(\Phi_t) = \mathbf{s}^i_{\text{edit}}(\Phi_t) - \mathbf{s}^i_{\text{base}}. \tag{8}$$

Here, $\Delta^i_{\text{edit}}(\Phi_t)$ measures the full-vocabulary predictive change produced by the editable KVs. FRD aligns this change with the reference factual residual:

$$\underbrace{\mathbf{s}^i_{\text{edit}}(\Phi_t) - \mathbf{s}^i_{\text{base}}}_{\Delta^i_{\text{edit}}(\Phi_t)} \approx \underbrace{\mathbf{s}^i_{\text{new}} - \mathbf{s}^i_{\text{old}}}_{\Delta^i_{\text{fact}}}. \tag{9}$$

This alignment directly guides the predictive change introduced by the editable KVs rather than requiring them to reconstruct the complete new-fact predictive state.

Training Objective. We optimize the residual alignment over the full vocabulary using

$$\mathcal{L}_{\text{FRD}}(\Phi_t) = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \text{Huber}\left(\Delta^i_{\text{edit}}(\Phi_t), \text{sg}\left(\Delta^i_{\text{fact}}\right)\right), \tag{10}$$

where $\mathcal{I}$ denotes the set of supervised answer positions, $\mathrm{sg}(\cdot)$ denotes stop-gradient, and the Huber loss (Huber, 1964) is applied element-wise and averaged over the vocabulary. While $\mathcal{L}_{\mathrm{FRD}}$ constrains the desired predictive transition, it does not by itself ensure successful prediction of the updated fact. We therefore complement it with the standard autoregressive objective

$$\mathcal{L}_{\mathrm{CE}}(\Phi_t) = -\frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \log p_{\theta,\Phi_t}\left(y_i \mid x, y_{<i}\right). \tag{11}$$

Here, $\mathcal{L}_{\mathrm{CE}}$ anchors the target prediction, while $\mathcal{L}_{\mathrm{FRD}}$ regulates the full-vocabulary change required to achieve it. The editable state for edit batch $\mathcal{E}_t$ is jointly optimized as

$$\Phi_t^* = \arg\min_{\Phi_t} \frac{1}{|\mathcal{E}_t|} \sum_{e \in \mathcal{E}_t} \left[\mathcal{L}_{\mathrm{CE}}(e; \Phi_t) + \mathcal{L}_{\mathrm{FRD}}(e; \Phi_t)\right]. \tag{12}$$

During optimization, we additionally apply a shared slot-wise stochastic mask across the selected editable-KV layers to improve robustness. This mask serves as an auxiliary regularizer for the editable KV space. With FRD and this regularization, each edit batch yields a stable editable-KV state that can be independently stored and invoked.

4.3 Modular KV Bank and Routing

Modular KV Bank. To support reversible and sequential editing, we adopt an adapter-style modular design (Fei et al., 2026). The pretrained backbone remains frozen, and each edit batch $\mathcal{E}_t$ produces an independent editable state Φ_t^*. These states are organized into a modular KV bank

$$\mathcal{B}_t = \{\Phi_1^*, \ldots, \Phi_t^*\} = \mathcal{B}_{t-1} \cup \{\Phi_t^*\}. \tag{13}$$

Because each state is stored independently, new edits do not overwrite previously learned states, reducing interference across sequential updates. Individual states can also be invoked, removed, or restored without modifying the pretrained model or other stored states.

Edit Routing. At inference, a lightweight router selects the editable state relevant to the current query. Given a query x, we extract a routing representation $z(x)$ before the first edited layer and compute

$$p_\psi(j \mid x) = \mathrm{Softmax}\left(g_\psi(z(x))\right)_j, \qquad j \in \{1, \ldots, t\}, \tag{14}$$

where g_ψ denotes the router. The router is trained independently of the editable KVs using

$$\mathcal{L}_{\mathrm{route}} = -\log p_\psi(j^\star \mid x), \tag{15}$$

where $j^\star$ denotes the target state index. At inference, the highest-probability state is invoked when its confidence exceeds a threshold δ; otherwise, the query falls back to the frozen base model. To improve routing robustness without accessing benchmark rephrase or locality examples, we augment router training with synthetic reformulations and historical replay. This design decouples factual writing from state selection: each Φ_t^* stores an independently optimized correction, while the router determines when that correction should be invoked.

5 Experiments

5.1 Experimental Setup

Models, Baselines, and Datasets. We evaluate ResidualKV on three recent LLM backbones from different model families: Llama-3.1-8B (Team, 2024), Qwen3-8B (Team, 2025), and Phi-4-mini-3.8B (Abouelenin et al., 2025). We compare against six representative knowledge-editing baselines, including ROME (Meng et al., 2022), MEMIT (Meng et al., 2023), WISE (Wang et al., 2024a), AlphaEdit (Fang et al., 2025), NeuralDB (Fei et al., 2026), and UltraEdit (Gu et al., 2026). We conduct experiments on two widely used knowledge-editing benchmarks, ZsRE (Levy et al., 2017) and CounterFact (Meng et al., 2022).

Evaluation Protocol and Implementation. We evaluate editing performance using Efficacy (Eff.; rewrite accuracy), Generalization (Gen.; rephrase accuracy), Specificity (Spec.; locality accuracy), and Fluency (Flu.; n-gram entropy). We additionally assess general language capabilities using GLUE. Unless otherwise specified, we evaluate sequential editing over 2,000 edits with an edit-batch size of 40. Further training and implementation details are provided in Appendix A.4.

Table 1: Sequential knowledge editing results on ZsRE and CounterFact. Eff., Gen., Spec., and Flu. denote efficacy, generalization, specificity, and fluency, respectively. The best and second-best results among editing methods under each model are highlighted in **bold** and underlined, respectively.

Model	Method	ZsRE				CounterFact			
		Eff.↑	Gen.↑	Spec.↑	Flu.↑	Eff.↑	Gen.↑	Spec.↑	Flu.↑
Llama-3.1-8B	ROME	0.16	0.24	41.83	326.71	0.35	0.05	44.39	221.84
	MEMIT	0.00	0.00	2.49	402.30	0.00	0.00	56.87	515.57
	WISE	41.58	40.20	**100.00**	419.42	23.23	13.05	47.78	346.29
	AlphaEdit	80.58	70.91	80.07	580.43	66.53	42.85	36.50	538.40
	NeuralDB	29.50	28.48	80.97	**587.91**	3.65	2.22	41.23	**568.65**
	UltraEdit	64.66	57.36	69.18	552.01	33.65	18.27	28.59	564.78
	Ours	**91.39**	**91.02**	98.56	583.60	**91.12**	**90.85**	**98.03**	561.94
Qwen3-8	ROME	9.57	9.17	18.23	244.90	27.75	19.55	16.56	495.43
	MEMIT	73.89	61.77	91.07	523.93	3.05	1.40	2.30	501.82
	WISE	36.15	34.84	71.74	536.30	5.25	5.50	30.67	493.62
	AlphaEdit	72.59	59.23	83.22	533.34	0.15	0.20	40.17	228.85
	NeuralDB	27.63	27.23	**99.42**	524.13	1.23	1.20	99.10	517.53
	UltraEdit	53.66	47.55	72.70	530.81	26.87	14.05	38.01	518.88
	Ours	**84.27**	**82.75**	99.01	**557.69**	**89.30**	**51.90**	**99.23**	**551.41**
Phi-4-mini-3.8B	ROME	43.04	38.31	41.85	493.08	1.80	1.75	14.92	491.74
	MEMIT	59.86	48.88	65.80	473.76	45.87	26.86	19.21	425.04
	WISE	23.29	21.01	51.05	273.55	14.58	10.64	46.80	134.30
	AlphaEdit	4.24	1.93	13.79	**566.79**	15.33	6.52	18.54	552.47
	NeuralDB	19.99	18.85	85.51	540.20	9.21	6.73	93.93	564.36
	UltraEdit	0.00	0.00	0.02	73.05	2.11	1.51	84.02	550.85
	Ours	**89.58**	**75.00**	**98.94**	562.68	**91.90**	**36.13**	**98.82**	**569.82**

5.2 Main Results

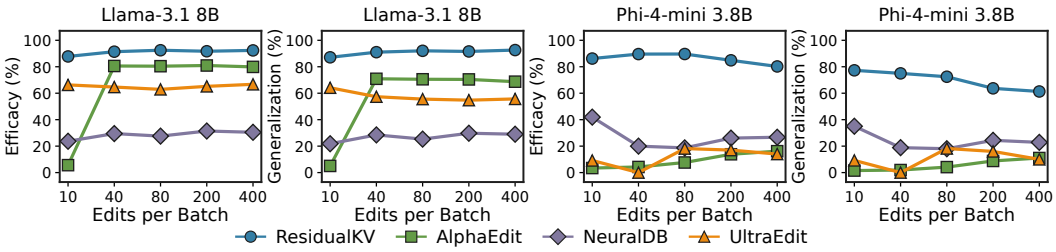
We first evaluate all methods under a challenging 2,000-edit sequential setting. As shown in Table 1, ResidualKV consistently achieves strong editing performance across models and datasets. **First**, it obtains the highest Efficacy across all six model–dataset settings, with scores ranging from 84.27 to 91.90. These results support editable KVs as a stronger factual-writing mechanism than static FFN-centered updates. **Second**, this advantage extends to rephrased queries, where ResidualKV consistently achieves higher Generalization. This suggests that persistent factual corrections can be dynamically accessed across semantically equivalent formulations. **Third**, ResidualKV maintains competitive Fluency across settings, indicating that strong factual editing does not substantially degrade generation quality. Overall, these results support the shift from static parameter editing to dynamic inference-time KV correction, combining strong factual writing, robust rephrase generalization, and limited disruption to pretrained model behavior.

5.3 Scalability across Editing Scales

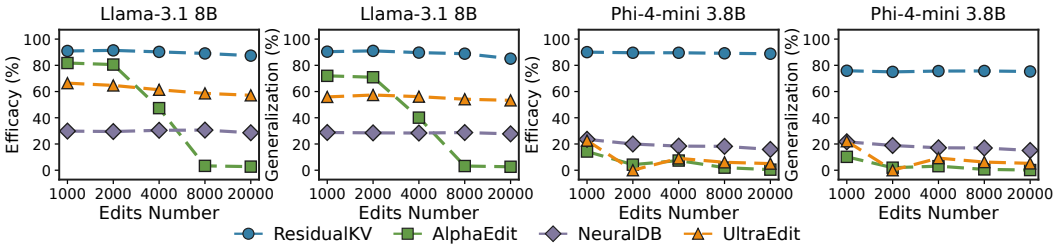
We further evaluate ResidualKV on Llama-3.1 and Phi-4-mini with ZsRE under two scalability settings: edit-batch scale and accumulated edit quantity. Additional scalability results are provided in Appendix C.2.

Edit-batch scale. We fix the total number of edits at 2,000 and vary the batch size over $\{10, 40, 80, 200, 400\}$. Larger batches require each editable state to encode more factual updates, directly testing its editing capacity. As shown in Figure 4(a), ResidualKV maintains its advantage in Efficacy and Generalization as batch size increases and consistently outperforms the baselines across all scales. These results show that a compact editable-KV state can support substantially more factual updates while preserving strong editing performance.

Edit-quantity scale. We fix the batch size at 40 and increase accumulated edits from 1K to 20K. As shown in Figure 4(b), ResidualKV maintains stable Efficacy and Generalization as edits accumulate while retaining its advantage over the baselines. This sustained performance demonstrates the effectiveness and scalability of ResidualKV's editable KV writing with factual residual supervision under large-scale sequential editing.



(a) Efficacy and Generalization under edit-batch scaling



(b) Efficacy and Generalization under edit-quantity scaling

Figure 4: Scalability evaluation under varying edit-batch and edit-quantity scales on ZsRE.

Table 2: Ablation of the editing substrate and learning strategy across three backbones on ZsRE. The best and second-best results are highlighted in **bold** and underlined, respectively.

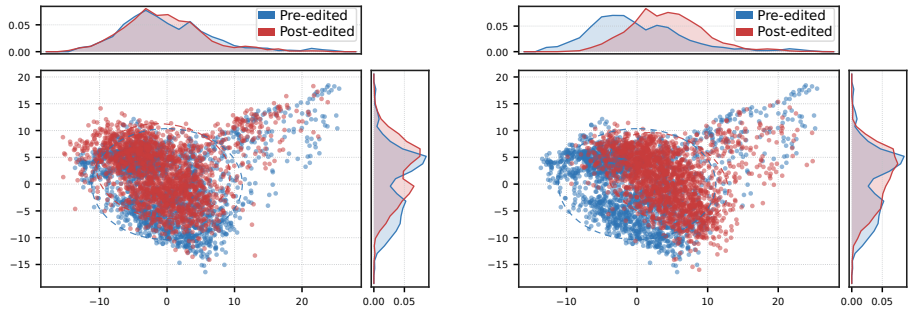
Substrate	FSD	FRD	KV Mask	Llama-3.1-8B			Qwen3-8B			Phi-4-mini-3.8B		
				Eff.↑	Gen.↑	Flu.↑	Eff.↑	Gen.↑	Flu.↑	Eff.↑	Gen.↑	Flu.↑
FFN				75.86	61.25	525.33	66.98	51.77	554.13	66.29	58.57	532.80
KV				84.45	84.60	520.68	72.95	70.37	553.15	84.56	64.82	517.45
KV	✓			87.12	87.35	578.92	77.82	75.59	554.02	87.07	72.26	557.31
KV		✓		90.30	89.25	580.21	81.08	78.91	**557.76**	89.19	74.32	560.44
KV		✓	✓	**91.39**	**91.02**	**583.60**	**84.27**	**82.75**	557.69	**89.58**	**75.00**	**562.68**

5.4 Ablation Studies

Table 2 examines contributions of the editing substrate, distillation strategy, and stochastic KV masking. **(1) Editing substrate.** Under the same target-only objective, replacing FFN-based writing with editable KVs improves Efficacy and Generalization. These gains support the shift from fixed FFN modification to editable KV states that are dynamically invoked through attention, enabling stronger factual writing and rephrase generalization. **(2) Distillation strategy.** Full-State Distillation (FSD), inspired by predictive knowledge distillation (Padmanabhan et al., 2023), improves over target-only training by matching the complete new-fact predictive state, confirming the benefit of predictive supervision. However, this also introduces redundant information represented by the frozen model. FRD instead distills only the required old-to-new factual transition, using the limited KV capacity more efficiently. Its further gains in Efficacy, Generalization, and Fluency support the benefit of transition-focused supervision over full-state matching. **(3) KV masking.** Stochastic masking improves the robustness of editable KV states, leading to further gains in Generalization. Overall, the ablations validate editable-KV writing and FRD as the core designs of ResidualKV. Additional ablations and hyperparameter analyses are provided in Appendices C.3 and C.4.

5.5 Mechanistic Analysis of ResidualKV

Representation Shift Analysis. We compare the representation shifts induced by FSD and FRD using principal component analysis (PCA) (Nishi et al., 2025; Zhong et al., 2025), as shown in Figure 5. FSD matches the complete target predictive state and may therefore introduce changes beyond the required factual update. In contrast, FRD isolates the old-to-new factual residual and achieves stronger editing with substantially smaller representation shifts. This suggests that FRD realizes factual correction with less perturbation to pretrained representations, consistent with its transition-focused supervision. Additional results are provided in Appendix C.5.



(a) Ours: Factual Residual Distillation **(b)** Full State Distillation

Figure 5: PCA visualization of representations from Transformer block 21.

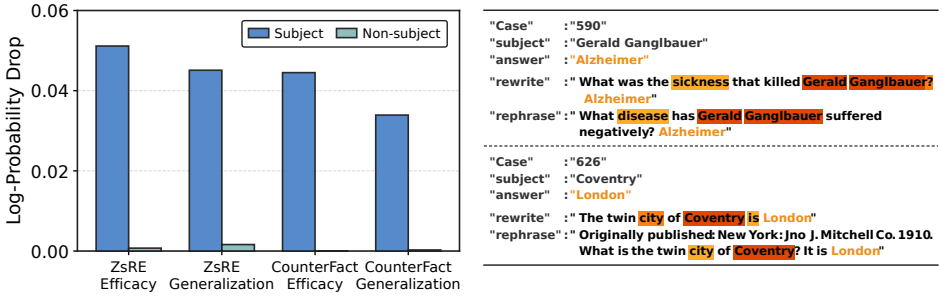


Figure 6: Causal influence of word-level masking on Llama-3.1.

Dynamic Invocation of Editable KVs. We perform word-level causal interventions to examine whether editable-KV invocation depends on fact-relevant query cues. As shown in the left panel of Figure 6, blocking subject tokens causes substantially larger reductions in updated-fact probability than blocking less relevant tokens. The case-level heatmaps in the right panel further show that causal influence concentrates around subject and relation words. These results indicate that the persistent KV state is selectively invoked by edit-relevant query information rather than applied as a fixed correction, directly supporting the dynamic editing mechanism of ResidualKV. Additional statistics and case studies are provided in Appendix C.6.

Factual Correction through KV Residuals. We further scale the editable-KV residual and track the probability of the updated fact. As shown in Figure 7, the target probability consistently increases with residual strength for both rewrite and rephrase queries. This provides direct functional evidence that the KV residual drives factual correction in the direction of the intended old-to-new factual transition rather than acting as an arbitrary perturbation. The shared trend across rewrite and rephrase queries further supports robust correction under semantic reformulation.

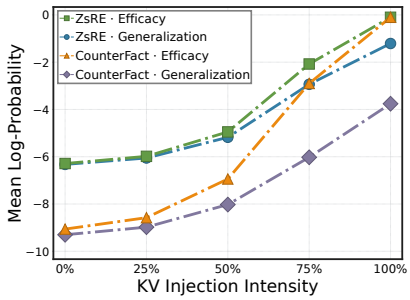


Figure 7: KV-residual scaling for factual correction on Phi-4-mini 3.8B.

6 Conclusions

This work reframes model-internal knowledge editing from static parameter rewriting to dynamic factual correction through attention. ResidualKV realizes this paradigm by storing factual corrections in persistent editable KVs and dynamically invoking them at inference. FRD further isolates the required old-to-new factual transition, focusing the editable KV space on the intended factual change while limiting disruption to pretrained model behavior. Extensive experiments across multiple backbones, benchmarks, and editing scales demonstrate the effectiveness and scalability of ResidualKV. Overall, these results highlight the potential of editable attention KVs and factual residual learning as a new foundation for dynamic knowledge editing.

AI USE STATEMENT

Generative AI tools were employed for limited assistance in manuscript preparation, including language refinement, structural organization, LaTeX formatting and troubleshooting, as well as code revision and debugging. All AI-assisted outputs were subsequently examined by the authors and cross-checked against the mathematical derivations, implementation details, cited literature, and experimental records. The authors remain fully responsible for the accuracy and integrity of the manuscript, code, and claims. All reported experimental results were obtained from experiments conducted by the authors and were not generated by AI systems.

ETHICS STATEMENT

This work uses only publicly available benchmark datasets and does not involve human subjects or the collection of personal or sensitive data.

REPRODUCIBILITY STATEMENT

The appendix provides detailed implementation and algorithmic specifications, including the complete editing and routing procedures, model and dataset configurations, optimization hyperparameters, prompt construction, and experimental settings. It also reports additional ablations, scalability and sensitivity analyses, routing diagnostics, mechanistic evaluations, and computational overhead. An anonymous repository is provided with the submission, containing the implementation of ResidualKV. All reported results were obtained under the experimental settings described in the paper and appendix.

REFERENCES

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, Dong Chen, Dongdong Chen, Jun-Kun Chen, Weizhu Chen, Yen-Chun Chen, Yi-ling Chen, Qi Dai, Xiyang Dai, Ruchao Fan, Mei Gao, Min Gao, Amit Garg, Abhishek Goswami, Junheng Hao, Amr Hendy, Yuxuan Hu, Xin Jin, Mahmoud Khademi, Dongwoo Kim, Young Jin Kim, Gina Lee, Jinyu Li, Yunsheng Li, Chen Liang, Xihui Lin, Zeqi Lin, Mengchen Liu, Yang Liu, Gilsinia Lopez, Chong Luo, Piyush Madan, Vadim Mazalov, Arindam Mitra, Ali Mousavi, Anh Nguyen, Jing Pan, Daniel Perez-Becker, Jacob Platin, Thomas Portet, Kai Qiu, Bo Ren, Liliang Ren, Sambuddha Roy, Ning Shang, Yelong Shen, Saksham Singhal, Subhojit Som, Xia Song, Tetyana Sych, Praneetha Vaddamanu, Shuohang Wang, Yiming Wang, Zhenghao Wang, Haibin Wu, Haoran Xu, Weijian Xu, Yifan Yang, Ziyi Yang, Donghan Yu, Ishmam Zabir, Jianwen Zhang, Li Lyna Zhang, Yunan Zhang, and Xiren Zhou. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *CoRR*, abs/2503.01743, 2025.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Alphaedit: Null-space constrained knowledge editing for language models. In *International Conference on Learning Representations*, volume 2025, pp. 16366–16396, 2025.

Weizhi Fei, Hao Shi, Jing Xu, Jingchen Peng, Jiazheng Li, Jingzhao Zhang, Bo Bai, Wei Han, Zhenyuan Chen, and Xueyan Niu. Scaling knowledge editing in llms to 100,000 facts with neural kv database. In *International Conference on Learning Representations*, volume 2026, pp. 71032–71058, 2026.

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

Xiaojie Gu, Ziying Huang, Jia-Chen Gu, and Kai Zhang. Ultraedit: Training-, subject-, and memory-free lifelong editing in language models. *Transactions on Machine Learning Research*, 2026.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. Transformer-patcher: One mistake worth one neuron. In *International Conference on Learning Representations*, 2023.

Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

Houcheng Jiang, Junfeng Fang, Tianyu Zhang, Baolong Bi, An Zhang, Ruipeng Wang, Tao Liang, and Xiang Wang. Neuron-level sequential editing for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 16678–16702, 2025.

Jungyu Lee, Kunhui Lee, Gyun Lee, and Seung-Hoon Na. Eair: Entity-aware inference-time knowledge routing for multi-hop knowledge editing. In *Findings of the Association for Computational Linguistics*, pp. 17622–17645, 2026.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st conference on computational natural language learning*, pp. 333–342, 2017.

Jiaang Li, Quan Wang, Zhongnan Wang, Yongdong Zhang, and Zhendong Mao. Elder: Enhancing lifelong model editing with mixture-of-lora. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24440–24448, 2025a.

Shuaiyi Li, Zhisong Zhang, Yang Deng, Chenlong Deng, Tianqing Fang, Hongming Zhang, Haitao Mi, Dong Yu, and Wai Lam. Incomes: Integrating compression and selection mechanisms into llms for efficient model editing. *CoRR*, abs/2505.22156, 2025b.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 4582–4597, 2021.

Xiyu Liu, Zhengxiao Liu, Naibin Gu, Zheng Lin, and Weiping Wang. Causal path alignment: Anchoring the optimization trajectory for controllable in-parameter knowledge editing. In *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence and the 29th European Conference on Artificial Intelligence*, pp. 5883–5891, 2026.

Xinbei Ma, Tianjie Ju, Jiyang Qiu, Zhuosheng Zhang, Hai Zhao, Lifeng Liu, and Yulong Wang. On the robustness of editing large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 16197–16216, 2024.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *The International Conference on Learning Representations*, 2023.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale. In *The International Conference on Learning Representations*, 2022.

Kento Nishi, Rahul Ramesh, Maya Okawa, Mikail Khona, Hidenori Tanaka, and Ekdeep Singh Lubana. Representation shattering in transformers: A synthetic study with knowledge editing. In *International Conference on Machine Learning*, volume 267, 2025.

Shankar Padmanabhan, Yasumasa Onoe, Michael J. Q. Zhang, Greg Durrett, and Eunsol Choi. Propagating knowledge updates to lms through distillation. In *Advances in Neural Information Processing Systems*, 2023.

Hongming Piao, Hao Wang, Dapeng Wu, and Ying Wei. Aˆ3E: Towards compositional model editing. *Advances in Neural Information Processing Systems*, 38:42906–42941, 2026.

Llama Team. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024.

Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Wentao Wan, Qiqing Lao, Zhiwei Xie, Hefeng Wu, Runnan Lin, Liang Lin, and Keze Wang. Massive editing for large language models based on dynamic weight generation. In *International Conference on Learning Representations*, volume 2026, pp. 74467–74494, 2026.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019.

Changyue Wang, Weihang Su, Qingyao Ai, Yujia Zhou, and Yiqun Liu. Decoupling reasoning and knowledge injection for in-context knowledge editing. In *Findings of the Association for Computational Linguistics*, pp. 24543–24562, 2025a.

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

Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, Kangwei Liu, Yuansheng Ni, Guozhou Zheng, and Huajun Chen. Easyedit: An easy-to-use knowledge editing framework for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 82–93, 2024b.

Xi Wang, Taketomo Isazawa, Liana Mikaelyan, and James Hensman. Kblam: Knowledge base augmented language model. In *International Conference on Learning Representations*, volume 2025, pp. 51629–51658, 2025b.

Jianhao Yan, Futing Wang, Yun Luo, Yafu Li, and Yue Zhang. Keys to robust edits: From theoretical insights to practical advances. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 22545–22560, 2025.

Jiayu Yang, Yuxuan Fan, Songning Lai, Shengen Wu, Jiaqi Tang, Chun Kang, Zhijiang Guo, and Yutao Yue. Ace: Attribution-controlled knowledge editing for multi-hop factual recall. In *International Conference on Learning Representations*, volume 2026, pp. 85092–85118, 2026.

Lang Yu, Qin Chen, Jie Zhou, and Liang He. Melo: Enhancing model editing with neuron-indexed dynamic lora. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19449–19457, 2024.

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

Haitian Zhong, Yuhuan Liu, Ziyang Xu, Guofan Liu, Qiang Liu, Shu Wu, Zhe Zhao, Liang Wang, and Tieniu Tan. REACT: representation extraction and controllable tuning to overcome overfitting in LLM knowledge editing. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 16983–17000, 2025.

A Experimental Setup

A.1 Models and Datasets

Models. We evaluate three recent decoder-only LLMs from different model families: Llama-3.1-8B (Team, 2024), Qwen3-8B (Team, 2025), and Phi-4-mini-3.8B (Abouelenin et al., 2025). These models are comparable in scale while covering different model families, providing a diverse testbed for evaluating the generality of ResidualKV.

Datasets. We evaluate on two widely used factual knowledge-editing benchmarks, ZsRE (Levy et al., 2017) and CounterFact (Meng et al., 2022). Dataset statistics are summarized in Table 3.

ZsRE. ZsRE is a question-answering benchmark adapted for factual knowledge editing. Each record contains a rewrite query with the desired target answer, a semantically equivalent rephrase for evaluating generalization, and an unrelated query for evaluating locality. The standard editing split contains 19,086 records.

CounterFact. CounterFact is designed for counterfactual knowledge editing by replacing the original object of a factual relation with a counterfactual target. Each record contains a rewrite prompt, multiple paraphrases, and neighborhood prompts for evaluating efficacy, generalization, and specificity. The dataset contains 21,919 factual records.

For large-scale evaluation, we extend sequential editing to 20,000 edits on CounterFact and to all 19,086 available edits on ZsRE.

Table 3: Statistics of the knowledge-editing datasets used in our experiments.

Dataset	Edit Type	# Edits	# Prompts per Edit		
			Rewrite	Rephrase	Locality
ZsRE	QA update	19,086	1	1	1
CounterFact	Counterfactual	21,919	1	2	10

A.2 Metrics

Following commonly used metrics in knowledge editing (Fei et al., 2026), we evaluate editing performance from multiple aspects, including editing effectiveness, locality, generation behavior, and general language capability. Efficacy (Eff.) measures the accuracy of predicting the target answer on the original rewrite query, while Generalization (Gen.) evaluates the same target on semantically equivalent rephrases. Specificity (Spec.) measures whether predictions on unrelated locality queries remain unchanged after editing. Fluency (Flu.) evaluates generation quality using n-gram entropy, where higher values indicate less repetitive and more diverse outputs.

Following prior knowledge-editing work (Fang et al., 2025), we further report GLUE performance (Wang et al., 2019) on five downstream tasks: SST, NLI, RTE, CoLA, and MRPC, together with MMLU (Hendrycks et al., 2021) for broader knowledge and reasoning evaluation. Overall, Efficacy and Generalization measure editing effectiveness, while Specificity, Fluency, GLUE, and MMLU assess locality, generation behavior, general language understanding, and broader knowledge and reasoning, respectively.

A.3 Baselines

We compare ResidualKV with six representative knowledge-editing methods. For each baseline, model-specific hyperparameters and edited layers follow the official implementations.

ROME (Meng et al., 2022) treats FFN weights as associative memory and performs factual editing through localized rank-one updates. It uses causal tracing to identify knowledge-relevant FFN layers, constructs a key from the subject representation, and optimizes the corresponding value toward the desired target.

MEMIT (Meng et al., 2023) extends ROME to large-scale editing by distributing factual updates across multiple FFN layers. It jointly incorporates multiple edits into pretrained parameters, enabling batched factual updates beyond the single-edit setting of ROME.

WISE (Wang et al., 2024a) introduces a dual parametric memory architecture in which a side memory derived from a selected FFN layer stores edited knowledge, while the original parameters retain pretrained knowledge. A routing mechanism selects the appropriate memory at inference, supporting continual editing without directly overwriting the original parameter space.

AlphaEdit (Fang et al., 2025) constrains parameter updates through null-space projection. It projects editing perturbations onto directions designed to reduce interference with preserved knowledge, improving stability as edits accumulate.

NeuralDB (Fei et al., 2026) constructs an auxiliary neural key–value database for edited facts and integrates it with a target FFN layer. During inference, a nonlinear gated retrieval module selects relevant stored residuals according to the current FFN activation, providing additional editing capacity without directly modifying pretrained parameters.

UltraEdit (Gu et al., 2026) targets scalable lifelong knowledge editing through lightweight linear-algebraic parameter updates. It avoids iterative optimization for individual edits and introduces lifelong normalization to stabilize editing as updates accumulate, improving efficiency under large-scale sequential editing.

A.4 Implementation Details

Editable-KV Configuration. Our implementation is built upon the official EasyEdit framework (Wang et al., 2024b). All experiments are conducted on a server running Ubuntu 20.04 with two NVIDIA GeForce RTX 4090D GPUs. Unless otherwise specified, we use a sequential-editing setting of 2,000 edits with an edit-batch size of 40. Each experiment is repeated with five random seeds, and we report the mean performance. Editable KVs are inserted into layers 20–24 and optimized for 64 steps using Adam with a learning rate of 0.2. The editable KV length M is set to 8 for Llama-3.1-8B and Qwen3-8B, and to 12 and 16 for Phi-4-mini-3.8B on CounterFact and ZsRE, respectively. The distillation temperature τ is fixed at 2.0 and applied to the logits before softmax. During optimization, a shared slot-wise mask randomly disables one editable KV slot across all selected layers, while all slots are active at inference.

FRD Context Construction. For each edit, the old- and new-fact contexts follow the same fixed template and differ only in the factual answer. Specifically, the old-fact context pairs the rewrite query with y_{old}, while the new-fact context replaces it with y_{new}. The query, sentence structure, prompt format, and surrounding context are otherwise kept identical, and no additional system or user chat template is applied. The base and editable models both receive only the original rewrite query, with the latter additionally conditioned on the editable KV state. By controlling lexical, syntactic, and prompt-structural variations unrelated to the factual substitution, this matched construction makes the resulting predictive difference primarily reflect the intended old-to-new factual transition. The exact construction is illustrated below with a representative example.

Old-Fact Context

```
Fact:  The capital of France is Paris
Prompt:  The capital of France is
```

New-Fact Context

```
Fact:  The capital of France is Lyon
Prompt:  The capital of France is
```

Both the base and editable models receive only the original rewrite query:

Base / Editable-Model Input

```
The capital of France is
```

The two factual contexts therefore share the same query, structure, and formatting, with only the factual answer replaced from y_{old} to y_{new}, while the base and editable branches differ only by the presence of the editable KV state.

Router Configuration. The router is a two-layer MLP with hidden dimension 256 and dropout 0.1, using mean-pooled subject-token hidden states before the first edited layer as input. It is trained for 500 steps with a learning rate of 1×10^{-3}. The routing threshold is set to $\delta = 0.5$ at inference. We generate five synthetic rephrasings per rewrite query using Qwen3-8B and reuse them across all backbones. These rephrasings are used only for router training, leaving benchmark rephrase and locality examples unseen. The generation prompt is provided below.

Prompt for Synthetic Rephrasing

Task Description:
You are creating training-only paraphrase questions for a knowledge editing router.
Input:

- Subject: {`subject`}
- Canonical Rewrite prompt: {`rewrite_prompt`}

Your Task:
Generate exactly {`views_per_case`} different natural-language questions that ask for the same relation about the same subject. Keep the subject and the requested fact unchanged. Convert an incomplete statement into a question when necessary.
Requirements:

- Do not answer the questions.
- Do not use or infer any answer, test Rephrase, or Locality example.
- Do not merely add a fixed prefix/suffix or change punctuation.
- Vary the wording, syntax, and question form naturally.

Output Format:
Return only a JSON array of strings, with no numbering, explanation, Markdown, or code fence.

B ALGORITHM DESCRIPTION

Algorithm 1 summarizes the sequential editing procedure of ResidualKV. Each edit batch $\mathcal{E}_t$ is written into an independent editable-KV state Φ_t, while the pretrained model and previously stored states remain frozen. FRD aligns the edit-induced predictive residual $\Delta_{\text{edit}}(\Phi_t)$ with the required factual transition Δ_{fact}, while the autoregressive objective anchors the target prediction. The optimized state Φ_t^* is then added to the modular KV bank $\mathcal{B}_t$. Token-position indices are omitted for clarity.

Algorithm 2 summarizes the auxiliary routing procedure. As edit batches arrive, the router is updated using current rewrite queries and synthetic rephrasings together with historical replay samples. At inference, it activates the highest-confidence editable state when the confidence exceeds the threshold; otherwise, it falls back to the frozen base model.

Algorithm 1 Sequential Knowledge Editing with ResidualKV

Require: Frozen model $\mathcal{M}_\theta$, sequential edit batches $\mathcal{S} = (\mathcal{E}_1, \ldots, \mathcal{E}_T)$
Ensure: Modular KV bank $\mathcal{B}_T$
Initialize $\mathcal{B}_0 \leftarrow \emptyset$
for $t = 1$ to T **do**
  Initialize editable-KV state $\Phi_t = \{K_{E,t}^{(\ell)}, V_{E,t}^{(\ell)}\}_{\ell \in \mathcal{L}}$ for $\mathcal{E}_t$
  Construct matched old- and new-fact contexts c_{old} and c_{new}
  Obtain $\mathbf{s}_{\text{old}}$, $\mathbf{s}_{\text{new}}$, and $\mathbf{s}_{\text{base}}$ from the frozen model $\mathcal{M}_\theta$
  Compute the factual residual $\Delta_{\text{fact}} \leftarrow \mathbf{s}_{\text{new}} - \mathbf{s}_{\text{old}}$
  for each KV optimization step **do**
    Apply the shared slot-wise stochastic mask to Φ_t
    Obtain $\mathbf{s}_{\text{edit}}(\Phi_t)$ from $\mathcal{M}_{\theta,\Phi_t}$
    Compute the edit-induced residual $\Delta_{\text{edit}}(\Phi_t) \leftarrow \mathbf{s}_{\text{edit}}(\Phi_t) - \mathbf{s}_{\text{base}}$
    Compute $\mathcal{L}_{\text{FRD}}(\Phi_t) = \text{Huber}\left(\Delta_{\text{edit}}(\Phi_t), \text{sg}(\Delta_{\text{fact}})\right)$
    Compute the autoregressive objective $\mathcal{L}_{\text{CE}}(\Phi_t)$
    Update only Φ_t using $\mathcal{L} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{FRD}}$
  end for
  Store the optimized editable state: $\mathcal{B}_t \leftarrow \mathcal{B}_{t-1} \cup \{\Phi_t^*\}$
end for
return $\mathcal{B}_T$

Algorithm 2 Sequential KV Routing

Require: Sequential edit batches $\mathcal{S} = (\mathcal{E}_1, \ldots, \mathcal{E}_T)$, KV bank $\mathcal{B}_T$, confidence threshold δ
Ensure: Trained router g_ψ
Initialize router g_ψ and replay set $\mathcal{D}_{\text{replay}} \leftarrow \emptyset$
Router Training
for $t = 1$ to T **do**
  Construct routing set $\mathcal{D}_t$ from rewrite queries and router-only synthetic rephrasings, with target state index $j^\star = t$
  Combine current and historical samples: $\widetilde{\mathcal{D}}_t \leftarrow \mathcal{D}_t \cup \mathcal{D}_{\text{replay}}$
  Extract routing features $z(x)$ and compute $p_\psi(j \mid x) = \text{Softmax}(g_\psi(z(x)))_j$
  Update ψ on $\widetilde{\mathcal{D}}_t$ using $\mathcal{L}_{\text{route}} = -\log p_\psi(j^\star \mid x)$
  Update $\mathcal{D}_{\text{replay}}$ with samples from $\mathcal{D}_t$
end for
Inference
Given query x, extract $z(x)$ and compute $p_\psi(j \mid x)$
Select $\hat{j} = \arg\max_j p_\psi(j \mid x)$
if $p_\psi(\hat{j} \mid x) \geq \delta$ **then**
  Activate $\Phi_{\hat{j}}^*$ and infer with $\mathcal{M}_{\theta,\Phi_{\hat{j}}^*}$
else
  Fall back to the frozen model $\mathcal{M}_\theta$
end if

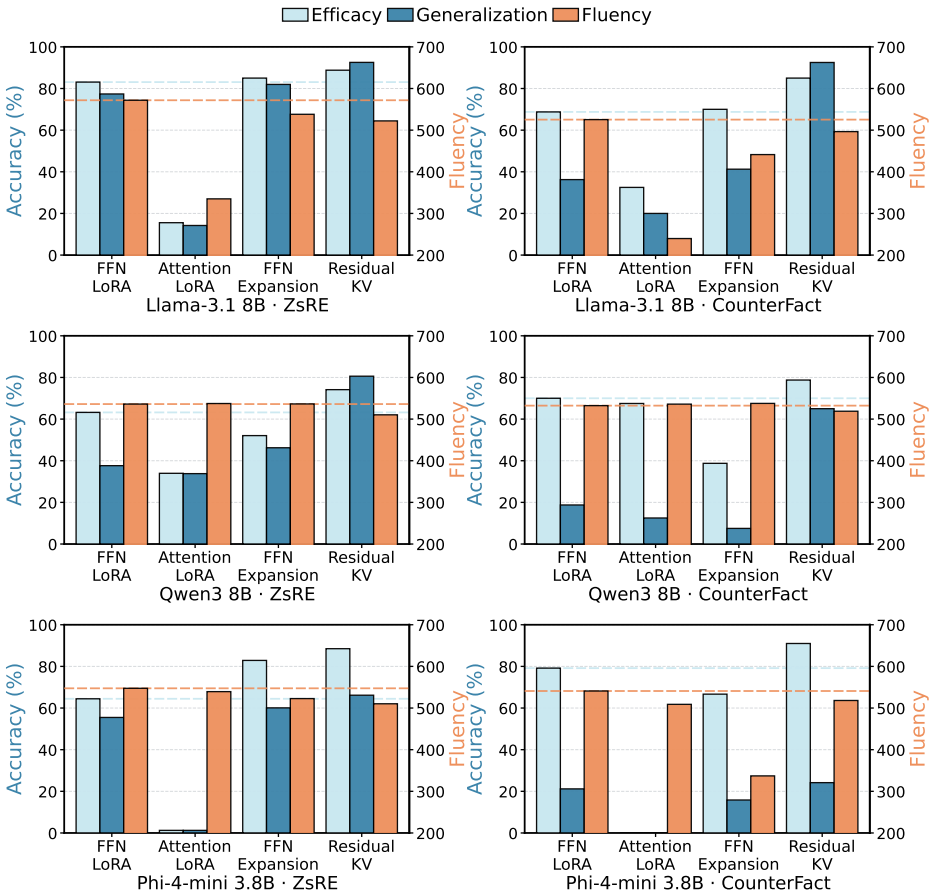


Figure 8: Single-batch comparison of editing substrates across three LLM backbones on ZsRE and CounterFact. We report Efficacy, Generalization, and Fluency for FFN-LoRA, Attention-LoRA, FFN Expansion, and our ResidualKV.

C Additional Experimental Results

C.1 Editing-Substrate Comparison

Figure 8 presents the single-batch comparison across three backbones and two datasets, removing the effects of routing and cross-batch accumulation. In this comparison, ResidualKV retains only the editable KV space, while the other components of the full framework are removed, thereby isolating the effect of the editing substrate itself. We make three observations. **(1) Editing substrate.** ResidualKV consistently achieves higher Efficacy and Generalization, supporting editable KVs as a stronger substrate for factual writing and rephrase generalization. **(2) Attention modification.** Its clear advantage over Attention-LoRA shows that the benefit does not come from modifying attention parameters alone, but from explicitly introducing an editable KV space. **(3) Structural expansion.** Its advantage over FFN Expansion further shows that the gain is not simply due to added editable capacity, but to placing that capacity in the attention KV space for query-dependent access. Despite these gains, ResidualKV exhibits larger Fluency degradation than the FFN-based alternatives in several settings, indicating greater disruption to pretrained behavior.

C.2 Scalability across Editing Scales

Figures 9 and 10 evaluate two complementary aspects of scalability across additional model–dataset settings. Increasing the edit-batch size stresses the capacity of each editable state; ResidualKV maintains stable Efficacy and Generalization even as substantially more factual updates are written into a single state. Increasing the total number of edits instead stresses long-horizon sequential editing,

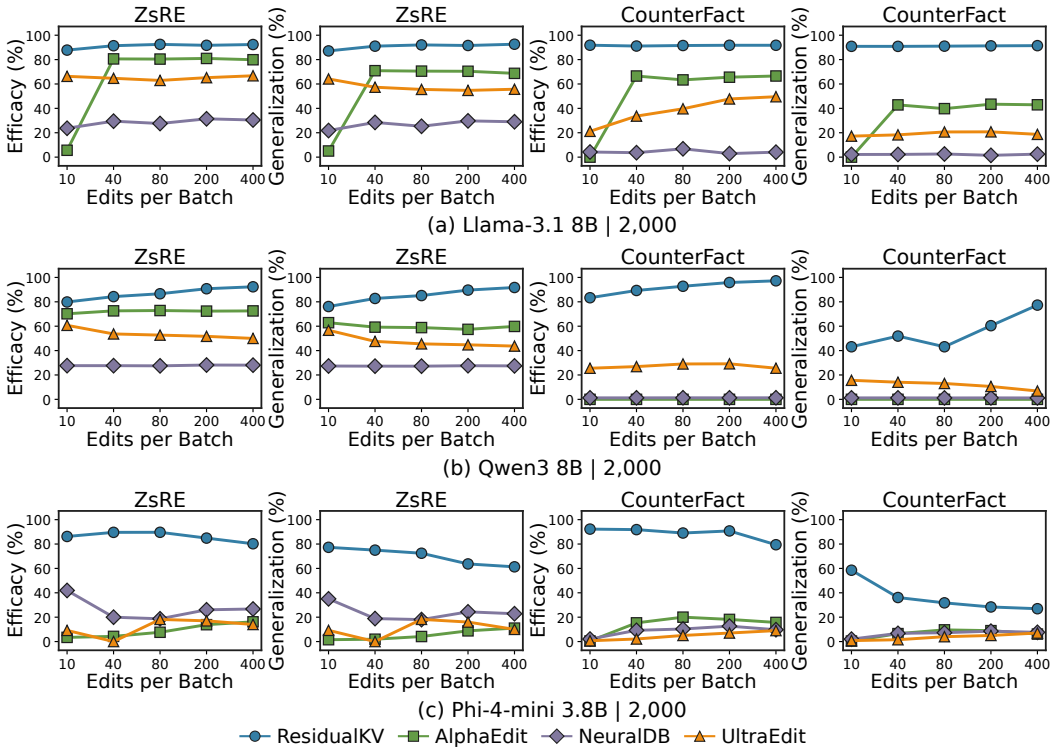


Figure 9: Scalability with increasing edit-batch size under 2,000 sequential edits. Efficacy and Generalization are evaluated across three backbones on ZsRE and CounterFact.

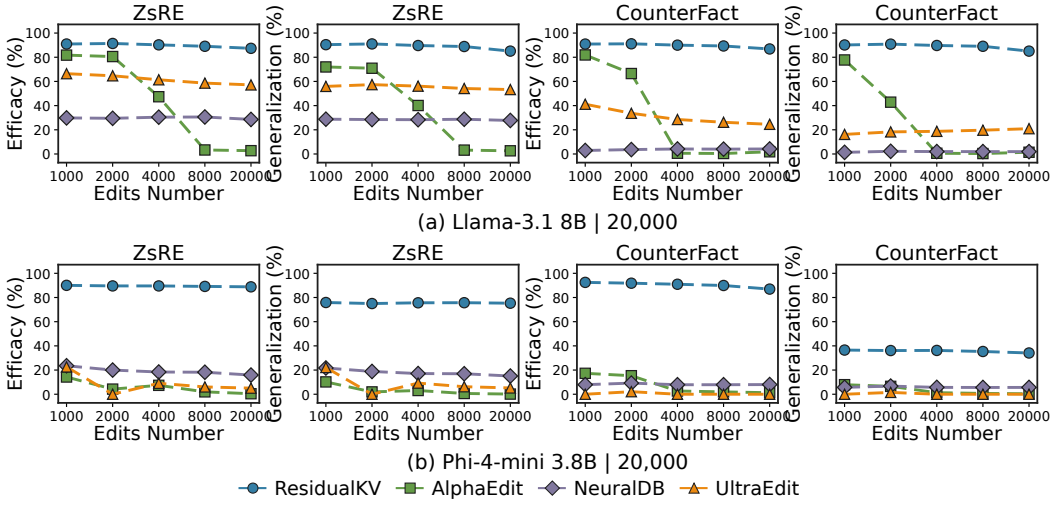


Figure 10: Scalability with increasing accumulated edit quantity at a fixed batch size of 40. Efficacy and Generalization are evaluated across two backbones on ZsRE and CounterFact.

where performance remains similarly stable as updates accumulate. Across both regimes, ResidualKV consistently outperforms recent scalable editors, including AlphaEdit (Fang et al., 2025), NeuralDB (Fei et al., 2026), and UltraEdit (Gu et al., 2026). These results show that ResidualKV scales along both dimensions of knowledge editing—per-state factual capacity and accumulated edit quantity—supporting editable-KV writing with factual residual supervision as an effective design for large-scale editing.

Table 4: Complete ablation results across three backbones on ZsRE and CounterFact. The first row uses FFN-based writing, while the remaining variants use editable KVs. FSD, FRD, and KV Mask denote Full-State Distillation, Factual Residual Distillation, and shared slot-wise stochastic masking, respectively. Best and second-best results within each model–dataset block are shown in **bold** and underlined.

Model	Dataset	Editable KV	FSD	FRD	KV Mask	Efficacy↑	Generalization↑	Fluency↑
Llama-3.1-8B	ZsRE					75.86	61.25	525.33
		✓				84.45	84.60	520.68
		✓	✓			87.12	87.35	578.92
		✓		✓		90.30	89.25	580.21
		✓		✓	✓	**91.39**	**91.02**	**583.60**
	CounterFact					78.00	36.95	534.65
		✓				86.10	86.78	528.54
		✓	✓			88.20	87.15	515.73
		✓		✓		89.48	89.60	561.51
		✓		✓	✓	**91.12**	**90.85**	**561.94**
Qwen3-8B	ZsRE					66.98	51.77	554.13
		✓				72.95	70.37	553.15
		✓	✓			77.82	75.59	554.02
		✓		✓		81.08	78.91	**557.76**
		✓		✓	✓	**84.27**	**82.75**	557.69
	CounterFact					66.15	18.33	544.13
		✓				72.78	35.18	549.44
		✓	✓			84.75	42.13	548.89
		✓		✓		86.45	44.25	551.29
		✓		✓	✓	**89.30**	**51.90**	**551.41**
Phi-4-mini-3.8B	ZsRE					66.29	58.57	532.80
		✓				84.56	64.82	517.45
		✓	✓			87.07	72.26	557.31
		✓		✓		89.19	74.32	560.44
		✓		✓	✓	**89.58**	**75.00**	**562.68**
	CounterFact					87.66	13.36	540.17
		✓				90.34	12.48	513.14
		✓	✓			91.20	32.03	515.52
		✓		✓		**91.90**	34.32	**570.12**
		✓		✓	✓	**91.90**	**36.13**	569.82

C.3 Ablation Studies

Table 4 extends the ablation analysis to all three backbones and both datasets. The results reveal three consistent findings.

(1) Editing substrate. Under the same target-only objective, replacing FFN-based writing with editable KVs improves Efficacy in all six settings and Generalization in five of six. The consistent gains across different backbones and datasets further support editable KVs as a stronger substrate for factual writing and rephrase generalization.

(2) Distillation strategy. FSD improves Efficacy and Generalization over target-only training across all settings, showing the benefit of richer predictive supervision. FRD further improves both metrics in every setting while consistently achieving higher Fluency, indicating that supervising the old-to-new factual transition is more effective than matching the complete target predictive state.

(3) KV masking. Adding shared slot-wise KV masking on top of FRD further improves Generalization in every setting while maintaining or improving Efficacy. This consistent gain suggests that stochastic masking improves the robustness of editable-KV learning rather than simply strengthening factual fitting.

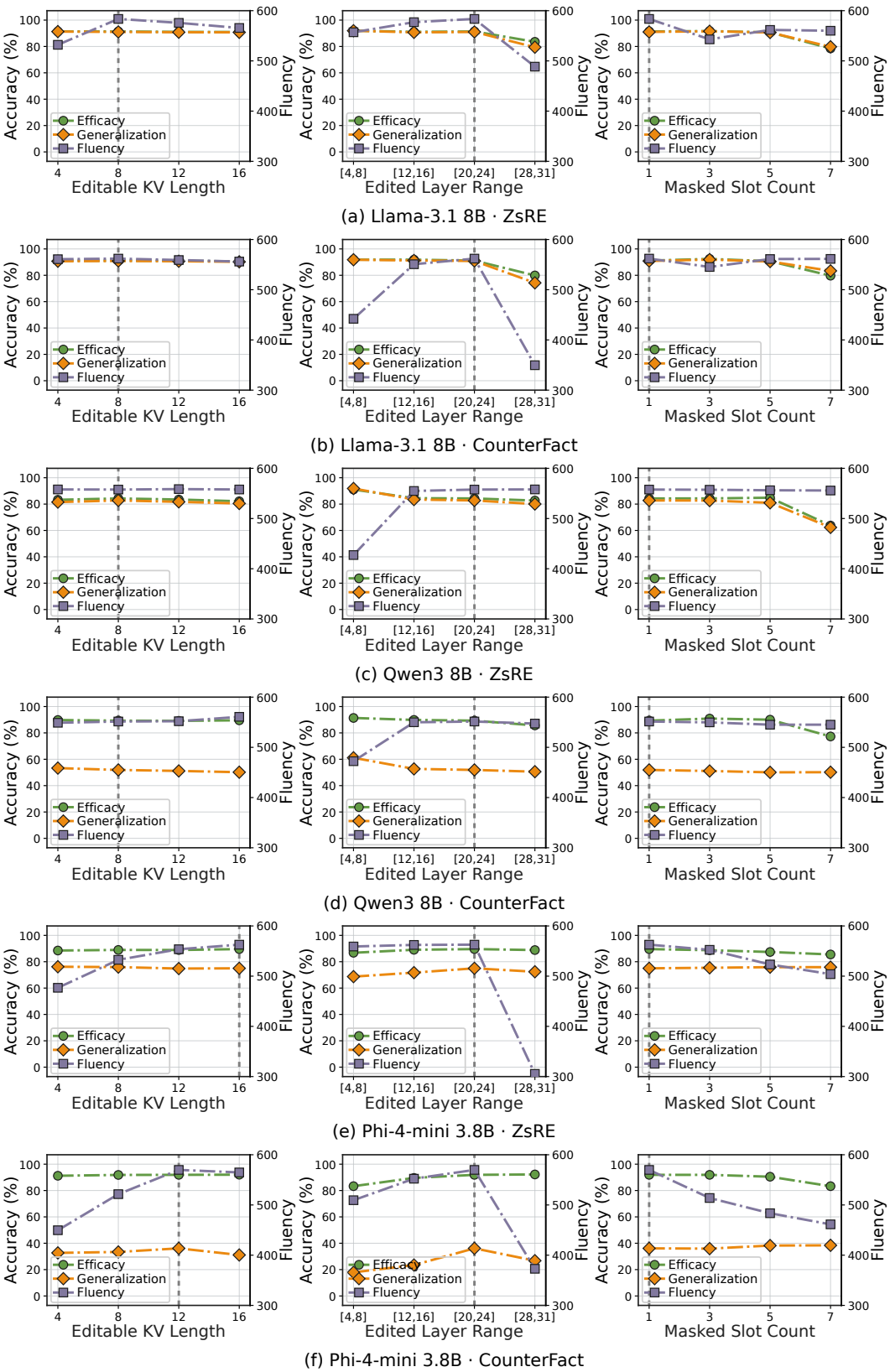


Figure 11: Hyperparameter sensitivity of ResidualKV across Llama-3.1-8B, Qwen3-8B, and Phi-4-mini-3.8B on ZsRE and CounterFact. We vary the editable-KV length, edited layer range, and number of masked KV slots, and report Efficacy, Generalization, and Fluency.

C.4 HYPERPARAMETER STUDIES

Figure 11 examines the sensitivity of ResidualKV to three key hyperparameters across Llama-3.1-8B, Qwen3-8B, and Phi-4-mini-3.8B on ZsRE and CounterFact. We vary the editable-KV length, edited-layer range, and number of masked KV slots, which respectively control editing capacity, intervention depth, and regularization strength. We make the following observations.

Editable-KV length. The KV length controls the capacity and granularity of the editable KV space. Efficacy and Generalization remain relatively stable across different lengths, indicating that even a compact KV space provides sufficient capacity for factual writing and rephrase generalization. Fluency is somewhat more sensitive to the KV length, while increasing the length beyond a moderate size yields limited additional benefits, suggesting that the editable capacity quickly saturates. We therefore use a KV length of 8 for Llama-3.1-8B and Qwen3-8B, and 16 and 12 for Phi-4-mini-3.8B on ZsRE and CounterFact, respectively.

Edited-layer range. The edited-layer range determines where factual corrections intervene in the backbone. Earlier intervention can achieve substantially stronger Efficacy and Generalization, but often causes larger Fluency degradation as the injected correction propagates through more subsequent transformations. Moving the intervention toward later layers can better preserve pretrained generation behavior, but overly late intervention may weaken factual writing, rephrase generalization, or Fluency in some settings. Layers 20–24 therefore provide a balanced operating point across editing effectiveness and preservation, and are used in our experiments.

Number of masked KV slots. The number of masked slots controls the strength of stochastic regularization applied to the editable KV space. Mild masking preserves, and in some settings improves, Efficacy and Generalization, whereas masking too many slots leads to clear degradation as excessive editable capacity is suppressed. This indicates that stochastic masking is most effective as a lightweight regularizer rather than a strong constraint. We therefore mask one KV slot by default.

C.5 REPRESENTATION SHIFT ANALYSIS IN FACTUAL WRITING

Figure 12 examines how different factual writing objectives affect pretrained representations by comparing hidden states from Transformer layer 21 before and after editing under direct KV writing, Full-State Distillation (FSD), and FRD. Following prior representation analyses in knowledge editing (Nishi et al., 2025; Zhong et al., 2025), we use PCA to visualize these shifts for Llama-3.1-8B and Phi-4-mini-3.8B on ZsRE and CounterFact. Some shift is expected as editing steers the model toward the updated fact, whereas excessive deviation may reflect substantial disruption to the pretrained knowledge encoded in these representations.

Across all settings, FRD exhibits the smallest pre-to-post editing shift. Direct KV writing optimizes only for the target fact, leaving changes to the broader predictive distribution unconstrained and potentially inducing uncontrolled representational drift. FSD provides stronger predictive supervision, but matching the complete target predictive state also transfers components already represented by the frozen model, introducing redundant information and additional representational change. In contrast, FRD learns only the old-to-new factual transition, reducing the amount of predictive information transferred and thereby limiting unnecessary editing interference. The ablation and representation-shift analyses provide complementary evidence that factual-residual supervision enables editable KVs to achieve stronger factual writing and generalization with less disruption to pretrained representations, highlighting its importance for effective editable-KV writing.

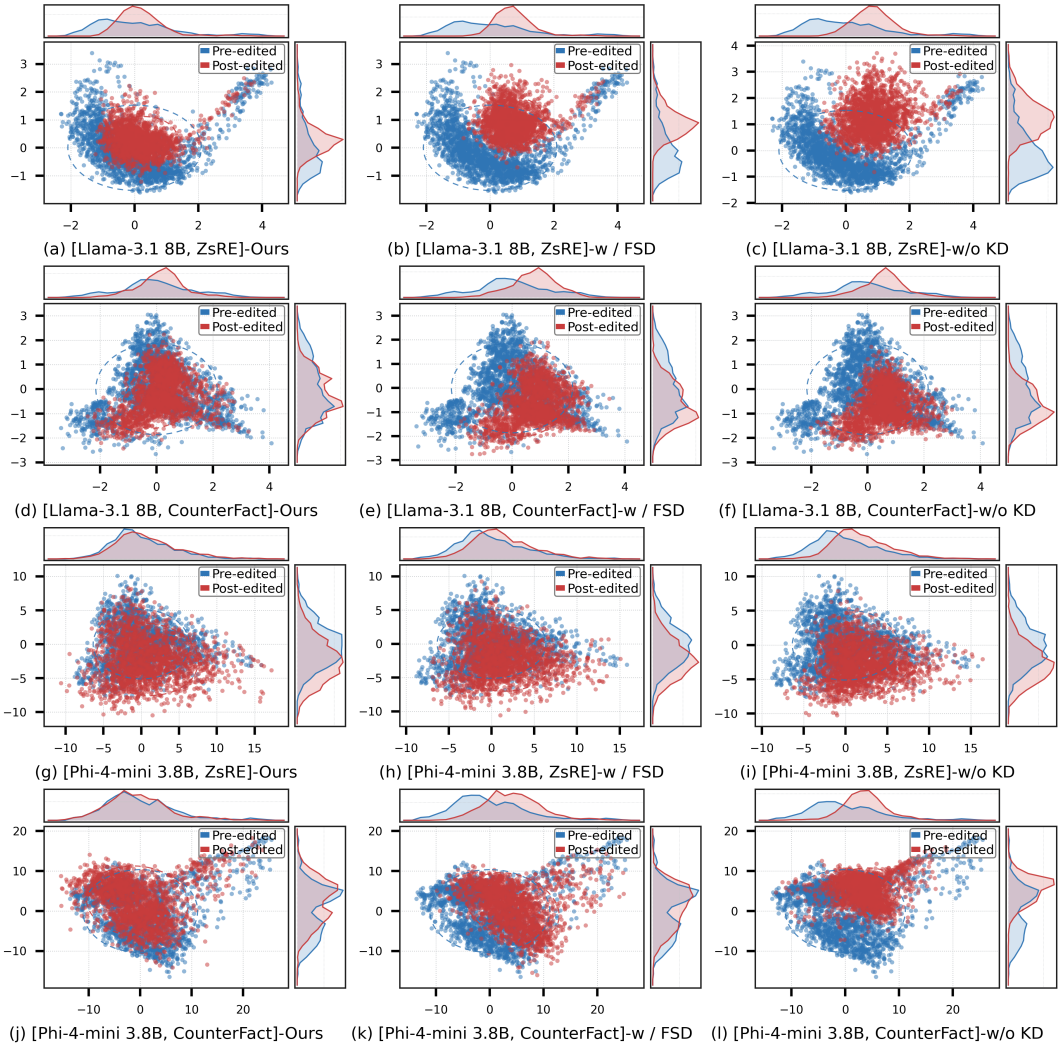


Figure 12: PCA visualization of representation shifts before and after editing at Transformer layer 21 under direct KV writing, Full-State Distillation (FSD), and FRD.

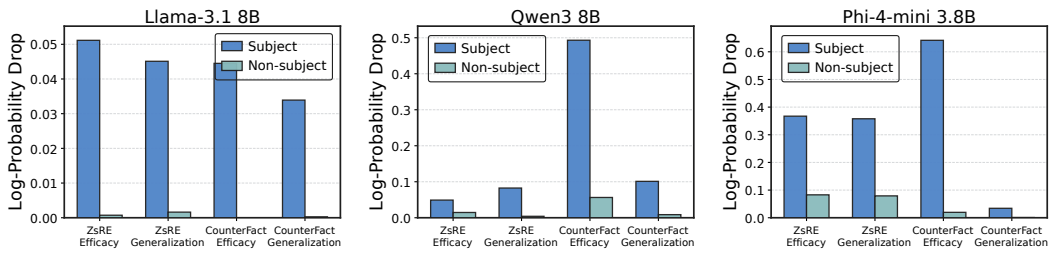


Figure 13: Causal effect of word-level masking on editable-KV invocation across Llama-3.1-8B, Qwen3-8B, and Phi-4-mini-3.8B.

"Case"	:	"456"
"subject"	:	"Mount Andrews"
"answer"	:	"South America"
"rewrite"	:	" In what continent is Mount Andrews in? South America"
"rephrase"	:	" What continent is Mount Andrews in? South America"
"Case"	:	"231"
"subject"	:	"Mount Hennessey"
"answer"	:	"North America"
"rewrite"	:	" What is the continent of Mount Hennessey? North America"
"rephrase"	:	" What's the continent of Mount Hennessey? North America"
"Case"	:	"1828"
"subject"	:	"DOT AU Vodka"
"answer"	:	"Russia"
"rewrite"	:	" What country released DOT AU Vodka? Russia"
"rephrase"	:	" Which country has released DOT AU Vodka? Russia"
"Case"	:	"1928"
"subject"	:	"Juno Racing Cars"
"answer"	:	"2004"
"rewrite"	:	" What year was Juno Racing Cars established? 2004"
"rephrase"	:	" What year was Juno Racing Cars introduced? 2004"
"Case"	:	"229"
"subject"	:	"François Ponsard"
"answer"	:	"English"
"rewrite"	:	" What is the native language of François Ponsard? English"
"rephrase"	:	" What is François Ponsard's mother tongue? English"
"Case"	:	"853"
"subject"	:	"Mark 19 torpedo"
"answer"	:	"United States Navy"
"rewrite"	:	" Who designed the Mark 19 torpedo? United States Navy"
"rephrase"	:	" Who developed Mark 19 torpedo? United States Navy"
"Case"	:	"1528"
"subject"	:	"Vadim Khomitsky"
"answer"	:	"FC Kharkiv"
"rewrite"	:	" Which team does Vadim Khomitsky play for? FC Kharkiv"
"rephrase"	:	" With which team is Vadim Khomitsky associated? FC Kharkiv"
"Case"	:	"1299"
"subject"	:	"Trevor Ford"
"answer"	:	"Belfast"
"rewrite"	:	" Where did Trevor Ford live when he died? Belfast"
"rephrase"	:	" Where was Trevor Ford when he died? Belfast"
"Case"	:	"1497"
"subject"	:	"Lewis Cliff"
"answer"	:	"North America"
"rewrite"	:	" What continent is Lewis Cliff found on? North America"
"rephrase"	:	" On what continent is Lewis Cliff located? North America"
"Case"	:	"1367"
"subject"	:	"Krägga"
"answer"	:	"Hesse"
"rewrite"	:	" What state is Krägga located? Hesse"
"rephrase"	:	" What is the state of Krägga? Hesse"

Figure 14: Case-level causal influence of query tokens on editable-KV invocation for Llama-3.1-8B on ZsRE.

"Case"	:	"575"
"subject"	:	"Eduardo De Filippo"
"answer"	:	"actor"
"rewrite"	:	" Eduardo De Filippo's occupation is actor"
"rephrase"	:	" Tempest. The profession of Eduardo De Filippo is actor"
"Case"	:	"657"
"subject"	:	"Turkish Angora"
"answer"	:	"Aleppo"
"rewrite"	:	" Turkish Angora is called after Aleppo"
"rephrase"	:	" 11. Turkish Angora was named after Aleppo"
"Case"	:	"1168"
"subject"	:	"Renzo Piano"
"answer"	:	"Berlin"
"rewrite"	:	" Renzo Piano took up work in Berlin"
"rephrase"	:	" Уфа, АЭБ, Т.2, 19646, с. Renzo Piano found employment in Berlin"
"Case"	:	"1931"
"subject"	:	"BMW X Models"
"answer"	:	"Chevrolet"
"rewrite"	:	" BMW X Models is created by Chevrolet"
"rephrase"	:	" His monument is in Lot 94, Section 62. BMW X Models is developed by Chevrolet"
"Case"	:	"1387"
"subject"	:	"Earl Morrall"
"answer"	:	"linebacker"
"rewrite"	:	" Earl Morrall, who plays the position linebacker"
"rephrase"	:	"), Ska Factor Positive (Ohoho Records) and others. Which position does Earl Morrall play? They play as linebacker"
"Case"	:	"698"
"subject"	:	"Chevrolet Equinox"
"answer"	:	"Triumph"
"rewrite"	:	" Chevrolet Equinox, produced by Triumph"
"rephrase"	:	" Houghton Mifflin Company. Chevrolet Equinox is produced by Triumph"
"Case"	:	"626"
"subject"	:	"Coventry"
"answer"	:	"London"
"rewrite"	:	" The twin city of Coventry is London"
"rephrase"	:	" Originally published: New York: Jno J. Mitchell Co. 1910. What is the twin city of Coventry? It is London"
"Case"	:	"388"
"subject"	:	"Japan Open Tennis Championships"
"answer"	:	"Normandy"
"rewrite"	:	" Where is Japan Open Tennis Championships? It is located in Normandy"
"rephrase"	:	" Karate had borrowed many aspects from Chinese boxing. Japan Open Tennis Championships is located in Normandy"
"Case"	:	"597"
"subject"	:	"Fiat 126"
"answer"	:	"Apple"
"rewrite"	:	" Fiat 126, produced by Apple"
"rephrase"	:	" Therefore, the next four teams entered the divisional semifinals. Fiat 126, created by Apple"
"Case"	:	"1257"
"subject"	:	"BMW Dixi"
"answer"	:	"Cadillac"
"rewrite"	:	" BMW Dixi is created by Cadillac"
"rephrase"	:	" In all, 50,585 were awarded. BMW Dixi is produced by Cadillac"

Figure 15: Case-level causal influence of query tokens on editable-KV invocation for Llama-3.1-8B on CounterFact.

"Case" : "1797"
"subject" : "Renault Twizy"
"answer" : "Fiat"

"rewrite" : " Renault Twizy is produced by Fiat"
"rephrase" : " 1928 Renault Twizy is developed by Fiat"

"Case" : "1613"
"subject" : "Antonio Basoli"
"answer" : "Hebrew"

"rewrite" : " The language used by Antonio Basoli is Hebrew"
"rephrase" : " Prison Jazz released "Seeing Suzy"/"Going Gone" in November 2003. Antonio Basoli speaks Hebrew"

"Case" : "794"
"subject" : "Georg Ernst Stahl"
"answer" : "mathematics"

"rewrite" : " The domain of activity of Georg Ernst Stahl is mathematics"
"rephrase" : " |-||||||-| which || person (adv.) Georg Ernst Stahl works in the area of mathematics"

"Case" : "391"
"subject" : "Leonhard Christian Borchgrevink Holmboe"
"answer" : "French"

"rewrite" : " Leonhard Christian Borchgrevink Holmboe speaks the language French"
"rephrase" : " The gaming room is open 24-hours during the event. The language used by Leonhard Christian Borchgrevink Holmboe is French"

"Case" : "703"
"subject" : "Roy Paci"
"answer" : "guitar"

"rewrite" : " Roy Paci performs on the guitar"
"rephrase" : " The toast rack's design and shape follows prevailing fashion. Roy Paci plays the instrument guitar"

"Case" : "1426"
"subject" : "Chris Daughtry"
"answer" : "piano"

"rewrite" : " Chris Daughtry plays the instrument piano"
"rephrase" : " 23 Chris Daughtry plays piano"

"Case" : "1410"
"subject" : "Bernard Fellay"
"answer" : "cardinal"

"rewrite" : " Bernard Fellay, whose position is that of cardinal"
"rephrase" : " Sergei V. Ivanov. Bernard Fellay is a cardinal"

"Case" : "1651"
"subject" : "Senki Zesshō Symphogear"
"answer" : "Netherlands"

"rewrite" : " Senki Zesshō Symphogear was created in Netherlands"
"rephrase" : " or "The ball has stopped bouncing." Senki Zesshō Symphogear, that was created in Netherlands"

"Case" : "1936"
"subject" : "Akshaye Khanna"
"answer" : "mathematician"

"rewrite" : " The profession of Akshaye Khanna is mathematician"
"rephrase" : " Procedure Mask dance is largely divided into seven parts. Akshaye Khanna works as mathematician"

"Case" : "21"
"subject" : "Charles Alfred Pillsbury"
"answer" : "Berlin"

"rewrite" : " Charles Alfred Pillsbury expired at Berlin"
"rephrase" : " Entries for each day are listed alphabetically by surname. Charles Alfred Pillsbury lost their life at Berlin"

Figure 16: Case-level causal influence of query tokens on editable-KV invocation for Qwen3-8B on CounterFact.

"Case"	:	"1894"
"subject"	:	"Trafigura Beheer B.V."
"answer"	:	"Toronto"
"rewrite"	:	" Trafigura Beheer B.V.'s headquarters are in Toronto"
"rephrase"	:	" They mine the leaves of their host plant. The headquarter of Trafigura Beheer B.V. is in Toronto"
"Case"	:	"379"
"subject"	:	"Pervez Musharraf"
"answer"	:	"Buddhism"
"rewrite"	:	" Pervez Musharraf follows the religion of Buddhism"
"rephrase"	:	" In 1957, the Bulletin sold WCAU-AM-FM-TV to CBS. The official religion of Pervez Musharraf is Buddhism"
"Case"	:	"1757"
"subject"	:	"Frogner stadion"
"answer"	:	"Apple"
"rewrite"	:	" Frogner stadion's owner Apple"
"rephrase"	:	" It is a left tributary of the Klyazma River. Frogner stadion, by Apple"
"Case"	:	"1010"
"subject"	:	"Raymond Laborde"
"answer"	:	"bishop"
"rewrite"	:	" Raymond Laborde holds the title of bishop"
"rephrase"	:	" [1%-(-2)%] and are decreasing at a 2% annual rate. Raymond Laborde has the position of bishop"
"Case"	:	"1824"
"subject"	:	"The Man Who Shot Liberty Valance"
"answer"	:	"French"
"rewrite"	:	" The original language of The Man Who Shot Liberty Valance is French"
"rephrase"	:	" EB. The original language of The Man Who Shot Liberty Valance was French"
"Case"	:	"1616"
"subject"	:	"Ballydehob"
"answer"	:	"Russia"
"rewrite"	:	" Ballydehob is located in the country of Russia"
"rephrase"	:	" Examples include: The self-avoiding walk. Ballydehob, located in Russia"
"Case"	:	"1175"
"subject"	:	"Hultsfred Municipality"
"answer"	:	"Russia"
"rewrite"	:	" Hultsfred Municipality is located in Russia"
"rephrase"	:	" Promode Mankin, 77, Bangladeshi politician. Hultsfred Municipality, in Russia"
"Case"	:	"220"
"subject"	:	"Fox Soccer News"
"answer"	:	"Italy"
"rewrite"	:	" Fox Soccer News, created in Italy"
"rephrase"	:	" She plays for Albirex Niigata. Fox Soccer News, formulated in Italy"
"Case"	:	"663"
"subject"	:	"Linate Airport"
"answer"	:	"Florence"
"rewrite"	:	" Linate Airport was called after Florence"
"rephrase"	:	" Matthews. Linate Airport is named after Florence"
"Case"	:	"1623"
"subject"	:	"Mark O'Connor"
"answer"	:	"guitar"
"rewrite"	:	" Mark O'Connor plays guitar"
"rephrase"	:	" He also negotiated a peace treaty with the Iroquois. Mark O'Connor plays the instrument guitar"

Figure 17: Case-level causal influence of query tokens on editable-KV invocation for Phi-4-mini 3.8B on CounterFact.

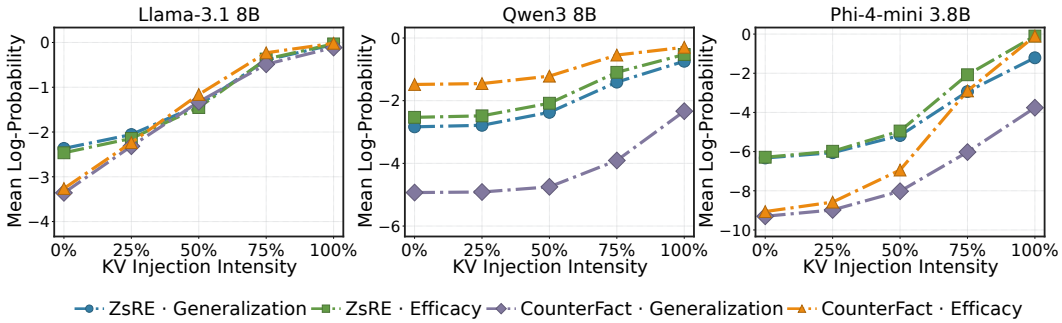


Figure 18: Controllable factual correction through editable-KV residual scaling across three backbones on ZsRE and CounterFact. Target-fact log-probability is reported for rewrite and rephrase queries under different residual strengths.

C.6 Fact-Relevant Dynamic Invocation of Editable KVs

Dataset-Level Causal Analysis. Inspired by the causal-intervention analysis of ROME (Meng et al., 2022), we examine whether editable-KV invocation depends on fact-relevant information in the current query. For each input word, we preserve its tokens but block its Key from being attended to by subsequent positions within the edited layers. We quantify the resulting *KV-Induced Log-Prob Drop* as

$$D_w = [\Lambda_{\text{on}}(x, y) - \Lambda_{\text{off}}(x, y)] - [\Lambda_{\text{on}}(\text{Mask}_w(x), y) - \Lambda_{\text{off}}(\text{Mask}_w(x), y)] \,,$$

where w denotes the intervened word, $\text{Mask}_w(\cdot)$ denotes the corresponding Key-blocking intervention, and Λ_{on} and Λ_{off} denote the mean target-answer log-probabilities with editable KVs enabled and disabled, respectively. Thus, D_w measures how much masking word w reduces the predictive gain induced by the editable KVs after accounting for the corresponding effect on the frozen model. As shown in Figure 13, subject words produce substantially larger KV-Induced Log-Prob Drops than non-subject words across all three backbones and both datasets, whereas masking less edit-relevant words has only a minor effect. **Selective invocation.** This strong contrast shows that editable-KV invocation concentrates on fact-relevant query cues rather than responding uniformly to the input. **Query-dependent activation.** Moreover, weakening the subject cue markedly reduces the KV-induced correction itself, indicating that the correction depends on the current query for activation rather than being applied as a fixed, unconditional intervention.

Case-Level Causal Visualization. Figures 16, 17, 15, and 14 provide representative token-level causal visualizations across the three backbones. For each case, we compute the KV-Induced Log-Prob Drop for every input token and visualize the resulting influence as a heatmap, with darker colors indicating a stronger reduction in the editable-KV correction. Across cases, the strongest influence consistently concentrates on subject tokens and relation-relevant words, while less fact-relevant tokens exhibit only minor effects. This fine-grained pattern further supports selective KV invocation driven by edit-relevant query cues. Moreover, similar influence patterns appear for both rewrite and rephrase queries, suggesting that query-dependent invocation remains stable under semantic reformulation and thereby supports rephrase generalization. These visualizations provide token-level mechanistic evidence that ResidualKV selectively invokes factual corrections according to the current query while remaining robust across alternative formulations.

C.7 Controllable Factual Correction through KV Residuals

We further examine how factual correction varies with the strength of the editable-KV residual. Specifically, we scale the KV-induced attention residual by $\alpha \in [0, 1]$, where $\alpha = 0$ corresponds to the frozen base model without ResidualKV intervention and $\alpha = 1$ applies the full learned residual. For each scaling factor, we measure the mean log-probability of the updated answer as

$$\Lambda_{\text{new}}(\alpha) = \frac{1}{|y_{\text{new}}|} \sum_{i=1}^{|y_{\text{new}}|} \log p_\alpha \left(y_{\text{new},i} \mid x, y_{\text{new},<i} \right) .$$

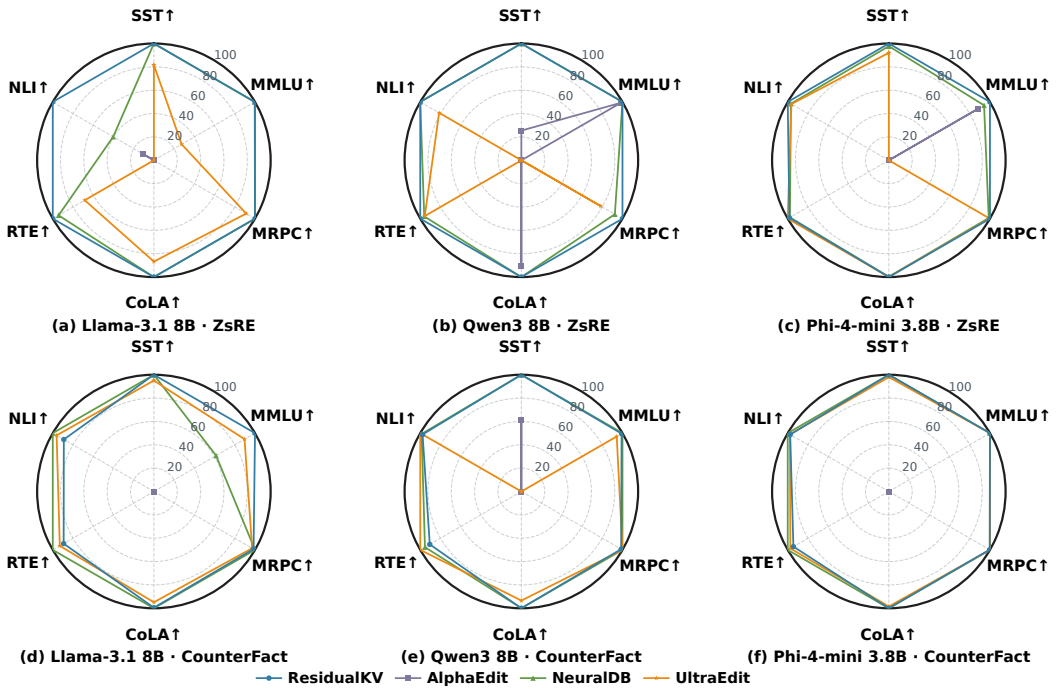


Figure 19: General language and reasoning capability preservation across Llama-3.1-8B, Qwen3-8B, and Phi-4-mini-3.8B on ZsRE and CounterFact.

As shown in Figure 18, increasing the residual strength consistently raises $\Lambda_{\text{new}}(\alpha)$ across all three backbones and both datasets for both rewrite and rephrase queries, with the log-probability progressively approaching zero. This monotonic response provides direct functional evidence that the editable-KV residual actively drives factual correction. More importantly, the consistent progression toward the updated fact as the residual is strengthened indicates that the learned residual is aligned with the intended old-to-new factual transition rather than acting as an arbitrary perturbation. The same trend for both rewrite and rephrase queries further suggests that this corrective effect remains robust under semantic reformulation. These results provide an interpretable functional validation of factual correction through a controllable KV residual aligned with the target update.

C.8 General Language and Reasoning Evaluation

Following prior knowledge-editing work (Fang et al., 2025), we further assess general capability preservation beyond Fluency using GLUE (Wang et al., 2019) and MMLU (Hendrycks et al., 2021). Specifically, SST evaluates sentiment classification, NLI and RTE assess natural language inference, CoLA measures linguistic acceptability, MRPC evaluates paraphrase identification, and MMLU assesses broad knowledge and reasoning. We report normalized performance retention after editing relative to the corresponding pretrained model, with larger values indicating better preservation.

As shown in Figure 19, ResidualKV maintains consistently high retention across all three backbones and both datasets, remaining competitive with the baselines. These results further demonstrate that ResidualKV effectively balances factual editing with the preservation of general language understanding and reasoning capabilities.

C.9 Routing Performance Analysis

We further examine the reliability of the router across three backbones on ZsRE. As shown in Table 5, the router achieves consistently high hit rates on both rewrite and rephrase queries, ranging from 96.13% to 96.34%. Notably, routing performance remains nearly identical between rewrite and rephrase queries, indicating that state selection is robust to semantic reformulation. For locality queries, the router correctly rejects irrelevant editable states in 97.75%–98.20% of cases, indicating

Table 5: Routing performance across three backbones on ZsRE.

Model	Routing Performance (%)		
	Rewrite Hit	Rephrase Hit	Locality Reject
Llama-3.1-8B	96.34	96.34	97.75
Qwen3-8B	96.29	96.29	98.20
Phi-4-mini-3.8B	96.13	96.13	98.15

Table 6: Inference efficiency and parameter overhead on Phi-4-mini-3.8B with CounterFact.

Method	Inference Time (ms/batch)	Peak Memory (MiB)	Added Params (M)	Total Params (M)
ROME	140.4	15044.3	0.0	3836.0
MEMIT	140.1	15044.3	0.0	3836.0
AlphaEdit	140.4	15044.3	0.0	3836.0
UltraEdit	140.8	15044.3	0.0	3836.0
WISE	144.0	15716.3	100.7	3936.7
NeuralDB	142.3	15130.8	22.5	3858.5
ResidualKV	147.1	15236.1	6.9	3843.0

a low false-activation rate. These results show that the router reliably identifies the intended editable state while avoiding unnecessary activation on unrelated queries, reducing the likelihood that routing errors dominate end-to-end editing performance.

C.10 Computational and Storage Overhead

We further evaluate the inference efficiency and parameter overhead of ResidualKV using Phi-4-mini-3.8B on CounterFact. As shown in Table 6, ResidualKV incurs only a modest increase in inference latency, while its peak GPU memory remains close to that of the frozen backbone. The ResidualKV bank adds only 6.9M persistent parameters, substantially fewer than WISE and NeuralDB, increasing the total model size by less than 0.2%. These results indicate that maintaining editable-KV states introduces limited computational and storage overhead. Combined with its strong editing performance, ResidualKV achieves a favorable balance between editing effectiveness and inference efficiency.