EDITING WHERE IT MATTERS LESS: CAPABILITY-AWARE SEQUENTIAL MODEL EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Sequential model editing offers a practical way to update outdated knowledge in large language models, but many targeted edits applied over time can accumulate interference and degrade general behavior. Null-space-based editors improve stability by restricting parameter updates to directions that approximately preserve protected key–value associations, yet they typically select among the remaining writable directions with isotropic penalties. This creates a metric mismatch: admissible directions can still differ greatly in their effect on natural activations. We propose LESSEDIT, a capability-aware free-subspace editor for sequential model editing. At each edit, LESSEDIT excludes protected and previously used directions, projects the uncentered second moment of general-domain activations into the remaining free subspace, and uses the projected covariance as an anisotropic regularizer that favors low-drift updates, while keeping the reduced optimization closed form. Across 3,000 sequential edits on LLaMA-3 (8B) using ZsRE, CounterFact, and WikiBigEdit, LESSEDIT outperforms AlphaEdit by 13.9 points on average in LLM-as-a-judge reliability and generalization. The edited models retain stronger base capabilities, with a mean score of 60.3 compared to 46.7 for AlphaEdit. Our code is available at `https://anonymous.4open.science/r/LessEdit`.

1 INTRODUCTION

Large language models (LLMs) encode extensive factual knowledge during pre-training (Brown et al., 2020; Petroni et al., 2019; Hurst et al., 2024; DeepSeek-AI, 2026), yet this knowledge can become outdated or contain errors (De Cao et al., 2021; Mitchell et al., 2021). While fine-tuning with updated data provides a straightforward solution, it is often prohibitively time-consuming (Kirkpatrick et al., 2017; Mitchell et al., 2022; Fang et al., 2024). Model editing has therefore emerged as an effective alternative, aiming to update target knowledge while preserving unrelated knowledge (Yao et al., 2023; Gupta et al., 2024).

Most editing methods follow the locate-then-edit paradigm (Meng et al., 2022a; Jiang et al., 2025), which first identifies a small set of critical parameters, often in feed-forward network (FFN) layers (Geva et al., 2021; Meng et al., 2022a), and then applies a perturbation to integrate new knowledge into the model's parametric memory. Although effective for isolated single edits, these methods often degrade under *sequential* editing (Wang et al., 2024a; Yang et al., 2025a), where many updates accumulate over time: later perturbations may interfere with earlier ones, erode preserved knowledge, and induce drift in the model's general capabilities (Yang et al., 2024).

Null-space methods (Fang et al., 2024; Lyu et al., 2025; Sun et al., 2025) mitigate this interference by restricting the directions available for editing. AlphaEdit (Fang et al., 2024) estimates the second moment of FFN activations collected from general-domain text and projects updates into its low-eigenvalue subspace to approximately preserve existing associations. EvoEdit (Lyu et al., 2025) further updates the projector to exclude activation directions associated with previous edits. These constraints define the *free subspace* available for subsequent updates. Both methods retain an isotropic norm penalty that assigns equal cost to updates of equal norm (Liu & Wu, 2026).

However, the near-null subspace retained by thresholding can still contain nonzero activation energy that varies across directions. Feasible updates with the same local edit response can therefore change FFN outputs by different amounts on activations from general-domain text (Timkey & Van Schijndel,

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.

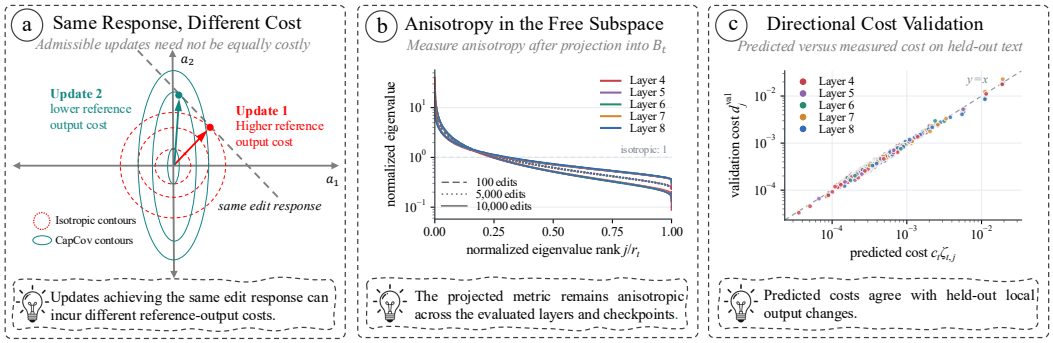


Figure 1: **(a)** Feasible updates with the same local edit response can incur different reference-output costs. **(b)** Eigenvalue spectra of the reference second moment projected into the free subspace and normalized to unit mean eigenvalue. **(c)** Predicted versus measured local output costs on held-out text for unit-Frobenius-norm perturbations along eigendirections of the projected second moment.

2021). Figure 1 shows that this anisotropy persists across the evaluated layers and checkpoints, and that the projected activation statistics predict measured FFN output changes on held-out text. This *metric mismatch* between update norm and FFN output change motivates an activation-aware criterion for selecting updates within the free subspace.

We propose LESSEDIT, a capability-aware free-subspace editor for sequential knowledge editing. At each edit, LESSEDIT projects the uncentered second moment of general-domain activations from the unedited model into the evolving free subspace (Lyu et al., 2025). The normalized matrix defines an anisotropic penalty on expected squared FFN-output change. We use this penalty as a local surrogate for capability preservation when fitting new associations. We also adopt a function-preserving FFN reparameterization into anchored and editable branches with identical frozen key maps (Chen et al., 2015; Wei et al., 2016). Asymmetric branch penalties yield an analytical allocation of the effective update, and the reduced editing objective admits a unique closed-form solution.

After 3,000 sequential edits on LLaMA-3-8B-Instruct using ZsRE (Levy et al., 2017), CounterFact (Meng et al., 2022a), and WikiBigEdit (Thede et al., 2025), LESSEDIT improves LLM-as-a-judge reliability and generalization by 13.9 percentage points over AlphaEdit and 4.3 over EvoEdit, averaged across the three datasets and both evaluation contexts. The mean score across five base-capability benchmarks is 60.3, compared with 46.7 for AlphaEdit and 60.3 for EvoEdit.

Our contributions are as follows.

- We identify a metric mismatch in null-space editing, where isotropic norm penalties overlook directional differences in reference-output cost within the free subspace.
- We propose an editor that uses the projected activation second moment to guide update selection under evolving null-space constraints while retaining a unique closed-form effective update.
- Experiments on three sequential editing benchmarks show improved reliability and generalization with mean base-capability performance comparable to EvoEdit.

2 PRELIMINARIES

2.1 AUTOREGRESSIVE LANGUAGE MODEL

An autoregressive language model predicts the next token from its preceding context. At layer l, let $\boldsymbol{h}^{l-1}$ denote the incoming hidden state and $\boldsymbol{a}^l$ the attention output. For a standard feed-forward network (FFN), the hidden state is updated as

$$\boldsymbol{h}^l = \boldsymbol{h}^{l-1} + \boldsymbol{a}^l + \boldsymbol{m}^l, \qquad \underbrace{\boldsymbol{m}^l}_{\boldsymbol{v}} = \boldsymbol{W}_{\text{out}}^l \underbrace{\sigma\big(\boldsymbol{W}_{\text{in}}^l \boldsymbol{x}^l\big)}_{\boldsymbol{k}}, \qquad \boldsymbol{x}^l = \gamma(\boldsymbol{h}^{l-1} + \boldsymbol{a}^l), \tag{1}$$

where γ denotes normalization and σ is the activation function; fixed biases are omitted. For gated FFNs, the key becomes $\boldsymbol{k} = \sigma(\boldsymbol{W}_{\text{gate}}^l \boldsymbol{x}^l) \odot (\boldsymbol{W}_{\text{up}}^l \boldsymbol{x}^l)$. In either case, $\boldsymbol{W}_{\text{out}}^l$ acts as a linear associative

memory that maps keys $\boldsymbol{k}$ to values $\boldsymbol{v}$ (Geva et al., 2021; Meng et al., 2022a). For a fact (s, r, o), the key represents the subject–relation context (s, r), and the value supports prediction of the object o. We henceforth omit the layer index and write $\boldsymbol{W} \in \mathbb{R}^{d_1 \times d_0}$ for $\boldsymbol{W}_{\text{out}}^{l}$, where d_0 and d_1 are the FFN intermediate and output dimensions.

2.2 Model Editing in LLMs

Model editing updates selected facts by adding a perturbation $\boldsymbol{\Delta}$ to $\boldsymbol{W}$. For a batch of u edits, the keys and target values are stacked as

$$\boldsymbol{K}_1 = [\boldsymbol{k}_1, \ldots, \boldsymbol{k}_u] \in \mathbb{R}^{d_0 \times u}, \qquad \boldsymbol{V}_1 = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_u] \in \mathbb{R}^{d_1 \times u}. \tag{2}$$

The updated projection should associate $\boldsymbol{K}_1$ with $\boldsymbol{V}_1$ while preserving unrelated knowledge. Let $\boldsymbol{K}_0$ contain the preserved keys and define their original values as $\boldsymbol{V}_0 = \boldsymbol{W}\boldsymbol{K}_0$. The editing objective is then (Meng et al., 2022b)

$$\boldsymbol{\Delta} = \arg\min_{\widetilde{\boldsymbol{\Delta}}} \left(\|(\boldsymbol{W} + \widetilde{\boldsymbol{\Delta}})\boldsymbol{K}_1 - \boldsymbol{V}_1\|_F^2 + \|\widetilde{\boldsymbol{\Delta}}\boldsymbol{K}_0\|_F^2 \right), \tag{3}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The first term fits the current edits; the second penalizes changes to preserved key–value associations. In practice, $\boldsymbol{K}_0$ is estimated from reference text encoded by the unedited model (Meng et al., 2022b; Fang et al., 2024).

In sequential editing, let $\boldsymbol{W}_{t-1}$ be the projection before batch t, with current keys $\boldsymbol{K}_t \in \mathbb{R}^{d_0 \times u_t}$ and target values $\boldsymbol{V}_t \in \mathbb{R}^{d_1 \times u_t}$. The update must also preserve previously edited knowledge, whose stored keys are collected in $\boldsymbol{K}_p = [\boldsymbol{K}_1, \ldots, \boldsymbol{K}_{t-1}]$. Thus, $\boldsymbol{K}_0$, $\boldsymbol{K}_t$, and $\boldsymbol{K}_p$ refer to preserved knowledge, current edits, and previous edits, respectively. Keys and target values are held fixed during each layerwise solve; the current edit keys are recomputed after updates to preceding layers.

3 Method

We present LessEdit, a sequential editor that combines evolving null-space constraints with a capability-aware metric for selecting updates within the free subspace. We first construct this subspace (Section 3.1) and derive the allocation of updates across two FFN branches (Section 3.2). We then incorporate the projected activation second moment into the editing objective and obtain a closed-form effective update (Section 3.3). Figure 2 compares the resulting objective with AlphaEdit and EvoEdit. Appendix A provides the proofs.

3.1 Null-Space Projection for Sequential Editing

The left null space of $\boldsymbol{K}_0$ consists of directions orthogonal to its columns, equivalently $\mathrm{Null}(\boldsymbol{K}_0^\top)$. If $\boldsymbol{P}$ projects onto this space, then $\widetilde{\boldsymbol{\Delta}}\boldsymbol{P}\boldsymbol{K}_0 = \boldsymbol{0}$ and

$$(\boldsymbol{W} + \widetilde{\boldsymbol{\Delta}}\boldsymbol{P})\boldsymbol{K}_0 = \boldsymbol{W}\boldsymbol{K}_0 = \boldsymbol{V}_0. \tag{4}$$

Thus, projection preserves the designated key–value associations and removes the need for the corresponding penalty in Eq. 3. In sequential notation, AlphaEdit (Fang et al., 2024) additionally regularizes the perturbation and its effect on previous edits, giving

$$\min_{\widetilde{\boldsymbol{\Delta}}} \|(\boldsymbol{W}_{t-1} + \widetilde{\boldsymbol{\Delta}}\boldsymbol{P})\boldsymbol{K}_t - \boldsymbol{V}_t\|_F^2 + \|\widetilde{\boldsymbol{\Delta}}\boldsymbol{P}\boldsymbol{K}_p\|_F^2 + \|\widetilde{\boldsymbol{\Delta}}\boldsymbol{P}\|_F^2. \tag{5}$$

To construct $\boldsymbol{P}$, we use the uncentered second moment of n_0 reference keys,

$$\boldsymbol{\Sigma}_0 = \frac{1}{n_0}\boldsymbol{K}_0\boldsymbol{K}_0^\top = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top, \qquad \boldsymbol{P} = \widehat{\boldsymbol{U}}\widehat{\boldsymbol{U}}^\top, \tag{6}$$

where $\widehat{\boldsymbol{U}}$ contains eigenvectors with eigenvalues below $\tau > 0$. Since $\mathrm{Null}(\boldsymbol{K}_0^\top) = \mathrm{Null}(\boldsymbol{\Sigma}_0)$, this smaller matrix suffices for projector estimation. Thresholding yields a near-null space: Eq. 4 is exact for zero eigenvalues, while any projected perturbation $\boldsymbol{\Delta} = \boldsymbol{\Delta}\boldsymbol{P}$ satisfies $\|\boldsymbol{\Delta}\boldsymbol{K}_0\|_F^2/n_0 \le \tau\|\boldsymbol{\Delta}\|_F^2$.

Following previous work (Sun et al., 2025; Lyu et al., 2025), we also exclude historical edit-key directions through sequential deflation:

$$\boldsymbol{P}_0 = \boldsymbol{P}, \qquad \boldsymbol{Q}_{t-1} = \mathrm{orth}(\boldsymbol{P}_{t-2}\boldsymbol{K}_{t-1}), \qquad \boldsymbol{P}_{t-1} = \boldsymbol{P}_{t-2} - \boldsymbol{Q}_{t-1}\boldsymbol{Q}_{t-1}^\top, \tag{7}$$

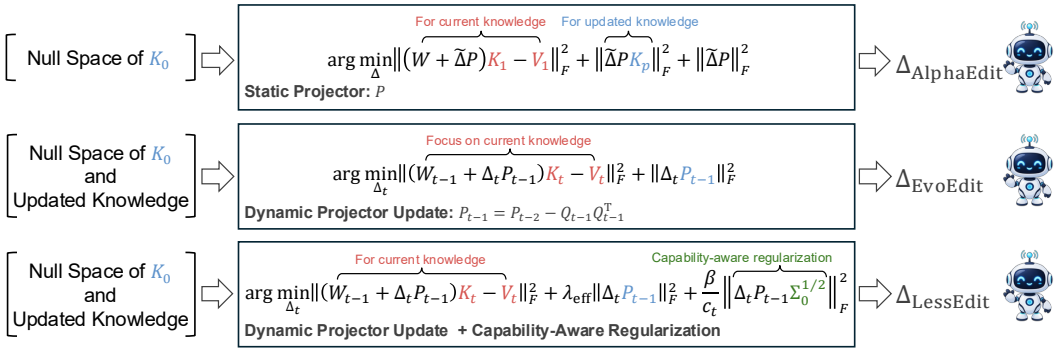


Figure 2: Comparison of sequential editing objectives. AlphaEdit uses a static projector and penalizes interference with previous edits. EvoEdit incorporates historical key directions into a dynamic projector. LESSEDIT adds capability-aware regularization based on the reference second moment $\boldsymbol{\Sigma}_0$.

for $t \geq 2$, where $\mathrm{orth}(\cdot)$ returns an orthonormal basis of the column space. The available space at batch t is $\mathcal{F}_t = \mathrm{Im}(\boldsymbol{P}_{t-1})$. In exact subspace arithmetic, $\boldsymbol{P}_{t-1}\boldsymbol{K}_p = \boldsymbol{0}$, so the historical penalty in Eq. 5 vanishes when $\boldsymbol{P}$ is replaced by $\boldsymbol{P}_{t-1}$. These identities concern stored keys; subsequent changes to upstream activations can change the keys produced by the same input. The remaining isotropic perturbation penalty assigns equal cost to equal-norm updates, motivating the activation-dependent metric in Section 3.3.

3.2 FUNCTION-PRESERVING FFN REPARAMETERIZATION

We now split the output projection into an anchored branch and an editable branch with identical, frozen key maps (Chen et al., 2015; Wei et al., 2016; Adila et al., 2026). Initializing both output weights to half of the original projection $\boldsymbol{W}_0$ preserves the FFN function:

$$\boldsymbol{W}_{\mathrm{A},0} = \boldsymbol{W}_{\mathrm{E},0} = \tfrac{1}{2}\boldsymbol{W}_0, \qquad \boldsymbol{W}_{\mathrm{A},0}\boldsymbol{k} + \boldsymbol{W}_{\mathrm{E},0}\boldsymbol{k} = \boldsymbol{W}_0\boldsymbol{k}. \tag{8}$$

At batch t, the effective projection is $\boldsymbol{W}_{t-1} = \boldsymbol{W}_{\mathrm{A},t-1} + \boldsymbol{W}_{\mathrm{E},t-1}$. Let $\boldsymbol{\Delta}_{\mathrm{LessEdit},t}$ denote the projected perturbation to this sum. For a fixed effective perturbation, we allocate the branch increments by minimizing their weighted squared norms (Boyd & Vandenberghe, 2004; Nocedal & Wright, 2006):

$$\min_{\boldsymbol{\Delta}_{t,\mathrm{E}},\boldsymbol{\Delta}_{t,\mathrm{A}}} \lambda_{\mathrm{E}}\|\boldsymbol{\Delta}_{t,\mathrm{E}}\|_F^2 + \frac{\lambda_{\mathrm{A}}}{\epsilon^2}\|\boldsymbol{\Delta}_{t,\mathrm{A}}\|_F^2 \quad \text{s.t.} \quad \boldsymbol{\Delta}_{t,\mathrm{E}} + \boldsymbol{\Delta}_{t,\mathrm{A}} = \boldsymbol{\Delta}_{\mathrm{LessEdit},t}, \tag{9}$$

where $\lambda_{\mathrm{E}}, \lambda_{\mathrm{A}}, \epsilon > 0$. The optimality conditions give $\boldsymbol{\Delta}_{t,b} = \alpha_b \boldsymbol{\Delta}_{\mathrm{LessEdit},t}$ for $b \in \{\mathrm{A},\mathrm{E}\}$, with

$$\alpha_{\mathrm{E}} = \frac{\lambda_{\mathrm{A}}}{\lambda_{\mathrm{A}} + \epsilon^2\lambda_{\mathrm{E}}}, \qquad \alpha_{\mathrm{A}} = \frac{\epsilon^2\lambda_{\mathrm{E}}}{\lambda_{\mathrm{A}} + \epsilon^2\lambda_{\mathrm{E}}}, \qquad \lambda_{\mathrm{eff}} = \frac{\lambda_{\mathrm{E}}\lambda_{\mathrm{A}}}{\lambda_{\mathrm{A}} + \epsilon^2\lambda_{\mathrm{E}}}. \tag{10}$$

The minimized cost is $\lambda_{\mathrm{eff}}\|\boldsymbol{\Delta}_{\mathrm{LessEdit},t}\|_F^2$. Choosing $\lambda_{\mathrm{A}}/\epsilon^2 \gg \lambda_{\mathrm{E}}$ assigns most of the perturbation to the editable branch. Because the key maps coincide, the branches reduce algebraically to an effective projection with ridge coefficient λ_{eff}; the allocation leaves $\mathcal{F}_t$ unchanged.

3.3 CAPABILITY-AWARE NULL-SPACE CONSTRAINED EDITING

As the projector determines admissible perturbations, we now select among them using their effect on reference activations. For a key $\boldsymbol{k}$ sampled from the empirical reference distribution, $\mathbb{E}[\boldsymbol{k}\boldsymbol{k}^\top] = \boldsymbol{\Sigma}_0$. The expected squared FFN-output change induced by a perturbation $\boldsymbol{\Delta}$ is

$$\mathbb{E}_{\boldsymbol{k}}\|\boldsymbol{\Delta}\boldsymbol{k}\|_2^2 = \mathrm{tr}(\boldsymbol{\Delta}\boldsymbol{\Sigma}_0\boldsymbol{\Delta}^\top) = \|\boldsymbol{\Delta}\boldsymbol{\Sigma}_0^{1/2}\|_F^2. \tag{11}$$

This quantity distinguishes updates of equal norm by their reference activation energy and serves as a local surrogate for capability preservation (Kirkpatrick et al., 2017; Martens & Grosse, 2015).

Let $\boldsymbol{B}_t \in \mathbb{R}^{d_0 \times r_t}$ be an orthonormal basis of $\mathcal{F}_t$, where $r_t = \dim \mathcal{F}_t$ and $\boldsymbol{P}_{t-1} = \boldsymbol{B}_t \boldsymbol{B}_t^\top$. For $r_t > 0$, projecting $\boldsymbol{\Sigma}_0$ into this space and normalizing its mean eigenvalue gives

$$\widetilde{\boldsymbol{\Sigma}}_t = \boldsymbol{B}_t^\top \boldsymbol{\Sigma}_0 \boldsymbol{B}_t, \qquad c_t = \frac{\operatorname{tr}(\widetilde{\boldsymbol{\Sigma}}_t)}{r_t}, \qquad \widehat{\boldsymbol{\Sigma}}_t = \frac{\widetilde{\boldsymbol{\Sigma}}_t}{c_t}. \tag{12}$$

We refer to the normalized second-moment metric as CapCov. For $r_t > 0$ and $c_t > 0$, incorporating Eq. 11 into the projected editing objective yields

$$\begin{aligned} \min_{\boldsymbol{\Delta}_t} \; & \|(\boldsymbol{W}_{t-1} + \boldsymbol{\Delta}_t \boldsymbol{P}_{t-1})\boldsymbol{K}_t - \boldsymbol{V}_t\|_F^2 + \lambda_{\text{eff}} \|\boldsymbol{\Delta}_t \boldsymbol{P}_{t-1}\|_F^2 \\ & + \frac{\beta}{c_t} \|\boldsymbol{\Delta}_t \boldsymbol{P}_{t-1} \boldsymbol{\Sigma}_0^{1/2}\|_F^2, \end{aligned} \tag{13}$$

where $\beta \geq 0$ controls capability-aware regularization. In Figure 2, the evolving projector constrains changes on stored edit keys, while CapCov weights the remaining admissible directions by their activation-dependent output cost.

To solve Eq. 13, write $\boldsymbol{\Delta}_t \boldsymbol{P}_{t-1} = \boldsymbol{A}\boldsymbol{B}_t^\top$ with $\boldsymbol{A} \in \mathbb{R}^{d_1 \times r_t}$ and define

$$\boldsymbol{R}_t = \boldsymbol{V}_t - \boldsymbol{W}_{t-1}\boldsymbol{K}_t, \qquad \widetilde{\boldsymbol{K}}_t = \boldsymbol{B}_t^\top \boldsymbol{K}_t. \tag{14}$$

For multiple edited layers, $\boldsymbol{R}_t$ is the share of the current target-representation residual assigned to this layer, with local target $\boldsymbol{V}_t = \boldsymbol{W}_{t-1}\boldsymbol{K}_t + \boldsymbol{R}_t$ (Meng et al., 2022b; Fang et al., 2024). The equivalent reduced objective is

$$\boldsymbol{A}_t = \arg\min_{\boldsymbol{A}} \left(\|\boldsymbol{A}\widetilde{\boldsymbol{K}}_t - \boldsymbol{R}_t\|_F^2 + \lambda_{\text{eff}} \|\boldsymbol{A}\|_F^2 + \beta \operatorname{tr}(\boldsymbol{A}\widehat{\boldsymbol{\Sigma}}_t \boldsymbol{A}^\top) \right). \tag{15}$$

Its normal equation is $(\boldsymbol{A}_t\widetilde{\boldsymbol{K}}_t - \boldsymbol{R}_t)\widetilde{\boldsymbol{K}}_t^\top + \lambda_{\text{eff}}\boldsymbol{A}_t + \beta \boldsymbol{A}_t \widehat{\boldsymbol{\Sigma}}_t = \boldsymbol{0}$. Since $\widehat{\boldsymbol{\Sigma}}_t \succeq 0$ and $\lambda_{\text{eff}} > 0$, the unique solution gives the final perturbation

$$\boldsymbol{\Delta}_{\text{LessEdit},t} = \boldsymbol{A}_t \boldsymbol{B}_t^\top = \boldsymbol{R}_t \widetilde{\boldsymbol{K}}_t^\top \left(\widetilde{\boldsymbol{K}}_t \widetilde{\boldsymbol{K}}_t^\top + \lambda_{\text{eff}} \boldsymbol{I} + \beta \widehat{\boldsymbol{\Sigma}}_t \right)^{-1} \boldsymbol{B}_t^\top. \tag{16}$$

We evaluate this expression by a linear solve, update $\boldsymbol{W}_{b,t} = \boldsymbol{W}_{b,t-1} + \alpha_b \boldsymbol{\Delta}_{\text{LessEdit},t}$, and reserve the current key directions through $\boldsymbol{Q}_t = \operatorname{orth}(\boldsymbol{P}_{t-1}\boldsymbol{K}_t)$ and $\boldsymbol{P}_t = \boldsymbol{P}_{t-1} - \boldsymbol{Q}_t \boldsymbol{Q}_t^\top$. The reservation depends on the projected keys, while the capability metric determines the perturbation within the current space. When $r_t = 0$, the update is zero; when $c_t \leq 10^{-12}$, we set $\widehat{\boldsymbol{\Sigma}}_t = \boldsymbol{I}$ in Eq. 15. In particular, an exact null space of $\boldsymbol{\Sigma}_0$ has zero reference energy; the anisotropic penalty exploits the nonzero energy retained by near-null projection. Setting $\beta = 0$ and $\lambda_{\text{eff}} = 1$ recovers EvoEdit's isotropic projected objective.

4 Experiment

4.1 Experimental Setup

We outline the experimental setup below, with full details in Appendix B.

Datasets, metrics, and evaluation. We perform 3,000 sequential edits on the base model (LLaMA-3-8B-Instruct) using three standard editing datasets: ZsRE (Levy et al., 2017), CounterFact (Meng et al., 2022a), and WikiBigEdit (Thede et al., 2025). Following prior work (Yang et al., 2025a; Ikram et al., 2026), we report two edit metrics: *reliability* (or efficacy) asks whether the edited model produces an acceptable answer to a given edit query, and *generalization* asks whether the effects of an edit extend to semantically related contexts. To measure capability degradation, we benchmark edited and base models on diverse tasks: MMLU (Hendrycks et al., 2020), IFEval (Zhou et al., 2023), TruthfulQA (Lin et al., 2022), ARC-Challenge (Clark et al., 2018), and GSM8k (Cobbe et al., 2021).

Because an edited LLM should express new knowledge through natural generation rather than teacher-forced likelihood (Ikram et al., 2026), we follow the WILD protocol (Yang et al., 2025a), which uses autoregressive decoding and LLM-as-a-judge evaluation. This avoids the content and length leakage in likelihood-based evaluation, which can overestimate editing performance (Yang et al., 2025a). We implement WILD with EasyEdit (Wang et al., 2024b) and evaluate prompts both with

Table 1: Comparison of LESSEDIT with existing methods on editing LLaMA-3-8B-Instruct. *Rel* and *Gen* denote reliability and generalization. We edit 3,000 samples from three datasets, evaluate edits with WILD, and measure base capability on five benchmarks. Best values are in bold.

Data	Method	Edited Capabilities				Base Capabilities				
		QA Context		No Context						
		Rel	Gen	Rel	Gen	MMLU	IFEval	TruthfulQA	ARC-C	GSM8K
ZsRE	LLaMA-3-8B-Instruct	2.2	1.9	2.9	2.2	69.5	69.3	50.7	**56.0**	73.5
	MEMIT	0.0	0.0	0.0	0.0	24.6	13.7	51.7	29.5	0.0
	Adam-NSCL	19.6	19.1	2.7	3.2	69.0	32.1	50.7	42.5	36.0
	LocBF-FT	69.3	59.5	26.0	22.3	**69.5**	70.2	51.6	51.5	**75.5**
	UltraEdit	19.8	16.2	23.7	18.8	69.3	**73.0**	51.0	52.0	72.0
	MEND	0.0	0.0	0.0	0.0	22.9	18.2	0.0	26.0	0.0
	FT	3.6	3.5	0.9	1.2	68.8	19.4	52.8	40.5	6.5
	LoRA	2.3	2.0	2.0	1.2	68.7	58.7	**53.3**	53.0	64.5
	AlphaEdit	70.5	59.4	39.6	31.6	53.3	51.2	44.1	45.5	47.0
	WISE	2.3	1.8	2.8	2.2	66.4	17.3	52.6	46.5	0.0
	EvoEdit	77.4	64.9	60.2	48.1	64.7	68.8	48.9	52.0	65.5
	CrispEdit	69.8	62.0	65.6	55.6	69.2	69.8	52.1	52.5	75.0
	LESSEDIT	**82.4**	**66.5**	**74.3**	**55.7**	63.2	65.5	48.0	53.0	69.0
CounterFact	LLaMA-3-8B-Instruct	1.2	1.0	0.3	0.6	**69.5**	69.3	50.7	**56.0**	73.5
	MEMIT	0.0	0.0	0.0	0.0	24.6	13.9	51.3	30.5	0.0
	Adam-NSCL	15.6	7.9	1.6	1.9	68.8	22.1	**57.3**	38.5	9.0
	LocBF-FT	61.4	41.4	11.1	13.3	69.5	64.8	51.3	49.0	**74.0**
	UltraEdit	19.7	12.1	12.4	9.9	69.4	**69.5**	49.0	55.5	73.0
	MEND	0.0	0.0	0.0	0.0	22.9	18.2	0.0	26.0	0.0
	FT	19.1	10.6	1.3	2.2	33.4	20.4	51.3	31.5	0.0
	LoRA	6.8	4.6	2.1	1.4	69.4	63.0	53.2	50.0	73.0
	AlphaEdit	74.8	59.0	53.0	45.8	48.3	33.4	37.8	43.5	41.5
	WISE	1.2	1.0	0.3	0.7	66.5	19.8	52.9	48.5	4.5
	EvoEdit	82.5	66.6	60.2	50.3	64.5	65.6	45.8	54.0	71.0
	CrispEdit	66.2	44.4	33.6	25.2	69.4	67.7	50.5	53.5	68.0
	LESSEDIT	**83.4**	**69.9**	**61.7**	**50.7**	64.8	64.4	47.6	54.5	69.0
WikiBigEdit	LLaMA-3-8B-Instruct	9.7	9.2	16.5	15.8	**69.5**	69.3	50.7	56.0	73.5
	MEMIT	0.0	0.0	0.0	0.0	24.6	0.0	50.9	26.0	0.0
	Adam-NSCL	12.1	11.8	4.0	4.1	68.9	45.4	48.4	42.5	37.5
	LocBF-FT	51.9	48.5	17.0	16.2	69.2	70.7	52.1	**58.0**	**74.5**
	UltraEdit	59.8	55.5	52.9	51.1	69.1	67.0	51.0	54.0	70.0
	MEND	0.0	0.0	0.0	0.0	22.9	18.2	0.0	26.0	0.0
	FT	13.4	12.6	1.8	1.5	68.1	34.5	51.8	43.0	29.5
	LoRA	12.9	11.7	10.8	7.7	69.1	66.1	51.4	53.0	73.0
	AlphaEdit	72.6	67.1	69.3	62.9	55.2	54.1	46.3	52.0	48.0
	WISE	9.9	9.6	16.4	15.8	68.3	23.7	**52.2**	49.0	40.5
	EvoEdit	81.9	73.2	82.1	73.1	65.4	**71.1**	48.5	54.0	65.0
	CrispEdit	66.0	57.9	40.8	39.2	69.4	69.8	50.7	53.0	73.0
	LESSEDIT	**83.9**	**76.8**	**87.8**	**79.3**	64.5	69.0	49.5	53.5	69.0

and without QA context (see Appendix B.3). Appendix C.1 compares teacher-forced results with this autoregressive evaluation.

Baselines. We compare LESSEDIT against several model editing baselines, including Fine-Tuning (FT) (Zhu et al., 2020), Adam-NSCL (Wang et al., 2021), MEND (Mitchell et al., 2021), LoRA (Hu et al., 2022), MEMIT (Meng et al., 2022b), WISE (Wang et al., 2024a), AlphaEdit (Fang et al., 2024), LocBF-FT (Yang et al., 2025b), UltraEdit (Gu et al., 2025), EvoEdit (Lyu et al., 2025), and CrispEdit (Ikram et al., 2026). Among them, AlphaEdit and EvoEdit are the most direct comparisons, as they also rely on null-space constraints to reduce interference during sequential editing. All baselines are evaluated on the same datasets and under the same WILD protocol whenever applicable. Additional details on implementation of LESSEDIT and baseline configurations are provided in Appendix B.4 and B.2.

4.2 MAIN RESULTS

Editing Performance. Table 1 shows that LESSEDIT achieves the highest WILD reliability and generalization on all three datasets under both prompt formats after 3,000 sequential edits. Compared with EvoEdit, which also uses evolving null-space constraints but regularizes updates isotropically,

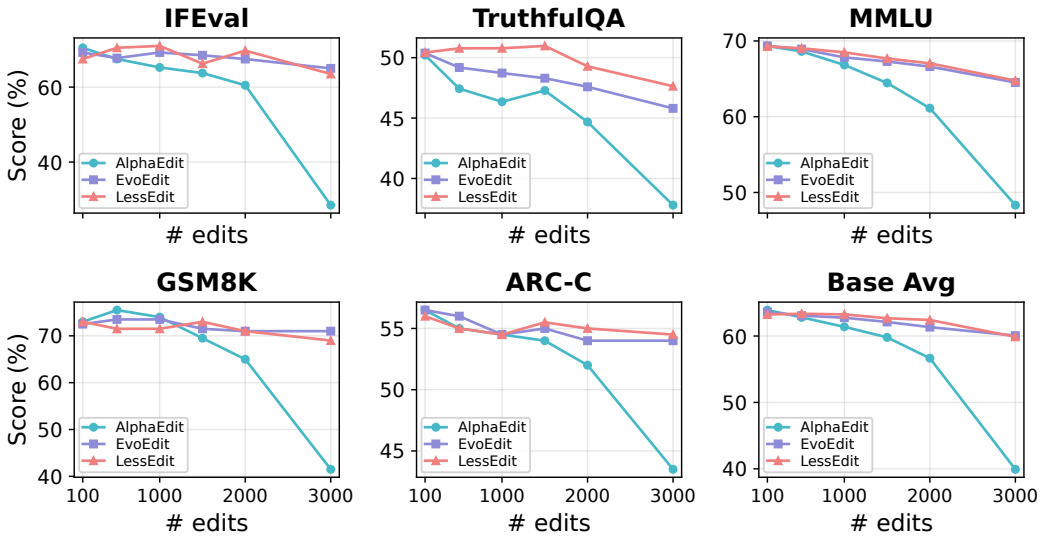


Figure 3: Base-capability scaling over the editing horizon. LESSEDIT remains close to EvoEdit across the five benchmarks, while AlphaEdit shows a sharper decline as edits accumulate.

Table 2: Ablation study on ZsRE after 3,000 edits. *USD*, *CapCov*, *Single*, and *Isotropic* denote used-subspace deflation, capability covariance, the single-tensor variant, and the isotropic variant that replaces the capability covariance with an identity matrix, respectively.

Variant	Edited Capabilities					Base Capabilities					
	QA Context		No Context		Avg						
	Rel	Gen	Rel	Gen		MMLU	IFEval	TruthfulQA	ARC-C	GSM8K	Avg
Full	82.4	66.5	74.3	55.7	69.7	63.2	65.5	48.0	53.0	69.0	59.7
w/o USD	0.0	0.0	0.0	0.0	0.0	24.6	5.5	50.7	28.0	0.0	21.8
w/o CapCov	82.1	65.5	60.9	44.5	63.3	54.9	52.3	42.0	43.0	46.0	47.6
Isotropic	79.8	65.3	62.7	47.7	63.9	60.6	62.3	47.3	53.5	61.5	57.0
Single	82.1	66.0	62.8	49.4	65.1	63.5	66.5	45.8	54.5	64.0	58.8

LESSEDIT improves these scores by an average of 4.3 percentage points. The gains average 5.9 points when the original dataset prompts are used without the QA template, compared with 2.7 points with the template. On ZsRE, reliability under No Context reaches 74.3%, compared with 65.6% for CrispEdit and 60.2% for EvoEdit. The improvement also extends to Qwen2.5-7B-Instruct, where LESSEDIT achieves the highest mean WILD score on each dataset (Appendix C.3). Representative generations in Appendix C.6 illustrate how these improvements appear in the model's answers.

Base-Capability Preservation. Table 1 evaluates the edited models on five benchmarks of general capabilities. After 3,000 sequential edits, LESSEDIT averages 60.3 across these benchmarks and the three editing datasets, compared with 63.8 for the unedited model. EvoEdit achieves a similar average of 60.3, while AlphaEdit drops to 46.7. On CounterFact, for example, LESSEDIT scores 64.4 on IFEval, compared with 33.4 for AlphaEdit. Figure 3 shows that LESSEDIT remains close to EvoEdit as edits accumulate, whereas AlphaEdit declines more sharply toward the end of the sequence. Together with the editing results, this shows that LESSEDIT improves reliability and generalization while retaining average base-capability performance comparable to EvoEdit.

Ablation Study. Table 2 shows ablations after 3,000 sequential edits on ZsRE. Without used-subspace deflation, which excludes historical edit-key directions, all four WILD scores fall to zero and the base-capability average drops to 21.8. Removing CapCov while retaining the original norm penalty reduces the mean WILD score from 69.7 to 63.3 and the base-capability average from 59.7 to 47.6. Replacing CapCov with an identity matrix instead raises these averages to 63.9 and 57.0, but both remain below the full model. The advantage over this isotropic control supports using activation

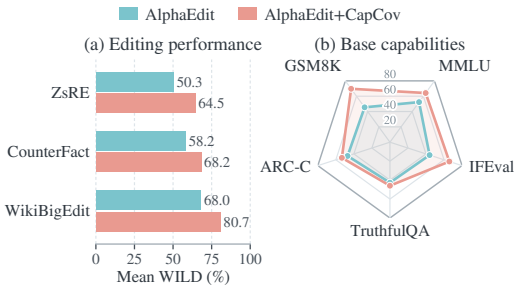


Figure 4: CapCov transfer after 3K LLaMA edits. (a) Mean WILD. (b) Averaged base scores (%).

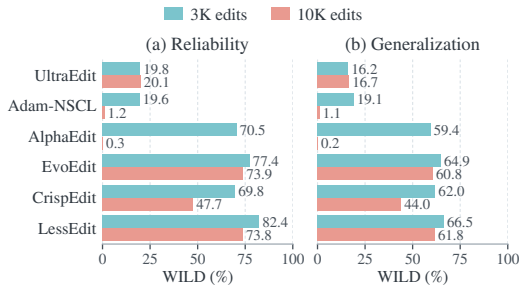


Figure 5: QA-context performance after 3K and 10K ZsRE edits.

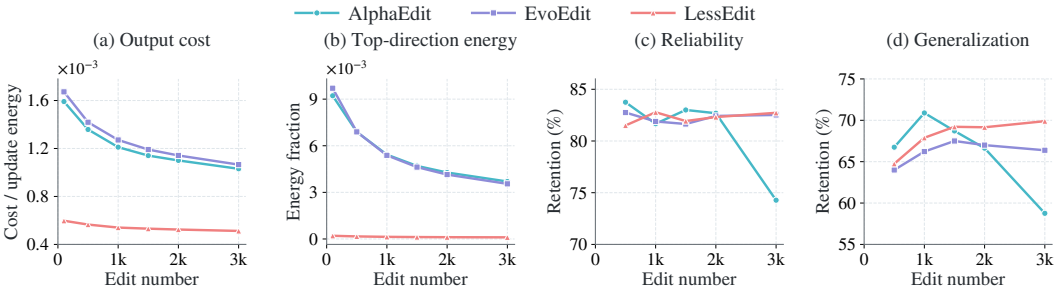


Figure 6: Update geometry (a,b) and previous-edit retention (c,d). Lower is better in (a,b).

statistics to guide update selection within the free subspace, rather than relying on an additional norm penalty alone. The single-tensor variant achieves similar scores with the QA template but lower reliability and generalization when prompted with the question alone. Branch equivalence and deployment costs are examined in Appendices E.1 and E.2.

Transferring CapCov to AlphaEdit. To test whether CapCov also benefits a static null-space editor, we add it to AlphaEdit while retaining its projector, historical-key penalty, and ridge coefficient. After 3,000 edits on LLaMA-3-8B-Instruct, this variant improves both editing performance and base capabilities on all three datasets (Figure 4). The gains average 12.3 percentage points in WILD and 13.5 points in base capability. Since the added penalty changes both directional weighting and regularization strength, we examine directional effects separately at fixed model states in Appendix D.1. Appendix D.2 reports the transfer results for each dataset.

Scaling to 10K Edits. We further evaluate LESSEDIT after 10,000 sequential ZsRE edits on LLaMA-3-8B-Instruct. Figure 5 compares performance with the QA template after 3,000 and 10,000 edits. At 10,000 edits, LESSEDIT achieves 73.8% reliability and 61.8% generalization, close to EvoEdit's 73.9% and 60.8%. Averaged over reliability and generalization under both prompt formats, its WILD score reaches 68.3%, exceeding EvoEdit by 2.5 percentage points. Its average score on the five base-capability benchmarks is 54.1, exceeding EvoEdit by 11.8 points. Appendix C.4 provides full results and analyzes the remaining free-subspace dimension and cumulative perturbation.

4.3 DISCUSSION

Capability-covariance diagnostics. For cumulative effective updates $\boldsymbol{D}_l = \boldsymbol{W}_{l,t} - \boldsymbol{W}_{l,0}$, we measure the normalized FFN output cost $\sum_l \mathrm{tr}(\boldsymbol{D}_l \boldsymbol{\Sigma}_{0,l} \boldsymbol{D}_l^\top) / \sum_l \|\boldsymbol{D}_l\|_F^2$ and the leading-eigenspace energy fraction $\sum_l \|\boldsymbol{D}_l \boldsymbol{U}_{\mathrm{top},l}\|_F^2 / \sum_l \|\boldsymbol{D}_l\|_F^2$, where $\boldsymbol{U}_{\mathrm{top},l}$ contains the top 64 eigenvectors of $\boldsymbol{\Sigma}_{0,l}$. Both remain lower for LESSEDIT throughout editing (Figure 6(a,b)). At 3,000 edits, its normalized output cost is roughly half that of either baseline, and its leading-eigenspace energy fraction is below 3% of theirs. This is consistent with CapCov favoring directions with smaller FFN output changes on general-domain activations per unit update energy. Appendix D.3 examines sensitivity to the sample size and source of the activation statistics.

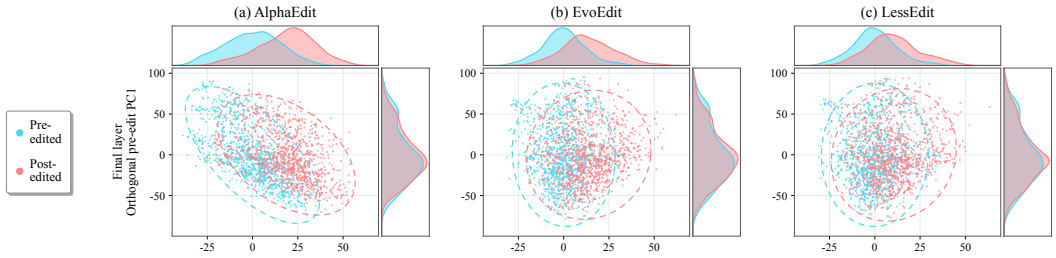


Figure 7: Final-layer representations before and after 3,000 ZsRE edits. Each method is shown in its own drift-aligned projection, with marginal densities and dashed 95% covariance ellipses.

Previous-edit retention. Figure 6(c,d) evaluates earlier CounterFact edits with the QA template. LESSEDIT and EvoEdit, both of which exclude historical edit-key directions, maintain stable reliability, whereas AlphaEdit declines near the end of the sequence. After 3,000 edits, LESSEDIT achieves reliability comparable to EvoEdit (82.7% versus 82.5%) and 3.5 percentage points higher generalization. Appendix C.2 reports the corresponding CounterFact editing trends.

Hidden-representation drift. We complement the local FFN diagnostics by examining final-layer representations at the last token of 1,000 held-out CounterFact prompts before and after 3,000 ZsRE edits (Figure 7). The centroid shift in the original representation space, normalized by the pre-edit centroid norm, is 12.2% for LESSEDIT, compared with 16.0% for EvoEdit and 21.7% for AlphaEdit. The projected distributions of LESSEDIT and EvoEdit show similar dispersion.

5 RELATED WORK

Knowledge editing aims to update specific factual associations in large language models while preserving unrelated behaviors. Early parameter-editing methods such as KNOWLEDGEEDITOR (De Cao et al., 2021), MEND (Mitchell et al., 2021), ROME (Meng et al., 2022a), and MEMIT (Meng et al., 2022b) edit model behavior by learning edit predictors or directly modifying localized feed-forward parameters; ROME interprets FFNs as key–value memories and applies rank-one updates, while MEMIT extends this idea to batched edits across multiple layers. These methods are effective for isolated or small-batch edits, but sequential editing introduces accumulated interference between new edits, previous edits, and preserved knowledge. Recent lifelong or continual editing methods, including WISE (Wang et al., 2024a), UltraEdit (Gu et al., 2025), and CrispEdit (Ikram et al., 2026), address this editing–preservation trade-off through auxiliary memories, online feature statistics, or constrained low-curvature updates. Closest to our work are null-space-based editors. AlphaEdit (Fang et al., 2024) projects updates into the null space of preserved key representations to reduce disruption, while EvoEdit (Lyu et al., 2025) and LangEdit (Sun et al., 2025) further adapt protected subspaces to mitigate interference from previous edits. These methods mainly define which directions should be avoided; after obtaining the remaining writable subspace, they typically rely on isotropic penalties to choose an update. In contrast, LESSEDIT keeps the same non-interference principle but treats free directions as unequally costly, using general-domain activation second moments as a capability-aware metric for selecting lower-impact directions within the current free subspace.

6 CONCLUSION

In this paper, we introduced LESSEDIT, a capability-aware free-subspace editor for sequential model editing. Null-space constraints restrict the available update directions, yet feasible updates can still differ in their effects on FFN outputs. Even for the same local edit response, the minimum-norm update need not produce the smallest output change on general-domain activations. LESSEDIT accounts for this distinction through a regularizer derived from the projected activation second moment. Historical-key deflation limits interference with stored edit keys, while CapCov guides update selection within the remaining space. The resulting objective retains a closed-form effective solution and complements parameter-norm regularization with a local surrogate for capability preservation.

LLM USAGE STATEMENT

This work used LLM assistance for literature search, language editing, coding, and mathematical formulation that was subsequently inspected and refined by hand. All LLM-assisted code and writing were validated by the authors, who take full responsibility for the final manuscript and released code. The use of LLMs to evaluate generated texts is documented separately in the experimental protocol.

REPRODUCIBILITY STATEMENT

We evaluate LESSEDIT on ZsRE, CounterFact, and WikiBigEdit, with five additional benchmarks assessing base capabilities. The main text and appendix describe the models, edited layers, target-value optimization, activation-statistic estimation, projector thresholds, regularization settings, and computational resources. We also provide the generation settings, prompt formats, LLM-based evaluation procedure, and benchmark scoring methods. Code and data are available at `https://anonymous.4open.science/r/LessEdit`.

ETHICS STATEMENT

This work studies sequential knowledge editing in large language models using existing benchmarks. Model editing can help correct outdated information, but it can also be misused to introduce false or misleading claims. Counterfactual targets in our experiments are used to assess editing behavior and should not be interpreted as verified factual corrections. Although LESSEDIT aims to limit unintended changes, its capability-aware regularizer is a local surrogate and does not guarantee preservation of unrelated knowledge, general capabilities, or safety behavior. Benchmark scores and LLM-based judgments do not establish the factual validity of arbitrary edit requests or the safety of edited models. Practical use requires verification of the requested changes and evaluation of their effects beyond the edited facts.

REFERENCES

Dyah Adila, Hanna Mazzawi, Benoit Dherin, and Xavier Gonzalvo. Grow, don't overwrite: Fine-tuning without forgetting. *arXiv preprint arXiv:2603.08647*, 2026.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. Net2net: Accelerating learning via knowledge transfer. *arXiv preprint arXiv:1511.05641*, 2015.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 6491–6506, 2021.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Shi Jie, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Alphaedit: Null-space constrained knowledge editing for language models. *arXiv preprint arXiv:2410.02355*, 2024.

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, 2021.

Xiaojie Gu, Guangxu Chen, Jungang Li, Jia-Chen Gu, Xuming Hu, and Kai Zhang. Ultraedit: Training-, subject-, and memory-free lifelong editing in large language models. *arXiv e-prints*, pp. arXiv–2505, 2025.

Akshat Gupta, Sidharth Baskaran, and Gopala Anumanchipalli. Rebuilding rome: Resolving model collapse during sequential model editing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21738–21744, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Zarif Ikram, Arad Firouzkouhi, Stephen Tu, Mahdi Soltanolkotabi, and Paria Rashidinejad. Crispedit: Low-curvature projections for scalable non-destructive llm editing. *arXiv preprint arXiv:2602.15823*, 2026.

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

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114 (13):3521–3526, 2017.

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 (CoNLL 2017)*, pp. 333–342, 2017.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 3214–3252, 2022.

Shuxin Liu and Ou Wu. Metake: Meta-learning aligned knowledge editing via bi-level optimization. *arXiv preprint arXiv:2603.12677*, 2026.

Sicheng Lyu, Yu Gu, Xinyu Wang, Jerry Huang, Sitao Luan, Yufei Cui, Xiao-Wen Chang, and Peng Lu. Evoedit: Evolving null-space alignment for robust and efficient knowledge editing. *arXiv preprint arXiv:2510.13851*, 2025.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pp. 2408–2417. PMLR, 2015.

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, 2022a.

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

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*, 2021.

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

Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 2006.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 2463–2473, 2019.

Wei Sun, Tingyu Qu, Mingxiao Li, Jesse Davis, and Marie Francine Moens. Mitigating negative interference in multilingual knowledge editing through null-space constraints. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 8796–8810, 2025.

Lukas Thede, Karsten Roth, Matthias Bethge, Zeynep Akata, and Tom Hartvigsen. Wikibigedit: Understanding the limits of lifelong knowledge editing in llms. *arXiv preprint arXiv:2503.05683*, 2025.

William Timkey and Marten Van Schijndel. All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4527–4546, 2021.

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. *Advances in Neural Information Processing Systems*, 37:53764–53797, 2024a.

Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, et al. 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 (Volume 3: System Demonstrations)*, pp. 82–93, 2024b.

Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 184–193, 2021.

Tao Wei, Changhu Wang, Yong Rui, and Chang Wen Chen. Network morphism. In *International conference on machine learning*, pp. 564–572. PMLR, 2016.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45, 2020.

Wanli Yang, Fei Sun, Xinyu Ma, Xun Liu, Dawei Yin, and Xueqi Cheng. The butterfly effect of model editing: Few edits can trigger large language models collapse. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 5419–5437, 2024.

Wanli Yang, Fei Sun, Jiajun Tan, Xinyu Ma, Qi Cao, Dawei Yin, Huawei Shen, and Xueqi Cheng. The mirage of model editing: Revisiting evaluation in the wild. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15336–15354, 2025a.

Wanli Yang, Rui Tang, Hongyu Zang, Du Su, Qi Cao, Jingang Wang, Huawei Shen, Xueqi Cheng, and Fei Sun. Fine-tuning done right in model editing. *arXiv preprint arXiv:2509.22072*, 2025b.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10222–10240, 2023.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*, 2020.

Appendix

A Proofs **15**
- A.1 Null-Space Projection and Historical-Key Preservation . . . 15
- A.2 Function-Preserving Reparameterization and Deployment . . . 16
- A.3 Optimal Branch Allocation . . . 16
- A.4 Capability-Aware Reduced Solution . . . 16
- A.5 Comparison at a Fixed Editing Response . . . 17

B Experimental Setup **17**
- B.1 Datasets . . . 17
- B.2 Baselines . . . 18
- B.3 Evaluation Protocol . . . 18
- B.4 Implementation Details . . . 19

C Additional Editing Results **20**
- C.1 Teacher-Forced Evaluation . . . 20
- C.2 Sequential Editing on CounterFact . . . 20
- C.3 Generalization to Qwen . . . 21
- C.4 Scaling to 10,000 Edits . . . 21
- C.5 Multi-Hop Questions . . . 22
- C.6 Case Study . . . 22

D Capability-Metric Analysis **24**
- D.1 Direction Selection at a Fixed Model State . . . 24
- D.2 CapCov Transfer to AlphaEdit . . . 24
- D.3 Sensitivity to Reference Statistics . . . 25

E Branch Reparameterization and Efficiency **26**
- E.1 Branch Interventions and Function Preservation . . . 26
- E.2 Efficiency . . . 27

F Limitations and Future Work **27**

A PROOFS

We use the notation of Sections 2 and 3. At a fixed edited layer, $\boldsymbol{W}_{t-1} \in \mathbb{R}^{d_1 \times d_0}$ denotes the effective projection before batch t. The current keys and target values are $\boldsymbol{K}_t$ and $\boldsymbol{V}_t$, and $\boldsymbol{K}_p$ collects stored keys from previous batches. The columns of $\boldsymbol{B}_t \in \mathbb{R}^{d_0 \times r_t}$ form an orthonormal basis of $\mathcal{F}_t = \mathrm{Im}(\boldsymbol{P}_{t-1})$. All layerwise statements below hold with these keys and targets fixed.

A.1 NULL-SPACE PROJECTION AND HISTORICAL-KEY PRESERVATION

Proposition 1 (Reference-key preservation)**.** *Let $\boldsymbol{\Sigma}_0 = \boldsymbol{K}_0\boldsymbol{K}_0^\top/n_0$, and let $\boldsymbol{P}$ be the spectral projector defined in Eq. 6. Then* $\mathrm{Null}(\boldsymbol{\Sigma}_0) = \mathrm{Null}(\boldsymbol{K}_0^\top)$*. Every perturbation satisfying $\boldsymbol{\Delta} = \boldsymbol{\Delta}\boldsymbol{P}$ obeys*

$$\frac{\|\boldsymbol{\Delta}\boldsymbol{K}_0\|_F^2}{n_0} \le \tau\|\boldsymbol{\Delta}\|_F^2. \tag{17}$$

If $\boldsymbol{P}$ retains only zero eigenvalues, the left-hand side is zero.

Proof. For any $\boldsymbol{x} \in \mathbb{R}^{d_0}$, $\boldsymbol{x}^\top\boldsymbol{\Sigma}_0\boldsymbol{x} = \|\boldsymbol{K}_0^\top\boldsymbol{x}\|_2^2/n_0$, which proves the null-space identity. By construction, $\boldsymbol{0} \preceq \boldsymbol{P}\boldsymbol{\Sigma}_0\boldsymbol{P} \preceq \tau\boldsymbol{P}$. Thus,

$$\frac{\|\boldsymbol{\Delta}\boldsymbol{K}_0\|_F^2}{n_0} = \mathrm{tr}(\boldsymbol{\Delta}\boldsymbol{P}\boldsymbol{\Sigma}_0\boldsymbol{P}\boldsymbol{\Delta}^\top) \le \tau\,\mathrm{tr}(\boldsymbol{\Delta}\boldsymbol{P}\boldsymbol{\Delta}^\top) = \tau\|\boldsymbol{\Delta}\|_F^2.$$

The exact-null case follows by replacing $\boldsymbol{P}\boldsymbol{\Sigma}_0\boldsymbol{P}$ with zero. □

Proposition 2 (Sequential deflation)**.** *With exact column-space extraction in Eq. 7, $\boldsymbol{P}_{t-1}$ is the orthogonal projector onto*

$$\mathcal{F}_t = \mathrm{Im}(\boldsymbol{P}) \cap \mathrm{Null}(\boldsymbol{K}_p^\top). \tag{18}$$

Consequently, $\boldsymbol{\Delta}_{\mathrm{LessEdit},t}\boldsymbol{K}_p = \boldsymbol{0}$, and Eq. 17 also holds for $\boldsymbol{\Delta}_{\mathrm{LessEdit},t}$.

Proof. The claim holds initially because $\boldsymbol{P}_0 = \boldsymbol{P}$ and the historical set is empty. Suppose it holds before reserving $\boldsymbol{K}_t$, and set $\boldsymbol{Q}_t = \mathrm{orth}(\boldsymbol{P}_{t-1}\boldsymbol{K}_t)$. Its columns lie in $\mathcal{F}_t$, so $\boldsymbol{P}_{t-1}\boldsymbol{Q}_t = \boldsymbol{Q}_t$ and $\boldsymbol{Q}_t^\top\boldsymbol{Q}_t = \boldsymbol{I}$. Therefore,

$$(\boldsymbol{P}_{t-1} - \boldsymbol{Q}_t\boldsymbol{Q}_t^\top)^2 = \boldsymbol{P}_{t-1} - \boldsymbol{Q}_t\boldsymbol{Q}_t^\top.$$

The matrix is also symmetric and projects onto $\mathcal{F}_t \cap \mathrm{Im}(\boldsymbol{Q}_t)^\perp$. For $\boldsymbol{x} \in \mathcal{F}_t$, $\boldsymbol{x}^\top\boldsymbol{P}_{t-1}\boldsymbol{K}_t = \boldsymbol{x}^\top\boldsymbol{K}_t$. Hence this intersection equals $\mathcal{F}_t \cap \mathrm{Null}(\boldsymbol{K}_t^\top)$, proving the induction step. Finally, $\boldsymbol{\Delta}_{\mathrm{LessEdit},t} = \boldsymbol{A}_t\boldsymbol{B}_t^\top$ annihilates $\boldsymbol{K}_p$ and has row space contained in $\mathrm{Im}(\boldsymbol{P})$. □

Numerical and functional scope. Thresholded column-space extraction gives approximate historical-key preservation. For any numerical basis,

$$\|\boldsymbol{\Delta}_{\mathrm{LessEdit},t}\boldsymbol{K}_p\|_F \le \|\boldsymbol{A}_t\|_2\,\|\boldsymbol{B}_t^\top\boldsymbol{K}_p\|_F. \tag{19}$$

The bound concerns stored activations. Upstream updates can change these activations, so the same prompt may still produce different predictions.

Corollary 3 (Available dimension)**.** *In exact subspace arithmetic,*

$$r_{t+1} = r_t - \mathrm{rank}(\boldsymbol{P}_{t-1}\boldsymbol{K}_t). \tag{20}$$

For fixed projected keys, this reduction is independent of the branch allocation and the capability regularizer.

Proof. The reservation removes exactly $\mathrm{Im}(\boldsymbol{Q}_t)$ from $\mathcal{F}_t$. Its dimension is $\mathrm{rank}(\boldsymbol{P}_{t-1}\boldsymbol{K}_t)$. □

A.2 Function-Preserving Reparameterization and Deployment

Proposition 4 (Branch equivalence)**.** *Suppose the anchored and editable branches use identical frozen key maps. For every layer input, their sum equals a single projection with weight*

$$\boldsymbol{W}_t = \boldsymbol{W}_{\mathrm{A},t} + \boldsymbol{W}_{\mathrm{E},t}. \tag{21}$$

The initialization in Eq. 8 preserves the original model function. Merging after editing also preserves the edited model function in exact arithmetic.

Proof. For every key $\boldsymbol{k}$, $\boldsymbol{W}_{\mathrm{A},t}\boldsymbol{k} + \boldsymbol{W}_{\mathrm{E},t}\boldsymbol{k} = (\boldsymbol{W}_{\mathrm{A},t} + \boldsymbol{W}_{\mathrm{E},t})\boldsymbol{k}$. At initialization, the sum is $\boldsymbol{W}_0$. Replacing each edited FFN by its merged form therefore preserves every layer output and, inductively, the network output. If an output bias is present, its branch contributions must sum to the original bias. The duplicated gate and up projections can be removed because their maps coincide. □

Choice of initialization. Any initial split whose sum is $\boldsymbol{W}_0$ preserves the function. The equal split additionally minimizes the unweighted sum of squared branch norms, since

$$\|\boldsymbol{W}_{\mathrm{A},0}\|_F^2 + \|\boldsymbol{W}_{\mathrm{E},0}\|_F^2 = \tfrac{1}{2}\|\boldsymbol{W}_0\|_F^2 + 2\|\boldsymbol{W}_{\mathrm{A},0} - \tfrac{1}{2}\boldsymbol{W}_0\|_F^2.$$

This initialization property is separate from the asymmetric penalty on subsequent increments.

A.3 Optimal Branch Allocation

Proposition 5 (Minimum branch cost)**.** *For a fixed effective perturbation $\boldsymbol{\Delta}_{\mathrm{LessEdit},t}$, Eq. 9 has the unique solution $\boldsymbol{\Delta}_{t,b} = \alpha_b \boldsymbol{\Delta}_{\mathrm{LessEdit},t}$, with coefficients from Eq. 10. The minimum cost is $\lambda_{\mathrm{eff}}\|\boldsymbol{\Delta}_{\mathrm{LessEdit},t}\|_F^2$.*

Proof. Eliminate $\boldsymbol{\Delta}_{t,\mathrm{A}}$ using the sum constraint. The derivative with respect to $\boldsymbol{\Delta}_{t,\mathrm{E}}$ gives

$$\lambda_{\mathrm{E}}\boldsymbol{\Delta}_{t,\mathrm{E}} - \frac{\lambda_{\mathrm{A}}}{\epsilon^2}(\boldsymbol{\Delta}_{\mathrm{LessEdit},t} - \boldsymbol{\Delta}_{t,\mathrm{E}}) = \mathbf{0}.$$

Solving yields the stated coefficients. Substitution into the objective gives $\lambda_{\mathrm{E}}\alpha_{\mathrm{E}}^2 + (\lambda_{\mathrm{A}}/\epsilon^2)\alpha_{\mathrm{A}}^2 = \lambda_{\mathrm{eff}}$. Both penalties are positive, ensuring uniqueness. □

With fixed coefficients across batches, $\boldsymbol{W}_{b,T} - \boldsymbol{W}_{b,0} = \alpha_b \sum_{t=1}^{T} \boldsymbol{\Delta}_{\mathrm{LessEdit},t}$. A small anchored-branch change follows from its allocation coefficient and need not imply a small effective perturbation. The branches support separate interventions while remaining equivalent to one projection.

A.4 Capability-Aware Reduced Solution

Proposition 6 (Reference-output cost)**.** *For $\boldsymbol{\Delta} = \boldsymbol{A}\boldsymbol{B}_t^\top$ and the empirical reference distribution,*

$$\mathbb{E}_{\boldsymbol{k}}\|\boldsymbol{\Delta}\boldsymbol{k}\|_2^2 = \mathrm{tr}(\boldsymbol{A}\widetilde{\boldsymbol{\Sigma}}_t\boldsymbol{A}^\top). \tag{22}$$

When $c_t > 0$, division by c_t preserves the eigendirections and sets the mean eigenvalue of $\widehat{\boldsymbol{\Sigma}}_t$ to one.

Proof. Linearity of expectation and trace gives

$$\mathbb{E}_{\boldsymbol{k}}\,\mathrm{tr}(\boldsymbol{A}\boldsymbol{B}_t^\top\boldsymbol{k}\boldsymbol{k}^\top\boldsymbol{B}_t\boldsymbol{A}^\top) = \mathrm{tr}(\boldsymbol{A}\boldsymbol{B}_t^\top\boldsymbol{\Sigma}_0\boldsymbol{B}_t\boldsymbol{A}^\top).$$

The normalization follows from $c_t = \mathrm{tr}(\widetilde{\boldsymbol{\Sigma}}_t)/r_t$. □

This exact identity for fixed reference keys defines a local capability surrogate. It omits downstream Jacobians and the output-loss curvature modeled by CrispEdit (Ikram et al., 2026). In an exact null space of the same $\boldsymbol{\Sigma}_0$, $\widetilde{\boldsymbol{\Sigma}}_t = \mathbf{0}$. The anisotropic signal therefore comes from the nonzero reference energy retained by near-null projection.

Theorem 7 (Unique effective update). *Assume $r_t > 0$, $\lambda_{\text{eff}} > 0$, $\beta \geq 0$, and $\widehat{\boldsymbol{\Sigma}}_t \succeq 0$. The reduced objective in Eq. 15 is strictly convex and has the unique minimizer*

$$\boldsymbol{A}_t = \boldsymbol{R}_t \widetilde{\boldsymbol{K}}_t^\top \boldsymbol{G}_t^{-1}, \qquad \boldsymbol{G}_t = \widetilde{\boldsymbol{K}}_t \widetilde{\boldsymbol{K}}_t^\top + \lambda_{\text{eff}} \boldsymbol{I} + \beta \widehat{\boldsymbol{\Sigma}}_t. \tag{23}$$

Consequently, $\boldsymbol{\Delta}_{\text{LessEdit},t} = \boldsymbol{A}_t \boldsymbol{B}_t^\top$ is unique.

Proof. For every nonzero $\boldsymbol{x}$, $\boldsymbol{x}^\top \boldsymbol{G}_t \boldsymbol{x} \geq \lambda_{\text{eff}} \|\boldsymbol{x}\|_2^2 > 0$. The directional second derivative along a nonzero matrix $\boldsymbol{H}$ is $2\,\mathrm{tr}(\boldsymbol{H}\boldsymbol{G}_t\boldsymbol{H}^\top) > 0$. Setting the first derivative to zero gives

$$(\boldsymbol{A}_t \widetilde{\boldsymbol{K}}_t - \boldsymbol{R}_t)\widetilde{\boldsymbol{K}}_t^\top + \lambda_{\text{eff}} \boldsymbol{A}_t + \beta \boldsymbol{A}_t \widehat{\boldsymbol{\Sigma}}_t = \boldsymbol{0},$$

which yields Eq. 23. Since $\boldsymbol{B}_t^\top \boldsymbol{B}_t = \boldsymbol{I}$, the map $\boldsymbol{A} \mapsto \boldsymbol{A}\boldsymbol{B}_t^\top$ is injective and preserves the Frobenius norm. □

The unconstrained representative $\boldsymbol{\Delta}_t$ in Eq. 13 need not be unique, because components annihilated by $\boldsymbol{P}_{t-1}$ do not affect the objective. Changing the basis to $\boldsymbol{B}_t\boldsymbol{O}$ for an orthogonal $\boldsymbol{O}$ changes $\boldsymbol{A}_t$ to $\boldsymbol{A}_t\boldsymbol{O}$ and leaves the effective update unchanged. When $r_t = 0$, the only admissible update is zero. When $c_t \leq 10^{-12}$, the identity fallback specified in Section 3.3 gives a ridge coefficient of $\lambda_{\text{eff}} + \beta$.

Corollary 8 (Isotropic special cases). *Setting $\beta = 0$ recovers isotropic projected ridge regression. Setting $\widehat{\boldsymbol{\Sigma}}_t = \boldsymbol{I}$ instead gives total ridge $\lambda_{\text{eff}} + \beta$. For a normalized metric, this isotropic control matches the trace of the full regularization matrix.*

Proof. Substitute into Eq. 23 and use $\mathrm{tr}(\widehat{\boldsymbol{\Sigma}}_t) = r_t$. □

With $\beta = 0$ and $\lambda_{\text{eff}} = 1$, the objective reduces to the isotropic EvoEdit formulation (Lyu et al., 2025). Identical editing trajectories further require matching target construction, keys, projector updates, numerical settings, and initial effective weights.

A.5 Comparison at a Fixed Editing Response

Proposition 9 (Directional benefit beyond norm reduction). *Let $\beta > 0$, and let $\boldsymbol{A}_t$ solve Eq. 15. Define the minimum-norm update with the same editing response by $\boldsymbol{A}_{\text{MN}} = (\boldsymbol{A}_t\widetilde{\boldsymbol{K}}_t)\widetilde{\boldsymbol{K}}_t^\dagger$, where $\dagger$ denotes the Moore–Penrose pseudoinverse, and set $\boldsymbol{N} = \boldsymbol{A}_t - \boldsymbol{A}_{\text{MN}}$. Then $\boldsymbol{N}\widetilde{\boldsymbol{K}}_t = \boldsymbol{0}$ and*

$$\|\boldsymbol{A}_t\|_F^2 - \|\boldsymbol{A}_{\text{MN}}\|_F^2 = \|\boldsymbol{N}\|_F^2, \tag{24}$$

$$\mathrm{tr}(\boldsymbol{A}_{\text{MN}}\widehat{\boldsymbol{\Sigma}}_t\boldsymbol{A}_{\text{MN}}^\top) - \mathrm{tr}(\boldsymbol{A}_t\widehat{\boldsymbol{\Sigma}}_t\boldsymbol{A}_t^\top) = \frac{2\lambda_{\text{eff}}}{\beta}\|\boldsymbol{N}\|_F^2 + \mathrm{tr}(\boldsymbol{N}\widehat{\boldsymbol{\Sigma}}_t\boldsymbol{N}^\top) \geq 0. \tag{25}$$

Proof. The matrix $\widetilde{\boldsymbol{K}}_t\widetilde{\boldsymbol{K}}_t^\dagger$ is the orthogonal projector onto $\mathrm{Im}(\widetilde{\boldsymbol{K}}_t)$. Thus $\boldsymbol{A}_{\text{MN}}$ and $\boldsymbol{N}$ are orthogonal in the Frobenius inner product, proving Eq. 24. Taking the inner product of the normal equation with $\boldsymbol{N}$ eliminates the fit term and gives $\beta\langle \boldsymbol{A}_t\widehat{\boldsymbol{\Sigma}}_t, \boldsymbol{N}\rangle_F = -\lambda_{\text{eff}}\|\boldsymbol{N}\|_F^2$. Expanding the quadratic cost of $\boldsymbol{A}_t - \boldsymbol{N}$ proves Eq. 25. □

At the same editing response, anisotropic regularization can lower reference-output cost even when the update norm increases. The fixed-state analysis in Appendix D.1 examines this distinction empirically. When a common numerical jitter $\eta\boldsymbol{I}$ is added to the solve, the same identity holds with $\lambda_{\text{eff}} + \eta$ in place of λ_{eff}.

B Experimental Setup

B.1 Datasets

Editing benchmarks. We evaluate sequential editing on three benchmarks.

- **ZsRE** (Levy et al., 2017) is a QA dataset with back-translated rephrasings for generalization and natural questions for locality.

- **CounterFact** (Meng et al., 2022a) contrasts counterfactual with factual statements and additionally evaluates fluency and consistency through multiple generation prompts.
- **WikiBigEdit** (Thede et al., 2025) is a large-scale lifelong editing benchmark derived from real-world Wikidata changes, comprising over 500K QA pairs across eight temporal batches with rephrased, persona-style, locality, and multi-hop questions.

Base capability evaluation. We evaluate the base capabilities of edited models using the *lm-evaluation-harness*. We benchmark performance on a diverse set of standard reasoning and knowledge tasks, including IFEval (Zhou et al., 2023), TruthfulQA (MC2) (Lin et al., 2022), MMLU (5-shot) (Hendrycks et al., 2020), GSM8K (Cobbe et al., 2021) with chain-of-thought prompting (8-shot), and ARC-Challenge (25-shot) (Clark et al., 2018). We apply the chat template and multi-turn few-shot formatting with a limit of 200 examples per task or subtask. MMLU therefore covers 9,183 questions across subjects. IFEval averages prompt-level strict and loose accuracy, ARC-C uses length-normalized accuracy, and GSM8K uses flexible answer extraction. The base-capability average is the unweighted mean of the five task scores.

B.2 Baselines

We use the original baseline implementations and their published hyperparameters, with the settings in Appendix B.4. MEMIT (Meng et al., 2022b) and AlphaEdit (Fang et al., 2024) apply closed-form updates to localized FFN projections, while EvoEdit (Lyu et al., 2025) adapts the protected subspace during sequential editing. WISE (Wang et al., 2024a) uses a parameter-efficient memory mechanism, and MEND (Mitchell et al., 2021) predicts updates with a learned hypernetwork. FT and LoRA (Zhu et al., 2020; Hu et al., 2022) provide full and low-rank fine-tuning baselines, while LocBF-FT (Yang et al., 2025b) restricts fine-tuning to a localized layer. UltraEdit (Gu et al., 2025) uses online statistics and sensitivity-aware updates, Adam-NSCL (Wang et al., 2021) performs projected optimization in a covariance-based null space, and CrispEdit (Ikram et al., 2026) constrains updates to preserve model capabilities.

B.3 Evaluation Protocol

We evaluate generated answers with WILD (Yang et al., 2025a) using EasyEdit (Wang et al., 2024b). Under *No Context*, the dataset question is used as-is. Under *QA Context*, the question is wrapped in the template "`Please answer the question:\n\nQ: {question}\nA:`", and the model generates autoregressively up to 40 tokens with stop tokens $[\texttt{.}, \texttt{\textbackslash n}, \texttt{eos}]$. Generated outputs are scored by `gpt-4o-mini` (see Figure 8 for the exact prompt). Given a model f_θ, a question q_i, and a gold answer a_i, the model first generates an answer under context format c:

$$\widehat{a}_{i,c} \;=\; \mathrm{Clean}\big(\mathrm{Decode}(f_\theta,\, \psi_c(q_i))\big), \tag{26}$$

where $\psi_c(\cdot)$ denotes the prompt template. The generated answer is then scored by a judge function

$$J(q_i,\, a_i,\, \widehat{a}_{i,c}) \;\in\; \{0, 1\}, \tag{27}$$

which marks an answer correct when it contains the key information in the gold answer without contradiction. Reliability and generalization are defined as follows.

- **Reliability** measures whether the edited model correctly answers the original edit question:

$$\mathrm{Rel}_c \;=\; \frac{1}{N_{\mathrm{rel}}} \sum_{i=1}^{N_{\mathrm{rel}}} J(q_i,\, a_i,\, \widehat{a}_{i,c}). \tag{28}$$

- **Generalization** measures whether the edit transfers to a semantically equivalent rephrase q_i^{r}:

$$\mathrm{Gen}_c \;=\; \frac{1}{N_{\mathrm{gen}}} \sum_{i=1}^{N_{\mathrm{gen}}} J(q_i^{\mathrm{r}},\, a_i,\, \widehat{a}_{i,c}^{\mathrm{r}}). \tag{29}$$

Here N_{rel} and N_{gen} are the numbers of rewrite and rephrase probes. Scores are reported as percentages for each context c.

Prompt for LLM-as-a-Judge

```
Your job is to look at a question, a gold target, and a predicted answer, and then assign a grade of either ["CORRECT", "INCORRECT"].

The following are examples of CORRECT predicted answers.
```
Question: What are the names of Barack Obama's children?
Gold target: Malia Obama and Sasha Obama
Predicted answer 1: sasha and malia obama
Predicted answer 2: Malia and Sasha Obama are the names of Barack Obama's children.
```
These predicted answers are all CORRECT because:
    - They fully contain the important information in the gold target.
    - They do not contain any information that contradicts the gold target.

The following are examples of INCORRECT predicted answers.
```
Question: What are the names of Barack Obama's children?
Gold target: Malia and Sasha
Predicted answer 1: Malia.
Predicted answer 2: Malia, Sasha, and Susan.
Predicted answer 3: Malia and Sasha, Malia and Sasha, Malia and Sasha, Malia and Sasha (repeated answer)
```
These predicted answers are all INCORRECT because:
    - A factual statement in the answer contradicts the gold target or contain repeated answer.


Here is a sample. Simply reply with either CORRECT or INCORRECT.

```
Question: {question}
Gold target: {target}
Predicted answer: {predicted_answer}
```

According to the gold target, please grade the predicted answer of this question as one of:
A: CORRECT
B: INCORRECT

Just return the letters "A" or "B", with no text around it.
```

Figure 8: The complete prompt used to employ a LLM as a judge for providing binary assessments (correct or incorrect) based on a given question, gold target answer, and predicted answer.

B.4 Implementation Details

We implement LessEdit with HuggingFace Transformers (Wolf et al., 2020) and follow AlphaEdit's target-value optimization (Fang et al., 2024). We edit FFN down-projections in LLaMA layers $[4, 5, 6, 7, 8]$ and Qwen layers $[7, 8, 9, 10, 11]$ using `subject_last`. Target values are optimized for 25 Adam steps with learning rate 0.1, KL factor 0.0625, weight decay 0.5, and clamp norm factor 0.75. We estimate the reference second moment from 100,000 Wikipedia documents, where n_0 counts activation observations, and set the initial-projector and free-subspace thresholds to 2×10^{-2} and 10^{-5}.

Table 3: Comparison of LESSEDIT with existing methods on editing LLaMA-3-8B-Instruct in the teacher-forcing evaluation pipeline. *Rel*, *Gen*, and *Spec* denote reliability, generalization, and specificity, respectively. We perform model editing on 3,000 samples from three representative datasets and evaluate editing performance following the teacher-forcing setup of (Meng et al., 2022b;a; Fang et al., 2024). Best values are in bold.

Method	ZsRE			CounterFact			WikiBigEdit		
	Rel	Gen	Spec	Rel	Gen	Spec	Rel	Gen	Spec
Llama 3 8B Instruct	25.7	25.1	41.2	0.9	1.2	57.6	34.0	34.8	32.8
MEMIT	0.1	0.1	1.6	0.0	0.0	50.8	0.3	0.3	0.4
Adam-NSCL	98.5	**92.0**	23.5	**99.7**	**81.3**	50.2	**99.9**	**98.1**	35.6
LocBF-FT	**99.1**	91.2	42.4	**99.7**	72.7	53.3	**99.9**	96.9	42.8
UltraEdit	62.1	57.7	44.5	27.8	17.7	54.2	87.3	83.8	**47.7**
MEND	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0
FT	79.7	76.6	16.8	78.6	59.8	51.6	93.5	90.5	29.0
LoRA	35.8	31.5	27.3	19.7	11.4	52.4	68.8	63.9	35.3
AlphaEdit	84.9	75.2	42.1	95.4	73.4	53.1	94.4	89.2	41.5
WISE	25.8	25.0	41.1	0.9	1.2	**57.6**	34.0	34.8	32.8
EvoEdit	93.1	82.4	**45.7**	98.2	70.0	56.1	97.2	90.3	45.3
CrispEdit	95.8	90.5	35.0	95.5	60.6	52.0	98.5	95.4	40.1
LESSEDIT	96.1	81.4	44.4	98.9	72.3	56.6	97.7	91.5	46.6

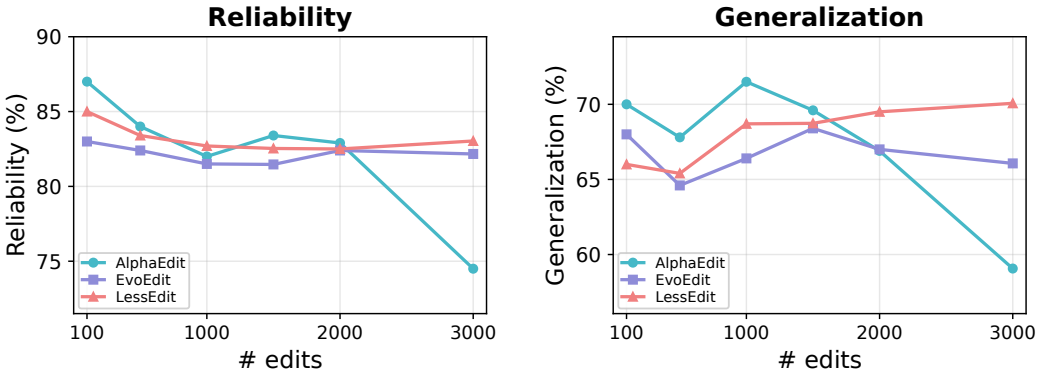


Figure 9: QA-context performance over the CounterFact editing sequence.

Unless specified otherwise, LESSEDIT uses two branches with frozen duplicated gate/up projections, $\beta = 1$, $\epsilon = 0.1$, $\lambda_{\mathrm{E}} = 1$, and $\lambda_{\mathrm{A}} = 10$. For LLaMA ZsRE10K, we use $\lambda_{\mathrm{E}} = 4.004$ and $\lambda_{\mathrm{A}} = 40.04$, corresponding to $\lambda_{\mathrm{eff}} = 4$. Edits are processed in batches of 100 on one A800 (80GB) GPU.

C ADDITIONAL EDITING RESULTS

C.1 TEACHER-FORCED EVALUATION

Table 3 evaluates editing under the teacher-forced protocol (Meng et al., 2022b;a; Fang et al., 2024). LESSEDIT achieves competitive reliability, generalization, and specificity across the three datasets. High teacher-forced scores for several baselines coincide with much lower WILD scores in Table 1, supporting the use of autoregressive evaluation in Section 4.1.

C.2 SEQUENTIAL EDITING ON COUNTERFACT

Figure 9 complements Table 1 by tracking QA-context performance throughout the CounterFact editing sequence. EvoEdit and LESSEDIT retain higher reliability and generalization than AlphaEdit near the end of the sequence, with LESSEDIT achieving the highest final generalization.

C.3 GENERALIZATION TO QWEN

Table 4 evaluates Qwen2.5-7B-Instruct after 3,000 edits on each dataset. Compared with EvoEdit, LESSEDIT improves nine of twelve WILD scores, increasing the mean WILD and base-capability scores by 1.92 and 0.77 points, respectively. UltraEdit retains stronger base capabilities, while LESSEDIT achieves the highest QA reliability and generalization on all three datasets.

Table 4: Editing and base capabilities of Qwen2.5-7B-Instruct after 3,000 edits.

Data	Method	Edited Capabilities				Base Capabilities				
		QA Context		No Context						
		Rel	Gen	Rel	Gen	MMLU	IFEval	TruthfulQA	ARC-C	GSM8K
ZsRE	UltraEdit	13.3	11.2	29.3	16.1	**75.7**	**68.5**	**60.0**	58.5	**79.0**
	AlphaEdit	12.9	13.2	7.9	8.1	43.0	41.3	47.1	39.0	58.5
	EvoEdit	57.3	41.8	**66.2**	37.7	72.7	63.3	54.9	**60.0**	74.5
	CrispEdit	51.1	33.6	58.9	37.0	73.4	61.5	57.6	52.5	73.5
	LESSEDIT	**60.6**	**45.5**	62.5	**39.8**	73.9	68.0	57.3	56.5	73.5
unterFact	UltraEdit	13.9	5.0	26.7	9.5	**75.5**	**67.5**	58.8	**58.5**	**79.5**
	AlphaEdit	0.0	0.0	0.0	0.0	23.3	9.5	53.0	27.5	0.0
	EvoEdit	70.3	39.7	57.1	**27.3**	71.5	62.5	**59.4**	54.5	74.5
	CrispEdit	27.9	8.4	51.2	11.9	73.9	65.8	53.9	52.5	74.5
	LESSEDIT	**72.4**	**42.0**	**59.5**	25.8	73.5	66.5	56.2	57.0	76.0
ikiBigEdit	UltraEdit	35.6	32.6	59.8	48.4	**75.7**	68.3	59.3	59.0	**79.0**
	AlphaEdit	0.4	0.4	0.1	0.1	24.4	6.3	48.8	24.5	0.0
	EvoEdit	69.3	59.8	**81.7**	59.9	73.6	62.8	55.9	58.5	76.0
	CrispEdit	45.8	37.0	43.7	38.6	74.2	64.5	**60.0**	**60.5**	73.0
	LESSEDIT	**75.1**	**65.2**	80.7	**62.1**	74.5	**69.5**	51.2	58.0	74.5

C.4 SCALING TO 10,000 EDITS

Table 5 reports performance after 10,000 ZsRE edits. On LLaMA-3-8B-Instruct, LESSEDIT achieves mean WILD and base scores of 68.28 and 54.06, exceeding EvoEdit by 2.52 and 11.78 points, respectively. On Qwen, mean WILD improves by 4.23 points, with a 0.93-point lower base average.

Table 5: Editing and base capabilities after 10,000 ZsRE edits.

Model	Method	Edited Capabilities				Base Capabilities				
		QA Context		No Context						
		Rel	Gen	Rel	Gen	MMLU	IFEval	TruthfulQA	ARC-C	GSM8K
LLaMA-3 8B Instruct	UltraEdit	20.1	16.7	12.6	10.4	67.9	**68.9**	49.8	46.0	**73.0**
	Adam-NSCL	1.2	1.1	0.4	0.7	68.2	14.8	**54.0**	35.0	2.0
	AlphaEdit	0.3	0.2	0.1	0.0	22.8	20.9	53.9	22.0	0.0
	EvoEdit	**73.9**	60.8	72.0	56.3	50.8	47.0	41.1	41.5	31.0
	CrispEdit	47.7	44.0	11.8	11.2	**68.8**	58.5	51.4	**53.0**	60.5
	LESSEDIT	73.8	**61.8**	**76.7**	**60.8**	58.7	59.3	42.9	49.0	60.5
Qwen2.5 7B Instruct	AlphaEdit	0.0	0.0	0.3	0.1	26.0	6.5	52.1	30.5	0.5
	EvoEdit	53.8	38.1	54.2	31.2	**61.1**	52.3	52.6	**53.5**	**63.0**
	CrispEdit	3.0	2.2	1.8	1.1	28.0	30.3	46.3	41.0	14.5
	LESSEDIT	**57.3**	**41.2**	**61.7**	**33.9**	59.2	**53.3**	**53.3**	52.0	60.0

Available dimension. Table 6 tracks the remaining free dimension during Qwen editing. Before the final 100 edits, 24.77% remains on average, with 1,932–4,454 directions per layer. Capability degradation therefore occurs before the free subspace is exhausted. This complements Corollary 3, which shows that dimension consumption depends on the projected keys rather than the metric.

Cumulative perturbation. For each edited layer l, let $\boldsymbol{D}_l = \boldsymbol{W}_{l,T} - \boldsymbol{W}_{l,0}$ and $C_l = \mathrm{tr}(\boldsymbol{D}_l \boldsymbol{\Sigma}_{0,l} \boldsymbol{D}_l^\top)$. Figure 10 compares 10K checkpoints and a common reference second moment.

Table 6: Remaining free dimensions during Qwen ZsRE 10K editing.

Prior edits	Layer 7	Layer 8	Layer 9	Layer 10	Layer 11	Mean (%)
0	13,471	13,001	11,820	13,337	14,350	100.00
1,000	12,471	12,001	10,820	12,337	13,350	92.39
5,000	8,474	8,003	6,822	8,338	9,350	61.97
9,000	4,488	4,015	2,829	4,344	5,352	31.60
9,900	3,590	3,119	1,932	3,445	4,454	24.77

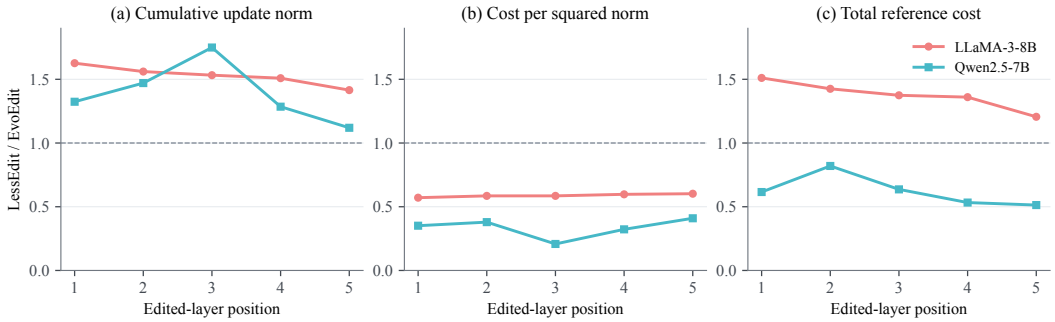


Figure 10: Weight diagnostics after 10K edits, normalized by EvoEdit.

LESSEDIT has lower $C_l/\|\boldsymbol{D}_l\|_F^2$ in every layer, but larger cumulative norms. The aggregate cost ratio $\sum_l C_l^{\text{LessEdit}} / \sum_l C_l^{\text{EvoEdit}}$ is 1.28 on LLaMA and 0.57 on Qwen. Thus the lower cost per unit update energy observed in Section 4.3 can coexist with larger cumulative perturbations.

C.5 MULTI-HOP QUESTIONS

We evaluate whether edited facts support multi-hop reasoning on 86 WikiBigEdit requests whose questions do not reveal the target or final answer. The full model and its variant without CapCov receive one batch of edits and use greedy decoding. Table 7 distinguishes strict teacher-forced recall from generated direct recall. Both variants reach 22.09% direct multi-hop accuracy and answer 18 of the 84 cases with shared strict recall correctly. The CoT difference is -1.16 points with a paired 95% interval of $[-5.81, 3.49]$. This comparison finds no measurable multi-hop benefit from CapCov.

Table 7: Use of edited facts on 86 paired multi-hop cases.

Model	Strict recall	Direct recall	Multi-hop	CoT multi-hop
Unedited	1.16	25.58	22.09	16.28
Without CapCov ($\beta = 0$)	97.67	58.14	22.09	24.42
LESSEDIT	98.84	56.98	22.09	23.26

C.6 CASE STUDY

The following generations illustrate target recall and answer quality across the three datasets, complementing the aggregate WILD comparison in Section 4.2.

Model Editing Case Study on ZsRE	
Editing Prompt	What species is FAM63A specific to?
Edit Target	species
Generation Output	
Adam-NSCL	species species species

Model Editing Case Study on WikiBigEdit	
Editing Prompt	In which historic county is 10 Marchmont Terrace located?
Edit Target	Lanarkshire
Generation Output	
Adam-NSCL	Lanarkshireshire (Lanarkshireshire is a historic county in Lanarkshireshireshire

weighting and regularization strength. Figure 4 summarizes the gains across datasets, while Appendix D.1 examines directional effects at fixed model states.

Table 8: CapCov transfer to AlphaEdit after 3K LLaMA edits.

Dataset	Method	QA Rel	QA Gen	NC Rel	NC Gen	Base
ZsRE	AlphaEdit	70.47	59.44	39.63	31.62	48.18
	AlphaEdit+CapCov	77.37	67.95	62.87	49.63	59.97
	LESSEDIT	82.43	66.48	74.33	55.74	59.73
CounterFact	AlphaEdit	74.77	59.00	53.00	45.83	40.90
	AlphaEdit+CapCov	87.20	71.07	62.23	52.27	58.78
	LESSEDIT	83.37	69.87	61.73	50.70	60.06
WikiBigEdit	AlphaEdit	72.60	67.16	69.30	63.02	51.12
	AlphaEdit+CapCov	82.97	74.70	86.60	78.60	60.64
	LESSEDIT	83.93	76.94	87.77	79.37	61.10

D.3 SENSITIVITY TO REFERENCE STATISTICS

Sample size. Table 9 compares nested Wikipedia estimates with the 100K-document reference at layer 6. Increasing the sample size from 100 to 10K documents reduces relative Frobenius error from 0.2890 to 0.0320 and raises top-32 eigenspace overlap from 0.7042 to 0.9926. The overlap is the mean squared cosine of the principal angles.

Table 9: Covariance convergence at layer 6 relative to Wikipedia 100K.

Documents	Tokens	Trace ratio	Rel. error	Cosine	λ_1 ratio	Top-32 overlap
100	62,044	1.0014	0.2890	0.9615	1.0252	0.7042
500	301,573	1.0036	0.1410	0.9903	1.0034	0.8687
1,000	683,131	1.0000	0.0964	0.9954	1.0004	0.9394
10,000	6,838,258	1.0013	0.0320	0.9995	1.0038	0.9926

Calibration source. We vary the covariance source while fixing the original base projector during CounterFact 3K editing. Each source contributes 683,131 activation observations, and the mixture pools all three sources. Table 10 reports covariance geometry averaged across edited layers relative to Wikipedia 100K, target-token accuracy on the final 500 edits, and mean base capability. Rewrite accuracy remains 98.22–98.42%, while base averages range from 59.47 to 62.01%.

Table 10: Calibration-source sensitivity after CounterFact 3K editing.

Source	Cosine	Rel. error	Top-32 overlap	TF Rel	TF Gen	Base
Wikipedia 1K	0.993	0.121	0.933	98.22	78.90	59.47
LMSYS	0.603	0.839	0.314	98.22	81.46	60.86
ZsRE	0.230	2.471	0.120	98.42	78.70	60.50
Mixture	0.634	0.914	0.432	98.42	76.92	62.01

Figure 12 reports output KL and relative FFN drift on 64 prompts per domain. The latter averages squared FFN-output changes normalized by original output energy across edited layers. ZsRE calibration gives the smallest ZsRE drift, and the mixture gives the smallest code-domain drift. The reference distribution therefore affects which behaviors the metric preserves.

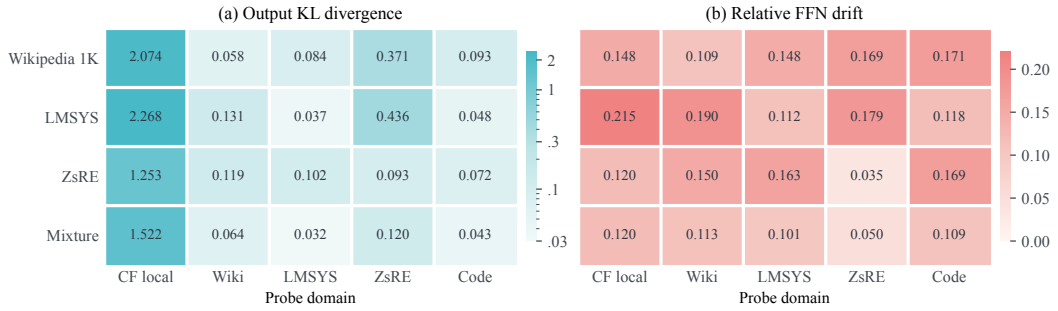


Figure 12: Cross-domain drift under four calibration sources. Lower values indicate less change.

E Branch Reparameterization and Efficiency

E.1 Branch Interventions and Function Preservation

Table 11 shows exact logit agreement on the evaluated prompts before and after FFN reparameterization. Table 12 then isolates branch increments on a CounterFact 3K checkpoint, with strict teacher-forced accuracy on the final 500 edits. The anchored-to-editable increment norm ratio is 0.001, consistent with Eq. 10. The anchored increment closely recovers base behavior, while the editable increment retains the edits, consistent with the asymmetric allocation in Section 3.2.

Table 11: Function-preserving growth sanity check before editing. Logit differences are measured before and after branch reparameterization on a small prompt set.

Function-Preserving Growth Check	
Growth configuration	
Growth layers	4, 5, 6, 7, 8
Growth time	8.73s
Functional invariance before editing	
Max logit difference	0.0
Mean logit difference	0.0
RMS logit difference	0.0
Top-1 agreement	100%

Table 12: Branch interventions on a CounterFact 3K checkpoint.

Active update	**TF Rel**	**TF Gen**	**MMLU**	**IFEval**	**TQA**	**ARC-C**	**GSM8K**	**Base**
Unedited	0.20	1.00	69.47	69.30	50.72	56.00	73.50	63.80
Anchor increment only	0.20	1.00	69.49	67.50	50.72	56.00	73.50	63.44
Editable increment only	98.40	78.20	65.19	65.25	46.74	53.50	70.50	60.24
LessEdit	98.40	78.20	64.81	64.40	47.63	54.50	69.00	60.07

Merging gives mean logit KL 2.05×10^{-8} and maximum logit difference 5.25×10^{-5} from the unmerged model, with unchanged rewrite/rephrase accuracy and agreement on all 100 greedy-generation probes. Redistributing the initial contribution of the same final checkpoint with proportions 0.25, 0.50, and 0.75 also preserves all tested greedy sequences. These redistributions support the functional equivalence established in Proposition 4.

E.2 EFFICIENCY

Editing stage. Table 13 compares editing cost on 300 CounterFact requests using LLaMA-3-8B-Instruct in FP32 on an A800, with batches of 100. Timing includes the first editor call and excludes model loading, evaluation, and saving. LESSEDIT takes 53m 52s, 1.6% longer than EvoEdit, while peak allocated memory increases from 33.4 to 47.2 GiB.

Table 13: Editing runtime and memory measurements on LLaMA-3-8B-Instruct.

Method	Extra Params	Time / 300 Edits	Time / 100 Edits	Peak Alloc. (GiB)	Peak Reserved (GiB)
AlphaEdit	0	53m 20s	17m 47s	36.9	38.0
EvoEdit	0	53m 02s	17m 41s	33.4	35.8
LESSEDIT	+880.8M	53m 52s	17m 57s	47.2	52.6

Deployment stage. Table 14 measures inference before and after merging a ZsRE10K checkpoint in FP32 with SDPA and 64 generated tokens. Each setting uses two warmups before timing. Merging removes 880.8M extra parameters, reduces latency by 8.5–9.7%, and lowers peak allocated memory by 3.28 GiB.

Table 14: Inference before and after merging dual branches.

Batch / input	Latency (s)	Tokens/s	Alloc. (GiB)	Reserved (GiB)
1 / 128	1.798 / 1.645	35.59 / 38.90	33.30 / 30.02	33.38 / 30.11
1 / 512	2.357 / 2.156	27.15 / 29.68	33.58 / 30.30	33.71 / 30.44
4 / 128	3.143 / 2.843	81.46 / 90.04	33.58 / 30.30	33.71 / 30.44
4 / 512	5.047 / 4.559	50.72 / 56.16	34.71 / 31.43	35.22 / 32.05

Each pair reports unmerged / merged measurements.

F LIMITATIONS AND FUTURE WORK

Capability covariance is a local, distribution-dependent surrogate, and the free subspace has finite capacity. Our evaluation covers two similarly sized model families, mostly with single editing trajectories. Future work should examine adaptive reference statistics and cumulative perturbation control across broader reasoning, multilingual, code, and multimodal settings.