

PRISM: Refracting Large-Batch Model Editing through Jacobian-Inferred FFN Subspaces

Ziyue Zhai¹, Qiyao Liu³, Hao Song¹, Heng Yu¹,
Guozhi Lin¹, Yulai Yang¹, Xuan Zhang^{1,2}

¹Zhongguancun Laboratory, ²Tsinghua University, ³Independent Researcher

Correspondence: xuanzhang@tsinghua.edu.cn

Abstract

Model editing updates factual knowledge in large language models (LLMs) without retraining, but simultaneous large-batch edits can degrade generalization and locality as edit interactions accumulate. We study this problem through a multi-scale analysis of feed-forward network (FFN) and multi-head self-attention (MHSA) geometry. Weak FFN–MHSA token-level alignment can coexist with non-trivial distributional coupling, while Jacobian-inferred FFN local-response geometry provides a distinct proxy for locally supported response directions. Based on this distinction, we propose **PRISM**, which replaces unconstrained residual-target fitting with controlled residual correction within edit-conditioned, Jacobian-inferred local-response subspaces. On the standard CounterFact large-batch benchmark, PRISM improves the baseline’s Score by 1.3–1.7 points and Generalization by 1.7–1.8 points at 10k edits, with larger Generalization gains in beyond-10k scaling tests at 20k edits while maintaining competitive Specificity.

1 Introduction

Large Language Models (LLMs) store substantial factual knowledge (Petroni et al., 2019; Dai et al., 2022), yet that knowledge is **static**: deployed models can become outdated or retain incorrect facts (Huang et al., 2025; Ji et al., 2023). Retraining or fine-tuning is expensive and can interfere with previously learned behavior (McCloskey and Cohen, 1989; Kirkpatrick et al., 2017).

Model editing objectives at scale. Model editing seeks targeted factual updates without retraining and is commonly evaluated by **Efficacy**, **Generalization**, and **Specificity** (Wang et al., 2024). We focus on simultaneous large-batch editing, where thousands of edits are written in one run rather than introduced sequentially over time. In this regime, the central difficulty is more specific than

a generic three-way trade-off: large-batch editing is not simply single-edit difficulty repeated many times, because interactions among edits accumulate and increasingly govern how transfer and locality degrade. For strong editors, efficacy on the edited prompts is often already high once the target fact is written; the harder problem is to preserve generalization under these accumulating interactions without collapsing specificity. This scale-conditioned tension drives our analysis and method design.

An open question in FFN-side editing. ROME rewrites FFN associations by rank-one updates, MEMIT extends this idea to large-batch editing, and PMET separates FFN- and MHSA-side target optimization while still writing back to FFN weights (Meng et al., 2022, 2023; Li et al., 2024). This FFN focus is not arbitrary: prior editing and mechanistic studies suggest that factual associations are often mediated by middle-layer FFN computations and can be effectively modified through FFN-side interventions (Meng et al., 2022, 2023; Geva et al., 2022). The practical question is therefore not only whether an FFN-side update can write the target fact, but whether its direction can sustain transfer without degrading Specificity as edit interactions densify. FFN-side updates may contain components aligned with directions shared with MHSA, motivating us to test whether stronger cross-component coupling is associated with larger response changes on matched reference prompts. Separately, we ask whether token-level targets align with locally supported FFN response directions or instead mix in prompt-specific components weakly supported by the FFN’s local response geometry.

A multi-scale geometric motivation. We study this question geometrically on WSC (Levesque et al., 2012) through three non-equivalent objects: token-level directional relations, distributional subspaces estimated from tokens, and local-response subspaces estimated from Jacobian–vector prod-

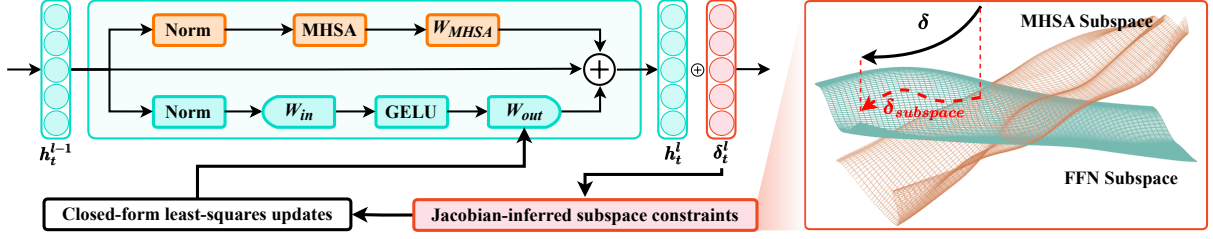


Figure 1: Motivating view of PRISM’s Jacobian-inferred constraints on FFN writeback.

uct (JVP) responses to random probe directions. Weak token-level alignment can coexist with non-trivial coupling between distributional subspaces, whereas local-response directions are consistently less aligned than the corresponding distributional subspaces. We therefore use distributional geometry as a diagnostic of cross-component coupling and local-response geometry as a proxy for directions that the local FFN can actually support. A matched-norm intervention test later links stronger distributional coupling to larger target-side response magnitudes and larger reference-KL changes, but we use this evidence as design motivation rather than as a causal proof.

PRISM: controlled residual correction in locally supported directions. Motivated by the separation between distributional coupling and local-response geometry, we propose **PRISM** (Projected Residual into Inferred Subspaces of Models). Existing FFN-side editors typically follow a target-then-writeback pattern: they fit layer-local residual targets and write them into FFN weights. Rather than fitting unconstrained layer-local targets, PRISM keeps the closed-form writeback but restricts each layerwise correction to an edit-conditioned, Jacobian-inferred local-response subspace of the FFN, then restores capacity through in-subspace refinement. Figure 1 summarizes this constrained target-construction view.

Contributions.

- **Multi-scale FFN–MHSA Geometry.** We distinguish FFN–MHSA token-level alignment, FFN–MHSA distributional coupling, and Jacobian-inferred local-response geometry, showing that weak token-level alignment can coexist with non-trivial distributional coupling while local-response geometry provides a separate proxy for locally supported FFN response directions.
- **Constrained Target Construction.** We in-

troduce PRISM, which constructs layerwise FFN residual targets by selecting and refining directions inside edit-conditioned, Jacobian-inferred local-response subspaces while retaining the standard closed-form writeback.

- **Scale-Stable Editing Quality.** Across the evaluated CounterFact and zsRE settings, PRISM improves large-batch editing quality relative to completed simultaneous-editor baselines, with stronger Generalization at 10k and slower degradation across the CounterFact scale sweep while maintaining competitive Specificity.

2 Related Work

Knowledge editing at scale. Knowledge editing seeks targeted updates that improve the desired prediction while preserving the rest of the model (Wang et al., 2024). Sequential editing studies interference across repeated rewrites, whereas our focus is the complementary large-batch regime, where many edits are written in one run and the key issue is how generalization and locality degrade as edit interactions accumulate (Meng et al., 2023; Li et al., 2024; Gupta et al., 2024). Recent continuous and lifelong editors study a related but different regime in which edits arrive over time and later updates may interfere with earlier ones (Li and Chu, 2025; Xu et al., 2025). Beyond direct rewriting, prior families also include learned editors such as MEND and MALMEN (De Cao et al., 2021; Mitchell et al., 2022a; Tan et al., 2024) and memory-based editors such as SERAC (Mitchell et al., 2022b).

Direct rewriting and geometry-aware constraints. ROME, MEMIT, and PMET edit factual associations through FFN-side writeback or FFN-centered target construction (Meng et al., 2022, 2023; Li et al., 2024). Recent constrained editors, especially for sequential or continual editing regimes, use regularization, null-space projection, or orthogonalization to reduce parameter-level or

history-dependent interference (Fang et al., 2025; Xu et al., 2025). Our contribution is not another writeback solver, but a target-construction rule for layerwise residuals within the same fixed edited-layer set: PRISM explicitly constrains the residual target before writeback to edit-conditioned, Jacobian-inferred FFN response subspaces, prioritizing locally realizable corrections and de-emphasizing weakly supported components that may mix with MHSA, rather than relying on preserved-example null spaces or past-edit caches. This is motivated by mechanistic and geometric work showing that transformer components are not a uniform update space, including FFNs as key-value memories, neuron-level factual units, attention-head specialization, and anisotropic contextual geometry (Geva et al., 2021; Dai et al., 2022; Olsson et al., 2022; Ethayarajh, 2019; Valeriani et al., 2023).

3 Preliminaries

Notation. We denote an autoregressive Transformer by f_θ with L layers and hidden size d . Given an edit set $\mathcal{E} = \{e_i\}_{i=1}^{|\mathcal{E}|}$ and edited layers $\mathcal{L} \subseteq \{1, \dots, L\}$, each edit request is a factual triple $e_i = (s_i, r_i, o_i)$, realized by a prompt family $\mathcal{X}_i = \{x_{i,c} : c \in \mathcal{C}\}$, where c indexes prompt/context realizations for edit i ; the fact is edited at the triple level, while optimization and evaluation operate on its textual realizations.

Transformer blocks and component outputs. Let $h_t^{\ell-1} \in \mathbb{R}^d$ be the residual stream at token position t entering layer ℓ . Using a pre-norm decomposition, we write the MHSA and FFN pre-residual outputs as

$$a_t^\ell = \text{Attn}^\ell(\text{LN}(h_t^{\ell-1})), \quad m_t^\ell = \text{MLP}^\ell(\text{LN}(\bar{h}_t^\ell)),$$

where $\bar{h}_t^\ell = h_t^{\ell-1}$ for parallel blocks and $\bar{h}_t^\ell = h_t^{\ell-1} + a_t^\ell$ for sequential blocks, so that $h_t^\ell = h_t^{\ell-1} + a_t^\ell + m_t^\ell$ in either case. This notation tracks residual increments only; it does not equate the two block organizations. Throughout, a_t^ℓ and m_t^ℓ denote the component outputs immediately before residual addition. We write the FFN output as $m_t^\ell = W^\ell z_t^\ell$, where W^ℓ denotes the second FFN linear map, i.e., the output projection edited throughout the paper.

Task objective and layer-local targets. At the task level, we follow the standard prediction-plus-KL locality objective (Meng et al., 2022, 2023): the edited model should increase $p_{\theta'}(o_i \mid x_{i,c})$ while

remaining close to the original model on unrelated prompts. Closed-form editors then replace this global objective by tractable layer-local targets. In our notation, a layer-local intervention adds a residual δ to the FFN pre-residual output at the edited site, and layerwise targets T^ℓ collect such corrections across edits. At this stage, T^ℓ is an output-space target; how its direction is chosen is separate from how it is written back into parameters.

Keys, covariances, and closed-form writeback.

Let k_i^ℓ be the realized input to the edited FFN output projection for edit e_i at layer ℓ , and let $K^\ell = [k_1^\ell, \dots, k_{|\mathcal{E}|}^\ell]$. Together with the corresponding second-moment matrix C^ℓ , these quantities are used only by the writeback solver: once T^ℓ has been specified, they translate it into a parameter update on W^ℓ but do not determine its direction. In the generic notation C^ℓ may be layer-specific; in our implementation, following MEMIT, we reuse a single precomputed second-moment matrix across edited layers. We then apply the standard MEMIT-style writeback

$$\Delta^\ell = T^\ell A^{\ell\top}, \quad A^\ell = (\lambda C^\ell + K^\ell K^{\ell\top})^{-1} K^\ell \quad (1)$$

Here, λ balances new and old associations in the closed-form writeback. Our contribution lies upstream, in how the targets T^ℓ are constructed before this solver is applied.

Jacobians and Jacobian-vector products. For a component operator F^ℓ , Jacobian-vector products (JVPs) $J_{F^\ell}^\ell(x)v$ provide local linear responses without explicitly forming the full Jacobian. Stacking normalized probe responses yields a low-dimensional proxy for dominant local-response directions near x ; the probe budget controls the quality-cost trade-off of this estimate.

4 Geometric Analysis

Token-level disagreement can mask subspace structure, so we compare token-level relations, *distributional* subspaces from retained tokens, and *local-response* subspaces from JVP responses. The first captures instantaneous cross-component alignment, the second captures which directions co-vary most under the observed token distribution, and the third is closer to directions that the component can locally realize and amplify around the current state. Accordingly, we use the distributional view as a diagnostic of shared variation and the local-response

view as the proxy that motivates the constrained update space.

Extraction points and semantic token slices. We use WSC and Quoref as diagnostic datasets because their localized semantic anchors reduce prompt-formatting noise in FFN–MHSA comparisons. Following Section 3, we record the same pre-residual MHSA and FFN component outputs at each layer. The geometry diagnostics use all 804 WSC examples and a fixed 1,000-example subset of Quoref validation prompts. To reduce prompt-formatting confounds, we restrict analysis to task-relevant semantic spans: antecedent/pronoun regions for WSC and annotated answer spans for Quoref, with a small local token window around each anchor. The same extraction interface is used for both parallel and sequential blocks, with schematics deferred to Appendix A.

Token-level FFN–MHSA alignment. We center MHSA and FFN outputs within each prompt and layer over the analyzed span, compute signed and absolute cosine similarities for all retained tokens, and summarize each layer by a token-level mean angle derived from the mean absolute cosine. On GPT-J / WSC, signed cosines stay near zero in many layers but remain depth-dependent, and the absolute cosines yield a high across-layer mean token-level angle of 86.67° . The representative distributions and all replications are deferred to Appendix A; Table 6 shows the same weak but non-uniform picture in the other settings, so token-level statistics alone do not rule out lower-dimensional cross-component overlap.

Distributional Subspaces. To test for shared statistical structure, we form centered token matrices for MHSA and FFN outputs and compare their principal subspaces by principal angles. We use a Gaussian-noise rank rule to focus the main comparison on structure above the matched noise floor, and summarize each layer by the mean of its smallest five available principal angles; alternative rank and top- k choices are treated as robustness checks in Appendix A.4. In the representative GPT-J / WSC case (Figure 2, top), the across-layer mean of this distributional summary is 40.53° , far below the token-level mean angle, indicating non-trivial overlap between the dominant distributional subspaces despite weak token-level alignment.

Local-Response Subspaces. Distributional overlap does not identify the directions each component

locally amplifies, and the FFN’s true functional update space is not directly observable from activations alone. We therefore use JVP-based local-response subspaces as an operational proxy for directions the component can locally realize near the current operating point. For the main comparison in Figure 2, we use the same Gaussian-noise rank rule and summarize each layer by its top-5 principal-angle mean.

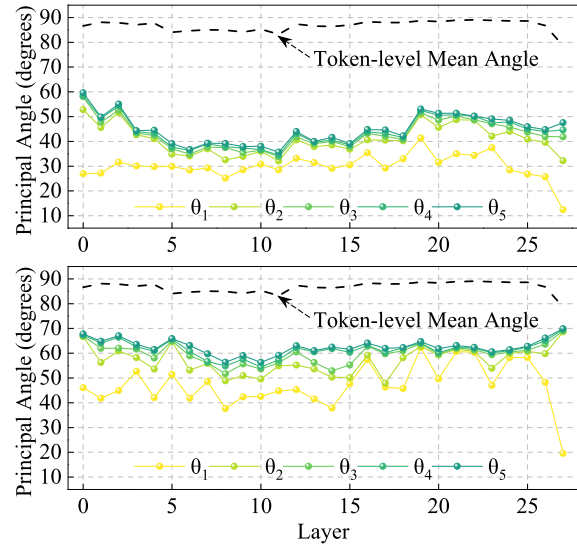


Figure 2: GPT-J on WSC: cross-component distributional (top) and local-response (bottom) geometry.

For GPT-J / WSC, the across-layer mean rises from 40.53° for the distributional summary to 57.68° for the local-response summary, and this upward shift appears in all 28 layers. Appendix A confirms the same ordering across settings, broader summaries, and control analyses. Taken together, token-level relations, distributional overlap, and local-response geometry are distinct objects; the stability controls provide additional evidence for using local-response geometry as an operational proxy for locally supported update directions within the PRISM framework.

Intervention Test: Is Distributional Coupling Associated with Reference-KL Sensitivity? As a separate controlled diagnostic, we connect this geometry to behavior through a weight-free inference-time intervention on GPT-J / CounterFact, using 300 relation-balanced intervention cases and 300 matched reference prompts. We define high- and low-coupling layer groups as the layers with the smallest (high coupling) and largest (low coupling) distributional top-5 angles from a pooled semantic-anchor analysis over CounterFact rewrite prompts.

Layer groups are fixed from these coupling statistics before perturbation evaluation, so intervention outcomes do not enter the selection. We then inject equal-norm Gaussian perturbations at the FFN output module before residual addition. We measure two outcomes: *target effect*, the magnitude of the target-margin change on intervention cases, and *reference KL*, the KL divergence between the original and intervened output distributions on matched reference prompts.

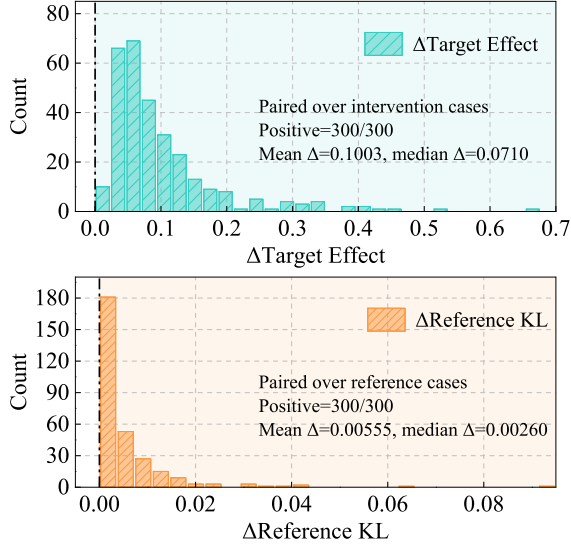


Figure 3: Matched-norm intervention on GPT-J.

In the canonical Gaussian-noise / top-5 setting, high-coupling layers produce larger target-side changes on average than low-coupling layers by 0.1003 and also larger reference-KL changes by 0.00555. Figure 3 shows this aggregate paired-difference signal for the 300 intervention and 300 reference cases. The aggregate high-minus-low differences remain positive across the rank-rule and top- k variants reported in Table 10. We therefore read the intervention as controlled diagnostic evidence: stronger distributional coupling is associated with larger target-margin and reference-KL changes under matched-norm FFN perturbations, while the local-response analysis remains the direct motivation for PRISM’s constrained update space.

5 Methodology

PRISM turns the geometric distinction into a constrained editor: distributional coupling remains diagnostic, while the update space is built from Jacobian-inferred directions that better approximate what the local FFN can support at the current operating point. The matched-norm intervention

Algorithm 1 The PRISM algorithm.

Data: edits $\mathcal{E} = \{e_i\}$; model f_θ ; layers $\mathcal{L}$; covariances C^ℓ
Result: edited model containing edits from $\mathcal{E}$

- 1: **for** $\ell \in \mathcal{L}$ **do** ▷ construct targets and update layer ℓ
- 2: **for** $e_i \in \mathcal{E}$ **do** ▷ build one PRISM target
- 3: $g_{i,\ell} \leftarrow \nabla_{\delta} \tilde{L}_i^\ell(0)$
- 4: $\delta_{i,\ell}^{\text{raw}} \leftarrow -\eta_{\delta} g_{i,\ell} / (\|g_{i,\ell}\|_2 + \epsilon)$
- 5: compute normalized JVP responses $\{r_{i,\ell}^{(j)}\}_{j=1}^{N_{\text{probe}}}$
- 6: $M_{i,\ell}^{\text{bundle}} \leftarrow [r_{i,\ell}^{(1)}, \dots, r_{i,\ell}^{(N_{\text{probe}})}] = U \Sigma V^\top$
- 7: $U_i^\ell \leftarrow U_{:,1:k_{i,\ell}}$ ▷ τ_σ -energy truncation
- 8: $\alpha_{i,\ell}^{(0)} \leftarrow U_i^{\ell \top} \delta_{i,\ell}^{\text{raw}}$
- 9: $g_{i,\ell}^\alpha \leftarrow \nabla_{\alpha} \tilde{L}_i^\ell(U_i^\ell \alpha)|_{\alpha=\alpha_{i,\ell}^{(0)}}$
- 10: $\alpha_{i,\ell}^{(1)} \leftarrow \alpha_{i,\ell}^{(0)} - \eta_{\alpha} g_{i,\ell}^\alpha / (\|g_{i,\ell}^\alpha\|_2 + \epsilon)$
- 11: $\delta_{i,\ell}^{\text{sub}} \leftarrow U_i^{\ell \top} \alpha_{i,\ell}^{(1)}$
- 12: **end for**
- 13: $T^\ell \leftarrow [\delta_{1,\ell}^{\text{sub}}, \dots, \delta_{|\mathcal{E}|,\ell}^{\text{sub}}]; K^\ell \leftarrow [k_1^\ell, \dots, k_{|\mathcal{E}|}^\ell]$
- 14: $A^\ell \leftarrow (\lambda C^\ell + K^\ell K^{\ell \top})^{-1} K^\ell$
- 15: $\Delta^\ell \leftarrow T^\ell A^{\ell \top}; W^\ell \leftarrow W^\ell + \Delta^\ell$
- 16: **end for**

test links stronger distributional coupling to larger target-margin and reference-KL changes.

Rationale: Controllability–Plasticity Tension.

Unconstrained residual updates are plastic but can include prompt-specific or weakly supported components; overly tight constraints improve control but can underfit. PRISM resolves this tension with four stages: a first-order proposal, an edit-conditioned Jacobian basis for the local FFN, one in-subspace refinement step, and standard closed-form writeback. For each edit $e_i = (s_i, r_i, o_i)$, layer $\ell \in \mathcal{L}$, and token position t^* , the optimization variable is a layer-local residual intervention δ . Figure 4 illustrates Steps 2–4 after the initial first-order proposal, while Algorithm 1 gives the complete procedure.

Step 1: First-Order Direction Estimation. Let $\tilde{L}_i^\ell(\delta)$ denote the layer-local surrogate obtained by intervening at (ℓ, t^*) . For compactness, write $p_{i,c}^{\ell,\delta} = p_\theta^{(\ell,t^*,\delta)}(o_i | x_{i,c})$ and $q_{i,\ell}^\delta = p_\theta^{(\ell,t^*,\delta)}(\cdot | x_i^{\text{KL}})$. We use the same surrogate throughout Steps 1 and 3:

$$\begin{aligned} \tilde{L}_i^\ell(\delta) = & \beta_{\text{NLL}} \sum_{x_{i,c} \in \mathcal{X}_i} \left[-\log p_{i,c}^{\ell,\delta} \right] \\ & + \beta_{\text{KL}} D_{\text{KL}}(q_{i,\ell}^0 \| q_{i,\ell}^\delta). \end{aligned} \quad (2)$$

Here β_{NLL} and β_{KL} weight the rewrite and locality terms. Step 1 does not solve this objective to convergence; it extracts a single normalized first-order

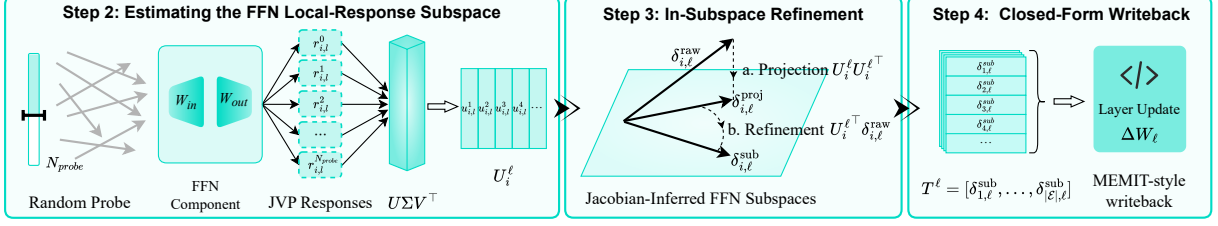


Figure 4: Overview of the PRISM large-batch model editing method for Steps 2–4.

proposal

$$g_{i,\ell} = \nabla_{\delta} \tilde{L}_i^{\ell}(0), \quad \delta_{i,\ell}^{\text{raw}} = -\eta_{\delta} \frac{g_{i,\ell}}{\|g_{i,\ell}\|_2 + \epsilon}. \quad (3)$$

Here η_{δ} is the normalized Step 1 step size, and ϵ is a numerical stabilizer. This yields one high-plasticity proposal per edit–layer pair with a single backward pass, which is key to scalability in large edit batches. Because the surrogate aggregates multiple prefix contexts, the gradient is less sensitive to per-prompt idiosyncrasies; nevertheless, at scale these raw proposals can still contain prompt-specific or weakly supported components. We therefore use $\delta_{i,\ell}^{\text{raw}}$ only as a proposal to be filtered and refined in Steps 2–3; Appendix D.2 verifies that these raw directions are structured rather than arbitrary gradient noise.

Step 2: Estimating the FFN Local-Response Subspace. Step 2 estimates an edit-conditioned FFN basis that captures dominant local-response directions and filters off-subspace, weakly supported components. For each edit–layer pair, we collect N_{probe} JVP responses, stack them into

$$M_{i,\ell}^{\text{bundle}} = [r_{i,\ell}^{(1)}, \dots, r_{i,\ell}^{(N_{\text{probe}})}] = U\Sigma V^{\top}, \quad (4)$$

and retain the leading columns of U until the explained-energy threshold τ_{σ} is reached, yielding an edit-conditioned basis U_i^{ℓ} . This explained-energy truncation is part of the editor construction; it is separate from the Gaussian-noise rank rule used only for the diagnostic geometry in Section 4. Smaller subspaces impose stronger control but can suppress transfer; larger ones recover more plasticity at the risk of reduced Specificity. In practice, the probe budget N_{probe} is the main quality-cost knob, with downstream effects reported in Appendix D.4.

Step 3: In-Subspace Refinement. Given the orthonormal basis U_i^{ℓ} , Step 3 reparameterizes the residual as $\delta = U_i^{\ell} \alpha$. We initialize the subspace

coordinates by orthogonally projecting the raw proposal:

$$\alpha_{i,\ell}^{(0)} = U_i^{\ell \top} \delta_{i,\ell}^{\text{raw}}, \quad \delta_{i,\ell}^{\text{proj}} = U_i^{\ell} \alpha_{i,\ell}^{(0)}. \quad (5)$$

We then restrict the same surrogate to this coordinate space and compute the subspace gradient:

$$\begin{aligned} \phi_{i,\ell}(\alpha) &= \tilde{L}_i^{\ell}(U_i^{\ell} \alpha), \\ g_{i,\ell}^{\alpha} &= \nabla_{\alpha} \phi_{i,\ell}(\alpha) \big|_{\alpha=\alpha_{i,\ell}^{(0)}}. \end{aligned} \quad (6)$$

The refined update is obtained by one normalized gradient step in subspace coordinates, where η_{α} is the Step 3 step size:

$$\begin{aligned} \alpha_{i,\ell}^{(1)} &= \alpha_{i,\ell}^{(0)} - \eta_{\alpha} \frac{g_{i,\ell}^{\alpha}}{\|g_{i,\ell}^{\alpha}\|_2 + \epsilon}, \\ \delta_{i,\ell}^{\text{sub}} &= U_i^{\ell} \alpha_{i,\ell}^{(1)}. \end{aligned} \quad (7)$$

Here $\delta_{i,\ell}^{\text{proj}}$ is the projection-only target used in the ablation, while $\delta_{i,\ell}^{\text{sub}}$ is the final target. Projection removes off-subspace components, and refinement restores capacity without leaving the same subspace constraint.

Step 4: Closed-Form Writeback (Layerwise Allocation). Finally, we form $T^{\ell} = [\delta_{1,\ell}^{\text{sub}}, \dots, \delta_{|\mathcal{E}|,\ell}^{\text{sub}}]$ and apply the MEMIT-style ridge-regularized writeback from Section 3. Step 4 is therefore only the realization stage: the key change in PRISM lies upstream, in how each layer’s target residual is constructed before writeback. Unlike fixed residual spreading, our layerwise targets are data-dependent: they are first semantically directed, then filtered to locally supported FFN directions, and finally refined within the same subspace.

6 Experiments

The empirical sections connect geometry to editing: distributional coupling diagnoses directions associated with larger target-margin/reference-KL changes, local-response geometry supplies the constrained space, matched-norm interventions provide diagnostic associations with behavior, and

Method	Score	Efficacy	Generalization	Specificity	Fluency	Consistency
GPT-J (6B), 10k edits						
<i>Contextual references (not protocol-matched)</i>						
Base model	22.4	15.2 (0.7)	17.7 (0.6)	83.5 (0.5)	622.4 (0.3)	29.4 (0.2)
FT-W	67.6	99.4 (0.1)	77.0 (0.7)	46.9 (0.6)	293.9 (2.4)	15.9 (0.3)
MEND	23.1	15.7 (0.7)	18.5 (0.7)	83.0 (0.5)	618.4 (0.3)	31.1 (0.2)
ROME	50.3	50.2 (1.0)	50.4 (0.8)	50.2 (0.6)	589.6 (0.5)	3.3 (0.0)
<i>Simultaneous large-batch editors (like-for-like)</i>						
MEMIT	85.8	98.9 (0.2)	88.6 (0.5)	73.7 (0.5)	619.9 (0.3)	40.1 (0.2)
PMET	86.2	99.5 (0.1)	92.8 (0.4)	71.4 (0.5)	620.0 (0.3)	40.6 (0.2)
PRISM	87.5	99.7 (0.1)	94.6 (0.4)	73.1 (0.5)	622.6 (0.3)	40.3 (0.2)
GPT-NeoX (20B), 10k edits						
<i>Contextual reference</i>						
Base model	23.7	16.8 (1.9)	18.3 (1.7)	81.6 (1.3)	620.4 (0.6)	29.3 (0.5)
<i>Simultaneous large-batch editors</i>						
MEMIT	82.0	97.2 (0.8)	82.2 (1.6)	70.8 (1.4)	606.4 (1.0)	36.9 (0.6)
PMET	84.3	98.4 (0.2)	89.4 (0.5)	70.3 (0.5)	598.1 (0.6)	38.9 (0.2)
PRISM	86.0	99.5 (0.1)	91.1 (0.5)	72.6 (0.5)	617.1 (0.3)	39.1 (0.2)

Table 1: CounterFact results for 10k edits on GPT-J (6B) and GPT-NeoX (20B).

Method	Efficacy	Generalization	Specificity
GPT-J (6B), 10k edits			
<i>Contextual references (not protocol-matched)</i>			
Base model	26.4 (0.6)	25.8 (0.5)	27.0 (0.5)
FT-W	69.6 (0.6)	64.8 (0.6)	24.1 (0.5)
MEND	19.4 (0.5)	18.6 (0.5)	22.4 (0.5)
ROME	21.0 (0.7)	19.6 (0.7)	0.9 (0.1)
<i>Simultaneous large-batch editors</i>			
MEMIT	96.7 (0.3)	89.7 (0.5)	26.6 (0.5)
PMET	96.9 (0.3)	90.6 (0.2)	26.7 (0.2)
PRISM	98.2 (0.2)	91.1 (0.5)	27.1 (0.5)
GPT-NeoX (20B), 10k edits			
<i>Contextual reference</i>			
Base model	30.2 (0.6)	29.5 (0.6)	31.1 (0.5)
<i>Simultaneous large-batch editors</i>			
MEMIT	91.7 (0.4)	85.8 (0.5)	28.9 (0.5)
PMET	Resource-limited: not completed (OOM)		
PRISM	95.5 (0.3)	89.0 (0.5)	31.3 (0.5)

Table 2: zsRE results for 10k edits on GPT-J (6B) and GPT-NeoX (20B).

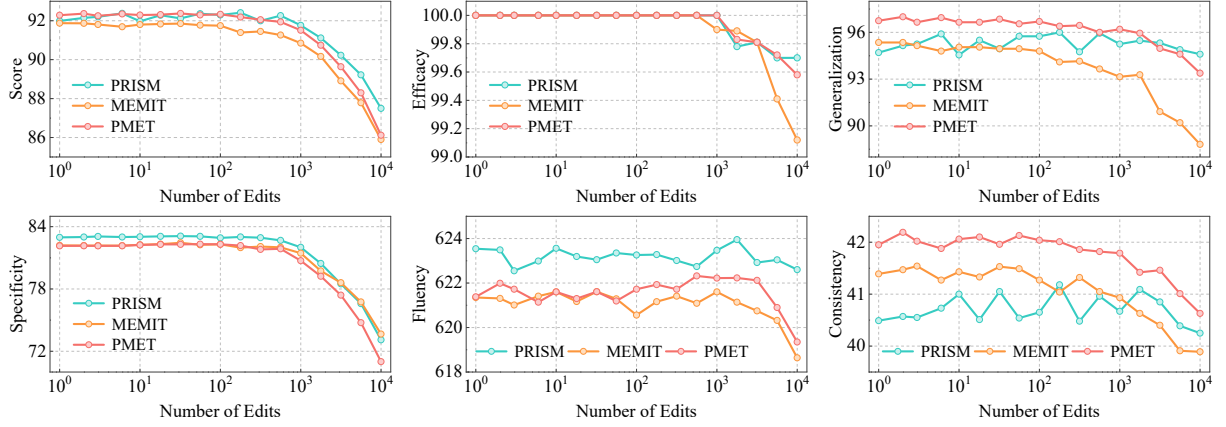


Figure 5: CounterFact scale stability.

ablations separate projection from refinement—projection imposes the subspace constraint while in-subspace refinement restores capacity.

6.1 Experimental Protocol

We evaluate on GPT-J (6B) and GPT-NeoX (20B), comparing like-for-like against the simultaneous large-batch editors MEMIT and PMET. FT-W (weight fine-tuning), MEND, ROME, and the unedited model are separated as contextual references because their procedures are not protocol-matched to simultaneous mass editing. Editing benchmarks are CounterFact and zsRE. Following the MEMIT/PMET mass-editing protocol, CounterFact uses the standard 17-point log-scale schedule from 1 to 10k edits, with 15k/20k stress tests; zsRE is evaluated at 10k. The WSC geometry diagnostics motivate the analysis, while the matched-norm CounterFact intervention assesses whether the coupling statistic is associated with the measured intervention responses; neither is used to benchmark

weight-updating editors. PMET did not complete GPT-NeoX/zsRE because the public implementation exhausted H100 memory during the 10k write-back, so we make no PMET comparison in that setting. To isolate direction selection, we keep the edited-layer sets fixed to those used by MEMIT for each backbone, so our contribution is in target construction within a fixed layer set rather than in layer selection.

Under this fixed-layer protocol, we report Efficacy, Generalization, Specificity, and their harmonic mean Score, with Fluency and Consistency as auxiliary diagnostics. Parenthesized component intervals are request-level 95% CI half-widths. Unless stated otherwise, Step 2 uses $N_{\text{probe}} = 1024$ JVP probes with $\tau_{\sigma} = 0.99$, computed by automatic differentiation in evaluation mode with deterministic seeds. All experiments use NVIDIA H100 cloud instances. Complete setting-specific configurations are reported in Appendix C.1.

6.2 10k Large-Batch Editing Results

Table 1 shows that PRISM attains the highest observed Score among protocol-matched editors on both base models in the evaluated CounterFact setting. On GPT-J, where Efficacy is already near saturation for the strongest editors, the main gain comes from Generalization: PRISM reaches 94.6 versus 92.8 for PMET and 88.6 for MEMIT, while keeping Specificity above PMET and close to MEMIT. On GPT-NeoX, the gain is broader: PRISM attains the best Score and improves all three core metrics over the strongest baselines.

As shown in Table 2, the transfer benefit of PRISM also extends to zsRE. On GPT-J, it achieves the best observed values on all three core metrics; on GPT-NeoX, where PMET is resource-limited, it improves over the completed MEMIT run on Efficacy, Generalization, and Specificity. Appendix C.2 reports protocol-labeled AlphaEdit and Llama-3-8B comparisons: at 10k, PRISM scores 87.49 versus 82.25 for protocol-matched one-shot AlphaEdit and 90.34 versus 86.11 for PMET on Llama-3-8B.

6.3 Scale Stability Tests

Figure 5 characterizes *scale stability*: how editing quality changes as the edit budget n increases and interference becomes denser. These are dedicated scaling runs, so the 10k values can differ slightly from the multi-run aggregates in Table 1. PRISM is close to PMET at small edit budgets, overtakes it as n grows, and by 10k edits reaches the best Score (87.49), mainly through slower Generalization degradation while keeping Specificity above PMET and close to MEMIT. The same ordering persists at 15k and 20k in Figure 6, where PRISM remains strongest overall with the best Score, Efficacy, and Generalization while retaining competitive Specificity. Exact values and request-level intervals are reported in Appendix C.3.

Runtime–quality trade-off. Table 3 reports editing-only wall-clock time and Score at representative 1k, 5.6k, and 10k budgets from GPT-J / CounterFact timing runs on one H100 80 GB; times exclude loading and evaluation. Appendix D.3 gives the full 17-point curves. The Cache entries report reuse time. PRISM-Cache reuses one layerwise basis built from a disjoint 1k-example calibration set; its one-time construction takes 804.5 ± 1.5 s over three complete runs (mean $\pm$ standard deviation). At 10k, full PRISM improves Score over PMET

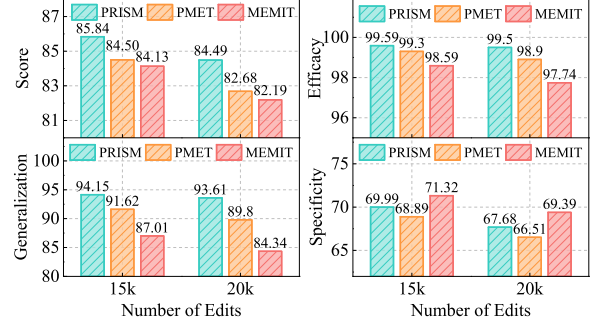


Figure 6: Beyond-10k CounterFact stress tests on GPT-J at 15k and 20k edits.

Table 3: Runtime–quality comparison on GPT-J / CounterFact.

Method	Runtime (ks) ↓			Score ↑		
	1k	5.6k	10k	1k	5.6k	10k
MEMIT	1.33	7.35	13.29	90.85	87.78	85.90
PMET	3.68	20.99	36.60	91.51	88.30	86.12
PRISM-Cache	1.89	10.48	18.87	92.33	88.30	86.19
PRISM	2.57	14.44	25.86	91.76	89.22	87.49

by 1.37 points while using 29% less time; relative to MEMIT, it gains 1.59 points at $1.95\times$ the time. Cache reuse is 27% faster than full PRISM but 1.30 points lower in Score; including construction, its first use takes 19.67 ks, still 24% less time. Thus, full PRISM suits transfer-prioritizing large batches, MEMIT suits strict compute limits, and Cache is a reuse-oriented middle ground.

6.4 Ablation Study

All ablations use CounterFact with 10k edits while fixing dataset, budget, and schedule. We isolate Jacobian projection and in-subspace refinement here; the Step 2 probe-budget sweep is deferred to Appendix D.4.

Projection, refinement, and stage-wise direction structure. Table 4 shows the core interaction. The unconstrained variant is highly plastic but weak on Specificity. Projection improves Specificity but underfits, whereas in-subspace refinement restores Efficacy and much of the lost Generalization while keeping Specificity above the unconstrained variant, producing the best overall Score. Thus, projection imposes the subspace constraint and refinement recovers expressivity within it; auxiliary generation metrics vary only slightly.

To relate this metric interaction to direction structure, we compare δ^{proj} and δ^{sub} after grouping edits by relation. Figure 7 shows that projection

Table 4: Controlled ablation on CounterFact with 10k edits.

Metric	Unconstrained	+Projection	+Refinement
Score	86.11	82.48	87.49
Efficacy	99.70 (0.1)	92.72 (0.5)	99.70 (0.1)
Generalization	97.20 (0.3)	79.84 (0.7)	94.62 (0.4)
Specificity	68.86 (0.5)	76.55 (0.5)	73.05 (0.5)
Consistency	40.81 (0.3)	37.60 (0.2)	40.25 (0.2)
Fluency	617.29 (0.4)	621.12 (0.3)	622.63 (0.3)

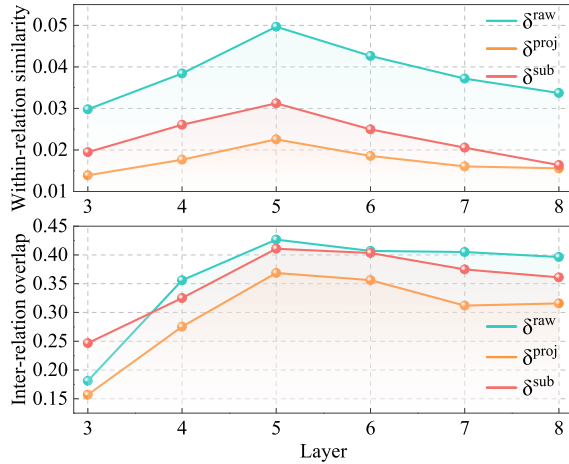


Figure 7: Stage-wise direction structure on CounterFact.

reduces inter-relation overlap but weakens within-relation coherence, while refinement partially restores coherence without fully undoing the separation gain. We treat these statistics as structural diagnostic evidence rather than a causal explanation of the metric gains. Appendix D.2 verifies that raw proposals already contain relation-level structure above shuffled controls.

Basis identity at matched rank. To separate basis identity from subspace capacity, an auxiliary GPT-J / CounterFact 10k study holds the edit requests, context templates, raw proposals, refinement, and writeback fixed and replaces only the projection basis. Table 5 reports changes from the edit-conditioned JVP control within this auxiliary protocol; these deltas are not mixed with the headline aggregates. Horizontal arrows mark the JVP zero reference, while upward and downward arrows indicate improvement and degradation (higher is better). Random averages three orthonormal bases, and both controls use the exact request-layer JVP ranks (mean 883.09).

At matched rank, JVP improves Generalization over Random by 2.08 points, whereas Random improves Specificity and Score by 3.29 and 0.96 points. Activation-PCA yields 0.48 points higher

Table 5: Matched-rank basis controls at 10k edits.

Metric	JVP	Random	Activation-PCA
Efficacy	$\leftrightarrow$ 0.00	$\downarrow$ 0.06	$\downarrow$ 0.02
Generalization	$\leftrightarrow$ 0.00	$\downarrow$ 2.08	$\uparrow$ 0.48
Specificity	$\leftrightarrow$ 0.00	$\uparrow$ 3.29	$\downarrow$ 1.63
Score	$\leftrightarrow$ 0.00	$\uparrow$ 0.96	$\downarrow$ 0.69

Generalization but 1.63 points lower Specificity than JVP. Together, these results show that basis identity changes the transfer-locality allocation even at fixed capacity: the edit-conditioned local-response construction preserves strong transfer under matched conditions, while Random and Activation-PCA yield different operating points. Appendix D.1 reports fixed-rank Random controls, uncertainty, seed variation, and construction details.

Jacobian probe ablation. The probe budget controls transfer-locality: 1024–1536 probes give the best balance on CounterFact 10k, while larger budgets improve Generalization at the cost of Specificity; see Appendix D.4.

7 Conclusion

Large-batch knowledge editing must sustain transfer as edit interactions densify without sacrificing locality. PRISM addresses this by constructing FFN targets in Jacobian-inferred local-response subspaces: distributional coupling diagnoses directions associated with larger target-margin/reference-KL changes, matched-norm interventions provide diagnostic behavioral evidence, and ablations show that projection imposes the subspace constraint while in-subspace refinement restores capacity. Within the evaluated CounterFact and zsRE settings, this design trades additional computation for stronger Generalization while maintaining competitive Specificity.

Limitations

Dense scale sweeps and the 15k/20k stress tests are centered on CounterFact; zsRE is evaluated only at 10k, and we have not completed protocol-aligned experiments on broader benchmarks such as KnowEdit or ReCoE. Consequently, the results do not establish broad cross-dataset generality or behavior under denser editing, longer contexts, or distribution shift. Standard Efficacy, Generalization, Specificity, neighborhood, and reference-KL measurements also do not exhaustively test post-

edit capabilities or ripple effects over related facts (Gu et al., 2024; Cohen et al., 2024). Finally, edit-conditioned JVP estimation adds substantial compute, and the cached approximation shifts the Generalization–Specificity trade-off while incurring an additional one-time construction cost.

Ethical Considerations

More controlled knowledge editing can be useful for localized factual repair without full retraining, but the same capability can also be used to alter model behavior at scale in ways that are difficult to audit. For that reason, stronger editing methods should be paired with careful logging, downstream evaluation, and access controls rather than treated as inherently benign.

Acknowledgments

We sincerely thank the anonymous reviewers and area chairs for their careful reading and constructive feedback, which helped improve the clarity and experimental presentation of this work. This work was supported by the National Cyber Security-National Science and Technology Major Project (2025ZD1502700) and the National Natural Science Foundation of China under Grant No. 62602604.

References

- Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, Usvsn Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. [GPT-NeoX-20b: An open-source autoregressive language model](#). In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 95–136.
- Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024. [Evaluating the ripple effects of knowledge editing in language models](#). *Transactions of the Association for Computational Linguistics*, 12:283–298.
- Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. [Knowledge neurons in pretrained transformers](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502.
- Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. [Quoref: A reading comprehension dataset with questions requiring coreferential reasoning](#). 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)*, pages 5925–5932.
- Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. [Editing factual knowledge in language models](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506.
- Kawin Ethayarajh. 2019. [How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings](#). 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)*, pages 55–65.
- Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2025. [Alphaedit: Null-space constrained knowledge editing for language models](#). In *The Thirteenth International Conference on Learning Representations*.
- Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. [Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45.
- Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. [Transformer feed-forward layers are key-value memories](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 540 others. 2024. [The llama 3 herd of models](#). Preprint, arXiv:2407.21783.
- Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. 2024. [Model editing harms general abilities of large language models: Regularization to the rescue](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16801–16819.
- Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. 2024. [Model editing at scale leads to gradual and catastrophic forgetting](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15202–15232.
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting

- Liu. 2025. [A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions](#). *ACM Transactions on Information Systems*, 43(2):1–55.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. [Survey of hallucination in natural language generation](#). *ACM Computing Surveys*, 55(12):1–38.
- James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. [Overcoming catastrophic forgetting in neural networks](#). *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.
- Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, pages 552–561.
- Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. [Zero-shot relation extraction via reading comprehension](#). In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342.
- Qi Li and Xiaowen Chu. 2025. [AdaEdit: Advancing continuous knowledge editing for large language models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4127–4149.
- Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2024. [Pmet: Precise model editing in a transformer](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18564–18572.
- Michael McCloskey and Neal J. Cohen. 1989. [Catastrophic interference in connectionist networks: The sequential learning problem](#). In *Psychology of Learning and Motivation*, volume 24, pages 109–165. Academic Press.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. [Locating and editing factual associations in gpt](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372.
- Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. 2023. [Mass-editing memory in a transformer](#). In *The Eleventh International Conference on Learning Representations*.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. [Pointer sentinel mixture models](#). In *International Conference on Learning Representations*.
- Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022a. [Fast model editing at scale](#). In *International Conference on Learning Representations*.
- Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. 2022b. [Memory-based model editing at scale](#). In *Proceedings of the 39th International Conference on Machine Learning*, pages 15817–15831.
- Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, and 7 others. 2022. [In-context learning and induction heads](#). *arXiv preprint arXiv:2209.11895*.
- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. [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)*, pages 2463–2473.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#).
- Chenmian Tan, Ge Zhang, and Jie Fu. 2024. [Massive editing for large language models via meta learning](#). In *The Twelfth International Conference on Learning Representations*.
- Lucrezia Valeriani, Diego Doimo, Francesca Cuturello, Alessandro Laio, Alessio Ansuini, and Alberto Cazzaniga. 2023. [The geometry of hidden representations of large transformer models](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 51234–51252. Curran Associates, Inc.
- Ben Wang and Aran Komatsuzaki. 2021. [GPT-J-6b: A 6 billion parameter autoregressive language model](#).
- Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2024. [Knowledge editing for large language models: A survey](#). *ACM Computing Surveys*, 57(3):1–37.
- Haoyu Xu, Pengxiang Lan, Enneng Yang, Guibing Guo, Jianzhe Zhao, Linying Jiang, and Xingwei Wang. 2025. [Knowledge decoupling via orthogonal projection for lifelong editing of large language models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13194–13213.

A Additional Geometry Results

To keep the main text focused, §4 centers on the representative parallel-block GPT-J / WSC case under the Gaussian-noise rank policy. This appendix checks whether the same geometry persists across extraction interfaces, additional model/dataset settings, and rank/control choices. Unless stated otherwise, all analyses use the same pre-residual MHSA/FFN outputs, semantic-span token pool, Gaussian-noise / top-5 principal-angle summary, and token-level mean-angle baseline as in the main text.

A.1 Extraction Interface

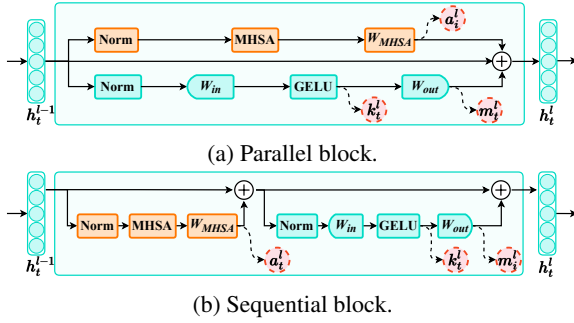


Figure 8: Extraction points for MHSA and FFN component outputs in parallel and sequential Transformer blocks.

Figure 8 shows the component-output extraction interface used for the geometry diagnostics. For both parallel and sequential blocks, we record MHSA and FFN outputs after their output projections and before dropout/residual addition, matching the notation in Section 3.

A.2 Token-Level and Layerwise Geometry Across Settings

Figure 9 shows the token-level cosine distributions for the representative GPT-J / WSC setting, complementing the aggregate alignment statistics reported in Section 4. The remaining three settings are summarized by a single token-level figure, Figure 10, which collects the absolute-cosine distributions for GPT-J / Quoref, GPT-2 XL / WSC, and GPT-2 XL / Quoref. Their corresponding layerwise geometry plots appear in Figures 11 to 13.

A.3 Cross-Setting Geometry Summary

The columns in Table 6 report the token-level mean angle, the across-layer means of the distributional and local-response top-5 angles, and the number of

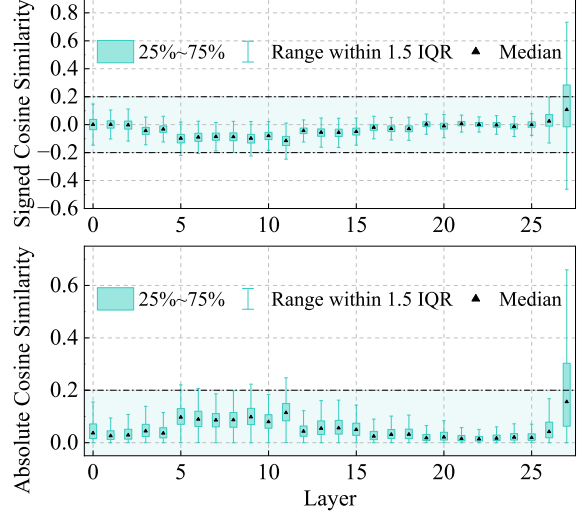


Figure 9: GPT-J / WSC token-level FFN-MHSA cosine distributions.

Table 6: Cross-setting geometry summary under the Gaussian-noise rank policy.

Setting	Token angle	Dist. top-5 angle	Local top-5 angle	Layers local > dist.
GPT-J / WSC	86.67°	40.53°	57.68°	28/28
GPT-J / Quoref	86.96°	41.12°	58.00°	28/28
GPT-2 XL / WSC	83.00°	32.88°	47.80°	48/48
GPT-2 XL / Quoref	83.10°	31.53°	43.34°	45/48

layers where the local-response angle exceeds the distributional one. Across all four settings, local-response angles are consistently larger than distributional angles, supporting the main-text claim that the two geometries capture distinct structure.

A.4 Rank and Control Robustness

This subsection tests whether the main ordering depends on the Gaussian-noise rank policy or on same-layer comparisons alone. We compare the default Gaussian-noise rank policy with two alternatives: 90% EV retains the minimum number of singular directions whose cumulative squared singular values explain at least 90% of the total energy, while Fixed- $k = 32$ keeps the top 32 singular directions regardless of explained energy. Rows labeled Noise use the default Gaussian-noise rank policy: the retained rank is the number of empirical singular values exceeding the corresponding threshold from same-shape Gaussian matrices scaled to the empirical standard deviation. Alternative rank rules and top-10 summaries are sensitivity checks; since they preserve the same distributional-local-response ordering, the tables report the top-5 sum-

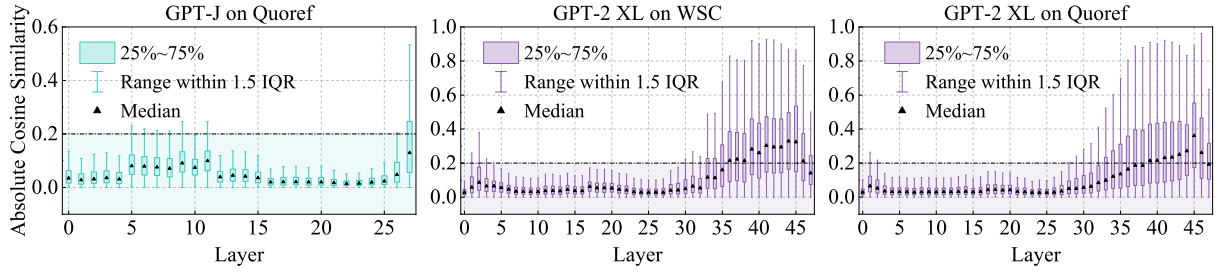


Figure 10: Token-level absolute-cosine distributions on the three additional settings.

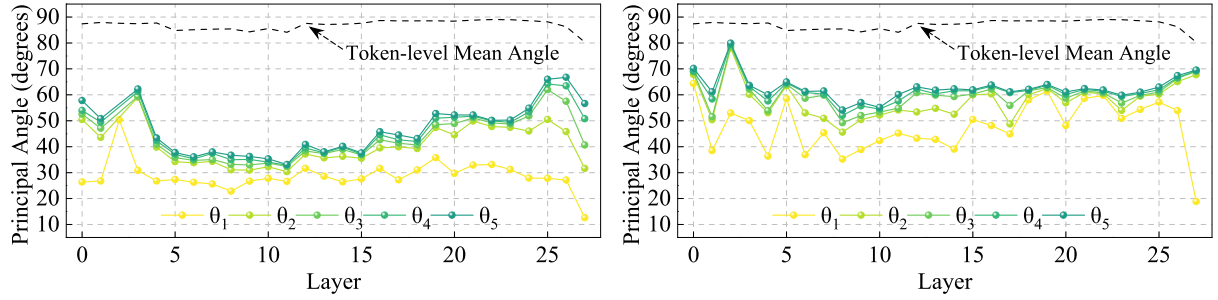


Figure 11: GPT-J on Quoref: distributional and local-response geometry.

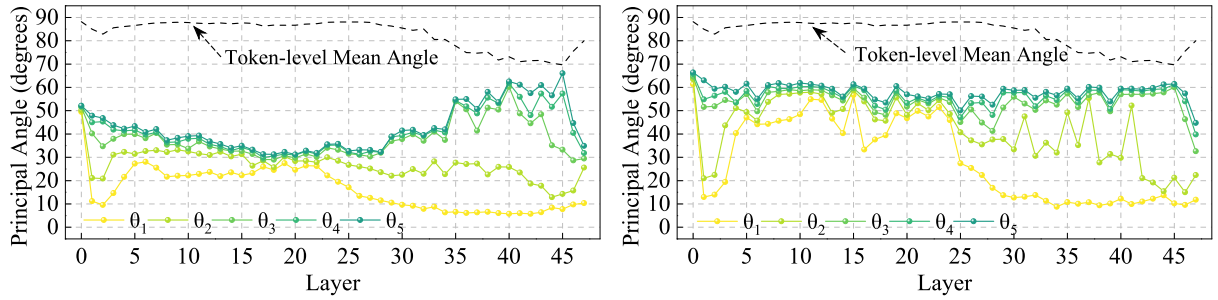


Figure 12: GPT-2 XL on WSC: distributional and local-response geometry.

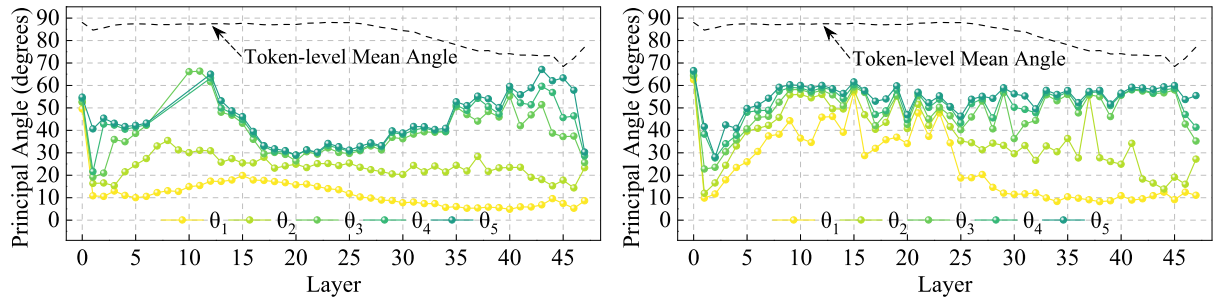


Figure 13: GPT-2 XL on Quoref: distributional and local-response geometry.

maries.

The rank-policy sensitivity checks and control comparisons appear in Tables 7 and 8, and the corresponding effective-rank and split-half summaries appear in Table 9. Here, *random-null* compares each observed MHSA-FFN pair against same-rank random orthonormal bases in the same ambient dimension, asking whether the observed overlap is stronger than chance given rank alone. *Cross-layer*

is a distance-matched, rank-matched control: for each layer, the MHSA subspace is compared with FFN subspaces from other layers within the same layer-distance buckets, so that the reported gap reflects same-layer specificity rather than arbitrary cross-layer overlap.

In Tables 7 and 8, ‘Obs. angle’ denotes the across-layer mean of the observed top-5 angles, ‘Rand.-null’ and ‘Cross-layer’ denote the corre-

Table 7: Distributional geometry robustness controls.

Setting / policy	Obs. angle	Rand.-null	Cross-layer	Frac. $< q_{0.05}$	Frac. $< \text{cross}$
GPT-J / WSC / Noise	40.53	56.08	53.85	1.000	0.922
GPT-J / WSC / 90% EV	31.70	42.64	47.97	1.000	0.926
GPT-J / WSC / Fixed- $k = 32$	64.27	81.41	73.72	1.000	0.932
GPT-J / Quoref / Noise	41.12	58.33	59.74	1.000	0.888
GPT-J / Quoref / 90% EV	29.28	41.60	51.29	1.000	0.870
GPT-J / Quoref / Fixed- $k = 32$	62.69	81.41	72.88	1.000	0.936
GPT-2 XL / WSC / Noise	32.88	51.28	45.93	1.000	0.921
GPT-2 XL / WSC / 90% EV	20.16	30.35	36.78	0.979	0.921
GPT-2 XL / WSC / Fixed- $k = 32$	53.51	76.22	62.08	1.000	0.883
GPT-2 XL / Quoref / Noise	31.53	56.19	48.60	1.000	0.886
GPT-2 XL / Quoref / 90% EV	18.67	33.13	39.59	0.979	0.885
GPT-2 XL / Quoref / Fixed- $k = 32$	47.79	76.22	55.61	1.000	0.933

Table 8: Local-response geometry robustness controls.

Setting / policy	Obs. angle	Rand.-null	Cross-layer	Frac. $< q_{0.05}$	Frac. $< \text{cross}$
GPT-J / WSC / Noise	57.68	65.56	82.66	0.964	0.920
GPT-J / WSC / 90% EV	44.92	47.20	85.96	0.821	0.917
GPT-J / WSC / Fixed- $k = 32$	71.22	81.41	76.52	1.000	0.909
GPT-J / Quoref / Noise	58.00	65.72	82.63	0.929	0.928
GPT-J / Quoref / 90% EV	43.70	46.01	85.89	0.821	0.894
GPT-J / Quoref / Fixed- $k = 32$	68.81	81.41	75.49	1.000	0.904
GPT-2 XL / WSC / Noise	47.80	61.56	76.80	1.000	0.921
GPT-2 XL / WSC / 90% EV	39.69	42.46	82.43	0.979	0.913
GPT-2 XL / WSC / Fixed- $k = 32$	57.46	76.22	66.51	1.000	0.904
GPT-2 XL / Quoref / Noise	43.34	60.41	74.43	1.000	0.909
GPT-2 XL / Quoref / 90% EV	37.88	40.54	82.19	0.958	0.897
GPT-2 XL / Quoref / Fixed- $k = 32$	52.27	76.22	63.12	1.000	0.928

Table 9: Effective-rank and split-half summaries.

Analysis	Setting	Eff. rank (MHSA)	Eff. rank (FFN)	Split-half (MHSA)	Split-half (FFN)
Distributional	GPT-J / WSC	268.7	646.7	0.858	0.770
Distributional	GPT-J / Quoref	387.1	691.0	0.807	0.767
Distributional	GPT-2 XL / WSC	164.4	415.3	0.896	0.804
Distributional	GPT-2 XL / Quoref	175.7	346.8	0.867	0.815
Local-response	GPT-J / WSC	11.6	1000.0	0.999	0.918
Local-response	GPT-J / Quoref	11.8	1071.7	0.999	0.908
Local-response	GPT-2 XL / WSC	13.1	548.8	0.999	0.952
Local-response	GPT-2 XL / Quoref	13.9	615.9	0.998	0.942

sponding control means, ‘Frac. $< q_{0.05}$ ’ reports the fraction of layers below the random-null 5th percentile, and ‘Frac. $< \text{cross}$ ’ reports the fraction below the cross-layer mean. This random-null percentile is a robustness threshold only; it is distinct from the Gaussian-noise rank rule used to select subspace dimensions in the main geometric analysis. In Table 9, the effective-rank columns use entropy effective rank computed from normalized squared singular values, and the split-half columns report mean subspace cosines under random half-splits of the retained-token or JVP-response pools.

Together, these tables support the main-text geometry claims under alternative rank rules and explicit stability controls.

B Intervention Sensitivity Checks

This appendix reports sensitivity checks for the main-text matched-norm diagnostic experiment. Table 10 gives paired high-minus-low differences under alternative top- k summaries and rank rules. Across all variants, high-coupling layers produce larger target-side changes than low-coupling layers; the reference-KL gap remains positive but varies in

magnitude, so we use the intervention as directional evidence rather than as a calibrated effect-size estimate.

For all rows in Table 10, the high- and low-coupling groups are formed from the three most and three least coupled valid layers after excluding layers whose usable rank is insufficient for the chosen top- k summary. The reported ΔTarget values are paired over intervention cases and use the absolute target-margin change, whereas ΔRefKL first averages over matched reference prompts within each coupling group and is then paired at the reference-prompt level.

Table 10: Intervention sensitivity checks.

Setting	ΔTarget	ΔRefKL
Noise / top-5	0.1003	0.00555
Noise / top-10	0.0434	0.000610
90% EV / top-5	0.0883	0.00543
90% EV / top-10	0.0912	0.00549

C Hyperparameters and Additional Comparisons

C.1 Hyperparameter Configurations

Table 11 reports the configuration used for each backbone–dataset setting. “Layers” specifies the edited layer set $\mathcal{L}$; N_{probe} and τ_{σ} control the JVP basis construction in Step 2; η_{δ} and η_{α} are the normalized step sizes in Steps 1 and 3, respectively; and λ weights the precomputed second-moment matrix in the MEMIT-style writeback of Step 4. Each row is selected by validation Score on a separate development set and then kept fixed across edit scales.

C.2 Protocol-Labeled Baseline and Backbone Results

Tables 12 and 13 report request-level 95% CI half-widths in parentheses; Score intervals use a joint request bootstrap.

Table 12 compares with AlphaEdit (Fang et al., 2025), which constrains the final parameter update through a preserved-knowledge null space rather than constraining residual targets before writeback. We use its official update and hyperparameters on the same GPT-J / CounterFact requests. The sequential row applies 100-edit chunks and is contextual; only the one-shot 10k row is protocol-matched to the one-shot PRISM setting.

Table 13 evaluates a more recent decoder-only backbone, Meta-Llama-3-8B-Instruct (Grattafiori et al., 2024), using shared 1k/10k CounterFact subsets and separately selected configurations fixed across the two scales. PRISM has the highest observed 10k Score and smallest observed drop, while MEMIT retains the highest Specificity. This is a backbone-transfer result, not a substitute for broader benchmark coverage.

C.3 Beyond-10k Stress-Test Results

Table 14 gives the complete component metrics and uncertainty estimates corresponding to the point estimates visualized in Figure 6. Parenthesized values are request-level 95% CI half-widths: component-metric intervals are SE-based, while Score uses a joint request bootstrap.

D Additional Ablation and Efficiency Details

This section complements the main ablation with basis-identity and rank controls, direction-structure diagnostics, the Step 2 probe-budget analysis, and runtime and cached-subspace results.

D.1 Basis-Identity and Rank Controls

The matched-rank study in Table 5 changes only the projection basis. Random bases are sampled as orthonormal bases and averaged over three seeds. Activation-PCA is estimated by layerwise SVD of centered, context-averaged FFN outputs from a disjoint 1k-example calibration slice, then truncated for each request and layer to the corresponding JVP rank. Table 15 adds two fixed-rank Random controls to show how capacity changes the same transfer-locality allocation. All metric entries are changes relative to the edit-conditioned JVP control; upward and downward arrows indicate improvement and degradation (higher is better), while horizontal arrows mark the zero reference. The final column reports request-level 95% CI half-widths for the corresponding absolute end-point estimates, in the same order as the four metric columns; these intervals do not quantify uncertainty in the displayed changes. For the three Random controls, the corresponding between-basis SDs, in the same metric order, are 0.02/0.14/0.14/0.11 at exact JVP ranks, 0.01/0.01/0.21/0.10 at fixed $k = 1,424$, and 0.01/0.08/0.04/0.02 at fixed $k = 2,304$.

Increasing the Random rank to 1,424 (61.3% above the matched mean) makes Generalization

Table 11: PRISM hyperparameters by backbone and dataset.

Setting	Layers	N_{probe}	τ_{σ}	η_{δ}	η_{α}	$(\beta_{\text{NLL}}, \beta_{\text{KL}})$	λ
GPT-J / CounterFact	3–8	1024	0.99	0.48	0.48	(1, 0.0625)	36k
GPT-J / zsRE	3–8	1024	0.99	0.47	0.47	(1, 0.0625)	30k
GPT-NeoX / CounterFact, zsRE	6–10	2048	0.99	0.52	0.52	(1, 0.0625)	40k
Llama-3-8B / CounterFact	4–8	1024	0.99	0.56	0.56	(1, 0.0625)	18k

Table 12: AlphaEdit comparison on GPT-J / CounterFact at 10k edits.

Method	Protocol	Efficacy	Generalization	Specificity	Score
AlphaEdit	sequential (100 × 100)	94.89 (0.43)	81.05 (0.64)	64.37 (0.55)	78.10 (0.32)
AlphaEdit	one-shot 10k	98.93 (0.20)	85.82 (0.58)	67.97 (0.56)	82.25 (0.30)
PRISM	one-shot 10k	99.70 (0.11)	94.62 (0.36)	73.05 (0.52)	87.49 (0.26)

Table 13: Meta-Llama-3-8B-Instruct results on CounterFact at 1k and 10k edits.

Method	Efficacy @10k	Generalization @10k	Specificity @10k	Score @1k	Score @10k	Drop
MEMIT	93.98 (0.47)	73.68 (0.75)	84.21 (0.40)	89.23 (0.92)	83.13 (0.38)	6.10
PMET	96.39 (0.37)	84.94 (0.61)	78.79 (0.47)	92.45 (0.69)	86.11 (0.30)	6.34
PRISM	99.22 (0.17)	91.32 (0.46)	82.10 (0.43)	93.79 (0.59)	90.34 (0.23)	3.45

Table 14: GPT-J / CounterFact stress-test results at 15k and 20k edits.

Edits	Method	Efficacy	Generalization	Specificity	Score
15k	MEMIT	98.59 (0.19)	87.01 (0.46)	71.32 (0.44)	84.13 (0.23)
	PMET	99.30 (0.13)	91.62 (0.37)	68.89 (0.42)	84.50 (0.23)
	PRISM	99.59 (0.10)	94.15 (0.31)	69.99 (0.43)	85.84 (0.23)
20k	MEMIT	97.74 (0.21)	84.34 (0.43)	69.39 (0.39)	82.19 (0.22)
	PMET	98.90 (0.14)	89.80 (0.35)	66.51 (0.37)	82.68 (0.21)
	PRISM	99.50 (0.10)	93.61 (0.28)	67.68 (0.38)	84.49 (0.20)

Table 15: Full basis-identity and rank controls on GPT-J / CounterFact at 10k edits.

Basis	Rank policy	Efficacy	Generalization	Specificity	Score	Endpoint CI half-widths
Edit-conditioned JVP	adaptive (mean 883.09)	$\leftrightarrow$ 0.00	$\leftrightarrow$ 0.00	$\leftrightarrow$ 0.00	$\leftrightarrow$ 0.00	0.10 / 0.31 / 0.51 / 0.26
Random (3-basis mean)	exact JVP ranks	$\downarrow$ 0.06	$\downarrow$ 2.08	$\uparrow$ 3.29	$\uparrow$ 0.96	0.11 / 0.38 / 0.51 / 0.25
Activation-PCA	exact JVP ranks	$\downarrow$ 0.02	$\uparrow$ 0.48	$\downarrow$ 1.63	$\downarrow$ 0.69	0.10 / 0.28 / 0.52 / 0.27
Random (3-basis mean)	fixed $k = 1,424$	$\downarrow$ 0.01	$\uparrow$ 0.13	$\uparrow$ 1.98	$\uparrow$ 0.99	0.10 / 0.30 / 0.51 / 0.25
Random (3-basis mean)	fixed $k = 2,304$	$\uparrow$ 0.02	$\uparrow$ 1.70	$\downarrow$ 0.17	$\uparrow$ 0.38	0.09 / 0.22 / 0.52 / 0.27

comparable to JVP, while rank 2,304 ($2.61 \times$ the matched mean) exceeds JVP Generalization by 1.70 points as Specificity decreases slightly. Rank therefore matters, but the exact-rank comparison still shows that direction choice changes how the same capacity is allocated. In this auxiliary setup, fixed- k Random is approximately 19% faster than the edit-conditioned JVP construction, making it a competitive lower-cost alternative rather than merely a negative control. A cached JVP control uses the same disjoint 1k calibration slice and retains 99% response energy (mean rank 3,797); its $\Delta(E, G, S, \text{Score}) =$

(+0.01, +2.60, −3.62, −1.16). Because it is not capacity matched, we exclude it from Table 15 and treat it as a runtime–quality approximation in Appendix D.3.

D.2 Direction-Structure Diagnostics

The stage-wise analysis in Figure 7 uses CounterFact edits grouped by relation ID at each edited layer. For every stage (δ^{raw} , δ^{proj} , δ^{sub}), we first row-normalize the edit directions within each relation group. The raw-direction signal in Figure 14 is measured by the explained-variance ratio of the top principal components (PCs) within each group, with split-half reproducibility computed against

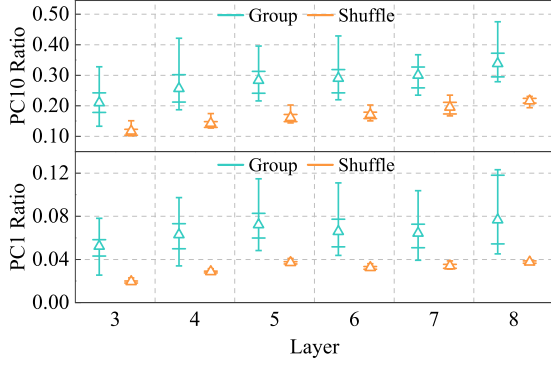


Figure 14: Raw-direction relation signal on CounterFact.

a shuffled matched-size baseline. For the stage-wise trade-off plots, *within-relation similarity* is the mean pairwise cosine within each relation group, aggregated across groups by relation-group size, and *inter-relation overlap* is the mean absolute cosine between normalized relation-group centroids. We exclude zero or near-zero raw directions and skip any relation group with fewer than 100 remaining examples at a layer, so that the directional statistics are computed only where the group structure is well-defined. For the raw-direction signal, PC1 signal is roughly $1.9\text{--}2.7\times$ the shuffled baseline across edited layers, PC10 signal remains about $1.5\text{--}1.8\times$ higher, and top-10 split-half stability is around $0.53\text{--}0.65$ versus $0.33\text{--}0.43$ for shuffled groups. For the stage-wise trade-off in Figure 7, projection reduces the average within-relation similarity from 0.0386 to 0.0174 and inter-relation overlap from 0.362 to 0.297; the refined stage restores within-relation similarity to 0.0231 while keeping overlap below the raw stage on five of six edited layers.

D.3 Runtime Cost and Cached-Subspace Performance

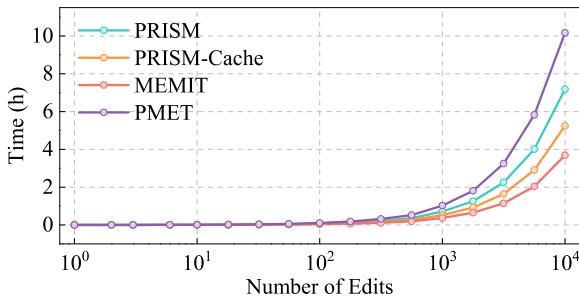


Figure 15: Wall-clock time across the 17-point CounterFact edit schedule on GPT-J.

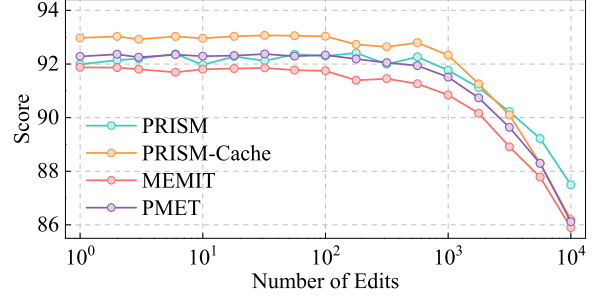


Figure 16: Score across the 17-point CounterFact edit schedule on GPT-J.

The main additional cost of PRISM comes from the edit-conditioned JVP probes used to estimate local-response subspaces in Step 2. We report wall-clock time on GPT-J / CounterFact across the standard 17-point edit schedule under the same experimental setup as the main editing results. To separate the cost of subspace estimation from the benefit of subspace-constrained targets, we also evaluate PRISM-Cache, an efficiency-oriented approximation that estimates one layerwise projection matrix from a disjoint 1k-sample calibration set and reuses it for all edit requests. Because these cached bases are no longer edit-conditioned, PRISM-Cache is not our default method; it is a diagnostic approximation for the runtime–editing-quality trade-off.

Figures 15 and 16 show that cached subspaces reduce a substantial fraction of the additional runtime overhead while preserving most of the editing quality at moderate edit budgets. Averaged across the schedule, PRISM-Cache reduces reuse-stage runtime by 20.8% relative to the full edit-conditioned variant; at 10k edits, wall-clock time decreases from 25.9 ks to 18.9 ks (-27.0%). From 3 edits onward, the reuse stage is also faster than PMET, although MEMIT remains the fastest method.

The Cache column in Table 3 excludes the one-time 1k-basis construction. Let T_{build} denote this one-time construction cost. CUDA-synchronized instrumentation gives $T_{\text{build}} = 804.5 \pm 1.5$ s over three complete builds on one H100 80 GB (mean $\pm$ standard deviation; coefficient of variation 0.19%). The timed interval includes per-layer MLP-input extraction, local JVP-basis construction, global SVD, CPU transfer, and cache-file writes, while excluding model, tokenizer, dataset, and template setup. This construction cost is the same at every edit scale. If the cache has not yet been built, adding the common $T_{\text{build}} = 0.8045$ ks to the scale-specific reuse times of 1.89 ks, 10.48 ks,

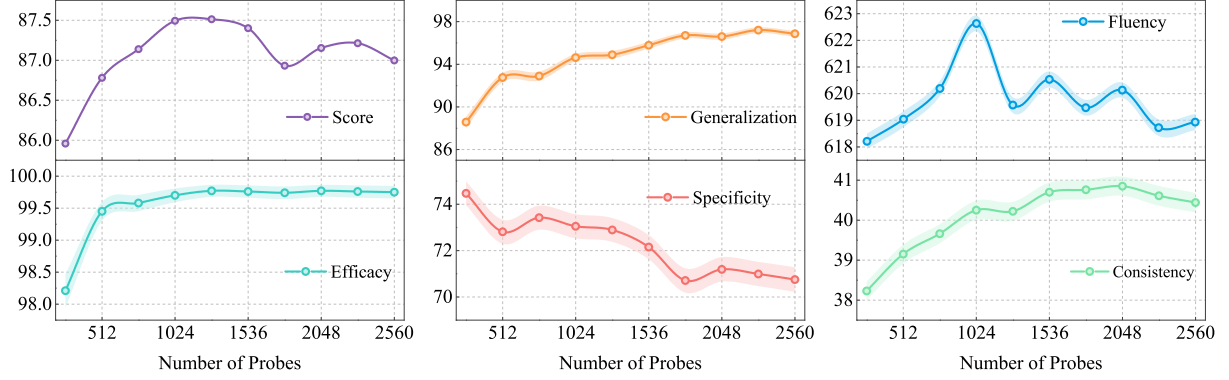


Figure 17: Jacobian probe-count sweep on CounterFact.

and 18.87 ks gives total first-use costs of 2.69 ks, 11.28 ks, and 19.67 ks at 1k, 5.6k, and 10k edits, respectively. For m workloads, let R_j denote the reuse-stage time of workload j ; the average per-workload cost is $T_{\text{build}}/m + \frac{1}{m} \sum_{j=1}^m R_j$.

The editing-quality trade-off becomes more pronounced as the edit batch grows. In the low-to-medium regime, cached subspaces remain competitive and are slightly better at 1k edits (92.33 vs. 91.76 Score). At 10k edits, however, the full edit-conditioned variant is stronger (87.49 vs. 86.19 Score), suggesting that request-specific local-response geometry becomes more important as edit interactions densify. These results support using cached subspaces as a practical runtime approximation, while retaining edit-conditioned subspaces as the default choice when large-batch transfer-locality control is the primary objective.

D.4 Probe-Budget Sweep

To complement the main ablation in Section 6.4, we vary the Step 2 JVP probe budget while keeping the rest of the pipeline fixed.

Figure 17 shows a stable transfer-locality trade-off: 1024–1536 probes provide the best overall balance on CounterFact 10k, whereas larger budgets continue to improve Generalization at the cost of Specificity.

E Artifact Statement

We use third-party scientific artifacts, including pre-trained language models, English-language benchmarks, diagnostic datasets, and baseline implementations. We also provide an accompanying software artifact of PRISM through the conference Software field, including the implementation and evaluation configurations. We do not collect new human-subject data or release a new dataset. Table 16

summarizes the main artifacts used or provided in this work.

Table 16: Scientific artifacts used or provided.

Artifact	License / terms
GPT-J-6B (Wang and Komatsuzaki, 2021)	Apache-2.0
GPT-NeoX-20B (Black et al., 2022)	Apache-2.0
GPT-2 XL (Radford et al., 2019)	Modified MIT
Meta-Llama-3-8B-Instruct (Grattafiori et al., 2024)	Llama 3 Community License
CounterFact (Meng et al., 2022)	Research release
zsRE (Levy et al., 2017)	Research release
WSC (Levesque et al., 2012)	CC-BY
Quoref (Dasigi et al., 2019)	CC-BY-4.0
WikiText (Merity et al., 2017)	CC BY-SA 3.0 / GFDL
ROME code (Meng et al., 2022)	MIT
MEMIT code (Meng et al., 2023)	MIT
PMET code (Li et al., 2024)	No explicit license
AlphaEdit code (Fang et al., 2025)	MIT
PRISM	MIT

The pretrained models are used as editing backbones or for geometry diagnostics; the editing benchmarks are CounterFact and zsRE, and WSC and Quoref support the geometry diagnostics. CounterFact follows the standard 17-point schedule from 1 to 10k edits, with 15k/20k stress tests; zsRE is evaluated at 10k edits. The diagnostic datasets use all 804 WSC examples and a fixed 1,000-example Quoref validation subset. Wiki-

Text is used only to estimate the precomputed second-moment matrix for the writeback solver. The accompanying software artifact contains our PRISM implementation, evaluation scripts, configurations, reproduction documentation, and the included ROME and MEMIT baseline implementations. PMET and AlphaEdit are used as external public implementations for baseline comparison. PMET is not redistributed because its repository does not specify an explicit license at the time of writing; the local AlphaEdit runner changes only run control and cache paths, not the update rule.

Intended use and redistribution. Third-party artifacts are used for research evaluation and analysis only, consistently with their public benchmark, model-release, or research-code purposes. The accompanying software artifact is intended for research reproduction and analysis. We do not redistribute modified versions of the original datasets or pretrained models. The software artifact documents external artifact dependencies and will not package third-party datasets or pretrained models whose licenses require separate access.

Privacy and content. The datasets may contain naturally occurring entity names because factual editing and coreference benchmarks require named entities. We do not scrape new text, introduce new personal data, infer protected attributes, or report individual-level results. No additional anonymization is applied to benchmark examples because entity names are part of the task definition; results are reported only in aggregate.

Documentation and coverage. We document both the data artifacts used in the experiments and the accompanying software artifact. All language data used in this work is English. CounterFact and zsRE provide factual-editing examples, WSC and Quoref provide coreference-style diagnostic examples, and WikiText is used only for covariance estimation. The subset sizes used in our experiments are reported above and in the main experimental protocol. The software artifact includes a README, quick-start instructions, a code-structure overview, dependency requirements, configuration files, and scripts for reproducing the main experiments.