CANONEDIT: CANONICAL QUOTIENT EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Sequential factual editing asks models to absorb requested facts over a long update stream while preserving behavior on the remaining knowledge. Yet each target specifies what behavior should change, not which parameter update should realize it. Many updates can write the same fact, but their side effects differ and can accumulate; the update retained at each step therefore shapes how long the model remains editable. We introduce CANONEDIT, which groups updates with the same target action into quotient-space equivalence classes, measures their cost on history and unrelated prompts, and commits the least-cost representative that passes nonlinear success and protection-budget checks. This quotient-space representative selection is a target-constrained Bregman projection with a closed-form KKT solution. We prove that CANONEDIT selects the successful local update with the smallest predicted protected change; under a stated budget, its feasible set retains every feasible hard-null update in the same trust region and admits additional bounded-cost directions. Across three datasets and four backbone families, seven 10,000-request streams achieve 99.24-100% online writing efficacy, 0-76 failed writes, and 99.97-100% measured specificity. These results show that quotient-space representative selection can control cumulative interference and keep models receptive to new knowledge over long update streams.

1 INTRODUCTION

Facts continue to change after a language model has been trained. Retraining for every correction is expensive, so model editing seeks a smaller intervention: change the answer to one requested fact while preserving measured behavior elsewhere. In a persistent stream, every accepted update changes the state from which the next request must be written. Sequential editing therefore asks not only whether the current fact can be written, but whether the model remains writable after many such updates.

Prior work addresses three parts of this process. Target-construction methods identify where to edit and what hidden state should encode the new fact (Meng et al., 2022a;b; Liu et al., 2026c). Protection methods restrict or penalize changes to selected model behavior (Fang et al., 2025; Liu et al., 2026a; Ikram et al., 2026). Sequential methods stabilize repeated updates, choose editable locations, or separate old and new memories (Wang et al., 2024a; Jiang et al., 2024; Gu et al., 2025; Ma et al., 2026; Liu et al., 2026b; Wang et al., 2026). These choices determine the target, the editable coordinates, and the behavior to measure. One question remains implicit: when several parameter updates all write the requested fact, which one should become part of the persistent model?

A factual request specifies which answer should change but imposes only a few constraints on a high-dimensional parameter space, so many parameter updates can realize the same target behavior. For example, suppose an edit changes the answer to "The headquarters of the European Space Agency are in" from "London" to "Paris". Two updates may both make "Paris" preferred to "London" for this prompt, while one changes scores on many earlier facts and the other leaves them almost unchanged. This underdetermination makes the choice among successful updates consequential in a persistent stream. Optimizing only the target does not distinguish these updates, so unnecessary changes can accumulate. Exact null-space protection avoids such changes by requiring zero first-order change on protected features, but accumulated independent constraints progressively shrink the admissible writing subspace; later targets can then become unreachable within that subspace even when they remain feasible outside it. The unresolved problem is therefore to select a successful update that minimizes protected change without sacrificing the writing space needed for later edits.

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.

We address this problem by separating what an edit must achieve from how it is stored. Rewrite and paraphrase margins define the feasible set of updates that successfully encode the requested fact. History-context and unrelated prompts induce a cost over this set that estimates each update's effect on protected behavior. This leads to CANONEDIT, an editor that selects the minimum-cost update from the feasible set. Updates that satisfy the same target constraints are grouped into an equivalence class, yielding a quotient space over alternative parameter realizations; its minimum-cost member under the protected cost defines the *canonical representative*. A finite budget on this protected cost defines the admissible region, retaining directions with sufficiently small protected effects and preserving additional degrees of freedom for subsequent edits.

Operationally, CANONEDIT performs this quotient-space selection in a small coordinate space constructed from target-responsive directions (Figure 1). The target constraints and protected cost form a constrained Bregman projection. Its closed-form solution moves through the inverse geometry induced by protected-output sensitivity only as far as the active target margins require. Forward evaluations then confirm that the proposed update succeeds in the actual nonlinear model.

This quotient-space construction yields two formal consequences. First, within a chosen local update space, CANONEDIT has a unique solution: among all updates that meet the target margins, it is the one with the smallest predicted change to protected responses. Second, when the protection budget includes hard-null updates with zero first-order protected effect under the same step-size limit, it retains all of them while admitting useful writing directions with small but nonzero protected effects.

We test both consequences over persistent trajectories of up to 10,000 requests across three datasets and four backbone families. Across seven 10K streams, CANONEDIT attains 99.24-100% online writing efficacy with 0-76 failed writes, while measured specificity remains 99.97-100%. The mechanism audit further shows that, when several candidates write successfully, canonical selection chooses the lower-change representative in 99.6% of cases.

The contributions of this work are as follows:

1. We identify a missing choice in sequential editing: when many updates can write the same fact, the editor must decide which one to keep. We group updates satisfying the same target constraints into quotient-space equivalence classes and select the least disruptive successful representative.
2. We introduce CANONEDIT, which makes target success a constraint and then minimizes predicted change to protected outputs. We derive its efficient local solution and show when it retains every feasible hard-null update while admitting additional low-cost directions with nonzero protected effects.
3. Our long-horizon evaluation shows that the additional feasible directions retained by the quotient formulation translate into sustained writing capacity while protected behavior remains stable. Canonical representative selection thereby turns the geometry of the quotient space into practical long-horizon gains and places continued writing and knowledge protection under one principle.

2 RELATED WORK

Persistent factual editing links three decisions: defining the target, measuring collateral change, and managing accepted updates over a long stream. Prior work addresses each, but leaves implicit which parameter realization to retain when several produce the same target action.

Target construction defines what must be written. ROME and MEMIT insert facts through low-rank updates to selected MLP weights (Meng et al., 2022a;b); MEND predicts updates with a learned editor, while SERAC stores edited behavior externally (Mitchell et al., 2022a;b). Forward Replay constructs downstream-compatible hidden targets (Liu et al., 2026c), and AnyEdit broadens the forms of knowledge being evaluated (Jiang et al., 2025). These methods specify a target or candidate update, but not the choice among target-equivalent successful realizations.

Protection supplies a cost for that choice. AlphaEdit protects selected representations through a null-space constraint (Fang et al., 2025), while BetaEdit studies leakage and history-aware updates in sequential null-space editing (Liu et al., 2026a). CrispEdit instead uses Bregman divergence and Gauss–Newton/K-FAC structure to project a candidate toward low-curvature directions (Bregman,

1967; Ikram et al., 2026; Martens & Grosse, 2015). These methods provide protection constraints or costs, but either require zero action or transform a chosen candidate rather than select across all successful realizations.

Sequential editing makes the choice persistent. Sequential editors manage accumulation through memory separation (Wang et al., 2024a), neuron or layer selection (Jiang et al., 2024; Li et al., 2025; Wang et al., 2026), normalization or structural correction (Gu et al., 2025; Ma et al., 2026; Liu et al., 2026b; Zhang et al., 2026), and localized fine-tuning (Yang et al., 2026). Yet every accepted step still commits one realization and changes all later edits. CANONEDIT connects the three lines above by grouping target-equivalent updates into quotient-space classes and selecting the successful representative with the lowest protected cost.

3 PROBLEM SETUP

Let f_{θ_0} be a pretrained language model with initial parameters θ_0. The model receives T edit requests in order. Request $e_t = (x_t, y_t^{\text{old}}, y_t^{\text{new}})$ asks the model to answer x_t with y_t^{new} instead of y_t^{old}. The updated parameters θ_t become the starting point for request $t+1$. For evaluation, $\mathcal{D}_t^{\text{para}}$ contains other phrasings of x_t, $\mathcal{D}_t^{\text{hist}}$ queries facts from earlier edits, and $\mathcal{D}_t^{\text{stat}}$ contains unrelated prompts that should not change. The target set $\mathcal{D}_{T,t} = \{x_t\} \cup \mathcal{D}_t^{\text{para}}$ contains the prompts that should express the new fact; the protected set $\mathcal{D}_{C,t} = \mathcal{D}_t^{\text{hist}} \cup \mathcal{D}_t^{\text{stat}}$ contains the prompts that should be preserved. We adopt aboving setting following standard factual and sequential editing protocols (De Cao et al., 2021; Mitchell et al., 2022a; Meng et al., 2022a; Yao et al., 2023; Wang et al., 2024b; Gupta et al., 2024; Wang et al., 2024a; Jiang et al., 2024; Fang et al., 2025).

For any prompt x, let $p_\theta(y \mid x)$ be the probability assigned to the full answer y, computed token by token. The new and old answers are compared by the margin

$$m_t(x;\theta) = \log p_\theta(y_t^{\text{new}} \mid x) - \log p_\theta(y_t^{\text{old}} \mid x). \tag{1}$$

The edit succeeds when $m_t(x_t;\theta_t) > 0$, because the new answer is then more likely than the old one. The same test on $\mathcal{D}_t^{\text{para}}$ measures generality; $\mathcal{D}_t^{\text{hist}}$ and $\mathcal{D}_t^{\text{stat}}$ measure retention and specificity.

4 SELECTING A PARAMETER UPDATE IN THE QUOTIENT SPACE

The success editing in Section 3 does not uniquely determine a parameter update: many updates can favor the new answer while affecting protected facts differently. Our work therefore treats target success as a constraint and selects among successful updates by their protected change.

4.1 CANDIDATE UPDATES AND THE EXACT OBJECTIVE

Following ROME and MEMIT (Meng et al., 2022a;b), the search uses d_t low-rank changes to selected feed-forward (MLP) output matrices. Index them by $\ell \in \{1, \dots, d_t\}$. The full-parameter tensor $E_{t,\ell}$ contains the ℓth low-rank matrix change and is zero elsewhere. Let $z_\ell \in \mathbb{R}$ be its strength and collect all strengths in $z = (z_1, \dots, z_{d_t}) \in Z_t := \mathbb{R}^{d_t}$, where Z_t is the coefficient space. The full parameter update is

$$\Delta\theta_t(z) = \sum_{\ell=1}^{d_t} z_\ell E_{t,\ell}. \tag{2}$$

Applying this update to the current parameters gives the candidate model $\theta_t(z) := \theta_{t-1} + \Delta\theta_t(z)$.

Different choices of z may all write the new fact but change protected answers differently. Let $s_t(\theta) \in \mathbb{R}^{r_t}$ collect the r_t monitored protected-answer log-likelihoods under parameters θ. For a protected prompt ξ, $p_\theta(\cdot \mid \xi)$ is the predictive output distribution. We measure protected change by

$$D_{C,t}^{\text{KL}}(z) = \sum_{\xi \in \mathcal{D}_{C,t}} D_{\text{KL}}\big(p_{\theta_{t-1}}(\cdot \mid \xi) \,\|\, p_{\theta_t(z)}(\cdot \mid \xi)\big)\,, \tag{3}$$

$$D_{C,t}^{\text{score}}(z) = \frac{1}{2r_t} \left\| s_t(\theta_t(z)) - s_t(\theta_{t-1}) \right\|_2^2. \tag{4}$$

The two quantities measure distribution-wide and protected-score changes, respectively; both vanish without an update, and smaller values mean less change. Below, $D_{C,t}$ denotes the selected measure.

The model-editing objective is therefore to satisfy every target margin while minimizing the change to protected behavior. Let $\gamma_t(x) \geq 0$ be the required margin for target prompt x, allowing the new answer to be preferred by a prescribed margin, and denote the optimal coefficient vector by z_t^{ideal}. The objective is

$$z_t^{\text{ideal}} \in \arg\min_{z \in Z_t} D_{C,t}(z) \quad \text{s.t.} \quad m_t(x; \theta_t(z)) \geq \gamma_t(x), \quad \forall x \in \mathcal{D}_{T,t}. \tag{5}$$

The constraints enforce the target fact on the rewrite prompt and its paraphrases; the objective selects among the successful updates by protected change.

4.2 From the exact objective to a local quotient problem

Equation (5) states the desired editing rule, but both its margins and its protected cost require running the nonlinear model. To obtain a small problem that can be solved at every edit, we keep their leading local terms at the current model, represented by $z = 0$. This step turns target success into a feasible set and protected change into a cost for choosing within that set.

For a target prompt x, define $a_t(x) := \nabla_z m_t(x; \theta_t(z))|_{z=0} \in \mathbb{R}^{d_t}$; it records how the target margin initially changes with each update coefficient. Under the local smoothness assumptions (details in Appendix A),

$$m_t(x; \theta_t(z)) = m_t(x; \theta_{t-1}) + a_t(x)^\top z + O(\|z\|_2^2). \tag{6}$$

Let $b_t(x) := [\gamma_t(x) - m_t(x; \theta_{t-1})]_+$ be the margin still missing before the edit. Stacking $a_t(x)^\top$ over the $n_t := |\mathcal{D}_{T,t}|$ target prompts gives $A_t \in \mathbb{R}^{n_t \times d_t}$; stacking $b_t(x)$ gives $\mathbf{b}_t \in \mathbb{R}^{n_t}$. The locally successful updates are therefore those satisfying $A_t z \geq \mathbf{b}_t$.

This constraint sees an update only through its predicted target effect $A_t z$. We therefore regard z_1 and z_2 as equivalent when $A_t z_1 = A_t z_2$, or equivalently when $z_1 - z_2 \in \ker A_t$. These equivalence classes form the quotient space $Z_t / \ker A_t$ (Absil et al., 2008). The target determines which classes are successful, but cannot choose among updates in the same class; that choice must come from their effect on protected behavior.

For either protected measure in Eqs. (3)–(4), $D_{C,t}(z)$ is minimized at the zero update. Its linear term therefore vanishes. Defining the local curvature by $G_{C,t} := \nabla_z^2 D_{C,t}(0) \in \mathbb{R}^{d_t \times d_t}$ gives

$$D_{C,t}(z) = \tfrac{1}{2} z^\top G_{C,t} z + O(\|z\|_2^3). \tag{7}$$

For output KL, $G_{C,t}$ is the Fisher curvature pulled back to the update coordinates; for protected scores, it is the corresponding Gauss-Newton matrix (Amari, 1998; Martens, 2020). To keep the update within the regime where these local approximations are reliable, let the positive-definite matrix $M_t \in \mathbb{R}^{d_t \times d_t}$ measure update size and let $\alpha > 0$. We use the combined cost matrix $B_t := G_{C,t} + \alpha M_t \succ 0$.

4.3 Minimum-cost successful update

The resulting local editing problem is

$$z_t^* = \arg\min_{z \in Z_t} \tfrac{1}{2} z^\top B_t z \quad \text{s.t.} \quad A_t z \geq \mathbf{b}_t. \tag{8}$$

The constraint requires a successful write, and the objective chooses the successful update with the smallest combined cost of protected change and update size. Since $B_t \succ 0$, $\phi_t(z) := z^\top B_t z / 2$ defines a Bregman distance from the zero update equal to this objective. Equation (8) is therefore the Bregman projection of zero onto the locally successful set (Bregman, 1967).

Theorem 1 (Lowest-cost successful update)**.** *If at least one z satisfies $A_t z \geq \mathbf{b}_t$, then Eq. (8) has a unique solution z_t^*, and every feasible z (every $z \in Z_t$ satisfying $A_t z \geq \mathbf{b}_t$) satisfies $(z_t^*)^\top B_t z_t^* \leq z^\top B_t z$. With one target prompt x,*

$$z_t^* = \begin{cases} 0, & b_t(x) = 0, \\ \dfrac{b_t(x)}{a_t(x)^\top B_t^{-1} a_t(x)} B_t^{-1} a_t(x), & b_t(x) > 0. \end{cases} \tag{9}$$

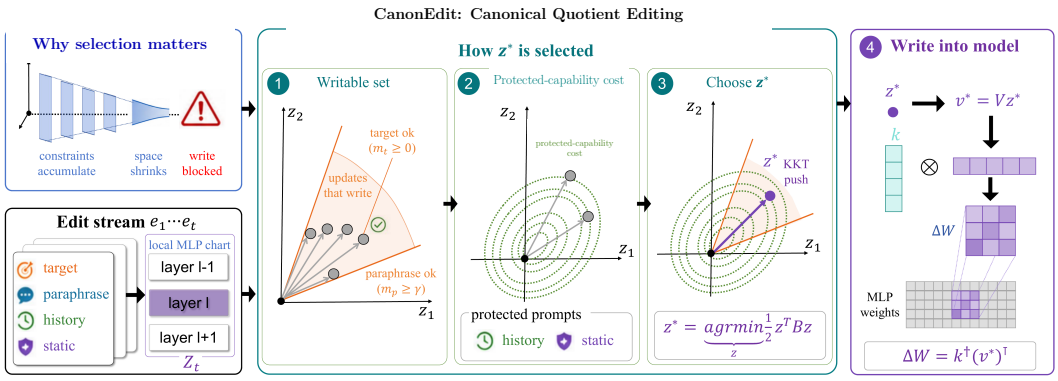


Figure 1: **CanonEdit selects one update from many ways to write the same fact.** Among updates that achieve the target, the constrained projection selects the minimum-cost canonical representative, maps it to a structured MLP update.

Thus z_t^* has the lowest local cost among all locally successful updates. With one target prompt, $b_t(x) = 0$ requires no update; otherwise B_t^{-1} suppresses costly components of $a_t(x)$ and the denominator scales the result to the required margin. Appendix B gives the proof and the multi-prompt solution. If no z satisfies $A_t z \geq \mathbf{b}_t$, which means the target constraints are infeasible for the current candidate update directions, CanonEdit tries another set of candidate directions, as described in Section 5.

5 CanonEdit

Section 4 defines the minimum-cost successful update in a local update space. CanonEdit realizes this rule over a small set of candidate charts, indexed by $j \in \mathcal{J}_t$; $E_{t,j,\ell}$, $A_{t,j}$, and $B_{t,j}$ are the chart-specific counterparts of $E_{t,\ell}$, A_t, and B_t above. Each chart produces a candidate update, which is verified on the full nonlinear model before the minimum-cost valid candidate is committed. The procedure has three steps.

Step 1: build target-responsive basic changes. The optimization in Eq. (8) can succeed only if its basic changes can move the target margin. A chart $j \in \mathcal{J}_t$ specifies editable MLP output matrices and d_t basic changes. For its ℓth basic change, let $W_{t,j,\ell}$ be the edited matrix. Running the target prompt gives the vector entering this matrix at the subject token; denote this input by the *key* $k_{t,j,\ell}$. CanonEdit also computes a *target residual* $v_{t,j,\ell}$, the change in the matrix output that makes the frozen downstream network assign a larger score to the new answer. Let $\eta > 0$ be a small numerical stabilizer. The full-parameter tensor $E_{t,j,\ell}$ is defined by

$$E_{t,j,\ell}|_{W_{t,j,\ell}} = \frac{v_{t,j,\ell} k_{t,j,\ell}^\top}{\|k_{t,j,\ell}\|_2^2 + \eta}, \qquad E_{t,j,\ell} = 0 \text{ elsewhere}, \tag{10}$$

where the tensor is zero outside $W_{t,j,\ell}$. This construction has a direct effect: multiplying the changed matrix by the same key gives $E_{t,j,\ell} k_{t,j,\ell} \approx v_{t,j,\ell}$. It therefore turns a desired hidden-state change into a low-rank weight change. Let $z = (z_1, \ldots, z_{d_t})$ contain the strengths of the d_t basic changes in chart j. Their combined parameter update is $\Delta\theta_{t,j}(z) = \sum_{\ell=1}^{d_t} z_\ell E_{t,j,\ell}$. Differentiating each rewrite or paraphrase margin with respect to z at $z = 0$ gives one row of the chart's target matrix $A_{t,j} \in \mathbb{R}^{n_t \times d_t}$, where n_t is the number of target prompts defined in Section 4. Thus, $A_{t,j} z$ predicts how the chosen combination closes the current target deficits $\mathbf{b}_t$.

In practice, CanonEdit computes $v_{t,j,\ell}$ directly from the current model. It keeps the model weights fixed, temporarily adds a trainable vector to the selected MLP output at the subject position, and optimizes this vector until the new answer outranks the old answer on the rewrite and paraphrase prompts. The optimized vector is used as $v_{t,j,\ell}$, and Eq. (10) converts it into a low-rank basic change. Using one such chart gives the Riesz realization: Eq. (8) selects its minimum-cost update, with Eq. (9) as the one-constraint solution. If one layer does not provide a direction that passes the full-network

target checks, the finite-atlas realization repeats the same construction over a small, fixed set of layers and prompt weightings. These charts differ only in their basic changes and target matrices; Step 2 constructs a cost matrix and solves the same problem for each one. Appendix C gives the residual objective and fixed chart settings. This construction follows hidden-target editing (Meng et al., 2022a;b) and downstream-compatible targets (Liu et al., 2026c).

Step 2: compute the cost matrix and solve the projection. The target matrix identifies combinations that can write the fact, but it does not say which combination changes protected answers least. Recall that $s_t(\theta) \in \mathbb{R}^{r_t}$ contains the r_t protected-answer scores. For chart j, define $C_{t,j} \in \mathbb{R}^{r_t \times d_t}$ as the Jacobian of these scores with respect to the same coefficient vector z:

$$C_{t,j} := \frac{1}{\sqrt{r_t}} \nabla_z s_t(\theta_{t-1} + \Delta\theta_{t,j}(z))|_{z=0}. \tag{11}$$

The vector $C_{t,j}z$ is the first-order change of the monitored scores, so $\|C_{t,j}z\|_2^2$ estimates their mean squared change. To also control the size of the parameter update, define $M_{t,j} \in \mathbb{R}^{d_t \times d_t}$ as the normalized Gram matrix of the basic changes: for $\ell, q \in \{1, \ldots, d_t\}$, its (ℓ, q) entry is proportional to the full-parameter inner product $\langle E_{t,j,\ell}, E_{t,j,q} \rangle$. Consequently, $z^\top M_{t,j} z$ measures the squared Frobenius size of $\Delta\theta_{t,j}(z)$ in a chart-independent scale. For a regularization weight $\alpha > 0$, the implemented cost matrix is

$$B_{t,j} = C_{t,j}^\top C_{t,j} + \alpha M_{t,j}, \tag{12}$$

which is the score-based Gauss–Newton term from Section 4 plus an update-size penalty (Martens & Grosse, 2015; Martens, 2020; Ikram et al., 2026). CanonEdit substitutes $A_{t,j}$, $B_{t,j}$, and the already defined deficit vector $\mathbf{b}_t$ into Eq. (8). A small active-set solver then returns the minimum-cost coefficient vector $z^*_{t,j}$ for chart j. The multi-constraint solver is detailed in Appendix B.

Step 3: verify and commit the update. The matrices $A_{t,j}$ and $B_{t,j}$ describe only the local behavior around the current model. Let $\varepsilon_t > 0$ be the allowed measured protected cost and let K be the maximum number of local corrections. CanonEdit temporarily applies $\Delta\theta_{t,j}(z^*_{t,j})$ and evaluates the actual rewrite margins, paraphrase margins, and protected cost with the full network. If a target margin is still short, the method recomputes the local quantities at the proposed model and solves the same minimum-cost problem for the remaining deficit, up to K times. Let $\widehat{\theta}_{t,j}$ be the resulting candidate for chart j, and let $\widehat{D}_{C,t}(\theta)$ be its exact protected cost against θ_{t-1}, evaluated by Eq. (3) or Eq. (4). CanonEdit commits

$$\theta_t \in \arg \min_{\theta \in \{\widehat{\theta}_{t,j} : j \in \mathcal{J}_t\}} \widehat{D}_{C,t}(\theta) \quad \text{s.t.} \quad m_t(x;\theta) \geq \gamma_t(x)\ \forall x \in \mathcal{D}_{T,t}, \qquad \widehat{D}_{C,t}(\theta) \leq \varepsilon_t. \tag{13}$$

If the constrained set is empty, CanonEdit sets $\theta_t = \theta_{t-1}$ and records a failed write.

Algorithm 1 CanonEdit at sequential edit t.

Require: Current model θ_{t-1}; target and protected prompts $\mathcal{D}_{T,t}, \mathcal{D}_{C,t}$; charts $\mathcal{J}_t$; budget ε_t; correction limit K

```
Compute the current target deficit b_t; set V_t ← ∅
for j ∈ J_t do
    Build E_{t,j,ℓ} and differentiate to obtain A_{t,j} and C_{t,j}
    Form B_{t,j} by Eq. (12) and solve Eq. (8)
    Temporarily apply the candidate; re-linearize and resolve at most K times if needed
    Add it to V_t if the actual target margins and protected budget pass
end for
return θ_{t−1} and record a failed write if V_t = ∅
Select the member of V_t with the smallest measured protected cost
return the model after applying the selected update
```

Computational cost. All linear algebra uses the d_t update coefficients rather than the full parameter dimension. Forming $C_{t,j}^\top C_{t,j}$ costs $O(r_t d_t^2)$, solving the coefficient-space system costs $O(d_t^3)$, and nonlinear verification uses a small fixed number of forward evaluations. The reported settings use at most 12 basic changes; Appendix C gives the full settings and Section 6 reports wall-clock time.

Table 1: **Online performance over 10,000 persistent requests on GPT-J-6B.** Rel. is online writing efficacy measured on θ_t immediately after request t; Gen. and Loc. denote generalization and measured locality, and Hist. is historical-context retention. All rows within a dataset block use the controlled $B{=}1$ evaluator above. **Bold** and underline mark the best and second-best edited-model results for each metric.

		Standard editing metrics			Long-horizon metrics	
Dataset	Method	Rel.↑	Gen.↑	Loc.↑	Hist.↑	P-MAE↓
CounterFact	Unedited	28.80	35.51	100.00	74.20	.000
	ROME (Meng et al., 2022a)	29.43	36.95	**100.00**	74.03	.016
	MEMIT (Meng et al., 2022b)	29.44	36.91	**100.00**	74.02	.016
	NSE (Jiang et al., 2024)	98.23	94.58	**100.00**	4.15	.340
	AlphaEdit (Fang et al., 2025)	25.17	35.35	**100.00**	**78.69**	**.009**
	RLEdit (Li et al., 2025)	92.19	74.11	99.99	18.04	.275
	UltraEdit (Gu et al., 2025)	79.63	65.06	**100.00**	30.58	.124
	MEMIT (FR target) (Liu et al., 2026c)	79.50	72.86	99.68	38.64	.452
	AlphaEdit (FR target) (Liu et al., 2026c)	88.32	67.38	99.69	60.73	.187
	NAS (Liu et al., 2026b)	98.47	96.52	**100.00**	3.13	.564
	BetaEdit (Liu et al., 2026a)	95.40	94.75	**100.00**	7.70	.522
	StableEdit (Ma et al., 2026)	70.76	60.31	**100.00**	38.77	.074
	HiEdit (Wang et al., 2026)	62.56	60.77	99.80	41.29	.262
	LocFT-BF (Yang et al., 2026)	91.88	84.89	99.48	11.97	.843
	CANONEDIT	**99.33**	**99.22**	**100.00**	74.65	.014
ZsRE	ROME (Meng et al., 2022a)	55.86	57.22	**100.00**	46.21	.011
	AlphaEdit (Fang et al., 2025)	48.72	52.66	**100.00**	**55.41**	.007
	MEMIT (FR target) (Liu et al., 2026c)	93.29	88.31	**100.00**	8.02	.503
	AlphaEdit (FR target) (Liu et al., 2026c)	78.04	69.95	**100.00**	51.08	.048
	CANONEDIT	**100.00**	**91.42**	**100.00**	47.09	**.005**
WikiFactDiff	MEMIT (FR target) (Liu et al., 2026c)	97.00	**90.05**	98.60	18.31	.322
	AlphaEdit (FR target) (Liu et al., 2026c)	98.51	79.71	**100.00**	52.61	.105
	CANONEDIT	**99.68**	79.47	99.99	**57.62**	**.089**

6 EXPERIMENTS

The experiments answer three questions. **Writability:** can the current target still be written after thousands of persistent updates? **Representative cost:** among successful candidates, does canonical selection reduce measured protected change? **Mechanism and cost:** which parts of target reachability, capability geometry, and nonlinear verification are needed, and what do they cost?

6.1 EXPERIMENTAL SETUP

Datasets and models. We use CounterFact (Meng et al., 2022a), ZsRE (Levy et al., 2017), and WikiFactDiff (Khodja et al., 2024). We evaluate sequences of 10,000 requests and use their first 3,000 requests for the shorter-horizon study. The 10K results cover GPT2-XL, GPT-J-6B, and Qwen2.5-1.5B-Instruct; the 3K matrix further covers Llama-3.1-8B-Instruct. Edits use batch size one, and every accepted update becomes the starting model for the next request.

Comparisons. Table 1 fixes GPT-J-6B and a 10K request horizon. It includes established locate-and-edit methods, recent sequential editors, Forward Replay target variants, and CANONEDIT. Broader 3K and 10K comparisons are reported in Appendix K.

Controlled persistent protocol. Within each dataset block, methods receive the same frozen 10K request pool with batch size one and are evaluated by the same sequence-likelihood, paraphrase, historical-context, and static-locality probes. Every accepted update persists into the next request. Appendix D records the row-level protocol fields used for direct comparison.

Metrics. We report online writing efficacy, the fraction of requests satisfying $m_t(x_t; \theta_t) > 0$ immediately after editing (Rel. in Table 1), benchmark paraphrase consistency as generality, and measured specificity on up to three unrelated prompts. Historical-context retention tests up to two prompts that explicitly query the pre-edit context. P-MAE and P-MSE are the protected-score diagnostics aligned with Eq. (4). These metrics follow factual-editing evaluation practice (Li et al., 2024), and sequence comparisons use teacher-forced likelihood (Yang et al., 2025); all accuracy and retention metrics are percentages.

6.2 LONG-HORIZON WRITING WITH BOUNDED PROTECTION

CanonEdit keeps the model writable while controlling side effects. Table 1 compares full 10K streams. On CounterFact, CANONEDIT improves the best competing efficacy and generality by 0.86 and 2.70 percentage points, while retaining 100% locality and .014 P-MAE. On ZsRE, the gains are 6.71 and 3.11 points, with no failed writes and the lowest P-MAE of .005. On WikiFactDiff, it obtains the highest efficacy and historical retention and the lowest P-MAE among the shown methods. The writing gains therefore coexist with 99.99-100% locality and low protected-score displacement, matching the intended effect of minimum-cost representative selection.

The same editing rule transfers across datasets and backbones. Table 2 broadens the 3K check to three datasets and four backbones. All 12 cells reach 98.57-100% online efficacy with only 0-43 failed writes. This narrow range across model families shows that the target-reachable quotient construction remains effective across differences in architecture, scale, and instruction tuning; Appendix K provides the corresponding protection and generality results.

The seven 10K streams extend this result to the full horizon: online efficacy is 99.24-100% and measured specificity is 99.97-100% (Appendix G). For the three GPT-J-6B streams, Figure 2 divides each run into ten 1K blocks to show when this performance is maintained.

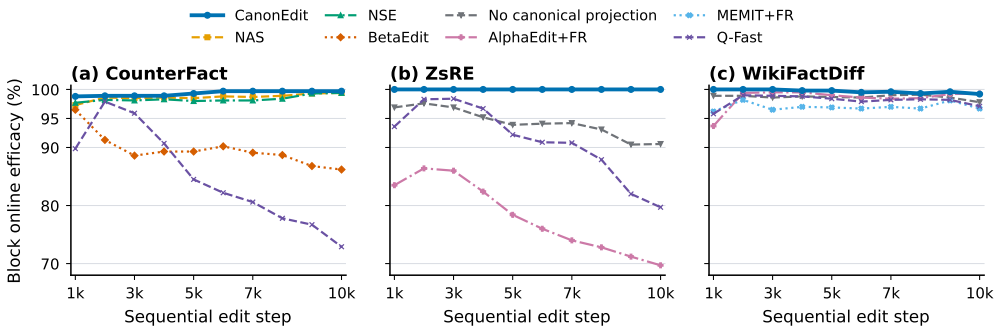


Figure 2: **Blockwise online writing efficacy over persistent edits.** The vertical axis is online efficacy within each non-overlapping 1K block. Markers are exact observations joined without smoothing, and each curve follows its method's persistent edit order on the shared request pool.

Reading the trajectories. Each marker in Figure 2 reports efficacy within one 1K block, exposing how writability changes along the stream. CANONEDIT remains at or above 98.8%, 100%, and 99.2% in every block of CounterFact, ZsRE, and WikiFactDiff, respectively. Its nearly flat trajectories indicate that target-reachable directions remain available throughout the edit stream, while several baselines progressively lose writability. Together with 99.97-100% measured specificity, this is direct evidence that canonical representative selection sustains future writes under bounded interference.

6.3 THEORY-ALIGNED ABLATIONS AND SENSITIVITY

Table 3: **Theory-aligned ablations and budget sensitivity** (N=600)**.** Left: one-principle removals on CounterFact–GPT-J. Right: capability budgets on ZsRE–Qwen2.5.

Variant	Eff.↑	Gen.↑	Fail↓	P-MAE↓	P-MSE↓
Hard-null protection	84.83	59.83	91	.473	.529
w/o canonical projection	96.33	59.50	22	.384	.369
w/o forward residual	97.17	67.50	17	**.157**	**.103**
w/o finite atlas	99.67	65.00	2	.226	.207
CANONEDIT (full)	**99.83**	**67.83**	**1**	.269	.243

Bud.	Eff.↑	Gen.↑	Fail↓	P-cost↓	sec/e.↓
100	98.50	83.17	9	.172	3.46
120	98.67	83.67	8	**.166**	3.47
150	98.67	**84.17**	8	**.166**	3.40
200	**99.00**	83.83	**6**	.172	**2.88**

Mechanism and cost. The ablations isolate the role of each principle. Replacing the finite capability budget with hard-null protection adds 90 failures, showing that exact protection removes useful target directions. Removing canonical projection adds 21 failures and raises P-MAE from .269 to .384, so

Table 2: **Breadth after 3,000 sequential edits across three datasets and four backbones.** E, G, R, and S are online efficacy, paraphrase generality, historical retention, and measured specificity (%); full diagnostics and baselines are in Appendix Tables 13 and 14.

	GPT-J-6B				GPT2-XL				Llama-3.1-8B				Qwen2.5-1.5B			
Dataset	E	G	R	S	E	G	R	S	E	G	R	S	E	G	R	S
CounterFact	99.90	80.67	51.57	100.00	99.40	70.50	60.17	100.00	99.27	84.60	41.43	99.93	99.53	78.93	65.80	99.97
ZsRE	99.60	85.13	38.60	99.93	98.83	87.10	45.70	99.97	98.57	88.17	34.60	99.77	99.73	89.37	49.03	99.93
WikiFactDiff	100.00	83.10	57.63	99.97	99.03	69.80	57.13	99.97	99.90	86.27	46.83	99.90	99.10	73.73	66.93	99.93

reachability alone cannot choose a stable update. Removing the forward residual gives the lowest displacement but leaves 17 writes unresolved, confirming that target success must define the feasible set before protected cost is minimized. The finite atlas resolves the remaining boundary failure. Across budgets 100-200, efficacy stays at 98.50-99.00%, protected cost at .166-.172, and time at 2.88-3.47 seconds per edit, showing a stable operating range.

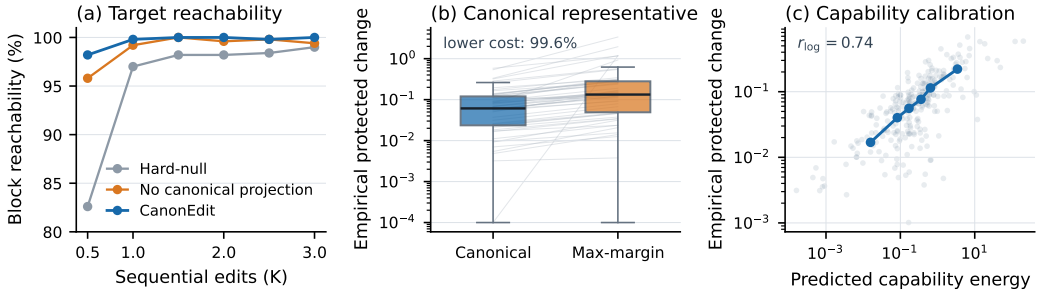


Figure 3: **Mechanism audit on CounterFact–GPT-J-6B.** (a) Reachability over six 500-edit blocks at N=3,000. (b) Protected change for 257 edits with multiple successful candidates; lines join representatives for the same request. (c) Capability energy versus measured change for 265 accepted updates; the line joins equal-count-bin medians ($r_{\log} = .74$, 95% bootstrap CI $[.65, .82]$).

Figure 3 tests three links behind the method. First, CANONEDIT raises hard-null reachability from 82.6% to 98.2% in the first block and reaches 100% in the final block, showing that the finite budget retains useful writing directions. Second, on 257 edits with multiple successful candidates, it chooses the lower-change representative in 99.6% of cases and reduces median change by 54.1%, verifying the canonical projection. Third, capability energy tracks nonlinear disturbance with $r_{\log} = .74$, validating $B_{t,j}$ as a local cost. At 5K, evaluation-only answer retention is 75.0% for CANONEDIT versus 61.3% for AlphaEdit with the Forward Replay target, with score MAE 3.59 versus 10.46 (Appendix F); this shows that protection extends beyond the prompts used to form the cost. Moving-block intervals remain at 99.13-99.53% (Appendix E), while the timing audit records $3.729 \pm .506$ seconds per edit and 93.9% use of the closed-form path (Appendix C). Together, the results connect the predicted geometry to reachability, representative quality, measured protection, and practical cost.

7 CONCLUSION AND LIMITATIONS

Sequential factual editing is underdetermined: many parameter updates can write the same target while changing protected behavior differently. CANONEDIT resolves this choice by selecting the minimum protected-capability lift over the target-success region. The projection is unique within a chart, has an active-set closed form, and, under the stated budget condition, retains admissible hard-null updates while allowing additional finite-cost directions. Across three datasets, four backbone families, and up to 10,000 edits, this principle gives 99.24-100% online writing efficacy, few failed writes, and 99.97-100% measured specificity. The mechanism audit further shows that canonical selection chooses the lower-change successful representative in 99.6% of paired cases. **Limitations.** The theoretical guarantees are local to the specified update charts and designated protected behavior. Nonlinear verification complements the local analysis by evaluating target success and protected change on the updated model.

ETHICS STATEMENT

Parameter editing is dual use: it can correct stale or harmful factual behavior, but it can also inject false information or obscure the provenance of a model's knowledge. Our experiments use established factual-editing benchmarks and do not involve human subjects. A deployed editor should authenticate edit requests, log parameter changes, evaluate downstream capability and safety, and retain a rollback mechanism. CANONEDIT controls how a requested target is written; it does not verify that the target is true.

REPRODUCIBILITY STATEMENT

Appendices B–C provide complete proofs, notation, target-coordinate construction, model-specific edit locations, budgets, and timing boundaries. The supplement also contains the 3K/10K matrices and trajectory summaries. Code, evaluation scripts, and processed results will be released with the paper.

AI USE STATEMENT

Generative AI tools were used solely for language editing. The authors verified the final manuscript and take full responsibility for the submission.

REFERENCES

P.-A. Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008. ISBN 978-0-691-13298-3. URL https://sites.uclouvain.be/absil/amsbook/.

Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998. doi: 10.1162/089976698300017746.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. ISBN 978-0-521-83378-3. URL https://web.stanford.edu/~boyd/cvxbook/.

Lev M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967. doi: 10.1016/0041-5553(67)90040-7.

John B Conway. *A course in functional analysis*. Springer, 2019.

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. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.522. URL https://aclanthology.org/2021.emnlp-main.522/.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. AlphaEdit: Null-space constrained knowledge editing for language models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.02355.

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 preprint arXiv:2505.14679*, 2025. URL https://arxiv.org/abs/2505.14679.

Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. Model editing at scale leads to gradual and catastrophic forgetting. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 15202–15232. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.902. URL https://aclanthology.org/2024.findings-acl.902/.

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. URL https://arxiv.org/abs/2602.15823.

Houcheng Jiang, Junfeng Fang, Tianyu Zhang, An Zhang, Ruipeng Wang, Tao Liang, and Xiang Wang. Neuron-level sequential editing for large language models. *arXiv preprint arXiv:2410.04045*, 2024. URL https://arxiv.org/abs/2410.04045.

Houcheng Jiang, Junfeng Fang, Ningyu Zhang, Mingyang Wan, Guojun Ma, Xiang Wang, Xiangnan He, and Tat-Seng Chua. AnyEdit: Edit any knowledge encoded in language models. In *International Conference on Machine Learning*, pp. 27510–27533, 2025. URL https://arxiv.org/abs/2502.05628.

Hichem Ammar Khodja, Frédéric Béchet, Quentin Brabant, Alexis Nasr, and Gwénolé Lecorvé. WikiFactDiff: A large, realistic, and temporally adaptable dataset for atomic factual knowledge update in causal language models. *arXiv preprint arXiv:2403.14364*, 2024. URL https://arxiv.org/abs/2403.14364.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning*, pp. 333–342, 2017. URL https://arxiv.org/abs/1706.04115.

Qi Li, Xiang Liu, Zhenheng Tang, Peijie Dong, Zeyu Li, Xinglin Pan, and Xiaowen Chu. Should we really edit language models? on the evaluation of edited language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0971. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/370fa2e691f57eb319bc263a07dad4a5-Abstract-Conference.html.

Zherui Li, Houcheng Jiang, Hao Chen, Baolong Bi, Zhenhong Zhou, Fei Sun, Junfeng Fang, and Xiang Wang. Reinforced lifelong editing for language models. *arXiv preprint arXiv:2502.05759*, 2025. URL https://arxiv.org/abs/2502.05759.

Bingqing Liu, Wei Liu, and Yuhua Li. BetaEdit: Null-space constrained sequential model editing. *arXiv preprint arXiv:2605.09285*, 2026a. URL https://arxiv.org/abs/2605.09285.

Mingda Liu, Zhenghan Zhu, Ze'an Miao, and Katsuki Fujisawa. Norm anchors make model edits last. *arXiv preprint arXiv:2602.02543*, 2026b. URL https://arxiv.org/abs/2602.02543.

Wei Liu, Hongkai Liu, Zhiying Deng, Yee Whye Teh, and Wee Sun Lee. From backward spreading to forward replay: Revisiting target construction in LLM parameter editing. *arXiv preprint arXiv:2605.00358*, 2026c. URL https://arxiv.org/abs/2605.00358.

Xin Ma, Wei Chen, Qi Liu, Derong Xu, Zhi Zheng, Tong Xu, and Enhong Chen. More edits, more stable: Understanding the lifelong normalization in sequential model editing. *arXiv preprint arXiv:2605.11836*, 2026. URL https://arxiv.org/abs/2605.11836.

James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International Conference on Machine Learning*, pp. 2408–2417, 2015.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, 2022a. URL https://arxiv.org/abs/2202.05262.

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. In *International Conference on Learning Representations*, 2022a. URL https://arxiv.org/abs/2110.11309.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Memory-based model editing at scale. In *International Conference on Machine Learning*, 2022b. URL https://arxiv.org/abs/2206.06520.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, NY, 2 edition, 2006. doi: 10.1007/978-0-387-40065-5.

WD Ray. Generalised inverses: Theory and applications, 1976.

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. *arXiv preprint arXiv:2405.14768*, 2024a. URL https://arxiv.org/abs/2405.14768.

Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, Kangwei Liu, Yuansheng Ni, Guozhou Zheng, and Huajun Chen. EasyEdit: An easy-to-use knowledge editing framework for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 82–93. Association for Computational Linguistics, 2024b. doi: 10.18653/v1/2024.acl-demos.9. URL https://aclanthology.org/2024.acl-demos.9/.

Yangfan Wang, Tianyang Sun, Chen Tang, Jie Liu, Wei Cai, and Jingchi Jiang. HiEdit: Lifelong model editing with hierarchical reinforcement learning. *arXiv preprint arXiv:2604.11214*, 2026. URL https://arxiv.org/abs/2604.11214.

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. *arXiv preprint arXiv:2502.11177*, 2025. URL https://arxiv.org/abs/2502.11177.

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. In *International Conference on Learning Representations*, volume 2026, pp. 71683–71702, 2026.

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. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.632. URL https://aclanthology.org/2023.emnlp-main.632/.

Taolin Zhang, Haidong Kang, Dongyang Li, Qizhou Chen, Xiaofeng He, Chengyu Wang, and Richang Hong. QueueEDIT: Structural self-correction for sequential model editing in LLMs. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 4706–4719, 2026. URL https://aclanthology.org/2026.findings-acl.231/.

A Notation and Assumptions

Table 4 collects the notation used in the paper. The local derivation assumes a finite update chart, twice-differentiable target margins, and a protected cost with the cubic remainder stated in Eq. (27) (Nocedal & Wright, 2006). The construction $B_t = G_{C,t} + \alpha M_t$ is positive definite because $G_{C,t} \succeq 0$, $M_t \succ 0$, and $\alpha > 0$; Theorem 1 applies when the local target-success set is nonempty. The nonlinear verification in Algorithm 1 checks the corresponding margins after the proposed parameter update is applied.

B Complete Proofs

B.1 Canonical representative

Consider

$$\min_{z \in \mathbb{R}^d} \frac{1}{2} z^\top B z \quad \text{subject to} \quad Az \geq b, \tag{14}$$

where $B \succ 0$ and the feasible set is nonempty. Coercivity of the objective and closedness of the feasible polyhedron imply existence; strict convexity implies uniqueness (Boyd & Vandenberghe, 2004).

Let z^* be the unique optimum, $S = \{j : a_j^\top z^* = b_j\}$ its active set, and $\lambda \geq 0$ the KKT multiplier. With the sign convention

$$\mathcal{L}(z, \lambda) = \frac{1}{2} z^\top B z - \lambda^\top (Az - b), \tag{15}$$

the KKT conditions are

$$Bz^* - A^\top \lambda = 0, \tag{16}$$

$$Az^* - b \geq 0, \qquad \lambda \geq 0, \tag{17}$$

$$\lambda_j (a_j^\top z^* - b_j) = 0. \tag{18}$$

Table 4: Symbol table.

Symbol	Meaning
θ_{t-1}, θ_t	Model parameters immediately before and after edit t.
$\mathcal{D}_{T,t}, \mathcal{D}_{C,t}$	Target prompts and designated protected prompts for edit t.
$Z_t = \mathbb{R}^{d_t}$	Coefficients of d_t allowed low-rank update patterns; $\Delta\theta_t(z)$ is their weighted sum in the model parameters.
$m_t(x;\theta), \gamma_t(x)$	Current and required margins for target prompt x.
$a_t(x), b_t(x)$	Local margin gradient and remaining margin deficit for target prompt x.
$A_t, \mathbf{b}_t$	Rows $a_t(x)^\top$ and deficits $b_t(x)$ stacked over $x \in \mathcal{D}_{T,t}$, giving $A_t z \geq \mathbf{b}_t$.
$G_{C,t}$	Protected output Fisher/Gauss–Newton metric in the edit chart.
M_t	Positive-definite chart trust metric.
B_t	Combined local cost matrix $G_{C,t} + \alpha M_t$.
H_t	Stacked linear operator used by a hard-null editor.
S	Active inequality indices at the optimum.
$D_{C,t}$	Protected output divergence, instantiated by output KL or protected-score displacement.
ε_t	Measured protected-cost budget used to admit a candidate update.
$\mathcal{J}_t, j$	Fixed set of candidate charts for edit t, and one chart index.
$E_{t,j,\ell}, k_{t,j,\ell}, v_{t,j,\ell}$	A basic low-rank change, its subject key, and its target residual.
$C_{t,j}, B_{t,j}$	Normalized protected-score Jacobian and combined local cost matrix in chart j.
$\widehat{\theta}_{t,j}, \widehat{D}_{C,t}$	Nonlinearly corrected candidate model and its measured protected cost.
$\mathcal{V}_t$	Finite set of chart representatives that pass nonlinear target and protected-cost checks.

Inactive multipliers vanish, so stationarity gives

$$z^* = B^{-1} A_S^\top \lambda_S. \tag{19}$$

Substituting the active equalities yields

$$A_S B^{-1} A_S^\top \lambda_S = b_S. \tag{20}$$

Let $Q_S = A_S B^{-1} A_S^\top \succeq 0$. Consistency follows from the existence of the KKT point. One solution is $\lambda_S = Q_S^\dagger b_S$ (Ray, 1976). If $v \in \ker Q_S$, then

$$0 = v^\top Q_S v = \|B^{-1/2} A_S^\top v\|_2^2, \tag{21}$$

so $A_S^\top v = 0$. Therefore every solution of the Schur system induces the same primal update, and

$$z^* = B^{-1} A_S^\top Q_S^\dagger b_S. \tag{22}$$

The active set is valid only if $\lambda_S \geq 0$ and all inactive inequalities hold; Algorithm 1 enumerates the small chart active sets and checks both. For one constraint, $Q = a^\top B^{-1} a$. When the nonnegative deficit b is zero, the origin is optimal; when $b > 0$, substitution gives the second case of Eq. (9). This is the finite-dimensional Riesz representation under the B inner product (Conway, 2019).

If z has the same active target action as z^*, then $A_S(z - z^*) = 0$. Using $Bz^* = A_S^\top \lambda_S$ gives $(z - z^*)^\top B z^* = 0$ and therefore

$$\tfrac{1}{2} z^\top B z = \tfrac{1}{2} (z^*)^\top B z^* + \tfrac{1}{2} \|z - z^*\|_B^2. \tag{23}$$

This decomposition shows that a component invisible to the active target action has nonnegative additional capability cost.

For any feasible z, convexity and KKT stationarity imply

$$\begin{aligned} \tfrac{1}{2} z^\top B z - \tfrac{1}{2} (z^*)^\top B z^* &= \tfrac{1}{2} \|z - z^*\|_B^2 + (z - z^*)^\top B z^* && (24) \\ &= \tfrac{1}{2} \|z - z^*\|_B^2 + (Az - Az^*)^\top \lambda && (25) \\ &\geq 0. && (26) \end{aligned}$$

This proves global minimum energy and uniqueness.

B.2 PROTECTED-DIVERGENCE BOUND

For output KL, the local curvature is $G_C = J_C^\top F_C J_C$, where J_C maps edit coordinates to protected outputs and F_C is their output Fisher matrix (Amari, 1998). For protected answer scores, $G_C = C^\top C$, where C is the normalized score Jacobian defined in Eq. (11). These are the two forms summarized by Eq. (7).

Assume that for $\|z\|_M \le \rho$,

$$\left|D_C(z) - \tfrac{1}{2} z^\top G_C z\right| \le \frac{L(\rho)}{6}\|z\|_M^3. \tag{27}$$

If $(z^*)^\top B z^* \le \mathcal{E}$ for $B = G_C + \alpha M$ and $\mathcal{E} \le \alpha\rho^2$, then nonnegativity of both terms gives

$$(z^*)^\top G_C z^* \le \mathcal{E}, \qquad \|z^*\|_M^2 \le \mathcal{E}/\alpha \le \rho^2. \tag{28}$$

Substitution into (27) yields

$$D_C(z^*) \le \frac{1}{2}\mathcal{E} + \frac{L(\rho)}{6}\left(\frac{\mathcal{E}}{\alpha}\right)^{3/2}. \tag{29}$$

This bound explains how the quadratic capability energy controls a smooth protected-output divergence inside the trust region. The score-based implementation admits the more direct expansion below.

B.3 PROTECTED-SCORE ALIGNMENT

Let $\Delta s(z) = s(\theta + \Delta\theta(z)) - s(\theta)$ and, as in Eq. (11), let $C = r^{-1/2}\nabla_z s(\theta + \Delta\theta(z))|_{z=0}$. Twice differentiability gives $\Delta s(z) = \sqrt{r}\, Cz + r_s(z)$ with $\|r_s(z)\|_2 = O(\|z\|_M^2)$. Therefore

$$D_C^{\text{score}}(z) = \frac{1}{2r}\|\sqrt{r}\, Cz + r_s(z)\|_2^2 \tag{30}$$

$$= \frac{1}{2} z^\top C^\top C z + O(\|z\|_M^3). \tag{31}$$

For coordinate changes $\Delta s_1, \ldots, \Delta s_r$, Cauchy–Schwarz further gives

$$\frac{1}{r}\sum_{c=1}^{r} |\Delta s_c| \le \sqrt{\frac{1}{r}\sum_{c=1}^{r} \Delta s_c^2} = \sqrt{2 D_C^{\text{score}}(z)}. \tag{32}$$

Thus the implemented $C^\top C$ metric is the second-order model of P-MSE and directly controls P-MAE locally.

B.4 SELECTION OVER THE VERIFIED ATLAS

Each chart returns at most one local canonical representative after refinement. The nonlinear checks in Algorithm 1 retain a finite set of candidates, all evaluated by the same output-space cost $\widehat{D}_{C,t}$. A minimum therefore exists whenever this set is nonempty. A fixed ordering resolves exact ties, making the committed update a deterministic minimum-cost member of the verified atlas. This determinism is distinct from the strict-convexity uniqueness proved within each chart.

B.5 HARD-NULL COMPARISON AND LONG-HORIZON SHRINKAGE

Let h_t be the number of protected features used by a hard-null editor, and let $H_t \in \mathbb{R}^{h_t \times d_t}$ map z to their first-order changes. Choose a radius $\rho > 0$, define $\|z\|_{M_t} := \sqrt{z^\top M_t z}$, and let $\varepsilon_t \ge 0$ be the protected-cost budget. Assume

$$\varepsilon_t \ge \sup_{H_t z = 0,\ \|z\|_{M_t} \le \rho} \tfrac{1}{2} z^\top G_{C,t} z. \tag{33}$$

Proposition 1 (Feasible-set inclusion). *Under Eq. (33), every $z \in Z_t$ satisfying $A_t z \ge \mathbf{b}_t$, $H_t z = 0$, and $\|z\|_{M_t} \le \rho$ also satisfies $z^\top G_{C,t} z/2 \le \varepsilon_t$. Moreover, the solution of Eq. (8) has no larger local cost than any such hard-null update.*

Proof. By the definition of the supremum in Eq. (33), every z satisfying $H_t z = 0$ and $\|z\|_{M_t} \le \rho$ obeys $z^\top G_{C,t} z/2 \le \varepsilon_t$. If it also satisfies $A_t z \ge \mathbf{b}_t$, it is feasible under the bounded-cost rule. It is also a competitor in Eq. (8), so Theorem 1 gives the selected representative no larger local cost. □

When protected history appends rows, $H_{t+1} = [H_t; \widetilde{H}_{t+1}]$. Consequently $\ker H_{t+1} = \ker H_t \cap \ker \widetilde{H}_{t+1} \subseteq \ker H_t$, and rank-nullity gives $\dim \ker H_{t+1} \le \dim \ker H_t$. A finite-cost target-responsive direction outside $\ker H_t$ can therefore remain available to the bounded-cost rule even though exact hard-null protection excludes it.

C Implementation Details

The two realizations use the same constrained objective and full-network verification; they differ in the candidate charts used to construct the basic changes. At the last token overlapping the matched subject span (or the final prompt token when no overlap is found), a chart writes a structured update to one MLP output matrix. A multi-token answer score is the sum of its teacher-forced token log probabilities, and every target margin compares this score with the old answer under the same prompt. The target and protected Jacobians are differentiated through the current model at every edit. Accepted updates persist, while the pre-edit keys and scores are reused only among candidates for that request.

Candidate-chart construction. The Riesz realization uses one chart. It inserts a hidden residual r at the subject position of the selected MLP and optimizes r through the frozen downstream network with

$$4[.25 - m_{\mathrm{rw}}(r)]_+^2 + .5\,\frac{1}{q-1}\sum_{k=2}^{q}[0 - m_k(r)]_+^2 + .002\|r\|_2^2. \tag{34}$$

Here m_{rw} is the rewrite margin and $m_2, \ldots, m_q$ are the paraphrase margins measured after injecting r. We use Adam with learning rate 2.0, four steps followed by early-stopped continuation up to eight steps, and clip $\|r\|_2$ at 40. Pairing the optimized residual with the subject key gives one rank-one basic change. The target functional induced by this change is then mapped through B_t^{-1}, giving the minimum-B_t-cost Riesz representative in Eq. (9).

The finite-atlas realization covers cases in which one rank-one chart contains no update that passes the nonlinear target conditions. At each fixed editable MLP location, it optimizes an inserted hidden residual through the frozen downstream network, as in the Riesz realization. Three fixed chart families vary the prompt weighting and residual scale: the rewrite-dominant family gives more weight to the rewrite prompt than to paraphrases, the balanced family weights rewrite and paraphrase prompts equally, and the low-drift family uses the rewrite-dominant residual with the smaller scale grid in Table 5. Equation (10) pairs each optimized residual with its subject key to form a basic matrix change. The MLP locations, prompt weightings, and scale grids are fixed before evaluation. Each chart has its own $A_{t,j}$ and $B_{t,j}$ but solves the same problem in Eq. (8). A temporarily applied update enters $\mathcal{V}_t$ only after it passes the full-network target margins and measured protected-cost budget.

Table 5: Implementation settings used by the reported CanonEdit runs. "Finite" and "Riesz" use different candidate chart sets but solve the same constrained representative problem.

Setting	Finite atlas	Riesz chart
Chart size	at most 12 basis vectors; at most 7 active constraints	one target basis; one active constraint
Regularization	$\alpha = .04$	$\alpha = .04$
Target condition	true rewrite margin ≥ 0; paraphrase floor .10	target margin .25; paraphrase floor .10
Protected-cost gates	P-MAE and KL-proxy thresholds 1.50	P-MAE and KL-proxy thresholds 1.50
Candidate scales	outer grid $\{2, 4, 6, 8, 10, 12, 16, 24, 32, 48\}$; interpolation $\{0, .25, .5, .75, 1\}$	$\{.5, .75, 1, 1.25, 1.5, 2\}$ with continuation factors up to 6
Nonlinear checks	full-network target and budget checks; at most four local corrections	the same checks; at most eight local corrections

Table 6: Backbone-specific edit locations. These locate-and-edit choices are fixed before evaluation and remain unchanged throughout each persistent stream.

Backbone	Edited module	Layer
GPT2-XL	`transformer.h.17.mlp.c_proj`	17
GPT-J-6B	`transformer.h.21.mlp.fc_out`	21
Qwen2.5-1.5B-Instruct	`model.layers.12.mlp.down_proj`	12
Llama-3.1-8B-Instruct	`model.layers.16.mlp.down_proj`	16

Protected prompts and nonlinear verification. Each request supplies up to two history-context prompts and three unrelated static prompts. Their pre-edit answer scores and coordinate gradients form $C_{t,j}$ for each chart and are cached across candidate checks for that edit; they are recomputed from θ_t for the next request. The linearized problem proposes updates satisfying the target half-spaces and local protected-cost budget. Each proposal is applied temporarily, scored on the rewrite, paraphrases, history contexts, and static prompts, and then removed. Only nonlinear-success proposals within budget enter the final deterministic minimum-cost selection. If none enters, the model is unchanged and the request contributes one failed write.

Persistent editing and timing. We use edit batch size one, maximum sequence length 128, FP16 model weights, evaluation and key batch size two, and a single designated GPU per process. The reported runs use NVIDIA RTX PRO 6000 Blackwell GPUs. The `sec/edit` field is the mean timed candidate-solve, nonlinear-check, and selection phase; it excludes one-time model loading, pre-edit key extraction, and reporting-only evaluation.

C.1 Observed solver path

Table 7 summarizes the operations in the controlled CounterFact–GPT-J-6B 10K trace. The solver usually commits the nonlinearly verified minimum-cost representative obtained from the closed-form active-set path; re-linearization is used on a small fraction of requests.

Table 7: Candidate evaluation statistics over 10,000 requests. Candidate probes share target computations; full evaluations apply a candidate and evaluate all target and protected prompts.

Operation	Mean/rate	95th percentile
Candidate target probes per request	6.24	10
Full candidate evaluations per request	1.40	3
Closed-form canonical path	93.90%	–
Nonlinear re-linearization triggered	6.13%	–

C.2 Isolated runtime

We separately time three methods on the same CounterFact–GPT-J-6B $N{=}300$ stream, GPU, process boundary, and evaluator. Table 8 reports three independent process repetitions. Edit time begins with an already tokenized request and ends after committing the update; end-to-end case time also includes request-level evaluation but excludes model loading.

Table 8: Controlled runtime at $N{=}300$ (mean $\pm$ sample standard deviation over three process repetitions). Accuracy values are online measurements in percent.

Method	Eff.	Gen.	edit s/case	end-to-end s/case	peak GPU MB
CanonEdit	99.33	99.33	$3.729 \pm .506$	$5.398 \pm .641$	23,272
AlphaEdit (FR target)	79.00	52.33	$.552 \pm .006$	$1.810 \pm .001$	23,133
MEMIT (FR target)	78.67	62.67	$.536 \pm .015$	$1.787 \pm .022$	23,133

D CONTROLLED-PROTOCOL AUDIT

For direct comparisons in Table 1, every method within a dataset block uses the same frozen 10K request pool, batch size one, GPT-J-6B revision, fixed layer 21, and evaluator. The evaluator computes teacher-forced sequence-log-probability margins on one rewrite, up to two paraphrases, up to two historical-context prompts, and up to three static prompts. Each accepted update persists. A row-level audit links every displayed value to its dataset-manifest hash, request-pool hash, method configuration, evaluator version, horizon, and source summary.

E REQUEST-LEVEL UNCERTAINTY

Table 9 reports moving-block bootstrap intervals from the controlled CounterFact–GPT-J-6B 10K stream. The same sampled blocks are used for paired differences because the two traces share case identifiers and order. These intervals quantify request-level variation within the fixed stream.

Table 9: Request-level 95% moving-block bootstrap intervals (1,000 resamples). Deltas are CANONEDIT minus AlphaEdit with the Forward Replay target.

Block	Quantity	Online Eff.	Gen.	Hist.
200	CANONEDIT	99.15–99.50	99.03–99.39	74.12–75.23
200	Paired delta	1.94–4.68	27.81–30.63	10.59–12.33
500	CANONEDIT	99.13–99.53	99.04–99.40	74.26–75.09
500	Paired delta	1.59–5.08	27.73–30.75	10.45–12.54

F PRIOR-EDIT AND HELD-OUT PROTECTION AUDIT

We freeze 256 evaluation-only prompts before editing; they are never used to construct any $C_{t,j}$, choose a chart, or select a representative. At the common 5K checkpoint, we also replay fixed cohorts written in edits 1–1,000 and 4,501–5,500. Table 10 shows that the measured protection advantage extends beyond the prompts used by the optimizer and that old rewrite and paraphrase behavior remains high at this checkpoint.

Table 10: Evaluation-only protection and prior-edit retention at the common 5K checkpoint. Early/middle entries are target/paraphrase retention (%).

Method	Held-out answer ret.	Held-out score MAE	Early target/para	Middle target/para
CANONEDIT	75.00	3.59	98.50/96.50	100.00/99.00
AlphaEdit (FR target)	61.33	10.46	96.50/70.50	99.00/72.00

G ADDITIONAL LONG-HORIZON RESULTS

Table 11: CANONEDIT after 10,000 sequential edits. Eff. is online writing efficacy; Gen., Ret., and Spec. are paraphrase generality, historical-context retention, and measured specificity (%). P-MAE and P-MSE are protected-score diagnostics. Both chart constructions solve the same canonical representative problem.

Model	Chart construction	Eff.	Gen.	Fail	Ret.	Spec.	P-MAE	P-MSE	sec/edit
CounterFact									
GPT2-XL	Finite atlas	99.24	70.47	76	63.55	100.00	.065	.033	4.32
GPT-J-6B	Finite atlas	99.33	99.22	67	74.65	100.00	.014	.005	2.80
Qwen2.5-1.5B	Finite atlas	99.29	79.19	71	60.56	99.99	.087	.048	3.67
WikiFactDiff									
GPT-J-6B	Finite atlas	99.68	79.47	32	57.62	99.99	.089	.026	4.41
ZsRE									
Qwen2.5-1.5B	Finite atlas	99.92	91.17	8	51.15	99.97	.036	.008	3.28
GPT-J-6B	Riesz chart	100.00	91.42	0	47.09	100.00	.005	.004	1.24
GPT2-XL	Riesz chart	99.99	96.18	1	40.57	100.00	.004	.003	1.83

Across these seven cells, online efficacy ranges from 99.24% to 100% and mean measured specificity from 99.97% to 100%. Both chart constructions therefore support the same canonical representative selection principle across multiple datasets and backbones.

H PROTECTED-SCORE SUMMARY

Table 12 summarizes the distribution of protected behavior in each highlighted 10K stream. Specificity includes the mean and final value; P-MAE and P-MSE include the mean and 95th percentile.

Table 12: Specificity and protected-score changes over three highlighted 10K streams.

Data	Chart construction	Specificity		P-MAE		P-MSE	
		mean	final	mean	p95	mean	p95
CounterFact	Finite atlas	1.000	1.000	.014	.086	.005	.013
ZsRE	Riesz chart	1.000	1.000	.004	.000	.003	.000
WikiFactDiff	Finite atlas	.9999	1.000	.089	.252	.026	.095

The 95th percentiles expose the tail of protected-score change beyond the mean.

I ONLINE EFFICIENCY FRONTIER

Figure 4 compares online efficacy with recorded solver-and-verification time for methods with 10K traces. On GPT-J-6B, CANONEDIT reaches 99.33%, 100%, and 99.68% efficacy at 2.80, 1.24, and 4.41 seconds per edit on CounterFact, ZsRE, and WikiFactDiff, respectively. Table 8 provides the controlled process-level comparison; this figure summarizes the quality–cost locations of the full 10K traces.

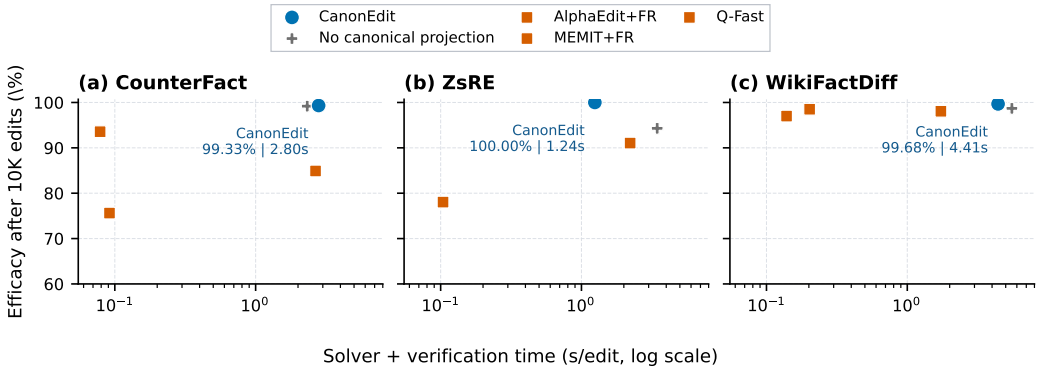


Figure 4: **Measured quality–time frontier after 10,000 edits.** Each panel uses case-level traces for one dataset on GPT-J-6B. Time covers candidate solving, nonlinear checks, and representative selection; the horizontal axis is logarithmic.

J BREADTH AT N=3,000

Table 13: Finite-atlas results after 3,000 sequential edits across three datasets and four backbones. Eff. is online writing efficacy; Gen., Ret., and Spec. are paraphrase generality, historical-context retention, and measured specificity (%). P-MAE and P-MSE are protected-score diagnostics.

Model	Eff.	Gen.	Fail	Ret.	Spec.	P-MAE	P-MSE	sec/edit
CounterFact								
GPT-J-6B	99.90	80.67	3	51.57	100.00	.297	.218	2.25
GPT2-XL	99.40	70.50	18	60.17	100.00	.119	.097	3.97
Llama-3.1-8B	99.27	84.60	22	41.43	99.93	.288	.298	5.35
Qwen2.5-1.5B	99.53	78.93	14	65.80	99.97	.166	.133	2.70
WikiFactDiff								
GPT-J-6B	100.00	83.10	0	57.63	99.97	.136	.049	3.61
GPT2-XL	99.03	69.80	29	57.13	99.97	.063	.033	25.54
Llama-3.1-8B	99.90	86.27	3	46.83	99.90	.214	.139	3.94
Qwen2.5-1.5B	99.10	73.73	27	66.93	99.93	.050	.023	5.09
ZsRE								
GPT-J-6B	99.60	85.13	12	38.60	99.93	.107	.053	3.18
GPT2-XL	98.83	87.10	35	45.70	99.97	.063	.025	3.40
Llama-3.1-8B	98.57	88.17	43	34.60	99.77	.143	.119	3.65
Qwen2.5-1.5B	99.73	89.37	8	49.03	99.93	.059	.021	2.24

K COMPLETE BASELINE COMPARISON MATRIX

Tables 14 and 15 report 96 runs at 3K and 37 at 10K. The comparisons include the no-edit control, ROME, MEMIT, AlphaEdit, and their Forward Replay target variants (Meng et al., 2022a;b; Fang et al., 2025; Liu et al., 2026c). The matrices also include two mechanism variants: "No canonical projection" isolates representative selection, while Q-Fast evaluates reduced nonlinear verification. Eff. denotes online writing efficacy; Gen., Ret., and Spec. denote paraphrase generality, historical-context retention, and measured specificity (%). P-MAE/P-MSE are protected-score diagnostics.

Table 14: Persistent-edit comparison matrix at $N = 3{,}000$. The corresponding CANONEDIT results are in the preceding table.

Method	Eff.	Gen.	Fail	Ret.	Spec.	P-MAE	P-MSE	sec/edit
CounterFact – GPT-J-6B								
Unedited	25.93	36.87	2222	76.23	100.00	.000	.000	0.00
ROME	30.13	39.80	2096	73.20	100.00	.036	.004	0.08
MEMIT	28.60	39.17	2142	74.17	100.00	.036	.004	0.09
AlphaEdit	28.53	37.60	2144	74.03	100.00	.015	.001	0.09
MEMIT+FR	84.10	72.77	477	35.57	99.93	.971	2.337	0.09
AlphaEdit+FR	95.67	74.60	130	61.43	99.97	.566	.727	0.09
No canonical projection	97.10	68.70	87	66.53	100.00	.267	.172	2.73
Q-Fast	95.63	75.83	131	63.20	100.00	.544	.671	1.53
CounterFact – GPT2-XL								
Unedited	31.87	41.60	2044	69.17	100.00	.000	.000	0.00
ROME	65.73	67.70	1028	36.90	100.00	.251	.086	0.07
MEMIT	65.73	67.70	1028	36.90	100.00	.251	.086	0.08
AlphaEdit	58.27	56.70	1252	65.13	100.00	.064	.014	0.08
MEMIT+FR	92.47	89.37	226	11.97	99.90	1.385	4.122	0.08
AlphaEdit+FR	82.87	68.33	514	60.77	100.00	.168	.075	0.08
No canonical projection	94.17	70.63	175	52.00	100.00	.119	.065	2.44
Q-Fast	86.00	69.53	420	60.23	100.00	.180	.075	1.44
CounterFact – Llama-3.1-8B								
Unedited	23.80	32.33	2286	82.27	100.00	.000	.000	0.00
ROME	48.87	53.17	1534	54.40	100.00	.061	.015	0.29
MEMIT	48.87	53.17	1534	54.40	100.00	.061	.015	0.08
AlphaEdit	42.33	50.93	1737	65.83	100.00	.039	.007	0.08
MEMIT+FR	96.20	94.57	114	5.10	100.00	2.618	13.653	0.08
AlphaEdit+FR	54.03	49.77	1383	75.03	96.57	.766	1.916	0.08
No canonical projection	99.33	82.60	20	40.23	99.93	.302	.314	2.88
Q-Fast	76.77	74.63	697	49.53	99.87	.276	.211	1.57
CounterFact – Qwen2.5-1.5B								
Unedited	26.87	40.57	2194	75.63	100.00	.000	.000	0.00
ROME	69.20	73.30	924	39.20	100.00	.312	.165	0.19
MEMIT	69.20	73.30	924	39.20	100.00	.312	.166	0.06
AlphaEdit	58.57	61.23	1243	75.87	100.00	.152	.070	0.06
MEMIT+FR	96.53	97.97	104	6.90	99.57	2.566	12.947	0.06

Continued on next page

Table 14 (continued)

Method	Eff.	Gen.	Fail	Ret.	Spec.	P-MAE	P-MSE	sec/edit
AlphaEdit+FR	90.10	80.40	297	69.23	99.80	.301	.322	0.06
No canonical projection	99.10	84.90	27	59.93	100.00	.182	.147	2.05
Q-Fast	92.03	84.07	239	67.67	99.90	.333	.375	1.23
WikiFactDiff – GPT-J-6B								
Unedited	46.13	48.23	1616	64.73	100.00	.000	.000	0.00
ROME	45.83	46.40	1625	57.23	100.00	.015	.001	0.35
MEMIT	45.83	46.30	1625	57.17	100.00	.015	.001	0.12
AlphaEdit	44.10	45.70	1677	62.50	100.00	.007	.000	0.12
MEMIT+FR	97.40	94.63	78	17.07	97.23	.634	1.627	0.12
AlphaEdit+FR	97.67	86.60	70	47.37	100.00	.191	.113	0.12
No canonical projection	99.00	84.67	30	53.10	100.00	.156	.074	5.04
Q-Fast	98.03	85.17	59	50.10	100.00	.170	.095	2.10
WikiFactDiff – GPT2-XL								
Unedited	31.43	43.07	2057	70.70	100.00	.000	.000	0.00
ROME	58.57	61.90	1243	44.67	100.00	.027	.003	0.38
MEMIT	58.47	61.90	1246	44.63	100.00	.027	.003	0.12
AlphaEdit	56.00	45.50	1320	61.30	100.00	.018	.001	0.11
MEMIT+FR	92.97	93.20	211	12.10	99.37	.779	1.875	0.11
AlphaEdit+FR	87.30	70.07	381	56.33	100.00	.103	.029	0.11
No canonical projection	96.03	72.03	119	55.57	99.97	.073	.021	4.62
Q-Fast	90.93	72.63	272	57.13	99.97	.114	.034	5.38
WikiFactDiff – Llama-3.1-8B								
Unedited	47.07	46.33	1588	64.53	100.00	.000	.000	0.00
ROME	49.70	49.43	1509	56.37	100.00	.046	.007	0.37
MEMIT	49.77	49.47	1507	56.37	100.00	.046	.007	0.10
AlphaEdit	52.83	54.17	1420	57.83	99.97	.035	.008	0.10
MEMIT+FR	98.57	98.87	43	6.13	92.53	2.476	16.528	0.10
AlphaEdit+FR	74.13	66.63	779	62.57	99.53	.252	.253	0.10
No canonical projection	99.77	87.03	7	47.00	99.53	.230	.204	3.91
Q-Fast	86.90	81.37	393	44.70	99.63	.228	.162	1.96
WikiFactDiff – Qwen2.5-1.5B								
Unedited	40.63	42.23	1781	66.33	100.00	.000	.000	0.00
ROME	48.33	48.83	1550	61.70	100.00	.022	.005	0.21
MEMIT	48.50	49.07	1545	61.63	100.00	.022	.005	0.10
AlphaEdit	46.57	48.00	1603	70.83	100.00	.010	.001	0.10
MEMIT+FR	90.53	88.80	284	22.00	88.53	1.870	8.266	0.13
AlphaEdit+FR	90.57	75.17	283	64.03	99.60	.182	.170	0.12
No canonical projection	96.93	74.13	92	62.93	100.00	.042	.017	3.90
Q-Fast	92.40	76.70	228	68.53	99.83	.149	.119	2.24
ZsRE – GPT-J-6B								
Unedited	42.07	44.23	1738	63.87	100.00	.000	.000	0.00
ROME	51.20	51.70	1464	50.33	100.00	.023	.002	0.23
MEMIT	51.17	51.73	1465	50.43	100.00	.023	.002	0.12
AlphaEdit	48.73	49.30	1538	56.17	100.00	.014	.001	0.12
MEMIT+FR	96.57	93.67	103	3.87	100.00	1.024	4.127	0.12
AlphaEdit+FR	84.73	76.13	458	48.30	100.00	.096	.043	0.11
No canonical projection	97.13	83.03	86	43.37	100.00	.087	.038	3.76
Q-Fast	94.50	84.73	165	46.20	100.00	.121	.061	1.97
ZsRE – GPT2-XL								
Unedited	52.27	52.60	1432	52.43	100.00	.000	.000	0.00
ROME	74.90	74.83	753	26.43	100.00	.079	.015	0.25
MEMIT	74.90	74.80	753	26.37	100.00	.079	.015	0.07
AlphaEdit	63.63	64.03	1091	42.07	100.00	.044	.005	0.07
MEMIT+FR	97.17	97.20	85	3.40	99.87	1.311	4.632	0.07
AlphaEdit+FR	88.60	85.60	342	42.60	99.93	.101	.033	0.08
No canonical projection	97.97	90.20	61	38.63	99.97	.068	.024	2.72
Q-Fast	91.83	88.83	245	40.97	100.00	.097	.030	1.54
ZsRE – Llama-3.1-8B								
Unedited	34.60	39.17	1962	67.40	100.00	.000	.000	0.00
ROME	54.47	54.73	1366	47.53	100.00	.036	.015	0.26
MEMIT	54.53	54.63	1364	47.53	100.00	.036	.015	0.08
AlphaEdit	57.03	58.50	1289	51.37	99.97	.026	.011	0.08
MEMIT+FR	98.87	99.03	34	2.17	94.10	3.165	28.131	0.08
AlphaEdit+FR	71.47	68.70	856	49.73	99.77	.413	.921	0.08
No canonical projection	99.37	89.33	19	27.90	99.87	.144	.097	3.89
Q-Fast	87.53	82.80	374	33.50	99.90	.247	.313	2.33
ZsRE – Qwen2.5-1.5B								
Unedited	44.13	45.00	1676	60.00	100.00	.000	.000	0.00
ROME	65.73	64.53	1028	37.23	100.00	.071	.019	0.13
MEMIT	65.73	64.53	1028	37.23	100.00	.071	.019	0.11
AlphaEdit	63.67	64.50	1090	69.30	99.97	.086	.024	0.08
MEMIT+FR	98.67	98.57	40	1.57	96.50	3.216	24.339	0.09
AlphaEdit+FR	90.13	86.50	296	52.87	99.70	.134	.059	0.09
No canonical projection	97.67	93.87	70	46.00	99.63	.104	.056	2.69
Q-Fast	97.53	96.03	74	48.13	99.77	.147	.060	1.05

Table 15: Persistent-edit comparison matrix at $N = 10{,}000$.

Method	Eff.	Gen.	Fail	Ret.	Spec.	P-MAE	P-MSE	sec/edit
CounterFact – GPT-J-6B								
Unedited	25.94	36.92	7406	76.26	100.00	.000	.000	0.00
MEMIT	35.38	44.81	6462	66.52	100.00	.019	.001	0.09
AlphaEdit	33.95	43.88	6605	70.44	100.00	.009	.000	0.08
MEMIT+FR	75.62	70.56	2438	41.16	99.98	.516	.946	0.09
AlphaEdit+FR	93.56	74.92	644	62.27	99.99	.302	.279	0.08
No canonical projection	91.63	69.05	837	65.78	100.00	.160	.087	2.39
Q-Fast	84.90	70.16	1510	68.83	99.99	.236	.217	2.66
CounterFact – GPT2-XL								
MEMIT+FR	90.58	87.26	942	13.11	99.97	.610	1.353	0.12
AlphaEdit+FR	92.48	74.11	752	65.99	100.00	.131	.043	0.12
No canonical projection	97.61	75.54	239	59.27	99.99	.068	.022	3.98
Q-Fast	93.41	74.68	659	64.23	100.00	.125	.039	1.81
CounterFact – Llama-3.1-8B								
AlphaEdit+FR	53.45	51.18	4674	69.01	99.03	.312	.530	0.32
CounterFact – Qwen2.5-1.5B								
MEMIT+FR	94.17	97.15	583	15.86	99.87	1.121	4.460	0.09
AlphaEdit+FR	93.58	83.18	642	66.82	99.88	.173	.151	0.12
No canonical projection	97.34	83.27	266	60.10	100.00	.075	.042	3.29
Q-Fast	95.11	86.50	489	63.11	99.97	.179	.156	1.05
WikiFactDiff – GPT-J-6B								
MEMIT+FR	97.00	90.05	300	18.31	98.60	.322	.606	0.14
AlphaEdit+FR	98.51	79.71	149	52.61	100.00	.105	.046	0.20
No canonical projection	98.70	78.44	130	59.39	99.99	.087	.032	5.53
Q-Fast	98.06	76.26	194	59.15	100.00	.086	.035	1.73
WikiFactDiff – GPT2-XL								
MEMIT+FR	88.56	87.80	1144	16.53	99.80	.318	.599	0.27
AlphaEdit+FR	86.48	64.28	1352	56.66	100.00	.062	.014	0.17
No canonical projection	95.86	62.17	414	58.22	100.00	.034	.006	4.46
Q-Fast	92.32	67.34	768	57.29	99.99	.063	.014	1.89
WikiFactDiff – Llama-3.1-8B								
AlphaEdit+FR	70.37	63.52	2974	57.84	99.81	.130	.098	0.44
WikiFactDiff – Qwen2.5-1.5B								
MEMIT+FR	96.36	92.57	364	13.93	93.91	1.330	4.671	0.10
AlphaEdit+FR	94.32	77.44	568	66.64	99.80	.181	.113	0.10
No canonical projection	97.79	74.70	221	61.19	100.00	.027	.004	3.40
Q-Fast	96.24	78.05	376	68.41	99.84	.153	.095	1.01
ZsRE – GPT-J-6B								
AlphaEdit+FR	78.04	69.95	2196	51.08	100.00	.048	.015	0.10
No canonical projection	94.30	78.78	570	42.52	99.99	.050	.023	3.45
Q-Fast	91.05	77.86	895	51.81	100.00	.075	.032	2.22
ZsRE – GPT2-XL								
AlphaEdit+FR	90.64	87.69	936	45.33	100.00	.054	.013	0.15
No canonical projection	98.80	90.27	120	43.38	99.98	.032	.008	6.42
ZsRE – Llama-3.1-8B								
AlphaEdit+FR	65.57	62.20	3443	52.25	99.87	.147	.262	0.28
ZsRE – Qwen2.5-1.5B								
AlphaEdit+FR	93.90	87.75	610	50.75	99.91	.055	.019	0.08
No canonical projection	98.76	94.07	124	50.07	99.87	.052	.019	3.12