PACE: PRE-WRITE EXPOSURE IN CONTINUAL KNOWLEDGE EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual knowledge editing has a blind spot: editors decide where and how to write a new fact but typically do not estimate which prior mappings that writing may expose. We formulate this problem as *pre-write exposure* and study sparse edit-conditioned feed-forward-network support as a prospective exposure signal. This leads to PACE, a bounded-state editor that extracts one activation path before each write and reuses it to coordinate localization, risk-aware replay, sparse solving, and preservation. Across GPT-J, LLaMA-3, and routed Qwen3, support overlap predicts interference more strongly than prompt semantics, while controlled interventions that change overlap under closely matched realized write strength produce corresponding changes in prior-edit damage. Under a fixed replay budget, better exposure ranking is strongly associated with lower forgetting, and the same ranking signal transfers to other editors. At reliability matched to UltraEdit, PACE improves locality by 1.9–2.0 points and reduces forgetting by 1.2–1.7 points on GPT-J and LLaMA-3. These results suggest a broader principle for continual editing: model which prior mappings a write may expose before deciding how to apply it.

1 INTRODUCTION

Large language models encode factual associations in their parameters, but those associations continually change. Knowledge editing aims to revise a targeted fact without retraining the model or disrupting unrelated behavior (De Cao et al., 2021; Mitchell et al., 2022a; Meng et al., 2022; Yao et al., 2023).Transformer feed-forward networks (FFNs) have been shown to encode factual associations (Geva et al., 2021), motivating many parametric editors to realize factual updates by modifying localized FFN parameters (Meng et al., 2022; 2023; Fang et al., 2025; Li et al., 2025). Continual editing introduces a different difficulty: every new write is applied on top of previous writes, so a locally successful update can damage mappings established earlier in the stream (Gupta et al., 2024b;a; Thede et al., 2025; Hu et al., 2025). Most existing parametric editors therefore emphasize a *write rule*: where should the incoming fact be written, and how should that update be constrained? We study a complementary variable that is largely implicit in this formulation: *before the write, which previous mappings are exposed to it?*

We call this quantity *pre-write exposure*. The idea follows from a simple local support principle. For the support-restricted FFN updates considered here, a realized incremental write can directly change a represented FFN mapping only through coordinates touched by that write. Consequently, disjointness from the active update union is sufficient for exact zero direct change to that represented mapping. This statement is deliberately local: it does not imply global Transformer non-interference, because upstream representation drift and downstream nonlinear computation may still alter behavior. More importantly, the exact active union is known only after replay and the update are constructed. A continual editor therefore needs a prospective estimate of exposure *before* the write is applied.

We obtain such an estimate from an *activation path*: a sparse, edit-conditioned set of layer–coordinate pairs extracted from the pre-write model. The path is not treated as a recovered causal circuit or an exact interference certificate. Instead, overlap between the incoming path and a stored path estimates whether the two edits may compete for the same editable support. Figure 1 illustrates the central empirical phenomenon: manipulating shared support changes cross-edit interference. On

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

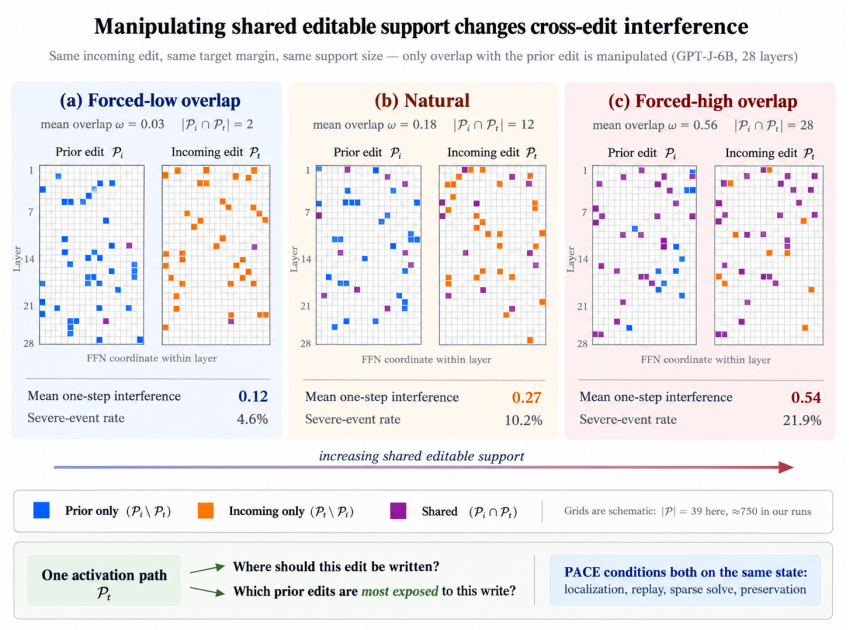

Figure 1: **Manipulating shared editable support changes cross-edit interference.** On GPT-J-6B, forcing mean overlap $0.03 \to 0.18 \to 0.56$ over $n = 1{,}000$ fixed prior–incoming pairs raises mean interference $0.12 \to 0.27 \to 0.54$ and severe-event frequency $4.6\% \to 10.2\% \to 21.9\%$. LLaMA-3 reproduces the ordering while closely matching realized incoming target gain and update norm. The intervention motivates using pre-write support overlap as a prospective exposure signal rather than as an exact damage estimator.

GPT-J-6B, forcing progressively greater overlap over the same prior–incoming pairs increases both mean interference and severe-event frequency; LLaMA-3 reproduces the ordering while closely matching realized write strength. In other words, continual editing benefits from an *exposure model*, not only a write rule.

This perspective leads to PACE (**P**ath-conditioned **A**daptive **C**ontinual **E**diting), a bounded-state strict-online editor that extracts one activation path before each write and reuses it as the common state for four decisions:

$$\mathcal{P}_t \Longrightarrow \left\{ \underbrace{\text{where to write}}_{\text{localization}}, \underbrace{\text{which mappings are exposed}}_{\text{risk ranking}}, \underbrace{\text{where to solve}}_{\text{sparse update}}, \underbrace{\text{what to preserve}}_{\text{protection}} \right\}.$$

The path selects edit-specific boundary layers, ranks retained edits by support overlap, restricts the online solve to the incoming and replayed supports, and defines path-restricted preservation keys. Thus the contribution is not localization, replay, least squares, or null-space preservation in isolation; it is the use of one pre-write exposure state to coordinate decisions that are otherwise made from separate representations. Preservation is preventive rather than restorative: the current write is constrained not to move selected protected keys, but previously accumulated drift is not reconstructed.

We test this exposure hypothesis through mechanism-identifying controls rather than relying only on final editing scores. First, support overlap is the strongest tested pre-write predictor of one-step interference, and its ordering persists prospectively on held-out GPT-J and LLaMA-3 streams and in-stream on routed Qwen3. Second, realized-write-matched intervention shows that increasing overlap increases damage even when incoming target gain and update norm remain closely matched. Third, under a fixed replay budget, better exposure ranking is strongly associated with lower forgetting across nine ranking policies. The same signal also transfers beyond PACE: replacing random replay ranking with path-overlap ranking reduces forgetting when wrapped around AlphaEdit, RLSEdit, and UltraEdit under otherwise matched replay conditions. Finally, separating the single exposure

state into independently extracted role states degrades the fit–preservation trade-off on both GPT-J and LLaMA-3. Together, these experiments distinguish two claims: support overlap is useful as a transferable pre-write risk signal, and PACE gains an additional benefit by coordinating the full editor through the same state.

We also test whether the apparent preservation gain is merely the consequence of weaker fitting. It is not. After prospectively selecting only the target-margin parameter on a record- and subject-disjoint development stream, PACE-MATCHED reaches essentially the same reliability as UltraEdit while improving generalization by 1.8 points, locality by 1.9–2.0 points, and forgetting by 1.2–1.7 points on GPT-J and LLaMA-3 CounterFact; the same frozen protocol transfers to LLaMA-3 ZsRE, and the operating-point differences persist across five matched stream orders. These results support a broader conclusion than the performance of one editor: *continual editing should model which prior mappings an incoming write may expose before deciding how to apply that write.*

Contributions. First, we formulate *pre-write exposure* as a distinct problem in continual knowledge editing and give local support-based statements that separate predicted exposure from the active update union that determines exact direct change. Second, we introduce PACE, which uses one pre-write edit-conditioned support state as a common interface for localization, risk-aware replay, sparse solving, and preservation. Third, we show through prospective prediction, controlled support intervention, fixed-budget ranking, cross-editor replay transfer, state decoupling, and matched-reliability replication that pre-write exposure is both operationally useful and consequential for continual retention across dense and routed language-model backbones.

2 RELATED WORK

Parametric editing and localization. Knowledge editing changes targeted model behavior while preserving unrelated behavior (De Cao et al., 2021; Mitchell et al., 2022a; Yao et al., 2023). Parametric editors realize such changes through localized rank-one, batched, or least-squares-style updates (Meng et al., 2022; 2023; Li et al., 2024; Gu et al., 2024; Gupta et al., 2024c; Fang et al., 2025; Li et al., 2025), while localization methods identify influential layers, neurons, or representations (Dai et al., 2022; Hase et al., 2023; Zhang et al., 2025; Datta et al., 2026; Guo et al., 2025). Adaptive-layer and sparse-support methods further reduce the intervention surface (Hu et al., 2024; Wang et al., 2025). A large class of these methods uses localization primarily to determine where the current edit should act. PACE asks a different question of the same local structure: which previously represented mappings are exposed *before* the new write is applied? Its sparse edit-conditioned support therefore serves not only as a write location, but also as a prospective risk signal. We treat this support as an operational state rather than a recovered causal circuit or exact interference certificate.

Continual editing and interference control. Sequential editing compounds interference across writes, motivating methods that stabilize parameter subspaces (Fang et al., 2025; Xu et al., 2025), adapt intervention locations (Hu et al., 2024), maintain auxiliary memory (Wang et al., 2024), accumulate recursive least-squares state (Wang et al., 2026), scale lifelong updates (Gu et al., 2026), or regularize representations and parameter geometry (Liu et al., 2026; Ma et al., 2025; Dai et al., 2026; Hu et al., 2025). These approaches address how past information should be retained while new edits are incorporated. PACE contributes a complementary pre-write variable: an exposure estimate over retained mappings. It uses support overlap to rank which prior edits are most at risk, then reuses the same state to restrict the sparse solve and define current-step preservation. Thus the novelty is not replay, least squares, or null-space projection individually, but the coordination of prospective interference estimation and the write mechanism through one edit-conditioned state.

Alternative state and access regimes. External-memory editors such as SERAC, GRACE, IKE, NeuralDB, and LightEdit retain or retrieve edit-specific state at inference time (Mitchell et al., 2022b; Hartvigsen et al., 2023; Zheng et al., 2023; Fei et al., 2026; Jung et al., 2026), while LocFT-BF revisits accumulated edits breadth-first (Yang et al., 2026). These methods operate under different state and intervention regimes from PACE's bounded-state strict-online parametric setting, so we use them as cross-class operating points rather than protocol-identical baselines. Recent work likewise emphasizes lifelong scaling and heterogeneous editing behavior (Thede et al., 2025; Chen et al., 2026; Gu et al., 2026). PACE instead tests a specific coordination hypothesis: one pre-write

exposure state should jointly determine where to write, which prior mappings to protect, where to solve, and what to preserve. Cross-editor replay transfer tests whether the exposure signal itself generalizes, while state-decoupling controls test the additional benefit of sharing that state across the full editor.

3 PRE-WRITE EXPOSURE FOR CONTINUAL EDITING

A continual editor faces two coupled questions: *where can the incoming write act, and which retained mappings can that write expose?* PACE answers both from one pre-write, edit-conditioned support state. Let $\mathcal{P}_t$ denote the activation path predicted before edit t is applied, and let $\mathcal{U}_t^\ell$ denote the active coordinate union formed after replay is selected. The first is a prospective estimate of exposure; the second is the support on which the represented update is actually solved. This distinction is the basis of both the method and its guarantees.

3.1 SETUP AND THE SUPPORT PRINCIPLE

At step t, an edit $e_t = (s_t, r_t, o_t \to o_t^\star)$ with prompt x_t arrives before future edits are known. Following the associative-memory view of Transformer FFNs (Geva et al., 2021; Meng et al., 2022; 2023), we edit the FFN output projection at the subject-token position:

$$a^\ell(x) = \psi(W_{\text{in}}^\ell h^{\ell-1}(x)), \qquad m^\ell(x) = W_{\text{out}}^\ell a^\ell(x), \qquad W_{\text{out},t}^\ell = W_{\text{out},0}^\ell + E_t^\ell, \tag{1}$$

where E_t^ℓ is the cumulative correction relative to the base output projection. At each layer, editing is restricted to a fixed coordinate universe $\mathcal{I}_{\text{edit}}^\ell$, selected once from held-out activation statistics and then frozen (Appendix A.2).

The support principle is simple. Let $\mathcal{U}_t^\ell$ be the active coordinate union on which the step-t increment is written. Because that increment is exactly zero outside $\mathcal{U}_t^\ell$, a prior materialized mapping whose stored key has zero embedding in the union undergoes exactly zero direct change at that step; when the key is exposed inside the union, Proposition 1 bounds its direct change by the realized increment norm and the exposed key energy. The exact condition therefore concerns $\mathcal{U}_t^\ell$, which is known only after replay is selected. PACE instead estimates exposure before writing through $\mathcal{P}_t$. Path overlap is consequently a prospective risk signal, whereas disjointness from the active union provides the sufficient exact zero-direct-change condition.

3.2 A PRE-WRITE EXPOSURE STATE

From one pre-write forward pass, PACE calibration-normalizes subject-token activations,

$$s_j^\ell(x_t) = \frac{|a_j^\ell(x_t)| - \mu_\ell}{\sigma_\ell + \varepsilon}.$$

A layer enters $\mathcal{A}_t$ when its top-k_{lay} mean score exceeds τ_{lay}. Within an active layer,

$$\mathcal{P}_t^\ell = \text{TopK}_{\lceil \varrho d_{\text{edit}}^\ell \rceil}\big(s^\ell(x_t); \mathcal{I}_{\text{edit}}^\ell\big) \cap \{j \in \mathcal{I}_{\text{edit}}^\ell : s_j^\ell(x_t) \geq \tau_{\text{coord}}\}, \qquad \mathcal{P}_t = \{(\ell, j) : \ell \in \mathcal{A}_t,\ j \in \mathcal{P}_t^\ell\}. \tag{2}$$

The percentile is an upper budget rather than a quota; deterministic minimum-support and empty-path fallbacks are given in Appendix A.2.

PACE uses this same pre-write state for two decisions before solving the update. First, it materializes the edit only at the first and last active layers,

$$\mathcal{B}_t = \{\min \mathcal{A}_t, \max \mathcal{A}_t\}, \tag{3}$$

while interior active layers remain part of the edit-level path used for exposure ranking. Exact stored-mapping guarantees therefore apply only to materialized boundary mappings.

Second, for retained edit i, PACE scores prospective exposure by path overlap,

$$\omega(i, t) = \frac{|\mathcal{P}_i \cap \mathcal{P}_t|}{|\mathcal{P}_i \cup \mathcal{P}_t|}. \tag{4}$$

From bounded memory $\mathcal{M}$ of capacity M, it selects the top-K positive-overlap records to form the global replay set $\mathcal{R}_t$. A boundary layer uses only selected records carrying a materialized constraint at that layer, $\mathcal{R}_t^\ell \subseteq \mathcal{R}_t$ (Appendix A.4). Thus exposure ranking is edit level, whereas replay and solving remain layer valid. Because replayed supports can enlarge the eventual active union, zero incoming-path overlap is not itself a certificate of zero direct change.

3.3 From Exposure to a Bounded Online Write

PACE next converts the selected path and replay set into a bounded update. A two-boundary edit should not request the full target change independently at both layers. Let $\mathcal{C}_t$ contain the first tokens of the new and old objects together with competing candidates, with $c^\star$ the new-object token and c^- the highest-scoring non-target token. PACE requests only the missing margin,

$$\delta_t = \max(0, \gamma - [z_{c^\star}(x_t) - z_{c^-}(x_t)]), \qquad b_t = \tfrac{\delta_t}{2}(e_{c^\star} - e_{c^-}), \tag{5}$$

where γ controls fitting strength. With $J_t^{\mathcal{B}}$ the candidate-logit Jacobian with respect to simultaneous additive interventions at the selected boundaries, the joint ridge target is

$$\mathbf{r}_t^{\mathcal{B}} = (J_t^{\mathcal{B}})^\top \left(J_t^{\mathcal{B}}(J_t^{\mathcal{B}})^\top + \gamma_r I\right)^{-1} b_t, \tag{6}$$

whose layer blocks are r_t^ℓ. This is a local first-order construction; Appendix A.3 separates the additive-intervention remainder from downstream-key shift under the physical weight write, and Appendix B.37 measures realized fidelity.

After replay selection, boundary layer ℓ is solved only on the incoming and layer-valid replay supports:

$$\mathcal{U}_t^\ell = \mathcal{P}_t^\ell \cup \bigcup_{i \in \mathcal{R}_t^\ell} \mathcal{P}_i^\ell, \qquad u_t^\ell = |\mathcal{U}_t^\ell| \le (|\mathcal{R}_t^\ell| + 1)\lceil \varrho d_{\text{edit}}^\ell \rceil \le (K+1)\lceil \varrho d_{\text{edit}}^\ell \rceil. \tag{7}$$

Hence, for fixed K and support budget, the solve width is independent of the number of edits already processed.

Let S_i^ℓ select the stored support of record i, let $k_i^\ell = (S_i^\ell)^\top a^\ell(x_i)$ be its restricted key, and let Q_t^ℓ select the current union. Its current-union embedding is

$$\kappa_{i,t}^\ell = (Q_t^\ell)^\top S_i^\ell k_i^\ell.$$

Each represented record carries a materialized target ρ_i^ℓ; for the incoming record,

$$\rho_t^\ell = E_{t-1}^\ell S_t^\ell k_t^\ell + r_t^\ell.$$

Let $\mathcal{H}_t^\ell$ denote the represented constraints with nonzero embedding into the current union. Conditional on this retained state, PACE solves the age-decayed weighted ridge objective

$$\Delta_{t,\mathcal{U}}^\ell = \arg\min_{\Delta} \sum_{i \in \mathcal{H}_t^\ell} \widetilde{w}_{i,t}^\ell \|\Delta \kappa_{i,t}^\ell - \rho_i^\ell\|_2^2 + \lambda \|\Delta\|_F^2 = B_t^\ell (A_t^\ell + \lambda I)^{-1}, \tag{8}$$

where A_t^ℓ and B_t^ℓ are the corresponding weighted sufficient statistics. The incoming record has unit weight; replayed records receive overlap-dependent weight before age decay. Exact removal–reinsertion, coordinate re-embedding, per-block aging, ownership, and eviction are given in Appendices A.4–A.5. For $\lambda > 0$, the displayed solution is the unique minimizer of the represented ridge objective.

By construction, the cumulative correction is unchanged outside the active union:

$$E_t^\ell \left(I - Q_t^\ell (Q_t^\ell)^\top\right) = E_{t-1}^\ell \left(I - Q_t^\ell (Q_t^\ell)^\top\right). \tag{9}$$

3.4 Path-Restricted Preservation

Replay steers the sparse solve toward selected prior targets but does not by itself guarantee invariance of protected mappings. PACE-Full therefore stacks path-restricted neutral calibration keys together

with nonzero history keys from layer-valid replay records into a preservation matrix $K_{0,t}^{\ell}$. Let $\Phi_{K,t}^{\ell}$ be an orthonormal basis for its nonzero column span and define

$$P_{0,t}^{\ell} = I - \Phi_{K,t}^{\ell}(\Phi_{K,t}^{\ell})^{\top}. \tag{10}$$

For the unconstrained cumulative solution, let

$$G_{t,\mathcal{U}}^{\ell} = \Delta_{t,\mathcal{U}}^{\ell} - E_{t-1}^{\ell} Q_t^{\ell}.$$

PACE-Full writes

$$E_t^{\ell} Q_t^{\ell} \leftarrow E_{t-1}^{\ell} Q_t^{\ell} + G_{t,\mathcal{U}}^{\ell} P_{0,t}^{\ell}. \tag{11}$$

The resulting incremental write is exactly zero on the selected preservation-key subspace. For a replayed history key in that span, PACE-Full therefore prevents *additional* direct drift at the current step; it does not repair drift accumulated earlier. PACE-Soft sets $P_{0,t}^{\ell} = I$ and retains target steering without this hard current-step invariance. The guarantee is local to the selected FFN mappings, and the post-solve projection need not equal the exact affine-constrained least-squares optimum; Appendix A.10 gives the compact implementation and objective-gap bound.

3.5 One State Coordinates the Editor

The central abstraction of PACE is that the same pre-write state connects exposure estimation to the update itself:

$$\mathcal{P}_t \Longrightarrow \left\{ \underbrace{\mathcal{B}_t}_{\text{where to write}}, \quad \underbrace{\omega(i,t)}_{\text{which mappings are exposed}}, \quad \underbrace{\mathcal{U}_t^{\ell}}_{\text{where to solve}}, \quad \underbrace{K_{0,t}^{\ell}}_{\text{what to preserve}} \right\}. \tag{12}$$

The activation path is therefore not merely a localization heuristic; it is the common coordinate system through which PACE connects prospective exposure, sparse updating, and preservation. Localization, replay, recursive least squares, and null-space projection are not individually new primitives. The contribution is their coordination through one edit-conditioned state available before the write. Section 4 tests these two claims separately: cross-editor replay transfer isolates the portability of the exposure signal, while state-decoupling controls test the additional value of sharing that state across the full editor.

3.6 Scope and Implementation

For fixed replay count K and path budget, Equation 7 bounds the active solve width independently of stream length; bounded replay capacity also bounds the persistent logical editor state (Appendix A.13). The formal guarantees are deliberately local. Equation 8 exactly minimizes the represented ridge objective, Equation 11 exactly prevents additional direct change on the selected preservation-key subspace, Proposition 1 gives a sufficient zero-direct-change condition for an unexposed materialized mapping, and Theorem 1 bounds finite-horizon direct mapping drift attributable to missed realized-support collisions. None of these statements implies invariance of arbitrary downstream behavior, and because PACE materializes only boundary-layer constraints, they apply only to represented materialized mappings. The empirical connection from missed collisions to realized mapping drift and behavioral forgetting is evaluated separately in Table 77.

For routed FFNs, support indices extend from (ℓ, j) to (ℓ, e, j) with expert-specific calibration and editable universes; the same exposure ranking, sparse solve, and preservation machinery then applies to the expert-weight coordinates actually edited (Appendix A.11). This extension does not cover router logits, attention, residual-stream interactions, KV-cache state, or future expert reassignment. We therefore interpret the routed-backbone experiment as transfer of the support mechanism under richer indexing, not as routing invariance or a causal routing claim.

4 Experimental Results

We organize the experiments around one question: *does pre-write exposure provide a useful control variable for continual editing?* We test four implications of this hypothesis: exposure should predict

Backbone	Method	Rel.↑	Gen.↑	Loc.↑	For.↓
GPT-J-6B	UltraEdit	93.4±0.4	82.2±0.5	93.6±0.3	7.2±0.4
	PACE-FULL	91.8±0.4	83.6±0.4	**96.9±0.2**	**4.8±0.3**
	PACE-MATCHED	93.3±0.4	**84.0±0.4**	95.5±0.3	6.0±0.3
LLaMA-3-8B	UltraEdit	93.7±0.3	83.4±0.4	94.6±0.3	6.7±0.4
	PACE-FULL	93.0±0.3	85.0±0.4	**97.1±0.2**	**4.5±0.3**
	PACE-MATCHED	93.7±0.3	**85.2±0.4**	96.6±0.2	5.0±0.3

Table 1: Matched-reliability comparison after 10K CounterFact edits, reported as mean ± standard deviation over five matched stream-order seeds. PACE-MATCHED differs from PACE-FULL only in γ, selected prospectively on a record- and subject-disjoint development stream and frozen before final evaluation. Full sweeps and order-level results are in Appendix B.4.

and modulate interference, better exposure ranking should improve retention under a fixed replay budget, the signal should transfer beyond PACE, and reusing one exposure state across the editor should matter beyond the quality of any individual component. Complete baselines, protocol audits, scaling, and sensitivity analyses are in Appendix B.

4.1 PRESERVATION PERSISTS AT MATCHED RELIABILITY

We evaluate 10K strict-online CounterFact edits on GPT-J-6B, replicate on LLaMA-3-8B-Instruct, and test routed support on Qwen3-30B-A3B-Base. Edit t is written before $t{+}1$ is revealed; PACE retains bounded replay/statistic state but never initiates a global revisit of accumulated history. Unless stated otherwise, replay memory has capacity $M = 64$ and selects $K = 8$ retained records. We report reliability (Rel.), generalization (Gen.), locality (Loc.), and forgetting (For.); protocol-matched comparisons use the same backbone, evaluator, prompt format, hardware, and edit stream.

A preservation method can appear strong simply by fitting new edits less aggressively. We therefore vary only PACE's target margin γ on a record- and subject-disjoint development stream, select an UltraEdit-matched reliability point under a pre-specified rule, and freeze it before final evaluation.

Table 1 shows that the preservation advantage survives matching. On GPT-J, PACE-MATCHED reaches nearly identical reliability to UltraEdit while gaining $+1.8$ Gen. and $+1.9$ Loc. and reducing For. by 1.2 points. On LLaMA-3, reliability is identical at 93.7, while PACE-MATCHED gains $+1.8$ Gen. and $+2.0$ Loc. and reduces For. by 1.7 points. The frozen protocol also transfers to LLaMA-3 ZsRE, and the differences persist across the same five stream orders (Appendix B.8). Thus PACE's retention behavior is not explained by weaker fitting or one favorable edit order.

External-memory editing provides a complementary operating regime rather than a protocol-identical baseline. LightEdit is stronger on several short-horizon ZsRE quality metrics, while the measured frontier changes by 100K edits; the full quality, selector, storage, and matched-serving analysis is reported in Appendix B.10. We therefore treat this comparison as a scale-dependent systems trade-off rather than evidence of universal dominance.

4.2 PRE-WRITE EXPOSURE PREDICTS AND MODULATES INTERFERENCE

For prior edit e_i, let m_i^{t-1} and m_i^t be its target-versus-old-object margins immediately before and after incoming edit e_t. We define

$$I(i,t) = \max\{0, m_i^{t-1} - m_i^t\},$$

and call an event *severe* when the prior target-supported margin crosses zero in the harmful direction or decreases by at least one logit unit. Path overlap is a prospective risk signal rather than an exact damage estimator: disjointness from the active update union is sufficient for exact zero direct change to the represented FFN mapping, while overlap need not produce nonzero change.

Across all three backbones, support-based signals predict interference more strongly than prompt semantics. On GPT-J, weighted path overlap reaches $r_s = 0.41$, AUROC 74.6, and R@8 59.1, compared with $0.20/61.9/30.5$ for prompt cosine. The ordering persists on LLaMA-3 and routed Qwen3, as well as on prospective held-out GPT-J and LLaMA-3 streams; on GPT-J, path overlap remains the strongest predictor after controlling lexical, semantic, and restricted-key similarity (Appendix B.6).

Backbone	Prompt cosine r_s/R@8	Path Jaccard r_s/R@8	Weighted Jaccard r_s/R@8
GPT-J-6B	$0.20/30.5$	$0.39/57.6$	$\mathbf{0.41/59.1}$
LLaMA-3-8B	$0.18/29.1$	$0.37/56.1$	$\mathbf{0.39/58.3}$
Qwen3-30B-A3B	$0.17/28.7$	$0.34/53.5$	$\mathbf{0.37/55.9}$

Table 2: In-stream predictive validity after 10K CounterFact edits. R@8 is recall, in percentage points, of the eight retained records with largest measured one-step margin loss within replay memory. Weighted Jaccard is diagnostic; deployed PACE uses binary Jaccard. Full controls are in Appendix B.6.

Condition	Mean ω	Target-gain ratio	$\|\Delta\|_F/\text{Natural}$	Mean $I(i,t)$	Severe rate
Forced-low	0.03	0.99	1.04	0.11	4.2%
Natural	0.18	1.00	1.00	0.25	9.6%
Forced-high	0.55	1.01	0.98	0.51	20.6%

Table 3: Realized-write-matched support intervention on LLaMA-3-8B over $n = 1{,}000$ fixed pairs. Arms share the incoming edit, requested margin, boundary/support budget, and pre-intervention state. Prior-edit damage is never used for pair selection, support construction, or matching. Full construction and paired uncertainty are in Appendix B.7.

Prediction alone does not establish that exposure is consequential. We therefore intervene on $n = 1{,}000$ fixed LLaMA-3 prior–incoming pairs while closely matching realized target gain and update norm.

As overlap increases from forced-low to natural to forced-high, mean interference rises $0.11 \to 0.25 \to 0.51$ and severe-event rate rises $4.2\% \to 9.6\% \to 20.6\%$, while target-gain ratios remain $0.99/1.00/1.01$ and normalized update norms $1.04/1.00/0.98$. Paired 95% bootstrap intervals for both adjacent contrasts exclude zero (Appendix B.7). Thus the lower damage at low overlap is not explained by a weaker realized write. At the retained-record level, realized algebraic mapping drift and missed-collision fraction are also associated with downstream forgetting ($r_s = 0.52$ and 0.49; Appendix B.36), empirically connecting the local support accounting to behavior without extending the algebraic theorem into a global behavioral guarantee.

4.3 Better Exposure Ranking Improves Retention

We next ask whether a better exposure estimate matters operationally. Across nine active replay policies we fix $M = 64$, $K = 8$, the solver, preservation mechanism, edit order, and all other hyperparameters, changing only the ranking signal. A separate no-replay ($K = 0$) condition is shown only as a boundary anchor.

Figure 2 shows a strong rank-level relationship between exposure recall and forgetting. The relationship is not deterministically pointwise—gradient-magnitude ranking slightly exceeds restricted-key cosine in R@8 while producing slightly more forgetting—so we use Kendall's τ rather than imposing an affine model. The full policy results and descriptive fit are reported in Appendix B.3. Under a fixed replay budget, therefore, *which* retained mappings are selected matters beyond the presence of replay itself.

Importantly, the exposure signal is not specific to PACE. Under matched $K = 8$ replay wrappers, replacing random ranking with path-overlap ranking reduces forgetting from 10.8 to 9.3 for AlphaEdit, 6.4 to 5.6 for RLSEdit, and 6.7 to 6.0 for UltraEdit (Table 24). The host editor, replay budget, edit stream, and update rule are otherwise unchanged. Pre-write exposure therefore transfers as a useful control signal beyond the editor in which it is introduced.

4.4 One Exposure State Is Load-Bearing

The transfer experiment establishes that the exposure signal is useful by itself. We now test the stronger architectural claim: does *sharing the same exposure state* across localization, replay, solving, and preservation matter? We keep the extraction family, sparsity budget, edit target, solver,

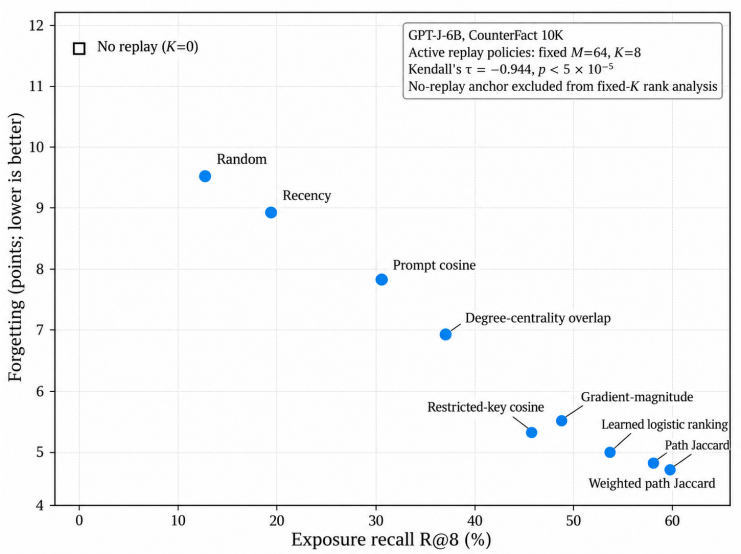

Figure 2: Fixed-budget replay-ranking control on GPT-J-6B after 10K CounterFact edits. Filled points denote $K = 8$ ranking policies under the same $M = 64$ and editing configuration; the no-replay point is excluded from the rank analysis. Across the nine active policies, higher exposure recall is strongly associated with lower forgetting (Kendall's $\tau = -0.944$, $p < 5 \times 10^{-5}$).

Condition	Rel.↑	Gen.↑	Loc.↑	For.↓	Avg.↑	$\overline{J}_{\text{role}}$
SHARED-CANONICAL	91.8	**83.6**	**96.9**	**4.8**	**90.8**	1.00
SHARED-VIEW	91.3	83.1	96.6	5.1	90.3	1.00
TWO-STATE	90.4	82.1	95.7	6.3	89.4	0.75
FOUR-STATE	89.6	81.1	94.9	7.4	88.5	0.59

Table 4: Shared-state control on GPT-J after 10K CounterFact edits. Alternative states use the same extraction family and matched sparsity. $\overline{J}_{\text{role}}$ averages the six role-assignment pairs and counts reused identical states as 1; support-size, boundary-agreement, and distinct-state audits are in Appendix B.5.

prompt, Jacobian, and evaluator fixed, changing only whether the four roles reuse one state or separately extracted states. TWO-STATE shares one state across localization/solving and another across replay/preservation; FOUR-STATE assigns a distinct extracted state to each role.

The pre-specified control separates state identity from extraction quality. SHARED-VIEW remains within 0.433 Avg. points of SHARED-CANONICAL, whereas FOUR-STATE loses 1.8 Avg. points and adds 2.3 For. points relative to SHARED-VIEW; LLaMA-3 reproduces the pattern (-1.6 Avg., $+1.9$ For.; Appendix B.9). Thus, within the matched extraction family, the same exposure state is useful not only for ranking risk but also as a common coordinate system for the write itself. This does not imply that every independently optimized role-specific estimator must underperform; rather, it isolates the additional benefit of coherent state sharing beyond the portability of the exposure signal.

5 CONCLUSION

We identify *pre-write exposure* as a missing variable in continual knowledge editing: before applying a new write, an editor should estimate which prior mappings it may expose. PACE operationalizes this principle with one edit-conditioned support state shared across localization, replay, solving, and preservation; support overlap predicts and experimentally modulates interference, and better exposure ranking reduces forgetting under a fixed replay budget. These effects persist at matched reliability across backbones and stream orders, suggesting a broader principle: continual editors benefit from modeling exposure before updating. The formal guarantees remain local to represented FFN mappings.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, manuscript organization, and feedback on technical presentation. The authors made all final scientific decisions and independently reviewed and verified the theoretical arguments, experimental methodology, reported results, and final manuscript. The authors take full responsibility for all claims and content.

REPRODUCIBILITY STATEMENT

Appendices A and B document the derivations, algorithms, implementation conventions, stream construction, evaluation definitions, hyperparameters, controlled protocols, ablations, and measured results underlying the reported findings.

REFERENCES

Qizhou Chen, Taolin Zhang, Xiaofeng He, Dongyang Li, Chengyu Wang, Longtao Huang, and Hui Xue. Lifelong knowledge editing for LLMs with retrieval-augmented continuous prompt learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 13565–13580, 2024. doi: 10.18653/v1/2024.emnlp-main.751. URL https://aclanthology.org/2024.emnlp-main.751/.

Qizhou Chen, Dakan Wang, Taolin Zhang, Zaoming Yan, Chengsong You, Chengyu Wang, and Xiaofeng He. Uniedit: A unified knowledge editing benchmark for large language models. *Advances in Neural Information Processing Systems*, 38, 2026.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, 2022. doi: 10.18653/v1/2022.acl-long.581. URL https://aclanthology.org/2022.acl-long.581/.

Yanbo Dai, Zhenlan Ji, Zongjie Li, and Shuai Wang. EAMET: Robust massive model editing via embedding alignment optimization. In *International Conference on Learning Representations*, 2026.

Shrestha Datta, Hongfu Liu, and Anshuman Chhabra. Golden layers and where to find them: Improved knowledge editing for large language models via layer gradient analysis. *arXiv preprint arXiv:2602.20207*, 2026.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6491–6506, 2021. doi: 10.18653/v1/2021.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.

Weizhi Fei, Hao Shi, Jing Xu, Jingchen Peng, Jiazheng Li, Jingzhao Zhang, Bo Bai, Wei Han, Zhenyuan Chen, and Xueyan Niu. Scaling knowledge editing in LLMs to 100,000 facts with neural KV database. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/73cadd87a4070ad4d836e9cacac22670-Abstract-Conference.html.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, 2021. doi: 10.18653/v1/2021.emnlp-main.446. URL https://aclanthology.org/2021.emnlp-main.446/.

Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. 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*, pp. 16801–16819, 2024. doi: 10.18653/v1/2024.emnlp-main.934. URL https://aclanthology.org/2024.emnlp-main.934/.

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

Phillip Huang Guo, Aaquib Syed, Abhay Sheshadri, Aidan Ewart, and Gintare Karolina Dziugaite. Mechanistic unlearning: Robust knowledge unlearning and editing via mechanistic localization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 20964–20992. PMLR, 2025. URL https://proceedings.mlr.press/v267/guo25k.html.

Akshat Gupta, Sidharth Baskaran, and Gopala Anumanchipalli. Rebuilding ROME: Resolving model collapse during sequential model editing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21738–21744, 2024a. doi: 10.18653/v1/2024.emnlp-main.1210. URL https://aclanthology.org/2024.emnlp-main.1210/.

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, 2024b. doi: 10.18653/v1/2024.findings-acl.902. URL https://aclanthology.org/2024.findings-acl.902/.

Akshat Gupta, Dev Sajnani, and Gopala Anumanchipalli. A unified framework for model editing. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15403–15418, 2024c. doi: 10.18653/v1/2024.findings-emnlp.903. URL https://aclanthology.org/2024.findings-emnlp.903/.

Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. Aging with GRACE: Lifelong model editing with discrete key-value adaptors. In *Advances in Neural Information Processing Systems*, 2023.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. In *Advances in Neural Information Processing Systems*, 2023.

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

Chenhui Hu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. WilKE: Wise-layer knowledge editor for lifelong knowledge editing. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3476–3503, 2024. doi: 10.18653/v1/2024.findings-acl.207. URL https://aclanthology.org/2024.findings-acl.207/.

Chenhui Hu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. Knowledge in superposition: Unveiling the failures of lifelong knowledge editing for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24086–24094, 2025.

Yuxin Jiang, Yufei Wang, Chuhan Wu, Wanjun Zhong, Xingshan Zeng, Jiahui Gao, Liangyou Li, Xin Jiang, Lifeng Shang, Ruiming Tang, Qun Liu, and Wei Wang. Learning to edit: Aligning LLMs with knowledge editing. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4689–4705, 2024. doi: 10.18653/v1/2024.acl-long.258. URL https://aclanthology.org/2024.acl-long.258/.

Dahyun Jung, Jaewook Lee, and Heuiseok Lim. Towards scalable lifelong knowledge editing with selective knowledge suppression. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3215–3231, 2026. doi: 10.18653/v1/2026.acl-long.145. URL https://aclanthology.org/2026.acl-long.145/.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The Winograd Schema Challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, 2012.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. PMET: Precise model editing in a transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18564–18572, 2024.

Xiaopeng Li, Shangwen Wang, Shasha Li, Shezheng Song, Bin Ji, Jun Ma, and Jie Yu. Rethinking residual distribution in locate-then-edit model editing. In *Advances in Neural Information Processing Systems*, 2025.

Qingyuan Liu, Jia-Chen Gu, Yunzhi Yao, Hong Wang, and Nanyun Peng. Energy-regularized sequential model editing on hyperspheres. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=CHsdtzCip6.

Zhenyan Lu, Daliang Xu, Dongqi Cai, Zexi Li, Wei Liu, Jian Luan, Fangming Liu, Shangguang Wang, and Mengwei Xu. MobiEdit: Resource-efficient knowledge editing for personalized on-device LLMs. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=fb7yTBOV3p.

Jun-Yu Ma, Hong Wang, Hao-Xiang Xu, Zhen-Hua Ling, and Jia-Chen Gu. Perturbation-restrained sequential model editing. In *International Conference on Learning Representations*, volume 2025, pp. 17302–17329, 2025.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, 2022.

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

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

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. Memory-based model editing at scale. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15817–15831. PMLR, 2022b.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534, 2016. doi: 10.18653/v1/P16-1144.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long and Short Papers)*, pp. 4149–4158, 2019. doi: 10.18653/v1/N19-1421.

Lukas Thede, Karsten Roth, Matthias Bethge, Zeynep Akata, and Thomas Hartvigsen. WikiBigEdit: Understanding the limits of lifelong knowledge editing in LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 59326–59354. PMLR, 2025.

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

Shiqi Wang, Qi Wang, Runliang Niu, He Kong, and Yi Chang. Microedit: Neuron-level knowledge disentanglement and localization in lifelong model editing. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 33882–33896, 2025. doi: 10.18653/v1/2025.emnlp-main.1719.

Xinyu Wang, Sicheng Lyu, Yu Gu, Jerry Huang, Peng Lu, Yufei Cui, and Xiao-Wen Chang. Beyond hard writes and rigid preservation: Soft recursive least-squares for lifelong LLM editing. In *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence*, pp. 2382–2390, 2026. doi: 10.24963/ijcai.2026/265.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 94–106, 2017. doi: 10.18653/v1/W17-4413.

Haoyu Xu, Pengxiang Lan, Enneng Yang, Guibing Guo, Jianzhe Zhao, Linying Jiang, and Xingwei Wang. 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)*, pp. 13194–13213, 2025. doi: 10.18653/v1/2025.acl-long.646.

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*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/74b8ccb5ad21871fd436867df19c35ac-Abstract-Conference.html.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10222–10240, 2023. doi: 10.18653/v1/2023.emnlp-main.632. URL https://aclanthology.org/2023.emnlp-main.632/.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, Xiaowei Zhu, Jun Zhou, and Huajun Chen. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*, 2024. URL https://arxiv.org/abs/2401.01286.

Tianyu Zhang, Junfeng Fang, Houcheng Jiang, Baolong Bi, Xiang Wang, and Xiangnan He. Explainable and efficient editing for large language models. In *Proceedings of the ACM on Web Conference 2025*, pp. 1963–1976, 2025. doi: 10.1145/3696410.3714835.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. Can we edit factual knowledge by in-context learning? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4862–4876, 2023. doi: 10.18653/v1/2023.emnlp-main.296. URL https://aclanthology.org/2023.emnlp-main.296/.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 15686–15702, 2023. doi: 10.18653/v1/2023.emnlp-main.971. URL https://aclanthology.org/2023.emnlp-main.971/.

Appendix

A Appendix: Methodological Details

This appendix provides the derivations, implementation conventions, proofs, and scope conditions behind Section 3. The central distinction is between the *predicted* activation path $\mathcal{P}_t$, available before writing, and the active update union $\mathcal{U}_t^\ell$ formed after replay is selected. Proposition 1 is an exact statement about direct parameter-mediated change to a materialized stored FFN mapping under the realized incremental write: zero embedding of the stored key in the active union is sufficient for exactly zero direct change. Incoming-path overlap is therefore a prospective ranking proxy rather than an exact exposure condition or a global non-interference certificate. We use $\mathcal{A}_t$ for active layers, $\mathcal{B}_t$ for boundary layers, K for replay count, and M for replay-memory capacity.

Notation reserved across this appendix. Several symbols are easily conflated and are therefore fixed once here. The residual target split across boundary layers is $r_t^\ell \in \mathbb{R}^{d_{\text{model}}}$, whereas the preservation-subspace rank is $\text{rk}_t^\ell \in \mathbb{N}_0$. The restricted activation key of edit i is $k_i^\ell = (S_i^\ell)^\top a^\ell(x_i)$ and is always a vector. The active-union width is u_t^ℓ, whereas the additive intervention variables of Appendix A.3 are v_t^ℓ. The replay memory is $\mathcal{M}$ with capacity M, whereas an individual materialized layerwise constraint is $\mathcal{D}_i^\ell$. The global replay set is $\mathcal{R}_t$; for layer ℓ, $\mathcal{R}_t^\ell \subseteq \mathcal{R}_t$ contains exactly those selected records that carry a stored constraint for that layer. Finally, Q_t^ℓ is always a coordinate selector and never an orthonormal factor: the orthonormal basis of the preservation subspace is $\Phi_{K,t}^\ell$. We write c_t^ℓ for the number of nonzero preservation-key columns after path-restricted embedding and p_i^{evict} for the eviction priority. Throughout, $\|\cdot\|_2$ denotes the Euclidean norm for vectors and the spectral norm for matrices.

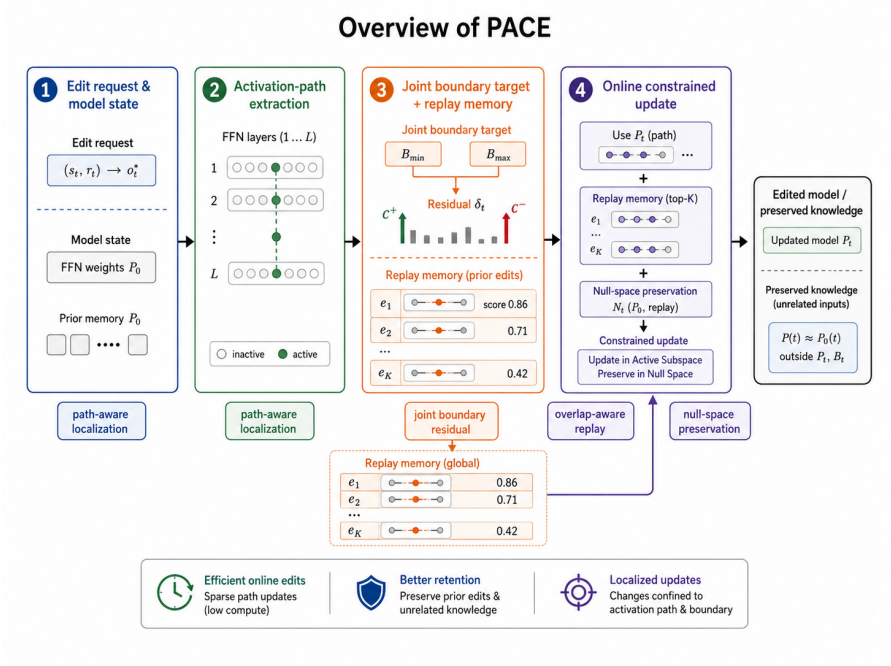

Figure 3: Operational overview of PACE. One activation path $\mathcal{P}_t$ determines the boundary layers $\mathcal{B}_t$, ranks prior edits by $\omega(i,t)$ to form the global replay set $\mathcal{R}_t$, and conditions the layer-valid sparse solve unions $\mathcal{U}_t^\ell$ and preservation keys $K_{0,t}^\ell$. The candidate-logit Jacobian is computed once for the joint boundary target; the preservation projector is the only algorithmic step distinguishing PACE-Full from PACE-Soft. This diagram describes the operational data flow; Figure 1 in the main paper provides the mechanism-level evidence motivating it.

A.1 Design Rationale and Implementation Choices

This subsection records the design choices that turn the support principle into a bounded online editor. Each choice is tied either to a falsifiable control in the experimental appendix or to an explicit scope condition below.

Editable universe. The editable coordinate universe is selected once by activation variance on a held-out calibration pool and then frozen. On GPT-J-6B the reported setting exposes 8192 of $D_{\mathrm{ff}} = 16384$ FFN coordinates per layer. All path fractions use d^{ℓ}_{edit} as denominator, while selector matrices retain D_{ff} rows so that embeddings remain dimensionally conformable. Appendix B.43 reports the width and selection-rule sensitivity.

Boundary placement. Writing every active layer increases the intervention surface. In the measured 10K CounterFact ablation, all-active-layer writing raises reliability by 0.3 points but costs 2.4 locality points and raises forgetting from 4.8 to 6.9; the difference comes from the 43.0% of edits with at least three active layers, because boundary and all-active writing coincide when $|\mathcal{A}_t| \leq 2$ (Table 25 and Table 56). This is why Equation 3 uses the two path boundaries rather than every active layer. Only these boundary layers materialize stored layerwise constraints; interior active layers contribute to the edit-level path and risk ranking but receive no edit-specific replay/preservation constraint unless they are themselves a boundary of another edit.

Incoming overlap as a pre-write proxy. The replay score $\omega(i,t) = J(\mathcal{P}_i, \mathcal{P}_t)$ is intentionally available before the current update union is formed. Proposition 1, however, is expressed in terms of the prior key embedded in the active union $\mathcal{U}^{\ell}_t$. Selected replay supports can enlarge this union, so an unselected record may become exposed through overlap with a replayed support even when its incoming-path overlap is zero. We call such an event a secondary realized-support collision; Theorem 1 counts it as missed whenever the affected record is not layer-valid replay at that step. Sparse support and bounded K cap the width of $\mathcal{U}^{\ell}_t$ through Equation 7, but do not eliminate such events. PACE therefore treats incoming-path overlap as a ranking signal rather than a certificate. This distinction is tested empirically by semantic and restricted-key controls, prospective GPT-J/LLaMA-3 prediction, nine-policy fixed-budget ranking, realized-write-matched intervention, and the algebraic-to-behavioral drift audit.

Target strength and matched reliability. The hinge target of Equation 5 makes γ the single declared fitting-strength knob: it enters the requested margin while the path extractor, replay rule, sparse-support rule, preservation mechanism, and all other hyperparameters are held fixed. Because changing earlier writes can alter later model states, realized future paths need not be identical across a γ sweep; the control claim is that no other mechanism or parameter is retuned. The matched-reliability experiments therefore sweep only γ on the development stream and freeze the selected point before final evaluation (Appendix B.4).

Why bounded state is plausible at long horizons. For fixed replay count K and path fraction, the per-step active-union width is bounded by Equation 7, so the cubic solve depends on u^{ℓ}_t rather than directly on stream length. Bounded replay capacity M also caps the number of retained record descriptors and canonical record contributions; blocks that lose their final retained-record owner are deleted. Empirically, GPT-J $p99$ union width grows from 880 to 1136 between 1K and 100K edits while additional resident state changes from 1.30 to 1.37 GB; the uninterrupted LLaMA-3 stream likewise shows gradual quality degradation through 100K edits (Appendix B.10). These measurements are not extrapolated beyond the evaluated horizon.

A.2 Activation-Path Construction and Fallback Rules

For every FFN layer ℓ, PACE computes a calibration mean μ_ℓ and standard deviation σ_ℓ from subject-token activations on a held-out calibration set. The normalized activation score is

$$s^{\ell}_j(x) = \frac{|a^{\ell}_j(x)| - \mu_\ell}{\sigma_\ell + \varepsilon}, \qquad j \in \mathcal{I}^{\ell}_{\mathrm{edit}}. \tag{A1}$$

PACE first computes the layer activity score,

$$g^\ell(x) = \frac{1}{k_{\text{lay}}} \sum_{j \in \text{TopK}_{k_{\text{lay}}}(s^\ell(x);\, \mathcal{I}^\ell_{\text{edit}})} s^\ell_j(x), \tag{A2}$$

and selects the active-layer set $\mathcal{A}_t = \{\ell : g^\ell(x_t) \geq \tau_{\text{lay}}\}$, where k_{lay} is the number of scores averaged and τ_{lay} is the layer-activity threshold. Both the layer score and coordinate selection below range over $\mathcal{I}^\ell_{\text{edit}}$ only, so layer selection and coordinate selection are defined on the same index set.

Full FFN width versus editable coordinate width. The full FFN activation has dimension D_{ff}, but the sparse path and online least-squares state are defined over the editable coordinate subset exposed to the editor, $\mathcal{I}^\ell_{\text{edit}} \subseteq \{1, \ldots, D_{\text{ff}}\}$, of size $d^\ell_{\text{edit}} = |\mathcal{I}^\ell_{\text{edit}}|$. For GPT-J-6B, the full MLP width is $D_{\text{ff}} = 16384$, while the editable coordinate width used in the reported runs is $d^\ell_{\text{edit}} = 8192$. All support percentages, percentile budgets, and path-size denominators are computed against d^ℓ_{edit}. Selector matrices are indexed by the full width: S^ℓ_t and Q^ℓ_t both have D_{ff} rows, so products such as $(Q^\ell_t)^\top S^\ell_i k^\ell_i$ are conformable.

Within-layer coordinate selection. Within each selected layer, PACE retains coordinates that are simultaneously high-ranking within the layer and salient in absolute normalized score:

$$\mathcal{P}^\ell_t = \underbrace{\text{TopK}_{\lceil \varrho\, d^\ell_{\text{edit}} \rceil}\big(s^\ell(x_t);\, \mathcal{I}^\ell_{\text{edit}}\big)}_{\text{layer-relative budget}} \cap \underbrace{\big\{j \in \mathcal{I}^\ell_{\text{edit}} : s^\ell_j(x_t) \geq \tau_{\text{coord}}\big\}}_{\text{absolute saliency floor}}. \tag{A3}$$

The percentile term caps how much of a layer may be written, while the score floor τ_{coord} prevents a layer that only marginally passes the activity threshold from spending its full budget on weakly activated units. If the intersection contains fewer than $k_{\min}$ coordinates, PACE restores the $k_{\min}$ highest-scoring editable coordinates of that layer. Assuming $k_{\min} \leq \lceil \varrho d^\ell_{\text{edit}} \rceil$, as in all reported configurations, the realized support size $d^\ell_t = |\mathcal{P}^\ell_t|$ obeys

$$k_{\min} \leq d^\ell_t \leq \big\lceil \varrho\, d^\ell_{\text{edit}} \big\rceil. \tag{A4}$$

For GPT-J-6B with $d^\ell_{\text{edit}} = 8192$ and $\varrho = 4\%$, the upper budget is $\lceil 0.04 \times 8192 \rceil = 328$ coordinates per selected active layer: a hard upper bound on d^ℓ_t for every selected layer, not an average target. The floor τ_{coord} makes realized support edit dependent, which explains why layers admitted near the activity threshold can contribute fewer coordinates and why path size grows sublinearly with active-layer count.

Deterministic fallback. If $\mathcal{A}_t$ is empty, PACE selects the single layer with the largest $g^\ell(x_t)$ and retains its $k_{\min}$ highest-scoring editable coordinates. This deterministic fallback prevents an empty update without converting PACE into a dense or fixed-layer editor. In the measured stream logs the fallback is rare—0.3% on CounterFact, 0.6% on ZsRE, 0.2% on MQuAKE, and 0.4% on WikiBigEdit.

Three quantities that must not be conflated. The total path size is

$$|\mathcal{P}_t| = \sum_{\ell \in \mathcal{A}_t} d^\ell_t, \tag{A5}$$

the per-active-layer support fraction is $d^\ell_t / d^\ell_{\text{edit}}$, and the active solve-block width is $u^\ell_t = |\mathcal{U}^\ell_t|$ after union with replayed supports. Because d^ℓ_t is edit dependent, $|\mathcal{P}_t|$ is not recoverable as $|\mathcal{A}_t|$ times a fixed constant. The overlap score is the Jaccard overlap between layer–coordinate pairs, not between layer-agnostic neuron indices.

Consequences for the boundary set. Since $\mathcal{B}_t = \{\min \mathcal{A}_t, \max \mathcal{A}_t\}$, a single active layer yields one boundary write and two or more yield two. The measured distribution of $|\mathcal{A}_t|$ implies that $\ell_{\min}(t) = \ell_{\max}(t)$ occurs for 4.0% of CounterFact edits, boundary and all-active editing coincide for the 57.0% of edits with $|\mathcal{A}_t| \leq 2$, and the non-boundary interior is nonempty only for the remaining 43.0% with $|\mathcal{A}_t| \geq 3$. The interior-only diagnostic is therefore defined only under the explicit execution convention reported with that ablation in Appendix B; no exact stored-constraint guarantee is claimed for interior layers of an edit because such layers are not materialized by the deployed boundary writer.

A.3 JOINT BOUNDARY RESIDUAL DERIVATION

Let $\mathcal{B}_t = \{\ell_1, \ldots, \ell_q\}$ denote the selected boundary layers in increasing order. Because FFN outputs at different layers are causally coupled, PACE does not treat their naturally occurring values as independent states. Instead, it introduces additive local intervention variables,

$$\mathbf{v}_t^{\mathcal{B}} = \begin{bmatrix} v_t^{\ell_1} \\ \vdots \\ v_t^{\ell_q} \end{bmatrix} \in \mathbb{R}^{qd_{\text{model}}}, \tag{A6}$$

where v_t^{ℓ} is injected into the FFN output of layer ℓ and the unedited forward pass corresponds to $\mathbf{v}_t^{\mathcal{B}} = \mathbf{0}$. PACE solves for a joint intervention

$$\mathbf{r}_t^{\mathcal{B}} = \begin{bmatrix} r_t^{\ell_1} \\ \vdots \\ r_t^{\ell_q} \end{bmatrix} \in \mathbb{R}^{qd_{\text{model}}}. \tag{A7}$$

The joint candidate-logit Jacobian is

$$J_t^{\mathcal{B}} = \left. \frac{\partial z_{\mathcal{C}_t}(x_t; \mathbf{v}_t^{\mathcal{B}})}{\partial \mathbf{v}_t^{\mathcal{B}}} \right|_{\mathbf{v}_t^{\mathcal{B}} = \mathbf{0}} \in \mathbb{R}^{|\mathcal{C}_t| \times qd_{\text{model}}}. \tag{A8}$$

It captures simultaneous first-order additive perturbations at all selected boundaries, including downstream logit effects of an intervention at an earlier boundary. Let $e_{c^\star}, e_{c^-} \in \mathbb{R}^{|\mathcal{C}_t|}$ be standard candidate-coordinate one-hot vectors and let $\gamma_r > 0$ denote the ridge damping parameter. With

$$b_t = \frac{\delta_t}{2}(e_{c^\star} - e_{c^-}), \tag{A9}$$

we have $(e_{c^\star} - e_{c^-})^\top b_t = \delta_t$. The ridge solution is

$$\mathbf{r}_t^{\mathcal{B}} = (J_t^{\mathcal{B}})^\top \left(J_t^{\mathcal{B}} (J_t^{\mathcal{B}})^\top + \gamma_r I \right)^{-1} b_t. \tag{A10}$$

The inner matrix is $|\mathcal{C}_t| \times |\mathcal{C}_t|$, so the solve is governed by candidate-set size rather than qd_{model}. This construction allocates one shared margin objective over the joint boundary subspace, minimizes a regularized local mismatch, and does not claim exact realization of the requested margin.

First-order coupling and physical downstream-key drift. The Jacobian in Equation A8 is evaluated at $\mathbf{v}_t^{\mathcal{B}} = \mathbf{0}$ with all additive boundary interventions varied jointly, so it captures first-order cross-boundary effects in that intervention model. A physical lower-boundary weight update can additionally change the representation arriving at a later boundary and therefore change the actual restricted key relative to the pre-write key used by the solver. That effect is distinct from the Taylor remainder of the additive intervention map; we do not subsume it into the bound below. Remark 1 bounds only the nonlinear remainder of the additive proxy, while Appendix B.37 measures realized fidelity after the actual weight write.

Remark 1 (Second-order remainder of the joint intervention target). Let $F(\mathbf{v}) = z_{\mathcal{C}_t}(x_t; \mathbf{v})$ denote the candidate-logit map under the stacked additive boundary intervention $\mathbf{v} = \mathbf{v}_t^{\mathcal{B}}$. Assume F is twice continuously differentiable on the line segment from $\mathbf{0}$ to $\mathbf{r}_t^{\mathcal{B}}$ and that the Hessian operator norm is bounded there by H_t. Taylor's theorem gives

$$\left\| F(\mathbf{r}_t^{\mathcal{B}}) - F(\mathbf{0}) - J_t^{\mathcal{B}} \mathbf{r}_t^{\mathcal{B}} \right\|_2 \leq \frac{H_t}{2} \left\| \mathbf{r}_t^{\mathcal{B}} \right\|_2^2. \tag{A11}$$

Thus Equation A8 captures all first-order cross-layer effects of the additive interventions, while the omitted intervention-space term is second order in the joint intervention norm. For two boundaries, the mixed-Hessian component is $O(\|r_t^{\ell_{\min}}\|_2 \|r_t^{\ell_{\max}}\|_2)$ and vanishes if either boundary intervention is zero; within-boundary curvature can remain even for a single-boundary edit. We therefore do not claim that the nonlinear remainder vanishes when $q = 1$, nor that this Taylor bound controls downstream-key drift under the realized weight update. Appendix B.37 measures realized first-order fidelity directly.

A.4 Coordinate-Union Embedding and Record Management

Replay is selected once at the edit level, but the online solve is layerwise. We therefore distinguish the global replay set $\mathcal{R}_t$ from

$$\mathcal{R}_t^\ell = \{i \in \mathcal{R}_t : \mathcal{D}_i^\ell \text{ is materialized}\} = \{i \in \mathcal{R}_t : \ell \in \mathcal{B}_i\},$$

the selected records that carry a stored constraint at layer ℓ. Only $\mathcal{R}_t^\ell$ enters the layer-ℓ union, replay reinsertion, and history-key preservation pool.

For each materialized layerwise constraint, an edit record stores

$$\mathcal{D}_i^\ell = \big(\underbrace{\mathcal{P}_i^\ell,\ S_i^\ell,\ k_i^\ell,\ \rho_i^\ell}_{\text{constraint}},\quad \underbrace{w_i^\ell,\ \tau_i^\ell,\ \chi_i^\ell}_{\text{canonical contribution}}\quad ,\underbrace{\nu_i,\ c_i}_{\text{eviction}}\big), \tag{A12}$$

where $k_i^\ell = (S_i^\ell)^\top a^\ell(x_i)$ is the stored restricted key, w_i^ℓ is its instantaneous materialization weight, τ_i^ℓ is the step of that materialization, and χ_i^ℓ identifies the unique canonical resident block that owns this record's sufficient-statistic contribution. The logical invariant is that every represented layerwise record contribution has exactly one canonical owner block. An implementation may physically coalesce blocks that share the same coordinate set and timestamp or may realize aging through an equivalent lazy scale, but it must preserve this unique-ownership invariant; an assembled union view is never itself committed as a second copy of surviving historical statistics.

A resident block b at layer ℓ has coordinate set $\mathcal{U}_b^\ell$, timestamp τ_b^ℓ, owner set $\mathcal{O}_b^\ell$, and native sufficient statistics

$$A_b^\ell = \sum_{i \in \mathcal{O}_b^\ell} w_i^\ell\, \kappa_{i,b}^\ell (\kappa_{i,b}^\ell)^\top, \qquad B_b^\ell = \sum_{i \in \mathcal{O}_b^\ell} w_i^\ell\, \rho_i^\ell (\kappa_{i,b}^\ell)^\top, \tag{A13}$$

where $\kappa_{i,b}^\ell = (Q_b^\ell)^\top S_i^\ell k_i^\ell$ and every owner in a block has $\tau_i^\ell = \tau_b^\ell$. Blocks created by the same stream step and coordinate union may be coalesced additively; contributions created at different logical timestamps are not combined unless an equivalent lazy-scaling representation preserves their individual ages.

Replay metadata are updated at two scopes:

$$\begin{aligned} &w_i^\ell \leftarrow 1 + \eta\, \omega(i,t), \quad \tau_i^\ell \leftarrow t, && i \in \mathcal{R}_t^\ell \quad \text{after old-contribution removal,} \\ &\nu_i \leftarrow t, && c_i \leftarrow c_i + \omega(i,t), \quad i \in \mathcal{R}_t \quad \text{once per stream step.} \end{aligned} \tag{A14}$$

Thus a record selected at several materialized layers can be rematerialized at each such layer, while its global replay utility is incremented only once.

For a new union $\mathcal{U}_t^\ell$, let $Q_t^\ell \in \{0,1\}^{D_{\text{ff}} \times u_t^\ell}$ select its coordinates. The current-union embedding of a record key is

$$\kappa_{i,t}^\ell = (Q_t^\ell)^\top S_i^\ell k_i^\ell \in \mathbb{R}^{u_t^\ell}. \tag{A15}$$

For each resident block b, let $T_{b\to t}^\ell$ map the coordinates shared with $\mathcal{U}_t^\ell$ into the current ordering. Its aged embedded view at step t is

$$\widetilde{A}_{b\to t}^\ell = \beta^{\,t-\tau_b^\ell} T_{b\to t}^\ell A_b^\ell (T_{b\to t}^\ell)^\top, \qquad \widetilde{B}_{b\to t}^\ell = \beta^{\,t-\tau_b^\ell} B_b^\ell (T_{b\to t}^\ell)^\top. \tag{A16}$$

This per-block age factor is essential because resident blocks can be materialized at different stream steps. A layer-level timestamp alone would be insufficient unless the implementation carries an equivalent per-block lazy scale.

Exact replay removal in the canonical source block. Before assembling the step-t state, every $i \in \mathcal{R}_t^\ell$ is removed from its unique owner block $b = \chi_i^\ell$ in that block's native coordinates:

$$\begin{aligned} A_b^{\ell,-} &= A_b^\ell - \sum_{\substack{i \in \mathcal{R}_t^\ell \\ \chi_i^\ell = b}} w_i^\ell\, \kappa_{i,b}^\ell (\kappa_{i,b}^\ell)^\top, \\ B_b^{\ell,-} &= B_b^\ell - \sum_{\substack{i \in \mathcal{R}_t^\ell \\ \chi_i^\ell = b}} w_i^\ell\, \rho_i^\ell (\kappa_{i,b}^\ell)^\top. \end{aligned} \tag{A17}$$

Because removal occurs in the same unaged native representation in which the contribution was stored, it is exact and does not require reconstructing a layer-global decay interval. The surviving contents of block b retain timestamp τ_b^ℓ and are aged only when embedded through Equation A16. Empty source blocks are deleted.

Insertion block and canonical commit. After removal, the current record and selected replay records are inserted at age zero into the current union. Define $\mathcal{I}_t^\ell = \{t\} \cup \mathcal{R}_t^\ell$ and

$$A_{\mathrm{ins},t}^\ell = \sum_{i \in \mathcal{I}_t^\ell} \bar{w}_{i,t}^\ell \kappa_{i,t}^\ell (\kappa_{i,t}^\ell)^\top, \qquad B_{\mathrm{ins},t}^\ell = \sum_{i \in \mathcal{I}_t^\ell} \bar{w}_{i,t}^\ell \rho_i^\ell (\kappa_{i,t}^\ell)^\top. \tag{A18}$$

The persistent cache commits *only* this insertion contribution as the new canonical block on $\mathcal{U}_t^\ell$ with timestamp t; it does not commit the full assembled solve statistics, because doing so would duplicate surviving contributions that remain owned by older blocks. Each $i \in \mathcal{I}_t^\ell$ updates χ_i^ℓ to this new block and τ_i^ℓ to t. This ownership rule is the cache-commit semantics omitted by an informal "assemble and solve" description and prevents double counting when future unions overlap several resident blocks.

Eviction policy. When the number of stored edit records exceeds capacity M, PACE evicts the record minimizing

$$p_i^{\mathrm{evict}} = \beta^{\,t-\nu_i}(1 + c_i). \tag{A19}$$

Before deleting the record, its unique canonical contribution is subtracted from every materialized layerwise owner block using Equation A17; the record metadata are then removed, and any block whose owner set becomes empty is deleted. Thus eviction removes a constraint from the represented objective rather than merely deleting metadata. Ties are broken by evicting the oldest record.

A.5 Weighted Online Least Squares

For the contributions inserted at step t, define

$$\bar{w}_{i,t}^\ell = \begin{cases} 1, & i = t, \\ 1 + \eta\, \omega(i,t), & i \in \mathcal{R}_t^\ell. \end{cases} \tag{A20}$$

After insertion, the stored layerwise metadata satisfy $w_i^\ell \leftarrow \bar{w}_{i,t}^\ell$ and $\tau_i^\ell \leftarrow t$. A represented record owned by block $b = \chi_i^\ell$ has effective step-t weight

$$\widetilde{w}_{i,t}^\ell = \beta^{\,t-\tau_b^\ell} w_i^\ell = \beta^{\,t-\tau_i^\ell} w_i^\ell.$$

For the current active coordinate block, the represented ridge objective is

$$\min_{\Delta} \sum_{i \in \mathcal{H}_t^\ell} \widetilde{w}_{i,t}^\ell \left\| \Delta \kappa_{i,t}^\ell - \rho_i^\ell \right\|_2^2 + \lambda \|\Delta\|_F^2, \tag{A21}$$

where $\mathcal{H}_t^\ell$ contains represented constraints with nonzero embedding into the current union after replay removal and reinsertion.

Let $\mathcal{B}_{t-1}^\ell$ denote the resident block set after the source-block removals of Equation A17. The exact assembled aged statistics used by the current solve are

$$\begin{aligned} A_t^\ell &= \sum_{b \in \mathcal{B}_{t-1}^\ell} \beta^{\,t-\tau_b^\ell} T_{b\to t}^\ell A_b^{\ell,-} (T_{b\to t}^\ell)^\top + A_{\mathrm{ins},t}^\ell, \\ B_t^\ell &= \sum_{b \in \mathcal{B}_{t-1}^\ell} \beta^{\,t-\tau_b^\ell} B_b^{\ell,-} (T_{b\to t}^\ell)^\top + B_{\mathrm{ins},t}^\ell. \end{aligned} \tag{A22}$$

Equivalently,

$$A_t^\ell = \sum_{i \in \mathcal{H}_t^\ell} \widetilde{w}_{i,t}^\ell \kappa_{i,t}^\ell (\kappa_{i,t}^\ell)^\top, \qquad B_t^\ell = \sum_{i \in \mathcal{H}_t^\ell} \widetilde{w}_{i,t}^\ell \rho_i^\ell (\kappa_{i,t}^\ell)^\top.$$

Thus Equation A22 is the exact sufficient-statistic realization of Equation A21, conditional on the represented retained records. The unconstrained solution is $\Delta_t^\ell = B_t^\ell (A_t^\ell + \lambda I)^{-1}$, written $\Delta_{t,\mathcal{U}}^\ell$

in the main text to emphasize restriction to the active union. The full assembled A_t^ℓ, B_t^ℓ are an ephemeral solve view; persistent storage commits only the insertion block of Equation A18, leaving surviving historical contributions uniquely owned by their source blocks. This distinction is what makes block embedding lossless without double counting.

Remark 2 (Lossless block re-embedding and unique ownership). Each represented layerwise record contribution has one canonical owner block. Embedding a source block through $T_{b\to t}^\ell$ creates only an aged current-union view and never creates a second persistent copy. Replay or eviction first removes the selected record from its owner block; current and replayed constraints are then committed exactly once through the new insertion block. Therefore surviving historical statistics and newly inserted statistics have disjoint logical ownership even when their coordinate sets overlap. Conditional on the retained records, the assembled statistics equal the sufficient-statistic sums of Equation A21 exactly on the current union. Information is discarded only through explicit bounded-memory eviction.

Lemma 1 (Well-posedness of the soft-preservation update). For a fixed boundary layer, active coordinate union, and represented retained constraint set, the unconstrained solution $\Delta_t^\ell = B_t^\ell(A_t^\ell + \lambda I)^{-1}$ uniquely minimizes Equation A21 whenever $\lambda > 0$.

Proof. Expanding Equation A21 and differentiating with respect to Δ gives $2\Delta A_t^\ell - 2B_t^\ell + 2\lambda\Delta = 0$. Hence $\Delta(A_t^\ell + \lambda I) = B_t^\ell$. Since A_t^ℓ is positive semidefinite and $\lambda > 0$, $A_t^\ell + \lambda I$ is positive definite, so the stationary point is the unique global minimizer by strict convexity. □

A.6 Why Direct Parameter Interference Is Support Mediated

The support claim is scoped to the FFN correction PACE actually writes. For a fixed layer and stream step, direct change to an earlier materialized mapping can arise only through coordinates selected by the current active update union. This is an exact algebraic statement about the support-restricted correction, not a statement that every downstream Transformer interaction or prompt-level effect is support mediated.

Proposition 1 (Support-mediated direct mapping change). Fix layer ℓ and steps $i < t$. Suppose the stored layerwise behavior of edit e_i is represented by the materialized correction mapping $E^\ell S_i^\ell k_i^\ell$. Let $\widehat{G}_{t,\mathcal{U}}^\ell$ be the step-t increment written on the active union and define the embedded prior key $\kappa_{i,t}^\ell = (Q_t^\ell)^\top S_i^\ell k_i^\ell$. Then

$$\left\|(E_t^\ell - E_{t-1}^\ell)S_i^\ell k_i^\ell\right\|_2 \le \left\|\widehat{G}_{t,\mathcal{U}}^\ell\right\|_2 \left\|\kappa_{i,t}^\ell\right\|_2. \tag{A23}$$

Thus prior-key energy exposed inside the active update union controls this direct-change bound. In particular, disjoint support is a sufficient exact zero-change condition: if $\mathcal{P}_i^\ell \cap \mathcal{U}_t^\ell = \emptyset$, then $\kappa_{i,t}^\ell = 0$ and $E_t^\ell S_i^\ell k_i^\ell = E_{t-1}^\ell S_i^\ell k_i^\ell$. The converse is not required: overlap may still yield zero direct change through the realized increment or preservation projection.

Proof. By Equation 9, $E_t^\ell - E_{t-1}^\ell = \widehat{G}_{t,\mathcal{U}}^\ell (Q_t^\ell)^\top$. Therefore $(E_t^\ell - E_{t-1}^\ell)S_i^\ell k_i^\ell = \widehat{G}_{t,\mathcal{U}}^\ell \kappa_{i,t}^\ell$, and submultiplicativity gives Equation A23. Because $S_i^\ell k_i^\ell$ is supported on $\mathcal{P}_i^\ell$, its embedding into $\mathcal{U}_t^\ell$ is zero when the supports are disjoint. □

Three consequences motivate the experimental design. *First*, the exact condition concerns $\mathcal{U}_t^\ell$, not merely $\mathcal{P}_t$. A zero incoming-path score does not certify zero direct risk because replay supports can enlarge the union; an unselected third record that overlaps such a replay support is a missed realized collision. *Second*, binary overlap cannot determine damage magnitude even when support is the correct mediator: Equation A23 also depends on embedded-key energy and on the magnitude and direction of the current increment. *Third*, a magnitude-aware overlap on the same path support is a natural diagnostic refinement; the weighted-Jaccard control yields a modest predictive gain but essentially unchanged rounded downstream performance, so binary Jaccard remains the deployed score.

Two scope conditions are essential. The proposition concerns one materialized layerwise FFN mapping and does not bound arbitrary downstream nonlinear behavior. Moreover, because deployed PACE materializes constraints only at boundary layers, the proposition supplies no stored-edit guarantee for an interior active layer at which that edit never wrote or stored a constraint.

A.7 Cumulative Interference and the Cost of a Missed Collision

Proposition 1 is one-step. Over a stream, PACE-Full uses replay selection in two coupled ways: a replayed record conditions the unconstrained ridge objective through its stored target, and its nonzero layerwise history key is included in the preservation pool. The latter gives the exact accounting needed for a cumulative bound: a selected colliding mapping incurs no additional direct change at that step, whereas an unselected realized collision may drift.

Definition 1 (Layer-wise collision set). Fix a materialized layerwise constraint $\mathcal{D}_i^\ell$ from edit e_i and horizon $T > i$. Its realized collision set is

$$\mathcal{C}_i^\ell(T) = \{t \in (i, T] : \ell \in \mathcal{B}_t,\ \mathcal{P}_i^\ell \cap \mathcal{U}_t^\ell \neq \emptyset\}.$$

It partitions into protected collisions

$$\mathcal{C}_{i,\mathrm{rep}}^\ell(T) = \{t \in \mathcal{C}_i^\ell(T) : i \in \mathcal{R}_t^\ell\}$$

and missed collisions $\mathcal{C}_{i,\mathrm{miss}}^\ell(T) = \mathcal{C}_i^\ell(T) \setminus \mathcal{C}_{i,\mathrm{rep}}^\ell(T)$. For a fixed realized collision sequence, define

$$\mu_i^\ell(T) = \frac{|\mathcal{C}_{i,\mathrm{miss}}^\ell(T)|}{|\mathcal{C}_i^\ell(T)|},$$

with $\mu_i^\ell(T) = 0$ when the denominator is zero.

Assumption 1 (Finite-horizon increment and key bounds). On the evaluated horizon, suppose constants Γ and $\kappa_\star$ satisfy

$$\|\widehat{G}_{t,\mathcal{U}}^\ell\|_2 \le \Gamma, \qquad \|\kappa_{i,t}^\ell\|_2 \le \kappa_\star$$

for realized collisions, and let

$$\|E_i^\ell S_i^\ell k_i^\ell - \rho_i^\ell\|_2 \le \epsilon_0$$

bound the stored-mapping residual immediately after edit i is inserted. These are finite-horizon boundedness assumptions; ridge regularization does not by itself imply universal constants independent of targets and activations.

Theorem 1 (Missed realized-support collisions carry cumulative direct drift)**.** *Under Assumption 1, for a prior constraint that remains represented over the horizon of interest, and provided each nonzero history key of a layer-valid replayed record is included in $K_{0,t}^\ell$ as specified in Appendix A.10,*

$$\left\|E_T^\ell S_i^\ell k_i^\ell - \rho_i^\ell\right\|_2 \le \epsilon_0 + |\mathcal{C}_{i,\mathrm{miss}}^\ell(T)|\,\Gamma\kappa_\star = \epsilon_0 + \mu_i^\ell(T)|\mathcal{C}_i^\ell(T)|\Gamma\kappa_\star. \tag{A24}$$

Steps outside $\mathcal{C}_i^\ell(T)$ add zero direct mapping change; replayed collisions whose history key is preserved also add zero; only missed collisions enter the additive drift term. The theorem does not extend the guarantee past deliberate eviction of record i.

Proof. Let $d_t = \|E_t^\ell S_i^\ell k_i^\ell - \rho_i^\ell\|_2$. At insertion, $d_i \le \epsilon_0$. If $t \notin \mathcal{C}_i^\ell(T)$, Proposition 1 gives $d_t = d_{t-1}$. If $t \in \mathcal{C}_{i,\mathrm{rep}}^\ell(T)$, the embedded history key lies in $\mathrm{col}(K_{0,t}^\ell)$ and $\widehat{G}_{t,\mathcal{U}}^\ell K_{0,t}^\ell = 0$, so the step adds zero direct change and again $d_t = d_{t-1}$. Importantly, this equality means the current replayed step freezes additional direct drift; it does not imply that d_{t-1} is repaired. If $t \in \mathcal{C}_{i,\mathrm{miss}}^\ell(T)$, then by the triangle inequality, Equation A23, and Assumption 1,

$$d_t \le d_{t-1} + \|(E_t^\ell - E_{t-1}^\ell)S_i^\ell k_i^\ell\|_2 \le d_{t-1} + \Gamma\kappa_\star.$$

Induction yields Equation A24. □

Corollary 1 (Conditional affine dependence of the drift bound). Condition on a fixed realized collision sequence $\mathcal{C}_i^\ell(T)$ and fixed finite-horizon scale constants. The right-hand side of Equation A24 is affine in $\mu_i^\ell(T)$, with slope $|\mathcal{C}_i^\ell(T)|\Gamma\kappa_\star$ and intercept ϵ_0.

The corollary does not assert that changing replay ranking leaves the realized collision sequence fixed: replay changes $\mathcal{U}_t^\ell$ and can therefore create or remove collisions, including secondary collisions induced by replay-expanded unions. Nor does an affine upper bound on a represented FFN mapping imply affine, deterministic, or exclusively support-mediated behavioral forgetting. The controlled ranking experiment therefore remains empirical. Table 77 separately tests the downstream connection: missed-collision fraction and realized algebraic mapping drift are positively associated with behavioral forgetting, but the relationship is imperfect and is not used to extend Theorem 1 beyond its local algebraic scope. R@K remains operationally useful because it measures the portion of a risk ranking that bounded replay can act on; the theorem's role is only to separate unexposed steps, protected collisions, and missed realized-support collisions for represented materialized mappings.

A.8 Conflict Resolution on a Shared Path

Lemma 1 guarantees a unique ridge solution but does not say which value is realized when retained constraints are mutually incompatible. The idealized coincident-key case admits a closed form for the *unconstrained ridge solution before the PACE-Full preservation projection*.

Proposition 2 (Bounded resolution of a coincident-key conflict). Fix layer ℓ and step t. In the idealized two-constraint case, suppose the only nonzero data terms in the active-block ridge objective are two represented constraints $i, j \in \mathcal{H}_t^\ell$ whose embedded keys coincide, $\kappa_{i,t}^\ell = \kappa_{j,t}^\ell =: \kappa$, with $s = \|\kappa\|_2^2 > 0$, whose targets differ, and $\lambda > 0$. Let $\widetilde{w}_{i,t}^\ell, \widetilde{w}_{j,t}^\ell > 0$ denote their effective age-decayed weights. Then

$$\Delta_t^\ell \kappa = \frac{\widetilde{w}_{i,t}^\ell \rho_i^\ell + \widetilde{w}_{j,t}^\ell \rho_j^\ell}{\widetilde{w}_{i,t}^\ell + \widetilde{w}_{j,t}^\ell + \lambda/s}. \tag{A25}$$

Hence the unconstrained value is a ridge-shrunk weighted compromise. In general,

$$\frac{\widetilde{w}_{i,t}^\ell}{\widetilde{w}_{j,t}^\ell} = \beta^{\tau_j^\ell - \tau_i^\ell} \frac{w_i^\ell}{w_j^\ell}.$$

Proof. Under the stated two-constraint assumption,

$$A_t^\ell = (\widetilde{w}_{i,t}^\ell + \widetilde{w}_{j,t}^\ell)\kappa\kappa^\top, \qquad B_t^\ell = (\widetilde{w}_{i,t}^\ell \rho_i^\ell + \widetilde{w}_{j,t}^\ell \rho_j^\ell)\kappa^\top.$$

Let $c = \widetilde{w}_{i,t}^\ell + \widetilde{w}_{j,t}^\ell$. By Sherman–Morrison,

$$(\lambda I + c\kappa\kappa^\top)^{-1}\kappa = \frac{\kappa}{\lambda + cs}.$$

Substituting into $\Delta_t^\ell = B_t^\ell (A_t^\ell + \lambda I)^{-1}$ gives Equation A25. □

As $\lambda \to 0$, Equation A25 approaches the weighted mean of the two targets; the ridge term shrinks that compromise toward zero. This proposition characterizes only the unconstrained/Soft solve. Under PACE-Full, if a conflicting replay-history key lies in the preservation span, the final projected increment is exactly zero on that direction, so the post-projection result need not realize the compromise in Equation A25; in the exact coincident-key case, preservation of the old history key can prevent the incoming write from changing that shared direction at all. Thus PACE-Full uses replay history primarily as a current-step invariance constraint on preserved directions, not as a guarantee of repairing an already drifted value. Resolving truly incompatible values requires arbitration or versioning beyond a single-writer parametric mapping. Appendix B.15 measures a high-overlap empirical proxy for this regime and is not a direct test of exact key coincidence.

A.9 Base-Relative Targets and Incremental Writes

The target for the current record is defined in base-relative coordinates. Let $E_{t-1,\mathcal{P}}^\ell = E_{t-1}^\ell S_t^\ell$. Then

$$\rho_t^\ell = E_{t-1,\mathcal{P}}^\ell k_t^\ell + r_t^\ell, \tag{A26}$$

where r_t^ℓ is the boundary-layer residual of Equation A7. A layerwise constraint $\mathcal{D}_t^\ell$ is materialized only for $\ell \in \mathcal{B}_t$; interior active layers remain part of the edit-level path used for overlap ranking but are not assigned a replay target because PACE does not write them. If the new cumulative correction satisfies $E_t^\ell S_t^\ell k_t^\ell \approx \rho_t^\ell$, then

$$E_t^\ell S_t^\ell k_t^\ell \approx E_{t-1}^\ell S_t^\ell k_t^\ell + r_t^\ell, \tag{A27}$$

which applies the desired residual once rather than re-adding the full edited output at every step.

A.10 Null-Space Preservation

The preservation-key matrix $K_{0,t}^\ell \in \mathbb{R}^{u_t^\ell \times n_0}$, with $n_0 = n_{\text{base}} + |\mathcal{R}_t^\ell|$, is assembled from path-restricted neutral calibration keys and nonzero history keys of layer-valid replayed records. Each neutral key originates from a calibration prompt whose own path is extracted by the same support rule. The canonical projector is

$$P_{0,t}^\ell = I - K_{0,t}^\ell \left((K_{0,t}^\ell)^\top K_{0,t}^\ell\right)^\dagger (K_{0,t}^\ell)^\top, \tag{A28}$$

which is the orthogonal projector onto $\mathrm{col}(K_{0,t}^\ell)^\perp$.

Implementation: compact rank-revealing factorization. After removing zero preservation columns, let $\widetilde{K}_{0,t}^{\ell} \in \mathbb{R}^{u_t^{\ell} \times c_t^{\ell}}$ and define

$$\mathrm{rk}_t^{\ell} := \mathrm{rank}(\widetilde{K}_{0,t}^{\ell}) \in \mathbb{N}_0, \qquad \mathrm{rk}_t^{\ell} \le \min\{u_t^{\ell}, c_t^{\ell}\} \le c_t^{\ell} \le n_0.$$

PACE computes a compact rank-revealing factorization, via pivoted thin QR or SVD,

$$\widetilde{K}_{0,t}^{\ell} = \Phi_{K,t}^{\ell} R_{K,t}^{\ell}, \qquad (\Phi_{K,t}^{\ell})^{\top} \Phi_{K,t}^{\ell} = I, \qquad \Phi_{K,t}^{\ell} \in \mathbb{R}^{u_t^{\ell} \times \mathrm{rk}_t^{\ell}}, \qquad R_{K,t}^{\ell} \in \mathbb{R}^{\mathrm{rk}_t^{\ell} \times c_t^{\ell}}, \quad \text{(A29)}$$

where any QR column permutation is absorbed into $R_{K,t}^{\ell}$. Thus $\mathrm{col}(\Phi_{K,t}^{\ell}) = \mathrm{col}(\widetilde{K}_{0,t}^{\ell})$ and

$$P_{0,t}^{\ell} = I - \Phi_{K,t}^{\ell} (\Phi_{K,t}^{\ell})^{\top}. \quad \text{(A30)}$$

If $c_t^{\ell} = 0$, we use the empty basis and $P_{0,t}^{\ell} = I$. For a general $u_t^{\ell} \times c_t^{\ell}$ preservation matrix, the rank-revealing factorization costs

$$O\big(u_t^{\ell} c_t^{\ell} \min\{u_t^{\ell}, c_t^{\ell}\}\big).$$

In the measured PACE regime, $u_t^{\ell} \ge c_t^{\ell}$, reducing this term to $O(u_t^{\ell}(c_t^{\ell})^2)$. Equation A30 is mathematically equivalent to Equation A28; no preservation direction is truncated.

Let $G_t^{\ell} = \Delta_t^{\ell} - E_{t-1}^{\ell} Q_t^{\ell}$. PACE applies

$$\widehat{G}_t^{\ell} = G_t^{\ell} P_{0,t}^{\ell} = G_t^{\ell} - (G_t^{\ell} \Phi_{K,t}^{\ell})(\Phi_{K,t}^{\ell})^{\top}$$

at cost $O(d_{\mathrm{model}} u_t^{\ell} \mathrm{rk}_t^{\ell})$. Since $P_{0,t}^{\ell} K_{0,t}^{\ell} = 0$, the projected increment satisfies

$$\widehat{G}_t^{\ell} K_{0,t}^{\ell} = 0,$$

and therefore

$$\left(E_{t-1}^{\ell} Q_t^{\ell} + \widehat{G}_t^{\ell}\right) K_{0,t}^{\ell} = E_{t-1}^{\ell} Q_t^{\ell} K_{0,t}^{\ell}. \quad \text{(A31)}$$

For a replayed history key included in $K_{0,t}^{\ell}$, Equation A31 means that the current step introduces no additional direct mapping change on that key. It does not imply that direct drift accumulated before the record was replayed is repaired. PACE-Soft, obtained by setting $P_{0,t}^{\ell} = I$, retains the unconstrained target-steering update but loses this exact current-step invariance.

Remark 3 (Objective-level cost of post-solve projection). Let $\mathcal{L}_t^{\ell}(\Delta)$ denote the weighted ridge objective in Equation A21, let $\Delta^{\star} = B_t^{\ell}(A_t^{\ell} + \lambda I)^{-1}$ be its unconstrained minimizer, let $E_{\mathrm{prev}} := E_{t-1}^{\ell} Q_t^{\ell}$, and define $G^{\star} := \Delta^{\star} - E_{\mathrm{prev}}$. Exact incremental preservation defines the affine feasible set

$$\mathcal{F}_t^{\ell} = \left\{\Delta : (\Delta - E_{\mathrm{prev}}) K_{0,t}^{\ell} = 0\right\}.$$

PACE-Full uses the feasible post-solve projection

$$\widehat{\Delta} = E_{\mathrm{prev}} + G^{\star} P_{0,t}^{\ell}.$$

Let Δ^{c} be the exact minimizer of $\mathcal{L}_t^{\ell}$ over $\mathcal{F}_t^{\ell}$ and define

$$\delta_t^{\ell} = \Delta^{\star} - \widehat{\Delta} = G^{\star}(I - P_{0,t}^{\ell}) = (G^{\star} \Phi_{K,t}^{\ell})(\Phi_{K,t}^{\ell})^{\top}.$$

Because $\mathcal{L}_t^{\ell}$ is quadratic with curvature $A_t^{\ell} + \lambda I$ and $\Delta^{\star}$ is stationary,

$$\mathcal{L}_t^{\ell}(\widehat{\Delta}) - \mathcal{L}_t^{\ell}(\Delta^{\star}) = \mathrm{tr}\left(\delta_t^{\ell}(A_t^{\ell} + \lambda I)(\delta_t^{\ell})^{\top}\right).$$

Since Δ^{c} minimizes the same objective over $\mathcal{F}_t^{\ell}$ and $\widehat{\Delta} \in \mathcal{F}_t^{\ell}$,

$$0 \le \mathcal{L}_t^{\ell}(\widehat{\Delta}) - \mathcal{L}_t^{\ell}(\Delta^{\mathrm{c}}) \le \mathcal{L}_t^{\ell}(\widehat{\Delta}) - \mathcal{L}_t^{\ell}(\Delta^{\star}) \le \|A_t^{\ell} + \lambda I\|_2 \|\delta_t^{\ell}\|_F^2, \qquad \mathrm{rank}(\delta_t^{\ell}) \le \mathrm{rk}_t^{\ell}.$$

Thus post-solve projection is exactly feasible for the preservation constraint but is not, in general, the exact affine-constrained least-squares optimizer; the displayed quadratic term upper-bounds its objective gap relative to that optimum.

Quantitative scope of the guarantee. Equation A31 gives exact invariance only on the selected preservation-key subspace. For a neutral calibration prompt $x_0^{(p)}$, let

$$a_p^\ell = (Q_t^\ell)^\top a^\ell(x_0^{(p)}) \in \mathbb{R}^{u_t^\ell},$$

and let $\mathcal{S}_{0,p,t}^\ell \subseteq \mathcal{U}_t^\ell$ denote the portion of that prompt's selected path that lies inside the current update union. Let $\Pi_{p,t}^\ell$ be the corresponding coordinate projector in the ordering of $\mathcal{U}_t^\ell$, with $\Pi_{p,t}^{\ell,\perp} = I - \Pi_{p,t}^\ell$. If the nonzero selected key $\Pi_{p,t}^\ell a_p^\ell$ is included as a column of $K_{0,t}^\ell$, then

$$\left\|\widehat{G}_t^\ell a_p^\ell\right\|_2 = \left\|\widehat{G}_t^\ell \Pi_{p,t}^{\ell,\perp} a_p^\ell\right\|_2 \le \left\|\widehat{G}_t^\ell\right\|_2 \left\|\Pi_{p,t}^{\ell,\perp} a_p^\ell\right\|_2. \tag{A32}$$

Moreover,

$$\left\|\Pi_{p,t}^{\ell,\perp} a_p^\ell\right\|_2 \le \sqrt{u_t^\ell - |\mathcal{S}_{0,p,t}^\ell|} \max_{j \in \mathcal{U}_t^\ell \setminus \mathcal{S}_{0,p,t}^\ell} |a_{p,j}^\ell|. \tag{A33}$$

Thus preservation is exact on the selected key subspace, whereas the complementary coordinates are controlled only through their residual activation energy and the norm of the projected increment. This is a local FFN-mapping guarantee and does not imply invariance of arbitrary prompts or downstream Transformer behavior.

A.11 Construction for Routed Architectures

This appendix completes the extension of Section 3.6. Throughout, $\mathcal{E}_t^\ell$ denotes the experts activated for the subject token of edit e_t at layer ℓ, and $k_{\mathrm{r}} = |\mathcal{E}_t^\ell|$ denotes routing width.

Expert-wise calibration. Normalization statistics are estimated per expert rather than per layer. For each (ℓ, e) we accumulate $\mu_{\ell,e}$ and $\sigma_{\ell,e}$ only over calibration tokens routed to expert e, discard experts receiving fewer than $n_{\min}^{\exp}$ calibration tokens from edit eligibility, and select an editable universe $\mathcal{I}_{\mathrm{edit}}^{\ell,e}$ within each eligible expert. All routed support percentages use $d_{\mathrm{edit}}^{\ell,e} = |\mathcal{I}_{\mathrm{edit}}^{\ell,e}|$ as denominator.

Layer activity under routing. The layer score is computed per activated expert and aggregated as $g^\ell(x_t) = \max_{e \in \mathcal{E}_t^\ell} g^{\ell,e}(x_t)$, so a layer is active when at least one routed expert is strongly engaged.

Routing drift and its measured scope. A record's routed path is extracted under the model state at its own edit time. Later expert-weight edits can alter the hidden representation presented to the router even though PACE does not edit router parameters, so the expert set activated by a stored subject can change. Deployed PACE does not re-extract historical materialized constraints during ordinary replay; their selectors, restricted keys, targets, expert identities, and canonical sufficient-statistic contributions remain attached to the original materialization. On the 10K Qwen3-30B-A3B CounterFact stream, stored paths retain 91.6% exact expert-set agreement, 0.972 mean set Jaccard, 98.4% dominant-expert agreement, and 99.0% retained gate mass at replay time. A one-time 5K refresh of a separate ranking-side path descriptor raises exact agreement to 96.8% but changes Rel./Gen./Loc./For. only from $93.9/86.0/97.4/3.7$ to $94.0/86.1/97.4/3.6$. The refresh does not move historical targets or sufficient-statistic contributions to newly routed experts; it measures ranking-state staleness rather than solving constraint rematerialization.

Baseline adaptations. AlphaEdit accumulates its preservation null space per activated expert, and UltraEdit applies its lifelong update to activated experts with expert-specific normalization. When several experts are activated, the baseline update is applied to each activated expert independently while router parameters remain untouched. No baseline receives access to PACE's path state.

Scope. Replacing dense indices (ℓ, j) by (ℓ, e, j) preserves the support algebra for the expert-weight coordinates actually written. It does not cover router logits, attention, residual-stream interactions, KV-cache state, or a future change in which expert serves a stored subject. The routed experiments therefore test transfer of the edited-weight support mechanism under a richer index rather than routing invariance.

Algorithm 1 PACE: one stream step. PACE-SOFT is obtained by setting $P^{\ell}_{0,t} = I$.

Require: edit e_t; corrections $\{E^{\ell}_{t-1}\}$; canonical block cache; replay memory $\mathcal{M}_{t-1}$

1: Extract $\mathcal{A}_t, \mathcal{P}_t$ from one pre-write forward pass
2: Set $\mathcal{B}_t \leftarrow \{\min \mathcal{A}_t, \max \mathcal{A}_t\}$ and select global replay set $\mathcal{R}_t$
3: Construct one joint boundary target and split it into $\{r^{\ell}_t\}_{\ell \in \mathcal{B}_t}$
4: **for** $\ell \in \mathcal{B}_t$ **do**
5: Set $\mathcal{R}^{\ell}_t = \{i \in \mathcal{R}_t : \mathcal{D}^{\ell}_i \text{ is materialized}\}$ and form $\mathcal{U}^{\ell}_t$
6: Construct current-union key embeddings $\kappa^{\ell}_{i,t}$ and source-block embeddings needed for replay removal
7: Remove every $i \in \mathcal{R}^{\ell}_t$ from its unique canonical source block using Equation A17; delete emptied blocks
8: Assemble aged surviving blocks in $\mathcal{U}^{\ell}_t$ using their own timestamps via Equation A16
9: Construct ρ^{ℓ}_t and add the current/replay insertion statistics of Equation A18
10: Solve $\Delta^{\ell}_{t,\mathcal{U}} = B^{\ell}_t (A^{\ell}_t + \lambda I)^{-1}$
11: Construct $P^{\ell}_{0,t}$ and write $E^{\ell}_t Q^{\ell}_t \leftarrow E^{\ell}_{t-1} Q^{\ell}_t + (\Delta^{\ell}_{t,\mathcal{U}} - E^{\ell}_{t-1} Q^{\ell}_t) P^{\ell}_{0,t}$
12: Commit only the insertion contribution as the new canonical block at timestamp t; repoint the inserted/replayed records to that block
13: **end for**
14: Update global replay utility once per selected record, insert e_t into $\mathcal{M}$, and evict by Equation A19 if needed

A.12 DETAILED ALGORITHM

Algorithm 1 gives one stream step; Algorithm 2 makes canonical contribution ownership, source-block removal, per-block aging, current-union assembly, insertion-block commit, and eviction explicit.

A.13 COMPLEXITY AND IMPLEMENTATION SCOPE

Let $u^{\ell}_t = |\mathcal{U}^{\ell}_t|$. For one materialized boundary layer, factorizing $A^{\ell}_t + \lambda I$ costs $O((u^{\ell}_t)^3)$, and solving for the d_{model} output rows costs $O(d_{\text{model}}(u^{\ell}_t)^2)$ after factorization. Let c^{ℓ}_t denote the number of nonzero preservation-key columns and

$$\text{rk}^{\ell}_t = \text{rank}(\widetilde{K}^{\ell}_{0,t}) \le \min\{u^{\ell}_t, c^{\ell}_t\}.$$

The compact rank-revealing factorization of Appendix A.10 has general cost

$$O\big(u^{\ell}_t c^{\ell}_t \min\{u^{\ell}_t, c^{\ell}_t\}\big),$$

which becomes $O(u^{\ell}_t (c^{\ell}_t)^2)$ in the empirically observed tall-and-thin regime $u^{\ell}_t \ge c^{\ell}_t$. Applying the preservation projector in factored form costs $O(d_{\text{model}} u^{\ell}_t \text{rk}^{\ell}_t)$. Excluding path extraction, canonical-block assembly, and candidate-logit Jacobian construction, the sparse-update cost of one edit is therefore

$$O\left(\sum_{\ell \in \mathcal{B}_t} \left[(u^{\ell}_t)^3 + d_{\text{model}}(u^{\ell}_t)^2 + u^{\ell}_t c^{\ell}_t \min\{u^{\ell}_t, c^{\ell}_t\} + d_{\text{model}} u^{\ell}_t \text{rk}^{\ell}_t\right]\right). \tag{A34}$$

Because

$$u^{\ell}_t \le \big(|\mathcal{R}^{\ell}_t| + 1\big)\left\lceil \varrho d^{\ell}_{\text{edit}} \right\rceil \le (K+1)\left\lceil \varrho d^{\ell}_{\text{edit}} \right\rceil,$$

the active solve width is bounded independently of the number of edits already processed when K, the support budget, and model dimensions are fixed. End-to-end latency additionally includes the pre-write forward pass, overlap lookup, canonical-block removal/assembly, and vector–Jacobian products for the candidate-logit Jacobian; these components are measured separately in Table 74.

The same bounded-state argument applies to persistent logical storage. A canonical block at layer ℓ stores $O((u^{\ell}_b)^2 + d_{\text{model}} u^{\ell}_b)$ values. Each retained edit materializes constraints at at most two boundary layers and, by the unique-ownership invariant of Appendix A.4, owns at most one canonical contribution at each such layer. With replay-memory capacity M, there are therefore at most

Algorithm 2 PACE: Detailed Online Editing Procedure

Require: Base weights $\{W^{\ell}_{\mathrm{out},0}\}$, edit stream $\{e_t\}_{t=1}^{T}$, fixed editable universes $\{\mathcal{I}^{\ell}_{\mathrm{edit}}\}$, path parameters $(\varrho, k_{\mathrm{lay}}, \tau_{\mathrm{lay}}, \tau_{\mathrm{coord}}, k_{\min})$, replay parameters (K, M, β, η), solver parameters $(\lambda, \gamma, \gamma_r)$, and preservation-pool size n_{base}

Initialize sparse corrections $E^{\ell}_0 \leftarrow 0$, canonical block dictionary with per-block timestamps and owner sets, record-to-block indices, and replay memory $\mathcal{M}_0 \leftarrow \emptyset$
for $t = 1$ to T **do**
  Carry forward $E^{\ell}_t \leftarrow E^{\ell}_{t-1}$ for all layers
  Run $f_{\theta_{t-1}}$ on x_t and collect FFN activations and candidate logits
  Construct $\mathcal{P}_t$, $\mathcal{B}_t$, and top-K positive-overlap replay set $\mathcal{R}_t$
  Form $J^{\mathcal{B}}_t$, solve for $\mathbf{r}^{\mathcal{B}}_t$, and split across boundary layers
  for each $\ell \in \mathcal{B}_t$ **do**
    Set $\mathcal{R}^{\ell}_t = \{i \in \mathcal{R}_t : \mathcal{D}^{\ell}_i \text{ is materialized}\}$ and form $\mathcal{U}^{\ell}_t$
    Construct Q^{ℓ}_t, current-union record embeddings $\kappa^{\ell}_{i,t}$, and native source-block keys $\kappa^{\ell}_{i,b}$ required for exact removal
    For each canonical source block owning a replayed record, subtract that record's unaged stored contribution via Equation A17; retain residual source statistics and delete blocks whose owner set becomes empty
    Embed every surviving resident block into $\mathcal{U}^{\ell}_t$ and apply its own age factor $\beta^{t-\tau^{\ell}_b}$ via Equation A16
    Construct ρ^{ℓ}_t, compute insertion weights $\bar{w}^{\ell}_{i,t}$, and form $A^{\ell}_{\mathrm{ins},t}, B^{\ell}_{\mathrm{ins},t}$ via Equation A18
    Form the ephemeral assembled solve statistics A^{ℓ}_t, B^{ℓ}_t via Equation A22
    Solve $\Delta^{\ell}_t = B^{\ell}_t (A^{\ell}_t + \lambda I)^{-1}$
    For PACE-FULL, assemble $K^{\ell}_{0,t}$, compute $\Phi^{\ell}_{K,t}$ by thin QR/SVD, and set $P^{\ell}_{0,t} = I - \Phi^{\ell}_{K,t}(\Phi^{\ell}_{K,t})^{\top}$; for PACE-SOFT, set $P^{\ell}_{0,t} = I$
    Set $\widehat{G}^{\ell}_t = (\Delta^{\ell}_t - E^{\ell}_{t-1} Q^{\ell}_t) P^{\ell}_{0,t}$ and update $E^{\ell}_t Q^{\ell}_t \leftarrow E^{\ell}_{t-1} Q^{\ell}_t + \widehat{G}^{\ell}_t$
    Commit only $(A^{\ell}_{\mathrm{ins},t}, B^{\ell}_{\mathrm{ins},t})$ as the new canonical block on $\mathcal{U}^{\ell}_t$ with timestamp t; set $(w^{\ell}_i, \tau^{\ell}_i, \chi^{\ell}_i)$ for every inserted/replayed record to its new materialization
  end for
  For each $i \in \mathcal{R}_t$, update (ν_i, c_i) once; insert current record with $(\nu_t, c_t) = (t, 0)$
  If $|\mathcal{M}| > M$, evict the record minimizing p^{evict}_i: subtract each of its canonical layerwise contributions, delete its metadata, and remove any newly ownerless blocks
end for
return f_{θ_T}

$2M$ retained record–layer contributions; every nonempty canonical block owns at least one such contribution, and ownerless blocks are deleted. Defining

$$\bar{u} = \max_{\ell} (K+1) \left\lceil \varrho d^{\ell}_{\mathrm{edit}} \right\rceil,$$

the canonical sufficient-statistic storage is consequently bounded by

$$O\big(M \left[\bar{u}^2 + d_{\mathrm{model}} \bar{u}\right]\big)$$

up to constant factors, plus $O(M)$ fixed-dimension sparse record metadata and the fixed calibration/preservation state. Thus, for fixed M, K, support budget, and model dimensions, PACE's persistent *logical* editor state is bounded independently of total stream length. Physical resident allocation can still vary with realized union widths, caching, and allocator overhead; accordingly, the measured 10K–100K memory increase is reported empirically and should not be extrapolated linearly beyond the evaluated horizon. We make no empirical 1M-edit memory claim.

A.14 LIMITATIONS OF THE METHOD

Locality of the target construction. PACE uses a first-order candidate-logit linearization around the current model state. Realization fidelity decreases with joint-residual magnitude: median $\hat{\delta}/\delta$ is $0.94/0.92$ overall on CounterFact/ZsRE and $0.84/0.80$ in the highest residual quartile. The residual-

stratified audit also shows higher forgetting in this tail, while sign inversions remain rare and locality remains high. These associations are descriptive because residual magnitude is not randomized. Remark 1 bounds only the additive-intervention Taylor remainder and does not bound physical downstream-key drift.

Locality of the preservation guarantee. Null-space projection preserves only selected path-restricted FFN keys in the current active coordinates. For replayed history keys this is a current-step drift-freezing guarantee, not an error-correction guarantee: previously accumulated direct drift need not be repaired. PACE-Full also projects after the unconstrained solve rather than solving the affine constrained problem exactly; Remark 3 bounds the resulting objective gap.

Boundary-materialization scope. Each edit stores constraints only at its written boundary layers. Interior active layers contribute to the path and risk score but have no edit-specific materialized key/target constraint. The support and cumulative theorems therefore apply only to materialized mappings and do not imply future protection at an interior layer that the original edit did not write.

Scope of support mediation and secondary collisions. Proposition 1 is exact for direct one-step change to a materialized FFN mapping and gives disjointness from the active union as a sufficient zero-change condition. It does not assert that every overlapping support produces damage, nor that prompt-level behavior is determined solely by this local mapping. Replay can enlarge $\mathcal{U}_t^\ell$ and expose an unselected third record through a selected replay support; such secondary realized-support collisions enter Theorem 1 as missed collisions. Bounded K and sparse support cap union width but do not eliminate this failure mode. The behavioral audit in Table 77 finds that missed-collision fraction and realized mapping drift are associated with downstream forgetting, but this empirical association does not convert the local algebraic theorem into a global behavioral guarantee.

Conflicting edits on a shared path. Proposition 2 describes the pre-projection ridge compromise in an idealized coincident-key case. PACE-Full can override that compromise along a preserved history-key direction by forcing zero incremental change there. Neither behavior guarantees simultaneous satisfaction of incompatible targets; arbitration/versioning is outside the method.

Staleness of stored path state. Each record's path is extracted at its own edit time. On routed Qwen3, standard stored paths retain 91.6% exact expert-set agreement, 0.972 mean Jaccard, and 98.4% dominant-expert agreement; a one-time 5K refresh of ranking-side metadata raises exact agreement to 96.8% but changes end-to-end quality only marginally. The refresh does not rematerialize historical constraints, so larger routing changes remain unresolved.

Fixed editable universe. The editable coordinate universe is selected once and frozen. The solver cannot act outside this universe, so interactions carried solely by unexposed coordinates are invisible to both the sparse write and preservation projector.

Strength and scope of the interference signal. Path overlap is the strongest tested pre-write predictor but remains moderate in absolute predictive strength ($r_s = 0.41$, AUROC 74.6 on GPT-J). This is consistent with Proposition 1: direct damage additionally depends on exposed-key energy and on the magnitude and direction of the realized increment, while downstream behavior can depend on further representation and nonlinear effects. PACE therefore uses overlap to rank a bounded replay set rather than as a calibrated damage estimate or non-interference certificate. The nine-policy fixed-budget control further shows a strong policy-level association between ranking recall and forgetting without requiring a deterministic pointwise relationship.

Scope of the shared-state claim. The independent-state control establishes that one coherent state is load-bearing within the matched extraction family used in the paper. It does not prove that every independently optimized role-specific estimator must underperform.

Architectural scope. Routed support indexes layer–expert–coordinate triples and covers only edited expert-weight mappings. It does not cover router logits, attention, residual-stream interactions, KV-cache state, or later changes in expert assignment.

Evaluation scope of the fit–preservation trade-off. Reliability matching rejects the explanation that PACE's reported preservation gain is solely a consequence of weaker default fitting, but it does not establish global Pareto dominance over every operating point of every baseline.

B Additional Experimental Results

This appendix provides the complete experimental record behind Section 4. We separate three evidence types throughout: *unified-protocol reruns* support numerical head-to-head claims; *source-native audits* provide literature coverage without ranking claims; and *diagnostic controls* test mechanism, sensitivity, or scope. Unsupported cells are left absent rather than estimated. The appendix contains the full baseline matrices, prospective and intervention controls, ablations, transfer settings, long-stream quality, cross-class external-memory scaling, selector-competition and deployment audits, robustness analyses, implementation details, and failure cases.

B.1 Experimental Protocol and Diagnostic Definitions

A *strict-online* PACE run consumes the externally supplied edit stream one record at a time: edit t is localized, solved, and written before edit $t{+}1$ is revealed. PACE may retain only the bounded replay/statistic state of Section 3; it never initiates a global or breadth-first revisit of the accumulated edit history. Repeated CounterFact records that occur after one complete benchmark pass in the long-horizon stress tests below are therefore *exogenous stream arrivals*, not editor-initiated replay. Methods that retain a growing external store (e.g., LightEdit) are treated as cross-class external-memory comparators rather than folded into this bounded-state definition. LocFT-BF is likewise reported as a reference operating regime because it explicitly revisits the accumulated edit set breadth-first.

Construction of CounterFact stress-test streams beyond one pass. Canonical CounterFact contains $21{,}919$ released edit records, so checkpoints beyond one complete pass cannot represent the same number of distinct facts. For run seed s, we construct the stress-test stream by concatenating deterministic full permutations $\pi_{s,p}$ of the canonical index set, one permutation per pass p:

$$\pi_{s,0} \| \pi_{s,1} \| \cdots ,$$

and truncate the resulting sequence at the requested checkpoint. Every canonical record therefore appears exactly once in each completed pass and at most once in the final partial pass. A revisit preserves the original subject, relation, prompt fields, and counterfactual target; we neither synthesize additional targets nor replace a record by a conflicting target. The edited model and all bounded PACE state continue across pass boundaries without reset. Consequently, CounterFact checkpoints beyond $21{,}919$ update operations measure repeated-update long-horizon stability under a fixed record universe, not 25K–100K distinct-fact capacity. Throughout these rows, "edits" denotes sequential update operations, and we explicitly call the beyond-one-pass trajectories *stress-test streams*.

Construction of long-horizon ZsRE streams. For checkpoints beyond one canonical pass, we use the same successive seeded-permutation construction as for CounterFact: deterministic full permutations of the canonical ZsRE record set are concatenated, revisited records retain their original targets, and neither the model nor method state is reset across pass boundaries. Accordingly, the 50K and 100K checkpoints denote sequential update operations rather than distinct facts.

Historical population used for long-horizon forgetting. For CounterFact stress-test checkpoints, forgetting is evaluated exhaustively over distinct canonical records rather than over a sampled history. Let $\mathcal{H}_t$ be the set of unique canonical record identities observed by checkpoint t. For record i, let $a_i^{\mathrm{post}} \in \{0,1\}$ denote correctness immediately after its first materialization in the uninterrupted stream and let $a_i^t \in \{0,1\}$ denote correctness at checkpoint t under the same generation-based evaluator. We report

$$\mathrm{For.}_t = 100 \, \frac{1}{|\mathcal{H}_t|} \sum_{i \in \mathcal{H}_t} \left(a_i^{\mathrm{post}} - a_i^t \right). \tag{B1}$$

A later revisit of the same canonical record neither creates a second history item nor resets a_i^{post}; it may change current correctness, but the first-materialization reference remains fixed. After the first complete CounterFact pass, $|\mathcal{H}_t|$ is therefore capped by the canonical record set even when the number of update operations grows to 50K or 100K. No historical subsampling or history-sampling seed enters these reported CounterFact For. values. This global checkpoint metric is distinct from R@8 below, which is deliberately an *intra-buffer* ranking diagnostic over the bounded replay memory.

For prior edit e_i, let m_i^{t-1} and m_i^t denote its target-versus-old-object margins immediately before and after incoming edit e_t. We define one-step interference as

$$I(i,t) = \max\{0, m_i^{t-1} - m_i^t\}.$$

A pair is a *severe event* when the prior target-supported margin crosses zero in the harmful direction or decreases by at least one logit unit. AUROC treats severe events as the positive class. R@8 is the recall, expressed in percentage points, of the eight prior records with the largest measured one-step margin loss *within the retained replay memory at that step*; it is not recall over the full historical stream. Thus 57.6 denotes recall 0.576. Unless a control explicitly changes replay, the default bounded state uses memory capacity $M = 64$ and replay count $K = 8$; when memory is full, a uniform random $K = 8$ selector has expected R@8 of $8/64 = 12.5\%$, providing the random-reference scale used in the ranking diagnostics. Spearman r_s uses the continuous one-step interference magnitude. Five-order CounterFact variability for UltraEdit, PACE-FULL, and PACE-MATCHED under matched stream-order seeds is reported in Appendix B.39. PACE-specific ZsRE and long-horizon order-sensitivity diagnostics are reported there separately. These measurements are used as descriptive stability evidence; inferential claims are reserved for experiments with an explicit inferential procedure.

We abbreviate LLaMA-3-8B-Instruct as LLaMA-3-8B and Qwen3-30B-A3B-Base as Qwen3-30B-A3B after their first mention. Where reported, Avg. is the unweighted mean of Rel., Gen., and Loc.; For. is kept separate.

The deployed replay score is binary path Jaccard. We additionally evaluate a magnitude-aware variant using the *same* selected support, so it changes only how shared coordinates are weighted. Let $c = (\ell, j)$ denote a dense layer–coordinate index and let $s_c(x_t)$ be the calibration-normalized activation salience used by the path extractor. Define

$$q_{t,c} = \mathbf{1}[c \in \mathcal{P}_t]\,[s_c(x_t)]_+, \qquad [z]_+ = \max(z, 0).$$

The weighted path Jaccard is

$$\omega_{\text{w}}(i,t) = \frac{\sum_{c \in \mathcal{P}_i \cup \mathcal{P}_t} \min(q_{i,c}, q_{t,c})}{\sum_{c \in \mathcal{P}_i \cup \mathcal{P}_t} \max(q_{i,c}, q_{t,c})}, \tag{B2}$$

with the zero-denominator case assigned score 0 (it is not encountered for the nonempty paths used in our experiments). For routed FFNs, $c = (\ell, e, j)$ and s_c uses the corresponding expert-wise calibration statistics. Binary Jaccard is recovered by replacing every nonzero $q_{t,c}$ with one. Weighted Jaccard is an auxiliary diagnostic/ranking alternative, not the deployed PACE score; binary Jaccard is retained because both obtain the same rounded downstream Avg. while the binary representation requires less stored state.

For both support-intervention experiments we evaluate $n = 1{,}000$ unique prior–incoming CounterFact pairs per backbone from the *operational replay population*, i.e., prior records co-resident with the incoming edit in the bounded memory before the write. We first sample distinct eligible incoming positions uniformly without replacement from the 10K stream. For a sampled position t, we then sample one prior record uniformly from the feasible records in the pre-write memory $\mathcal{M}_{t-1}$. Feasibility is purely mechanical: the records must be distinct, their natural paths must be nonempty, and the editable universe must admit capacity-matched forced-low and forced-high supports under the same boundary-layer and per-layer support-size constraints. If a sampled incoming position has no feasible predecessor, it is rejected before any intervention outcome is evaluated and the next position in the seeded sampling order is used. We do not stratify or filter by natural overlap magnitude, lexical or semantic similarity, target gain, $I(i,t)$, severe-event status, or the prior edit's post-write margin. This memory-conditioned sampling frame is deliberate: it matches the bounded candidate population from which PACE can operationally select replay records. Pair identities and corresponding pre-incoming states are frozen before post-edit interference is evaluated and are reused unchanged

across all three intervention arms. Each arm uses the same incoming fact, target, requested margin, boundary layers, and per-layer support-size budget. The intervention therefore identifies the effect of support manipulation conditional on this fixed operational population; it is not used to estimate pair frequencies over the full $O(T^2)$ historical pair space.

B.2 Full Protocol-Matched and Relocated Main-Paper Results

The main paper retains the shortest claim-supporting chain: matched reliability, cross-architecture predictive validity, the LLaMA-3 realized-write-matched intervention, fixed-budget replay ranking, and the GPT-J shared-state control. This section reports the complete protocol-matched matrices and secondary controls. Table 5 contains the full GPT-J 10K reruns including RLSEdit; Table 13 adds random, lexical, restricted-key, and prospective controls; and Table 26 gives the complete downstream metrics for the fixed-budget ranking experiment. Unified-protocol rows support head-to-head numerical claims; source-native rows below are explicitly contextual and support no cross-row ranking claim.

Dataset	Method	Rel.	Gen.	Loc.	For.↓	Time/edit↓
CounterFact, 10K	AlphaEdit (rerun)	90.9	78.6	88.9	12.4	3.58
	LightEdit (rerun)	92.2	76.4	**97.5**	5.6	0.88
	UltraEdit (rerun)	**93.4**	82.2	93.6	7.2	**0.68**
	RLSEdit (rerun)	92.6	82.9	91.4	6.3	0.69
	PACE-Full	91.8	**83.6**	96.9	**4.8**	0.74
ZsRE, 10K	AlphaEdit (rerun)	94.0	86.0	86.5	11.6	3.47
	LightEdit (rerun)	**94.8**	**91.1**	**99.4**	5.1	0.84
	RLSEdit (rerun)	94.1	87.3	90.8	6.1	**0.67**
	PACE-Full	92.3	84.1	97.2	**4.7**	0.71

Table 5: Full completed 10K continual-editing reruns on GPT-J-6B. Every displayed row uses the same backbone, evaluator, prompt format, hardware, and canonical edit stream. PACE does not dominate raw fit. Against the parameter-editing baselines, its separation is concentrated on locality and forgetting; LightEdit occupies a distinct external-memory operating point: its reported For. under the common evaluator reflects retrieval/selection failures rather than base-weight drift. A unified-protocol GPT-J ZsRE UltraEdit measurement is not present in the completed experimental record, so that row is left absent rather than copied from the LLaMA-3 experiment or estimated. Five-order CounterFact variability for UltraEdit, PACE-Full, and PACE-Matched is reported in Appendix B.39. Because corresponding five-order reruns are not available for every baseline in this table, cross-method conclusions are reported as measured effect sizes rather than as a table-wide significance test.

Reading the complete LLaMA-3 CounterFact block. The completed matrix exposes complementary operating points rather than a single dominating method. UltraEdit has the highest raw reliability (93.7), LightEdit the highest locality (98.1), and PACE-Full the strongest generalization (85.0) and lowest forgetting (4.5) among the sequentially evaluated rows. The cross-class comparison with LightEdit is particularly informative because reliability is already nearly matched (93.0 versus 93.3): PACE gains 4.1 generalization points and reduces forgetting by 0.7, while LightEdit retains a 1.0-point locality advantage. RLSEdit is the closest mechanistic parametric control because it shares recursive least-squares structure without PACE's edit-conditioned support; PACE improves over it by $+0.9$ Gen., $+4.5$ Loc., and -1.3 For. The table therefore supports a different fit–preservation frontier, not universal metric dominance.

GPT-J support-intervention protocol. We evaluate $n = 1{,}000$ fixed prior–incoming pairs. For each pair, Natural uses the extracted incoming support. Forced-Low selects a capacity-matched support that minimizes feasible overlap with the prior support inside the same editable universe, and Forced-High selects a capacity-matched support that maximizes feasible overlap. The edit request, requested target margin, boundary layers, support cardinality, and pre-intervention model state are identical across arms. The construction never uses prior-edit damage. This GPT-J

Method (GPT-J, our evaluator)	Rel.	Gen.	Loc.	For.↓	Time↓
Single-edit CounterFact					
AlphaEdit (rerun)	98.7	88.1	95.4	0.0	3.42
LightEdit (rerun)	**99.0**	74.8	97.6	0.0	0.81
PACE-FULL	98.1	**89.7**	**97.8**	0.0	**0.74**
10K sequential CounterFact					
AlphaEdit (rerun)	90.9	78.6	88.9	12.4	3.58
LightEdit (rerun)	92.2	76.4	**97.5**	5.6	0.88
UltraEdit (rerun)	**93.4**	82.2	93.6	7.2	**0.68**
PACE-FULL	91.8	**83.6**	96.9	**4.8**	0.74

Table 6: Unified-protocol comparison on GPT-J-6B. All rows share the same backbone, evaluator, prompt format, hardware, and edit-stream setting. Rel. and Gen. are generation-based, Loc. is agreement-based. Time is measured edit-insertion and state-update latency in seconds per edit on the same NVIDIA A6000; it excludes downstream answer-generation latency, so for LightEdit it does not include the per-query selector and decoding-time suppression overhead. In the single-edit setting no prior edit exists, so For. is 0.0 by convention and excluded from comparison. Five-order CounterFact variability for UltraEdit, PACE-FULL, and PACE-MATCHED is reported in Appendix B.39. AlphaEdit and LightEdit were not rerun under that same five-order protocol, so comparisons involving those methods remain descriptive effect-size comparisons rather than shared-variance significance tests. Full CounterFact and ZsRE results are in Appendix B.12.

Method	Source setting	CounterFact				ZsRE			
		Eff./Rel.	Gen.	Spe./Loc./Cap.	Avg.	Eff./Rel.	Gen.	Spe./Loc./Cap.	Avg.
FT *[source]*	GPT-J, 2K seq., batch 100	92.15	72.38	43.35	69.3	72.37	68.91	19.66	53.6
MEND Mitchell et al. (2022a) *[ICLR'22]*	GPT-J, 2K seq., batch 100	46.15	46.22	53.90	48.8	0.71	0.71	0.52	0.6
SERAC Mitchell et al. (2022b) *[ICML'22]*	GPT-J, 2K seq., batch 100	55.88	51.39	53.78	53.7	49.48	1.59	8.84	20.0
ROME Meng et al. (2022) *[NeurIPS'22]*	GPT-J, 2K seq., batch 100	57.50	54.20	52.05	54.6	56.42	54.65	9.86	40.3
MEMIT Meng et al. (2023) *[ICLR'23]*	GPT-J, 2K seq., batch 100	98.55	95.50	63.64	85.9	94.91	90.22	30.39	71.8
ECE Zhang et al. (2025) *[WWW'25]*	GPT-J, 2K seq., batch 100	100.00	96.35	79.41	91.9	99.74	96.88	28.49	75.0
AlphaEdit Fang et al. (2025) *[ICLR'25]*	GPT-J, 2K seq., batch 100	99.75	96.38	75.48	90.5	99.79	96.00	28.29	74.7
AlphaEdit+BLUE Li et al. (2025) *[NeurIPS'25]*	GPT-J, 2K seq., batch 100	99.77	97.13	75.23	90.7	99.63	95.96	28.67	74.8
NeuralDB Fei et al. (2026) *[ICLR'26]*	GPT-J, 2K source-native	99.70	94.60	80.00	91.4	99.20	95.90	27.50	74.2
LightEdit Jung et al. (2026) *[ACL'26]*	GPT-J, 1K test split	99.10	73.20	97.50	89.9	99.39	94.05	99.84	97.8
LocFT-BF Yang et al. (2026) *[ICLR'26]*	LLaMA-3-8B-Instruct, 3K lifelong	99.73	33.23	57.13†	63.4	98.97	75.83	57.04†	77.3
PACE-SOFT	GPT-J, single edit, ours	97.8	88.5	97.2	94.5	96.0	89.4	97.3	94.2
PACE-FULL	GPT-J, single edit, ours	98.1	89.7	97.8	95.2	96.4	90.2	98.0	94.9

Table 7: Source-native audit on CounterFact and ZsRE. Source rows are verbatim with per-row settings (Eff./Spe. are probability-style; †LocFT-BF reports Capability); PACE rows are single-edit under our evaluator. No bolding is used and no ranking is implied: the rows differ in backbone, stream length, and metric family. The FT/MEND/SERAC/ROME/MEMIT rows are the unified 2K-sequential measurements from the AlphaEdit evaluation rather than those papers' original single-edit tables. Avg. is taken within each row's native metric tuple.

experiment matches the requested write budget but does not explicitly match realized target gain; the LLaMA-3 replication below adds that control.

Routing-drift audit on the routed backbone. We directly measure routing-path staleness on Qwen3-30B-A3B. For a layer-valid replay event (i, t, ℓ), let $\mathcal{E}^{\ell}_{i,\mathrm{ref}}$ denote the expert set represented by the ranking-side path descriptor currently stored for record i, and let $\mathcal{E}^{\ell}_{i,t}$ denote the expert set obtained by re-running that record's subject token under the current pre-write model.

For the standard stored-path condition, $\mathcal{E}^{\ell}_{i,\mathrm{ref}}$ is the route recorded when the edit was materialized. In the refresh condition, only a separate ranking-side path descriptor is re-extracted once at 5K edits for records then present in the bounded replay memory. The materialized constraint itself—including

Method	CounterFact, source-native 10K setting			ZsRE, source-native 10K setting		
	Eff./Rel.	Gen.	Spe./Loc.	Eff./Rel.	Gen.	Spe./Loc.
MEMIT (NeuralDB rerun) (Fei et al., 2026)	48.8	49.3	51.9	0.2	0.1	0.2
RECT (NeuralDB rerun) (Fei et al., 2026)	76.3	70.6	54.9	53.5	49.6	21.9
AlphaEdit (NeuralDB rerun) (Fei et al., 2026)	91.6	79.6	60.3	94.2	86.1	22.5
NeuralDB Fei et al. (2026) *[ICLR'26]*	99.1	93.2	75.7	98.2	95.0	27.0
PACE-SOFT	90.4	82.1	96.5	91.1	83.0	96.9
PACE-FULL	91.8	83.6	96.9	92.3	84.1	97.2

Table 8: Literature-context audit for 10K sequential editing. Baseline rows are NeuralDB's unified 10K source metrics (Eff./Gen./Spe.), whereas PACE is scored by our Rel./Gen./Loc. evaluator; in particular, source specificity and our locality are not the same metric. The table is therefore contextual and is not used for head-to-head ranking. Forgetting under our evaluator is PACE-FULL $4.8/4.7$ and PACE-SOFT $6.3/5.8$ on CounterFact/ZsRE. The PACE-SOFT row is exactly the $P^{\ell}_{0,t} = I$ configuration of Section 3.4 and is reused verbatim as the *w/o null-space projection* ablation of Table 25.

Method (GPT-J, WikiBigEdit 10K)	Rel.	Gen.	Loc.	For.↓	Time/edit↓
AlphaEdit (rerun)	89.4	77.6	88.5	13.1	3.61
UltraEdit (rerun)	**92.0**	81.0	93.0	8.0	**0.70**
PACE-FULL	90.3	**81.9**	**96.4**	**5.7**	0.76

Table 9: Measured end-to-end continual-editing results on WikiBigEdit with 10K chronologically ordered edits on GPT-J-6B. Among these three protocol-matched methods, the same qualitative CounterFact pattern persists on an untuned stream: UltraEdit has higher reliability and lower edit time, while PACE-FULL has higher generalization and locality and lower forgetting.

Backbone	Method / update regime	Rel.↑	Gen.↑	Loc.↑	For.↓	Time/edit↓
LLaMA-3-8B	AlphaEdit, sequential	90.8	80.3	91.8	9.8	**0.78**
	LightEdit, external-memory online*	93.3	80.9	**98.1**	5.2	0.93
	UltraEdit, sequential	**93.7**	83.4	94.6	6.7	0.81
	RLSEdit, sequential	93.2	84.1	92.6	5.8	0.82
	LocFT-BF, breadth-first‡	96.0	87.8	96.4	3.5	1.55
	PACE-FULL, strict online	93.0	**85.0**	97.1	**4.5**	0.90
Qwen2.5-7B	AlphaEdit, sequential	91.2	80.8	92.1	9.4	**0.75**
	UltraEdit, sequential	**94.0**	83.9	94.9	6.4	0.78
	LocFT-BF, breadth-first‡	96.4	88.2	96.6	3.2	1.48
	PACE-FULL, strict online	93.4	**85.5**	**97.2**	**4.3**	0.88

Table 10: Modern-backbone continual-editing comparison under the unified 10K CounterFact evaluation. All non-LocFT-BF rows consume the same edits in the same sequential order. *LightEdit is a cross-class comparator: it is parameter preserving and accumulates an external edit store, so it does not share PACE's bounded-state constraint even though the evaluation stream and evaluator are matched. Its edit-time value excludes the query-time selector and decoding overhead quantified separately in Figure 6. ‡LocFT-BF revisits the accumulated edit set breadth-first and is reported as a contemporary reference operating point rather than a strict-online match. Bold marks the best value per column among the sequentially evaluated non-LocFT-BF rows; the caption and text state the intervention-class differences explicitly.

LLaMA-3-8B stream length	Rel.↑	Gen.↑	Loc.↑	For.↓
10,000	93.0	85.0	97.1	4.5
12,500	92.7	84.6	97.0	4.7
15,000	92.2	84.0	96.8	5.1
17,200	91.6	83.3	96.6	5.7

Table 11: Measured PACE-FULL checkpoints along the single uninterrupted budget-matched LLaMA-3-8B CounterFact stream; the model and editor state are not reset between checkpoints. The 10K row reproduces Table 10, and the 17.2K row is the endpoint used in Table 12. Reliability and generalization decrease monotonically while forgetting increases over this measured interval; absolute values need not match the GPT-J trajectory of Table 23 because the backbone differs.

Method	Edits in budget	Rel.	Gen.	Loc.	For.↓	Wall-clock
LocFT-BF, breadth-first	10,000	96.0	87.8	96.4	3.5	4.31 h
PACE-FULL, strict online	17,200	91.6	83.3	96.6	5.7	4.30 h

Table 12: Measured equal-wall-clock comparison on LLaMA-3-8B. LocFT-BF is evaluated after 10K edits, PACE after the budget-matched 17.2K-edit stream implied by the measured per-edit latency ratio; intermediate checkpoints of that stream are in Table 11. Because the two rows are evaluated after different numbers of edits, this table characterizes a throughput–quality trade-off and does not replace the equal-edit-count comparison.

Signal	Spearman r_s ↑	AUROC↑	R@8↑
In-stream (10K CounterFact, GPT-J-6B)			
Random score	0.00	50.0	12.5
Lexical overlap	0.12	56.8	22.0
Prompt-embedding cosine	0.20	61.9	30.5
Restricted-key cosine	0.31	68.7	45.8
Path Jaccard	0.39	73.8	57.6
Weighted path Jaccard	**0.41**	**74.6**	**59.1**
Prospective held-out stream, disjoint subjects/relations			
Random score	0.00	50.0	12.5
Lexical overlap	0.10	55.4	20.1
Prompt-embedding cosine	0.16	59.0	27.2
Restricted-key cosine	0.25	65.1	40.3
Path Jaccard	0.32	69.6	51.1
Weighted path Jaccard	**0.34**	**70.8**	**52.8**

Table 13: Predictive validity of edit-similarity signals on GPT-J-6B. The upper panel uses the 10K CounterFact stream; the lower panel evaluates a held-out stream with disjoint subjects and relations, with every signal computed before the incoming write. R@8 is intra-buffer recall of the eight retained records with the largest realized one-step margin loss and is not global-history recall. AUROC treats severe interference events as positives. Weighted Jaccard is diagnostic; deployed PACE uses binary Jaccard. All displayed outcome values are measured.

Split	Predictor	Coefficient	95% CI	Significance
In-stream	Path Jaccard	0.31	[0.27, 0.35]	$p < 10^{-4}$
	Restricted-key cosine	0.17	[0.13, 0.21]	$p < 10^{-4}$
	Prompt semantic similarity	0.09	[0.05, 0.13]	$p < 10^{-3}$
	Lexical overlap	0.03	[-0.01, 0.07]	n.s. ($p > 0.05$)
Prospective	Path Jaccard	0.25	[0.20, 0.30]	$p < 10^{-4}$
	Restricted-key cosine	0.13	[0.08, 0.18]	$p < 10^{-3}$
	Prompt semantic similarity	0.06	[0.01, 0.11]	$p < 0.05$
	Lexical overlap	0.02	[-0.03, 0.07]	n.s. ($p > 0.05$)

Table 14: Controlled-interference regression in-stream and on the prospective subject/relation-disjoint split. The four listed non-random predictors are entered jointly, so each coefficient is conditional on the other lexical, semantic, key-space, and path signals. Path Jaccard remains the largest positive coefficient in both splits; its reduction from 0.31 to 0.25 is reported descriptively under distribution shift. All displayed outcome values are measured.

Write condition	Mean overlap ω	Mean $I(i,t)$	Severe-event rate
Forced-low (non-path coords)	0.03	0.12	4.6%
Natural (unmodified path)	0.18	0.27	10.2%
Forced-high (shared coords)	0.56	0.54	21.9%

Table 15: Support intervention on GPT-J-6B over $n = 1{,}000$ fixed prior–incoming pairs. The edit request, requested target margin, boundary layers, support-size budget, and pre-intervention model state are shared across arms; support overlap with the prior edit is manipulated. Realized target gain is not explicitly matched in this GPT-J experiment, so the forced-low arm may partly weaken the incoming write. The LLaMA-3 replication in Appendix B.7 controls this confound.

Method	Source setting	Multi-hop Acc.
Base (unedited, CoT) (Zhong et al., 2023)	GPT-J, MQuAKE-cf	39.5
MEMIT Meng et al. (2023)		
[ICLR'23]	GPT-J, k=1	11.8
MEND (Mitchell et al., 2022a)	GPT-J, k=1	11.7
MeLLo (Zhong et al., 2023)	GPT-J, k=1	38.9
MeLLo (Zhong et al., 2023)	GPT-3, k=1	52.1
MEMIT Meng et al. (2023)		
[ICLR'23]	GPT-J, k=3000	5.4
MeLLo (Zhong et al., 2023)	GPT-J, k=3000	14.2
MeLLo (Zhong et al., 2023)	GPT-3, k=3000	30.9
PACE-SOFT	GPT-J, per-instance (k=1), ours, direct answer	41.6
PACE-FULL	GPT-J, per-instance (k=1), ours, direct answer	44.3

Table 16: Source-native multi-hop editing on MQuAKE-cf (updated arXiv revision). Rows span grouping sizes k and prompting regimes (Base/MEMIT/MEND: CoT; MeLLo: native decomposition; PACE: direct answer). The PACE rows are therefore not compared numerically with the source-native CoT rows in this table. Regimes differ, so this is dataset-compatible context rather than a unified rerun; the prompt-matched comparison is Appendix B.45.

Cell	Localization	Replay ranking	Rel.	Gen.	Loc.	For.↓	Avg.
A	Path boundary $\mathcal{B}_t$	Path Jaccard	91.8	83.6	96.9	4.8	90.8
B	Path boundary $\mathcal{B}_t$	Prompt embedding	89.5	80.8	96.4	7.8	88.9
C	Fixed middle layer ℓ=14	Path Jaccard	88.7	79.4	94.0	8.9	87.4
D	Fixed middle layer ℓ=14	Prompt embedding	87.6	78.4	93.8	9.8	86.6

Table 17: 2×2 factorial control on CounterFact, 10K edits, GPT-J-6B. Sparse active coordinates $\mathcal{U}_t^\ell$ and preservation support $K_{0,t}^\ell$ remain path conditioned in every cell; only localization and replay ranking vary. Cells A and B reproduce the corresponding path-boundary ranking conditions in Table 26, while Cell C reproduces the fixed-middle-layer/path-Jaccard operating point in Table 62. All displayed outcomes are measured.

Parameter	Range tested	Avg. range	ΔAvg.	Best / default
Load-bearing (trade-off knobs)				
Path percentile ϱ	1%–16%	88.7–90.8	2.1	4%
Decay factor β	0.90–1.00	88.7–90.8	2.1	0.99
Replay capacity M	16–128	88.9–90.8	1.9	64
Robust (Avg. varies ≤ 0.8)				
Layer threshold τ_{lay}	0.5–1.5	90.0–90.8	0.8	1.0
Preservation pool n_{base}	0–4,000	90.1–90.8	0.7	2,000
Editable width d_{edit}	4,096–16,384	90.1–90.8	0.7	8,192
Minimum support $k_{\min}$	16–64	90.3–90.8	0.5	32
Target margin γ	1.0–2.0	90.5–91.0	0.5	1.0
Replay size K	4–16	90.4–90.8	0.4	8
Replay weight η	0.5–2.0	90.4–90.8	0.4	1.0
Score width k_{lay}	32–128	90.4–90.8	0.4	64

Table 18: Hyperparameter sensitivity on GPT-J-6B after 10K sequential CounterFact edits. Avg. is the mean of Rel., Gen., and Loc.; ΔAvg. is the spread across the tested range. Each row condenses a full sweep: ϱ (Table 64), M and β (Table 65), τ_{lay} and k_{lay} (Table 67), K (Table 43), η (Table 44), $k_{\min}$ (Table 45), γ over $\{1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}$ (Table 28), n_{base} (Table 66), and d_{edit} (Table 85). The $M = 0$ setting is excluded here because it removes replay entirely and is reported as a component ablation in Table 25.

Backbone	Method	Rel.↑	Gen.↑	Loc.↑	For.↓	Time/edit↓
Dense LLaMA-3-8B	AlphaEdit	90.8	80.3	91.8	9.8	**0.78**
	UltraEdit	**93.7**	83.4	94.6	6.7	0.81
	PACE-FULL	93.0	**85.0**	**97.1**	**4.5**	0.90
Routed Qwen3-30B-A3B	AlphaEdit	92.0	81.4	92.6	7.9	1.42
	UltraEdit	**94.4**	84.5	95.2	5.2	**1.34**
	PACE-FULL	93.9	**86.0**	**97.4**	**3.7**	1.50

Table 19: Continual editing after 10K sequential CounterFact edits on a dense and a routed backbone; the LLaMA-3 block reproduces Table 10. The two blocks use different pretrained backbones, so their absolute difference does not identify a causal routing effect. Bold marks the best value per column within each block.

Backbone	Support (%)	Mean ω	$\lvert\mathcal{A}_t\rvert$	Experts/edit	$p99\ u_t^\ell$
GPT-J-6B (dense)	3.8	0.184	2.4	—	1016
LLaMA-3-8B (dense)	3.6	0.176	2.3	—	1092
Qwen3-30B-A3B (routed)	3.4	0.092	2.2	17.1	846

Table 20: Activation-path statistics on the 10K CounterFact stream. Support is the mean fraction of the editable universe retained per active FFN, computed within each activated expert for the routed backbone; mean ω is the average pairwise overlap among records considered by the bounded replay memory. $|\mathcal{A}_t|$ is the mean number of active layers and is *not* the boundary-layer count $|\mathcal{B}_t|$ reported in Table 57 (e.g., GPT-J has mean $|\mathcal{A}_t| = 2.4$ but mean $|\mathcal{B}_t| = 1.96$). "Experts/edit" counts activated expert instances across active layers. The final column is the 99th percentile of the replay-expanded per-layer solve width u_t^ℓ, not total path cardinality; the GPT-J value is the 10K entry of Table 58. The GPT-J support and mean-overlap entries reproduce the corresponding quantities in Table 57.

its support selector, restricted key, stored target, and removal bookkeeping—is left unchanged. Thus the refresh changes only the path used by overlap-based replay ranking; it does not rematerialize historical constraints on newly routed experts.

We report four complementary quantities: exact expert-set agreement, set Jaccard similarity, dominant-expert agreement, and retained router mass. For the last quantity, let $p^{\ell}_{i,t}(e)$ denote the current normalized router weight on activated expert e. We define

$$G_{\mathrm{ret}}(i,t,\ell) = \frac{\sum_{e \in \mathcal{E}^{\ell}_{i,\mathrm{ref}} \cap \mathcal{E}^{\ell}_{i,t}} p^{\ell}_{i,t}(e)}{\sum_{e \in \mathcal{E}^{\ell}_{i,t}} p^{\ell}_{i,t}(e)}.$$

Any drift is one minus the exact-set-agreement indicator. All routing-agreement statistics are measured before the incoming edit and never use post-edit damage. To make the two conditions directly comparable, the agreement statistics are evaluated over the same post-refresh window (steps 5,001–10,000).

To test whether stale ranking state materially affects editing quality, we run the complete 10K stream with the one-time 5K ranking-metadata refresh. The analysis-only forward passes used to measure routing agreement are excluded from deployment latency; the refresh-condition Time/edit includes the one-time refresh cost amortized over the full stream.

Condition	Exact route-set agr.↑	Route Jaccard↑	Dominant-expert agr.↑	Retained gate mass↑	Any drift↓
Stored paths	91.6	0.972	98.4	99.0	8.4
Refresh metadata at 5K	96.8	0.989	99.5	99.6	3.2

Table 21: **Routing-drift audit on the 10K Qwen3-30B-A3B CounterFact stream.** Routing agreement is computed over the common post-refresh evaluation window (steps 5,001–10,000) for both conditions. Exact route-set agreement, dominant-expert agreement, retained gate mass, and Any drift are percentages; route Jaccard is reported on $[0, 1]$. The refresh condition updates only the ranking-side path descriptor once at 5K; historical materialized constraints are not changed.

Condition	Rel.↑	Gen.↑	Loc.↑	For.↓	Time/edit↓
PACE-FULL, stored paths	93.9	86.0	97.4	3.7	1.50
Refresh metadata at 5K	94.0	86.1	97.4	3.6	1.51

Table 22: **End-to-end sensitivity to a one-time routing-path refresh on Qwen3-30B-A3B.** The stored-path row is standard PACE-FULL. In the refresh condition, ranking-side path metadata for the bounded replay memory are re-extracted once at 5K edits, while historical materialized constraints remain unchanged. Time/edit includes the one-time refresh overhead amortized over the 10K stream and excludes analysis-only routing-audit forward passes.

The audit shows measurable but limited routing drift at the evaluated horizon. Under standard stored paths, exact expert-set agreement is 91.6%, while mean route Jaccard remains 0.972, dominant-expert agreement is 98.4%, and 99.0% of current router mass remains on experts represented by the stored ranking state. Thus most exact-set changes affect lower-weight routed experts rather than the dominant route.

A one-time 5K ranking-state refresh increases exact agreement to 96.8% and route Jaccard to 0.989, while reducing Any drift from 8.4% to 3.2%. End-to-end quality changes only marginally: Rel./Gen./Loc./For. moves from $93.9/86.0/97.4/3.7$ to $94.0/86.1/97.4/3.6$, and amortized Time/edit from 1.50 to 1.51 seconds. The result therefore quantifies routing-path staleness and shows limited practical impact on replay ranking over this 10K stream.

This diagnostic does not establish routing invariance and does not rematerialize historical targets when expert assignments change. Full constraint rematerialization under larger routing shifts remains outside the current method.

Replay plug-in execution. The transfer experiment wraps each host editor at the request level while leaving its released optimization rule unchanged. At incoming step t, replay candidates are

Edits	Rel.↑	Gen.↑	Loc.↑	For.↓	Avg.↑
1K	96.7	87.6	97.8	1.4	94.0
5K	94.1	85.5	97.3	3.1	92.3
10K	91.8	83.6	96.9	4.8	90.8
25K	89.3	80.8	96.1	7.0	88.7
50K	86.4	77.6	94.9	10.3	86.3
100K	81.7	72.4	92.8	15.8	82.3

Table 23: Measured PACE-FULL performance as the GPT-J-6B CounterFact stress-test stream grows without resetting the model or bounded editor state. Checkpoints beyond one complete CounterFact pass are sequential update operations generated by successive seeded full-dataset permutations with the original target retained on every revisit; they are not distinct-fact counts. For. is evaluated exhaustively over unique previously encountered canonical records using the first-materialization reference of Equation B1. For reliability at 1K, 5K, 10K, and 25K edits, the reported values are the means over the same five random edit orders used in Figure 7; the corresponding order-induced standard deviations are reported there. The 2K projection diagnostic of Table 84 (94.0) and the 5K reliability reported here (94.1) differ by less than the measured edit-order variability and should not be interpreted as non-monotone long-stream behavior.

ranked from the $M = 64$ records already present in memory using the pre-write model; the top $K = 8$ records form $\mathcal{R}_t$. The wrapper then supplies the replay-augmented request set $\{e_t\} \cup \mathcal{R}_t$ to the host editor, which is invoked *once* for that stream step through its native multi-request update interface, preserving the implementation's own internal batching, normalization, regularization, and parameter-update order. We add no PACE-specific replay weight, decay factor, null-space projector, sparse residual solver, or post-update correction to the host. Only after the native host update completes is e_t inserted into the bounded replay memory and its path metadata recorded, so the incoming record can never replay itself. Random-ranked and overlap-ranked runs share the initial checkpoint, edit sequence, M, K, path-extraction workload, replay-request count, host hyperparameters, and execution schedule; within a host, only the identities of the selected replay records differ.

Host editor	Replay condition	Rel.	Gen.	Loc.	For.↓	Time/edit↓
AlphaEdit	none (native)	90.9	78.6	88.9	12.4	3.58
	random-ranked, K=8	91.0	79.0	88.7	10.8	3.88
	overlap-ranked, K=8	91.1	79.6	89.2	9.3	3.88
RLSEdit	none (native)	92.6	82.9	91.4	6.3	0.69
	random-ranked, K=8	92.6	82.9	91.3	6.4	0.93
	overlap-ranked, K=8	92.7	83.1	91.9	5.6	0.93
UltraEdit	none (native)	93.4	82.2	93.6	7.2	0.68
	random-ranked, K=8	93.4	82.3	93.4	6.7	0.93
	overlap-ranked, K=8	93.4	82.6	93.9	6.0	0.93
PACE-FULL	overlap-ranked (native)	91.8	83.6	96.9	4.8	0.74

Table 24: Plug-in transfer under the unified 10K CounterFact protocol on GPT-J-6B. Native rows reproduce their measured unified-protocol operating points. At each incoming step, the wrapper selects $K = 8$ records from the pre-write $M = 64$ memory, supplies the resulting replay-augmented request set once to the host's unchanged native multi-request update interface, and inserts the incoming record only after that update completes. Random-ranked and overlap-ranked conditions share the same checkpoint, edit order, memory capacity, replay count, path-extraction workload, host hyperparameters, and execution schedule; only replay-candidate identity differs. PACE-FULL is a reference operating point, not a host-editor condition.

Component ablation relocated from the main paper. The ablation should be read together with the coupling controls rather than as a leaderboard of components. Removing replay causes the

Variant	Rel.	Gen.	Loc.	For.↓	Avg.
PACE-FULL	91.8	**83.6**	**96.9**	**4.8**	**90.8**
w/o activation path (path-free pipeline)	88.5	79.8	94.2	8.7	87.5
all active layers (w/o boundary restriction)	**92.1**	83.2	94.5	6.9	89.9
w/o replay	86.9	77.5	96.2	11.6	86.9
w/o null-space projection (≡ PACE-SOFT)	90.4	82.1	96.5	6.3	89.7
w/o hinge margin	89.6	81.1	95.6	7.4	88.8

Table 25: Component ablation on CounterFact, 10K continual edits. The *w/o activation path* row removes $\mathcal{P}_t$ from all four roles and replaces each with a capacity-matched path-free counterpart—fixed middle FFN layer $\ell = 14$, prompt-embedding replay, a fixed 328-coordinate block from the same editable universe, and the same key pools in that block—rather than disabling replay, the solve, or the projection. It therefore still replays, which is why it degrades *less* than *w/o replay*; the two rows are not nested. The *w/o null-space projection* row is the $P^{\ell}_{0,t} = I$ configuration of Table 8.

largest continual drop; removing the hinge margin lowers fit uniformly; and replacing boundary-only materialization by all-active-layer writing buys only 0.3 reliability while losing 2.4 locality and raising forgetting to 6.9. Removing the path from all four roles lowers Avg. from 90.8 to 87.5. A simpler localization rule does not substitute for the shared state: LGA reaches 89.7 Avg. versus 90.8 (Table 62). The 2×2 control in Table 17 isolates localization and replay, while Table 4 in the main paper tests the stricter same-state hypothesis.

B.3 FULL FIXED-BUDGET REPLAY-RANKING CONTROL

We evaluate replay ranking while holding replay capacity ($M = 64$), replay count ($K = 8$), solver, preservation keys, edit order, and all remaining hyperparameters fixed. The original core control reports full downstream metrics for five ranking policies; an expanded control adds recency, degree-centrality overlap, gradient-magnitude ranking, and a learned non-path logistic ranker, yielding nine active $K = 8$ policies in total. The no-replay condition sets $K = 0$ and is reported only as an external boundary anchor. Weighted Jaccard uses Equation B2; deployed PACE uses binary path Jaccard.

Replay ranking	Rel.↑	Gen.↑	Loc.↑	For.↓	Avg.↑	R@8↑
No replay	86.9	77.5	96.2	11.6	86.9	0.0
Random replay	88.0	79.0	96.3	9.5	87.8	12.5
Prompt-embedding replay	89.5	80.8	96.4	7.8	88.9	30.5
Restricted-key cosine	91.5	83.2	96.4	5.3	90.4	45.8
Path Jaccard	91.8	83.6	**96.9**	4.8	**90.8**	57.6
Weighted path Jaccard	**91.9**	**83.7**	96.8	**4.7**	**90.8**	**59.1**

Table 26: Core controlled replay-ranking experiment after 10K sequential CounterFact edits on GPT-J-6B. The five active policies use identical $M = 64$ and $K = 8$ budgets and differ only in ranking; no replay is a $K = 0$ boundary anchor. Full downstream metrics are reported here, while Table 27 expands the ranking audit to nine fixed-budget policies.

Across the nine $K = 8$ policies, higher R@8 is strongly associated with lower forgetting (Kendall's $\tau = -0.944$, $p = 4.96 \times 10^{-5}$). We use this rank statistic as the primary policy-level summary because it does not assume an affine relationship and is less sensitive than a least-squares fit to the specific policies included. The corresponding descriptive fit is

$$\text{For.} = 10.96 \left(1 - \frac{\text{R@8}}{100}\right) - 0.04, \qquad R^2 = 0.980,$$

for the nine fixed-budget policies. Including the separately identified $K = 0$ boundary anchor gives

$$\text{For.} = 11.63 \left(1 - \frac{\text{R@8}}{100}\right) - 0.40, \qquad R^2 = 0.984,$$

Replay condition	R@8↑	For.↓
No replay ($K = 0$)	0.0	11.6
Random	12.5	9.5
Recency	19.3	8.9
Prompt cosine	30.5	7.8
Degree-centrality overlap	37.1	6.9
Restricted-key cosine	45.8	5.3
Gradient-magnitude ranking	48.7	5.5
Learned logistic ranking[†]	53.2	5.0
Path Jaccard	57.6	4.8
Weighted path Jaccard	**59.1**	**4.7**

[†]The logistic ranker combines lexical overlap, prompt-semantic similarity, and restricted-key similarity and is fitted only on a disjoint development stream; it receives no path-overlap feature.

Table 27: Expanded fixed-budget replay-ranking control on GPT-J-6B after 10K CounterFact edits. All nine active policies share $M = 64$, $K = 8$, the solver, preservation mechanism, edit sequence, and remaining hyperparameters; only the ranking signal changes. No replay is reported separately as a $K = 0$ boundary anchor and is excluded from the primary fixed-budget rank statistic.

over all ten displayed conditions; this wider-range fit is descriptive only and is not evidence that the no-replay condition differs solely in ranking.

The expanded control also shows why the relationship should not be interpreted as deterministic. Gradient-magnitude ranking attains higher R@8 than restricted-key cosine (48.7 versus 45.8) but slightly greater forgetting (5.5 versus 5.3). Nevertheless, the policy-level association remains strong, and the two path-overlap variants occupy the strongest measured retention end of the fixed-budget ranking set. The learned logistic ranker reaches 53.2 R@8 and 5.0 forgetting despite combining lexical, semantic, and restricted-key signals on a disjoint development stream, but does not recover the $57.6/4.8$ and $59.1/4.7$ operating points of binary and weighted path overlap. Thus the retention benefit is not attributable merely to replay being present: under an identical replay budget and editing protocol, which retained mappings are selected is consequential. Binary Jaccard remains the deployed score because weighted Jaccard changes the rounded downstream Avg. negligibly while requiring additional magnitude state.

B.4 Matched-Reliability Selection and Target-Margin Sweeps

The matched-reliability control of Table 1 tests whether PACE's preservation advantage can be explained simply by weaker edit fitting. Among the reported controls, this is the only one that selects an operating-point hyperparameter. We therefore separate three resources: (i) the existing calibration pool used for activation normalization and construction of the editable universe; (ii) a development stream used only to select the target margin γ; and (iii) the final 10K CounterFact stream. The development and final streams share neither edit records nor subjects.

Before final-stream evaluation, we fix the candidate grid and matching tolerance

$$\gamma \in \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}, \qquad \delta_{\mathrm{rel}} = 0.4,$$

while every other PACE hyperparameter remains at its main-paper default:

$$\varrho = 0.04, \quad M = 64, \quad K = 8, \quad \beta = 0.99, \quad \eta = 1, \quad \lambda = 10^{-3}, \quad \gamma_r = 10^{-2}, \quad n_{\mathrm{base}} = 2000.$$

For an operating point h, let $\mathrm{Rel}_{\mathrm{dev}}(h)$ denote development reliability and $\mathrm{Rel}_{\mathrm{dev}}^{\mathrm{UE}}$ the UltraEdit value evaluated on the same development stream. Eligible points satisfy

$$\mathcal{H}_{\mathrm{match}} = \left\{ h : \left| \mathrm{Rel}_{\mathrm{dev}}(h) - \mathrm{Rel}_{\mathrm{dev}}^{\mathrm{UE}} \right| \le 0.4 \right\}. \tag{B3}$$

Among eligible points we select

$$h^\star = \arg \min_{h \in \mathcal{H}_{\mathrm{match}}} \mathrm{For}_{\mathrm{dev}}(h), \tag{B4}$$

breaking ties first by higher locality and then by the smaller γ. Development scores are averaged over three fixed edit permutations. The selection rule, candidate grid, and selected operating point are fixed before the final stream is evaluated.

No parameter is changed after observing final-stream Rel., Gen., Loc., or For. Had no development point satisfied the reliability window, or had the frozen point failed to remain near UltraEdit's reliability operating point on the final stream, we would have reported that mismatch rather than selecting a second point retrospectively. The full final-stream γ sweeps reported below are diagnostic characterizations of the fitting–preservation trade-off; they are not used to revise the prospectively selected operating point.

Importantly, matching moves PACE toward *stronger*, not weaker, fitting relative to its default configuration. On GPT-J, $\gamma = 1.0 \rightarrow 1.6$ raises reliability from 91.8 to 93.3; on LLaMA-3, $\gamma = 1.0 \rightarrow 1.4$ raises reliability from 93.0 to 93.7. Thus the matched control does not obtain a preservation advantage by weakening PACE's fit to the incoming edits.

The matched operating points are additionally evaluated over five fixed stream-order seeds using the same order set for UltraEdit, PACE-FULL, and PACE-MATCHED on each backbone (Appendix B.39). On GPT-J-6B, the corresponding standard deviations are 0.3–0.5 points for UltraEdit and 0.2–0.4 points for the PACE variants; on LLaMA-3-8B they are 0.2–0.4 points. The mean matched-fit differences in generalization, locality, and forgetting are therefore several times larger than the individual between-order standard deviations, whereas the reliability difference remains negligible (93.3 versus 93.4 on GPT-J and exactly 93.7 on LLaMA-3). We treat these five-order measurements as descriptive stability evidence rather than as a formal cross-method hypothesis test.

The matched-fit conclusion is therefore supported by both the prospectively frozen operating-point protocol and the multi-order evaluation: PACE remains at essentially the same reliability operating point as UltraEdit while retaining higher generalization and locality and lower forgetting. This addresses the weaker-fitting and single-favorable-order explanations without implying global Pareto dominance over all possible operating points.

Full target-margin sweeps. Only γ changes in Tables 28 and 29; the edit stream, path extractor, editable universe, replay memory and count, preservation pool, solver regularization, and evaluator remain fixed. These final-stream sweeps are reported to expose the shape of the trade-off and do not alter the frozen matched point.

γ	Rel.↑	Gen.↑	Loc.↑	For.↓	Avg.↑
1.0	91.8	83.6	96.9	4.8	90.8
1.2	92.3	83.8	96.6	5.1	90.9
1.4	92.8	84.0	96.1	5.5	91.0
1.6	93.3	84.0	95.5	6.0	90.9
1.8	93.6	83.8	94.9	6.5	90.8
2.0	93.8	83.5	94.2	7.0	90.5

Table 28: Measured final-stream target-margin sweep on GPT-J-6B after 10K sequential CounterFact edits. $\gamma = 1.0$ is the default point; the prospectively development-selected and frozen matched point is $\gamma = 1.6$.

The GPT-J sweep shows a smooth fitting–preservation trade-off. Increasing γ from 1.0 to 2.0 raises reliability from 91.8 to 93.8, while locality decreases from 96.9 to 94.2 and forgetting increases from 4.8 to 7.0. The frozen $\gamma = 1.6$ point reaches 93.3 reliability, 84.0 generalization, 95.5 locality, and 6.0 forgetting.

LLaMA-3 exhibits the same monotone fitting–preservation movement. Increasing γ from 1.0 to 1.4 moves PACE from its default 93.0 reliability to exact reliability parity with UltraEdit at 93.7, while retaining 85.2 generalization, 96.6 locality, and 5.0 forgetting. The continuation of the sweep beyond the matched point exposes the same trade-off rather than identifying an alternative headline operating point.

Cross-class near-reliability stress test with LightEdit. The primary matched-fit experiment uses UltraEdit because both methods are strict-online parametric editors and PACE's operating point is

γ	Rel.↑	Gen.↑	Loc.↑	For.↓
1.0	93.0	85.0	97.1	4.5
1.2	93.4	85.2	96.9	4.8
1.4	93.7	85.2	96.6	5.0
1.6	94.0	85.1	96.2	5.4
1.8	94.2	84.9	95.8	5.8
2.0	94.4	84.6	95.3	6.3

Table 29: Measured final-stream target-margin sweep on LLaMA-3-8B after 10K sequential CounterFact edits. $\gamma = 1.0$ is the default point; the prospectively development-selected and frozen matched point is $\gamma = 1.4$.

selected prospectively on a disjoint development stream. LightEdit belongs to a different intervention and state class: it leaves the base weights unchanged, retains a growing external edit store, and invokes retrieval plus edit-aware decoding at query time. We therefore treat it as a descriptive cross-class stress test rather than a matched operating-point comparison. On LLaMA-3 CounterFact, LightEdit's measured reliability happens to lie only 0.3 points from PACE's *default* point (PACE 93.0 versus LightEdit 93.3); no PACE hyperparameter is selected or changed for this comparison.

Method	Rel.↑	Gen.↑	Loc.↑	For.↓
LightEdit	93.3	80.9	**98.1**	5.2
PACE-FULL ($\gamma = 1.0$)	93.0	**85.0**	97.1	**4.5**
Δ(PACE–LightEdit)	-0.3	$+4.1$	-1.0	-0.7

Table 30: Descriptive near-reliability stress test on LLaMA-3-8B CounterFact. The measured operating points differ by only 0.3 reliability points. Unlike the UltraEdit matched control, this comparison uses PACE's unchanged default $\gamma = 1.0$ configuration and involves no development-based operating-point selection.

At these measured operating points, PACE gains 4.1 generalization points and reduces forgetting by 0.7 points, while LightEdit retains a 1.0-point locality advantage. Thus the retention pattern observed against UltraEdit is also present against a non-parametric external-memory comparator at this particular CounterFact operating point, although the intervention classes and state representations differ substantially.

We do not construct a corresponding matched headline on ZsRE. At 10K, LightEdit occupies a substantially different measured point there ($95.1/92.5/99.6/4.6$), and moving PACE along its target-margin sweep after observing that result would amount to post-hoc operating-point selection. We therefore report the 10K matrix directly in Table 35 and, without retuning either default configuration, extend both methods to 50K and 100K in Appendix B.10.

B.5 Shared-State Construction and Distinct-State Audit

The main shared-state control changes state agreement while keeping the underlying extraction family, sparsity budget, edit target, and downstream solver fixed. Every state used by a condition is extracted from the same pre-write model, so no role observes a post-update representation. The canonical prompt remains the actual editing input. Alternative views are used *only* for path extraction, in fixed order: (i) a benchmark-provided paraphrase when available; (ii) a fixed factual-query instruction prefix; (iii) a second fixed instruction-style rendering; and (iv) a fixed semantically neutral contextual rendering. No target object, relation, or factual content is added or removed, and the rendering rules are frozen before the test stream. This control tests state coherence; it is not used to make a claim about extraction-time efficiency.

For each view q, let $P_t^{(q)} = \mathrm{Path}(x_t^{(q)})$. FOUR-STATE assigns different extracted states to localization, replay, solving, and preservation,

$$P_t^{\mathrm{loc}} = P_t^{(1)}, \quad P_t^{\mathrm{rep}} = P_t^{(2)}, \quad P_t^{\mathrm{solve}} = P_t^{(3)}, \quad P_t^{\mathrm{pres}} = P_t^{(4)}.$$

TWO-STATE reuses one state for localization/solve and another for replay/preservation. The actual edit prompt, target object, candidate-token set, target margin, Jacobian, and evaluation queries remain unchanged.

Two agreement statistics are intentionally different. The main table reports

$$\overline{J}_{\mathrm{role}} = \frac{1}{6} \sum_{a<b} J(P_t^{(a)}, P_t^{(b)}),$$

over the six *role-assignment* pairs, with a pair of roles reusing the same extracted state contributing 1. Table 31 instead reports *distinct-state Jaccard*, averaged only over different extracted paths.

Condition	Unique states	Distinct-state Jac.↑	Boundary ±1 agr.↑	Support ratio
Shared-Canonical	1	–	100.0	1.00
Shared-View	1	–	100.0	1.03
Two-State	2	0.62	88.0	1.04
Four-State	4	0.59	86.0	1.04

Table 31: Measured state-quality audit for the shared-state experiment. Distinct-state Jaccard averages only over *different* extracted paths and is therefore undefined for the one-state conditions. It is not the same statistic as $\overline{J}_{\mathrm{role}}$ in Table 4, which averages all six role-assignment pairs and counts reused identical states as 1.

For TWO-STATE, the 0.62 entry is the Jaccard between its two unique paths. Because each is reused by two roles, the role-assignment statistic is

$$\frac{2(1) + 4(0.62)}{6} = 0.7467 \approx 0.75,$$

exactly the value in Table 4. For FOUR-STATE, all four role states are distinct, so the two averages coincide at approximately 0.59.

The shared-state criterion was pre-specified on full-precision aggregates, not the rounded Avg. cells: SHARED-VIEW had to remain within 0.5 Avg. points of SHARED-CANONICAL, while FOUR-STATE had to lose at least 1.0 Avg. point and add at least 0.6 forgetting points relative to SHARED-VIEW. The measured component scores give

$$\frac{91.8 + 83.6 + 96.9}{3} - \frac{91.3 + 83.1 + 96.6}{3} = 0.433\ldots,$$

and FOUR-STATE is 1.8 Avg. points lower and 2.3 forgetting points higher than SHARED-VIEW; both conditions are satisfied.

Interpretation scope. SHARED-VIEW controls the largest view-quality confound: replacing the canonical prompt view by one alternative view while preserving a single state costs only 0.433 Avg. points, whereas breaking state identity across four roles costs 1.8 points and adds 2.3 forgetting points. The audit also shows similar support size and high boundary agreement. We therefore interpret the experiment as evidence that state coherence is load-bearing under the matched extraction family, not as a claim that every alternative prompt view is equally informative.

B.6 PROSPECTIVE INTERFERENCE PREDICTION ON LLAMA-3-8B

The main cross-backbone table is in-stream. We therefore repeat the stricter subject-disjoint prospective protocol on LLaMA-3-8B without re-tuning any signal definition, overlap score, interference threshold, or replay budget. All similarities are computed before the incoming edit is written.

The prospective effect attenuates relative to the in-stream LLaMA-3 measurement ($r_s = 0.39$, AUROC 74.0, R@8 58.3), but the ordering is preserved. Weighted path Jaccard remains strongest on all three criteria, followed by unweighted path Jaccard and restricted-key similarity. This extends the prospective evidence beyond GPT-J without changing the signal or evaluation definition.

Signal	Spearman r_s ↑	AUROC↑	R@8↑
Random score	0.00	50.0	12.5
Lexical overlap	0.09	54.9	19.5
Prompt-embedding cosine	0.15	58.7	26.3
Restricted-key cosine	0.23	64.0	39.0
Path Jaccard	0.30	68.9	49.8
Weighted path Jaccard	**0.32**	**70.1**	**51.6**

Table 32: Measured prospective interference prediction on a subject-disjoint LLaMA-3-8B CounterFact stream. Signal definitions are frozen before evaluation.

B.7 Realized-Write-Matched Support Intervention on LLaMA-3-8B

The intervention uses the same $I(i,t)$ and severe-event definitions as Appendix B.1.

The GPT-J overlap intervention manipulates support at a fixed requested margin and coordinate budget, but its forced-low arm can move the update away from coordinates the incoming edit naturally engages. We therefore repeat the intervention on LLaMA-3-8B while additionally matching the *realized* incoming target-margin gain. This is a diagnostic control only; it does not modify the deployed PACE algorithm.

Pair and support construction. We evaluate $n = 1{,}000$ unique prior–incoming CounterFact pairs sampled from the same 10K LLaMA-3 stream used by the predictive analysis. The sampling frame is the pre-write bounded replay population defined in Appendix B.1: incoming positions are sampled uniformly without replacement from positions admitting at least one feasible co-resident predecessor, and one prior record is then sampled uniformly from the feasible records in $\mathcal{M}_{t-1}$. Eligibility uses only memory membership and intervention feasibility; it does not use natural overlap magnitude, lexical or semantic similarity, realized target gain, $I(i,t)$, severe-event status, or any prior-edit post-write outcome. Pair identities and pre-incoming model states are frozen before the three arms are run. For every pair, all arms share the incoming edit, target, requested margin, boundary layers, and per-layer support-size budget. With P_i the prior support and P_t the natural incoming support, NATURAL uses P_t; FORCED-LOW chooses a capacity-matched support minimizing feasible overlap with P_i inside the same editable universe; and FORCED-HIGH chooses a capacity-matched support maximizing feasible overlap subject to the same boundary and support-size constraints.

Realized-write matching. After the support-specific sparse solve, arm c receives a scalar $\alpha_c > 0$ selected by a one-dimensional search using only the incoming edit. Let $\Delta m_c^{\text{incoming}}(\alpha)$ be the realized change in the incoming target-vs-old-object margin after scaling that arm by α. We choose the feasible scalar minimizing

$$\left|\Delta m_c^{\text{incoming}}(\alpha) - \Delta m_{\text{natural}}^{\text{incoming}}\right|$$

subject to

$$\frac{\Delta m_c^{\text{incoming}}(\alpha)}{\Delta m_{\text{natural}}^{\text{incoming}}} \in [0.95, 1.05].$$

The search never observes $I(i,t)$, the prior edit's post-write margin, or severe-event status. The prespecified replication criterion requires the target-gain ratio to remain in this interval and both mean interference and severe-event rate to increase strictly from forced-low to natural to forced-high.

Condition	Mean ω	Target-gain ratio	$\|\Delta\|_F/\|\Delta_{\text{natural}}\|_F$	Mean $I(i,t)$	Severe rate
Forced-low	0.03	0.99	1.04	0.11	4.2%
Natural	0.18	1.00	1.00	0.25	9.6%
Forced-high	0.55	1.01	0.98	0.51	20.6%

Table 33: Audit of the LLaMA-3-8B support intervention on $n = 1{,}000$ fixed pairs. Target-gain and update-norm ratios verify that the manipulated arms do not merely weaken the incoming write. Matching observes only the incoming edit; prior-edit damage is never exposed to the search.

Paired uncertainty analysis. Because all three intervention arms are evaluated on the same $n = 1{,}000$ fixed prior–incoming pairs, we quantify uncertainty using *within-pair contrasts* rather than treating the intervention arms as independent samples. For pair i under condition $c \in \{\mathrm{low}, \mathrm{nat}, \mathrm{high}\}$, let

$$I_i^c$$

denote the measured one-step interference, and let

$$S_i^c = \mathbf{1}[\text{pair } i \text{ is a severe event under condition } c]$$

denote the corresponding severe-event indicator.

We compute paired nonparametric bootstrap confidence intervals using $B = 20{,}000$ replicates. In each replicate, $n = 1{,}000$ pair indices are sampled with replacement, and the *same resampled indices are applied jointly to all three intervention arms*. This preserves the matched experimental design: the incoming edit, prior edit, pre-intervention model state, requested target margin, boundary layers, and support-size budget remain aligned across conditions for every resampled pair.

For adjacent conditions a and b, bootstrap replicate r computes

$$\Delta I_{b-a}^{(r)} = \frac{1}{n} \sum_{i \in \mathcal{B}_r} \left(I_i^b - I_i^a \right), \qquad \Delta S_{b-a}^{(r)} = \frac{100}{n} \sum_{i \in \mathcal{B}_r} \left(S_i^b - S_i^a \right),$$

where $\mathcal{B}_r$ denotes the bootstrap multiset of pair indices and ΔS is expressed in percentage points (pp). Two-sided 95% confidence intervals are given by the 2.5th and 97.5th percentiles of the corresponding paired-bootstrap distribution.

Pair identities are fixed before post-intervention interference is evaluated. Neither pair selection, support construction, nor realized-write matching uses $I(i, t)$, severe-event status, the prior edit's post-write margin, or any other post-intervention damage quantity. Thus, the uncertainty analysis is applied only after outcomes for all three intervention arms have been obtained and does not condition pair selection on the measured interference outcome.

Paired contrast	Observed difference	95% paired-bootstrap CI
Mean I: Natural − Forced-Low	$+0.14$	$[+0.117, +0.164]$
Mean I: Forced-High − Natural	$+0.26$	$[+0.227, +0.295]$
Severe rate: Natural − Forced-Low	$+5.4$ pp	$[+3.9, +6.9]$ pp
Severe rate: Forced-High − Natural	$+11.0$ pp	$[+8.9, +13.2]$ pp

Table 34: Paired uncertainty analysis for the LLaMA-3-8B support intervention over $n = 1{,}000$ experimentally evaluated prior–incoming pairs. All three conditions use identical pair identities and corresponding pre-intervention model states. Confidence intervals are computed by jointly resampling pair identities across the three intervention arms, thereby preserving their within-pair dependence. Positive contrasts indicate greater prior-edit damage as incoming-support overlap increases.

The intervention also satisfies the pre-specified realized-write matching criterion. Relative to the natural arm, the measured target-gain ratios are 0.99, 1.00, and 1.01 for FORCED-LOW, NATURAL, and FORCED-HIGH, respectively, all within the required $[0.95, 1.05]$ interval. The corresponding update-norm ratios are 1.04, 1.00, and 0.98 (Table 33). Hence, increasing support overlap does not track a progressively stronger realized incoming write.

Across the 1,000 matched pairs, mean one-step interference increases monotonically from

$$0.11 \to 0.25 \to 0.51$$

for FORCED-LOW, NATURAL, and FORCED-HIGH, respectively. The adjacent-arm increases are $+0.14$ for NATURAL−FORCED-LOW and $+0.26$ for FORCED-HIGH−NATURAL. Their paired 95% confidence intervals,

$$[+0.117, +0.164] \qquad \text{and} \qquad [+0.227, +0.295],$$

are both strictly above zero.

Severe-event frequency exhibits the same monotonic pattern,

$$4.2\% \to 9.6\% \to 20.6\%.$$

The corresponding paired increases are $+5.4$ percentage points from FORCED-LOW to NATURAL and $+11.0$ percentage points from NATURAL to FORCED-HIGH, with 95% paired-bootstrap confidence intervals of

$$[+3.9, +6.9] \text{ pp} \qquad \text{and} \qquad [+8.9, +13.2] \text{ pp},$$

respectively.

All four paired contrasts are therefore strictly above zero, and both outcomes obey the same ordering,

$$\text{FORCED-LOW} < \text{NATURAL} < \text{FORCED-HIGH}.$$

Because realized target gain and update norm remain closely matched, the ordering disfavors the explanation that the low-overlap arm is safer merely because it realizes a weaker incoming write. Under this intervention, forcing greater overlap between incoming and prior edited FFN support increases both mean one-step interference and severe-event frequency. The claim is deliberately local: this experiment identifies a support-mediated interference channel inside the edited FFN weights, not all interaction pathways of the full Transformer.

B.8 LLAMA-3-8B ZSRE MATCHED-RELIABILITY REPLICATION

Before applying the matched-fit control, we complete the LLaMA-3-8B ZsRE baseline matrix under the same 10K sequential evaluator used by PACE. This makes the non-dominant raw operating point explicit.

Method	Rel.↑	Gen.↑	Loc.↑	For.↓	Time/edit↓
AlphaEdit	94.4	87.6	89.3	9.1	**0.76**
LightEdit	**95.1**	**92.5**	**99.6**	4.6	0.91
UltraEdit	94.4	85.6	95.0	6.3	0.81
RLSEdit	94.7	88.5	92.1	5.6	0.80
PACE-FULL	93.5	85.6	97.4	**4.4**	0.90

Table 35: Unified 10K LLaMA-3-8B ZsRE comparison. Every row is measured under the same backbone, evaluator, prompt format, hardware, and sequential edit stream. LightEdit gives the strongest raw fit/locality operating point, while PACE gives the lowest forgetting; the table is intentionally reported without claiming universal dominance.

The 10K matrix exposes two distinct operating regimes. LightEdit leads raw reliability, generalization, and locality ($95.1/92.5/99.6$), whereas PACE has the lowest forgetting (4.4). Relative to the closest recursive parametric baseline, RLSEdit, PACE trades lower raw fit (-1.2 Rel., -2.9 Gen.) for $+5.3$ locality and -1.2 forgetting. This supports the intended preservation role while delimiting the 10K claim: LightEdit occupies the stronger raw-fit/locality point, whereas PACE minimizes forgetting. We do not extrapolate this ordering across stream length; the uninterrupted 50K and 100K ZsRE continuation in Appendix B.10 shows that the frontier changes with store size.

We next repeat the CounterFact matched-reliability protocol against UltraEdit. Calibration, development, and final ZsRE streams are separated so that the development and final streams share neither records nor subjects. Before final evaluation we fix

$$\gamma \in \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}, \qquad \delta_{\text{rel}} = 0.4,$$

and keep all other LLaMA-3 PACE hyperparameters fixed. Among development points within 0.4 reliability of UltraEdit, we select the point with lowest forgetting, breaking ties by higher locality and then smaller γ, and freeze it before the 10K final stream. The selected point is $\gamma = 1.4$.

At the frozen point PACE is within 0.2 reliability points of UltraEdit while retaining $+0.4$ generalization, $+2.0$ locality, and -1.5 forgetting. Thus the matched-fit preservation separation is not specific to CounterFact.

Method	Rel.↑	Gen.↑	Loc.↑	For.↓
UltraEdit	94.4	85.6	95.0	6.3
PACE-FULL (default)	93.5	85.6	**97.4**	**4.4**
PACE-MATCHED	94.2	**86.0**	97.0	4.8

Table 36: LLaMA-3-8B ZsRE matched-reliability result after 10K sequential edits. The PACE operating point is selected on the disjoint development stream and frozen before final evaluation.

γ	Rel.↑	Gen.↑	Loc.↑	For.↓	Avg.↑
1.0	93.5	85.6	97.4	4.4	92.2
1.2	93.9	85.8	97.2	4.6	92.3
1.4	94.2	86.0	97.0	4.8	92.4
1.6	94.5	85.9	96.9	5.1	92.4
1.8	94.8	85.7	96.5	5.5	92.3
2.0	95.0	85.4	96.0	6.0	92.1

Table 37: Measured LLaMA-3-8B ZsRE target-margin sweep. Increasing γ raises reliability while gradually spending locality and retention; the development-selected matched point is $\gamma = 1.4$.

B.9 SHARED-STATE REPLICATION ON LLAMA-3-8B

We repeat the four-condition coupling test using the same path-view construction, extraction family, sparsity budget, edit target, solver, and pre-specified criteria as on GPT-J. Alternative prompts are used only to extract state; the canonical edit prompt, target, Jacobian, margin, and evaluation queries remain fixed, and every state used by a condition is extracted from the same pre-write model. As in the GPT-J audit, this experiment tests the effect of state identity rather than extraction-time efficiency.

Condition	$\overline{J}_{\mathrm{role}}$	Rel.↑	Gen.↑	Loc.↑	For.↓	Avg.↑
Shared-Canonical	1.00	93.0	85.0	97.1	4.5	91.7
Shared-View	1.00	92.7	84.7	96.9	4.8	91.4
Two-State	0.77	91.9	83.8	96.2	5.7	90.6
Four-State	0.61	91.1	82.9	95.5	6.7	89.8

Table 38: LLaMA-3-8B shared-state replication after 10K CounterFact edits. $\overline{J}_{\mathrm{role}}$ averages the six role-assignment pairs and counts reused identical states as one.

Shared-View stays within 0.3 Avg. points of Shared-Canonical. Relative to Shared-View, Four-State loses 1.6 Avg. points and adds 1.9 forgetting points, exceeding the pre-specified $1.0/0.6$ thresholds.

For auditability we again separate role-assignment agreement from agreement among distinct extracted states. Two-State has distinct-state Jaccard 0.65, which implies

$$\overline{J}_{\mathrm{role}} = \frac{2(1) + 4(0.65)}{6} = 0.7667 \approx 0.77.$$

For Four-State all four states are distinct, so the two averages coincide.

Condition	Unique states	Distinct-state Jac.↑	Boundary ± 1 agr.↑	Support ratio
Shared-Canonical	1	–	100.0	1.00
Shared-View	1	–	100.0	1.02
Two-State	2	0.65	89.0	1.03
Four-State	4	0.61	87.0	1.03

Table 39: State-quality audit for the LLaMA-3 shared-state replication. Distinct-state Jaccard averages only different extracted paths and is therefore not numerically the same statistic as $\overline{J}_{\mathrm{role}}$.

B.10 Long-Stream Scaling and External-Memory Trade-offs on LLaMA-3-8B

PACE on the long-horizon CounterFact stress-test stream. We continue the same executed LLaMA-3-8B CounterFact stress-test stream used for the 10K and 17.2K checkpoints. The stream follows the successive-full-permutation construction of Appendix B.1: pass boundaries do not reset the model or bounded editor state, revisited canonical records retain their original targets, and no hyperparameter is re-tuned after the 10K configuration is fixed. The 25K–100K checkpoints therefore count sequential update operations rather than distinct facts. Long-horizon For. is evaluated exhaustively over distinct canonical records using the first-materialization reference in Equation B1; repeated occurrences are deduplicated by canonical record identity.

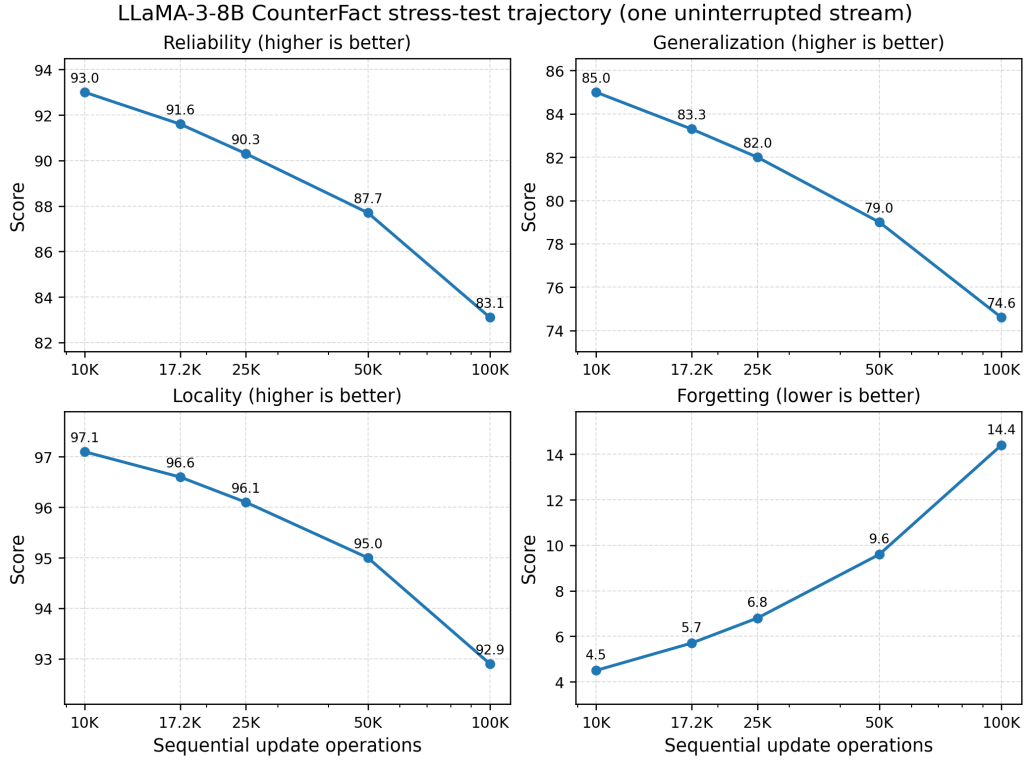

Figure 4: Measured LLaMA-3-8B CounterFact trajectory from one uninterrupted stress-test stream. Beyond one complete CounterFact pass, update operations continue through successive seeded full-dataset permutations with unchanged record targets; neither the model nor bounded editor state is reset. For. is computed over unique canonical records rather than repeated operation instances. The 10K and 17.2K checkpoints coincide with the previously reported measurements, and later checkpoints continue the same model state.

Figure 4 shows progressive degradation with stream length rather than an abrupt collapse at a single checkpoint. Locality is the most stable reported axis, decreasing from 97.1 at 10K to 92.9 at 100K, while generalization declines from 85.0 to 74.6. Reliability follows the same overall trend, dropping from 93.0 to 83.1, and forgetting rises from 4.5 to 14.4. The trend continues through 100K update operations without an abrupt single-checkpoint failure. These measurements provide an empirical long-horizon stress test of one uninterrupted editor state under repeated-update conditions; they are not extrapolated as an asymptotic stability guarantee.

Cross-class scaling against LightEdit. The strongest measured cross-class challenge to PACE occurs on ZsRE. At 10K edits under the unified LLaMA-3-8B evaluation, LightEdit reaches 95.1/92.5/99.6/4.6 Rel./Gen./Loc./For., compared with 93.5/85.6/97.4/4.4 for PACE-Full. LightEdit therefore has the stronger reliability–generalization–locality operating point at 10K, while PACE has slightly lower forgetting.

We continue both methods along identical prefixes of the same uninterrupted ZsRE update sequence and evaluate checkpoints at 10K, 50K, and 100K operations. No checkpoint resets the corresponding method state, and no hyperparameter is re-tuned after the 10K configuration is fixed. The 10K rows reproduce Table 35; later rows continue the corresponding method states. We report these checkpoint indices as sequential update counts and do not interpret them as a claim about distinct-fact cardinality.

Edits	Method	Rel.↑	Gen.↑	Loc.↑	For.↓
10K	LightEdit	**95.1**	**92.5**	**99.6**	4.6
	PACE-FULL	93.5	85.6	97.4	**4.4**
50K	LightEdit	**90.2**	**84.6**	**99.1**	**9.0**
	PACE-FULL	88.3	79.8	95.3	9.2
100K	LightEdit	82.6	73.9	**98.6**	16.0
	PACE-FULL	**83.8**	**75.3**	93.2	**14.1**

Table 40: Measured long-horizon LLaMA-3-8B results on the uninterrupted ZsRE update sequence. LightEdit leads Rel./Gen./Loc. at 10K and all four reported metrics at 50K. At 100K, PACE has higher reliability and generalization and lower forgetting, while LightEdit retains higher locality. These are measured operating-point comparisons under the fixed configurations, not claims of uniform or asymptotic dominance.

The measured frontier changes with scale. At 10K, LightEdit leads by 1.6 reliability and 6.9 generalization points; at 50K the corresponding gaps are 1.9 and 4.8 points. At 100K, PACE reaches 83.8 reliability and 75.3 generalization versus $82.6/73.9$ for LightEdit, giving differences of $+1.2$ Rel. and $+1.4$ Gen.; PACE also has 1.9 points less forgetting. LightEdit remains substantially stronger in locality and retains a 5.4-point advantage at 100K. We therefore interpret these results as a scale-dependent fit–locality–retention frontier rather than universal dominance by either method. Because the comparison follows one uninterrupted stream per method, we report effect sizes directly and do not derive cross-method significance from PACE-only edit-order variability.

Selector competition accompanies the ZsRE crossover. To diagnose one system-level factor associated with LightEdit's scaling behavior, we log its method-native selector. For query q, let $\mathcal{E}^\star(q)$ denote the set of stored entries corresponding to the same canonical edit identity as q. We define

$$s^+(q) = \max_{e \in \mathcal{E}^\star(q)} s(q,e), \qquad s^-(q) = \max_{e \notin \mathcal{E}^\star(q)} s(q,e), \qquad m_{\text{sel}}(q) = s^+(q) - s^-(q),$$

where $s(\cdot,\cdot)$ is the LightEdit selector score. Top-1 selector accuracy counts retrieval of any entry in $\mathcal{E}^\star(q)$ as correct.

Figure 5 shows a clear competitive pattern as the LightEdit store grows. From 10K to 100K edits, the mean correct-edit score remains 0.84 at the reported precision, whereas the strongest-negative score rises from 0.64 to 0.79. The selector margin therefore contracts from 0.20 to 0.10 to 0.05. Over the same checkpoints, top-1 selector accuracy falls from 95.8% to 89.0% to 80.2%, and downstream generalization falls from 92.5% to 84.6% to 73.9%. These measurements are consistent with increasing competition among stored edits, but they remain observational diagnostics rather than a causal attribution of the quality crossover.

PACE exhibits a different scaling signature: it does not perform per-query selection over a growing historical edit store, although sequential parameter updates remain exposed to accumulated interference. Its ZsRE generalization decreases from 85.6 at 10K to 75.3 at 100K, a 10.3-point reduction versus 18.6 points for LightEdit over the same reported checkpoints. We use this contrast only to characterize the two measured operating regimes; it does not identify the causal source of either method's degradation.

Query-time deployment scaling against LightEdit. The quality and selector trajectories above do not by themselves characterize serving cost. In a separate matched serving benchmark, we measure end-to-end query latency after $N \in \{1\text{K}, 10\text{K}, 50\text{K}, 100\text{K}\}$ processed edits. LightEdit timing

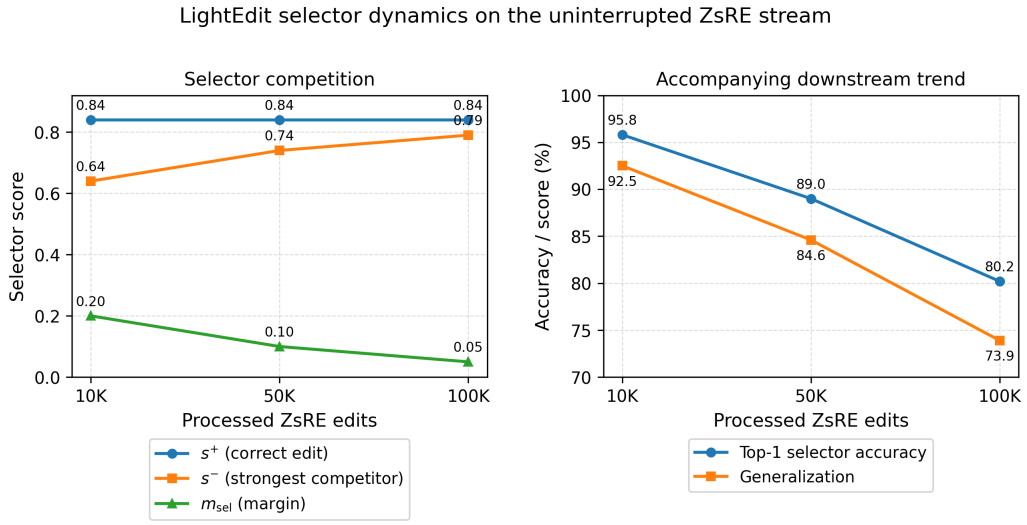

Figure 5: Measured LightEdit selector dynamics on the uninterrupted ZsRE stream. **Left:** the mean score assigned to the correct edit remains 0.84 at the reported precision, while the strongest competing score increases from 0.64 to 0.79, contracting the selector margin from 0.20 to 0.05. **Right:** over the same checkpoints, top-1 selector accuracy decreases from 95.8% to 80.2%, while downstream generalization decreases from 92.5% to 73.9%. These quantities are diagnostic and observational; they are not used to identify a causal explanation for the cross-method quality crossover.

includes query embedding, retrieval, edit-aware selection, and decoding-time suppression; PACE timing includes ordinary generation from the edited model. Generation length, batch size, precision, warm-up, and A6000 hardware are held fixed. This benchmark measures deployment cost independently of the quality analysis and is not used to explain the ZsRE crossover.

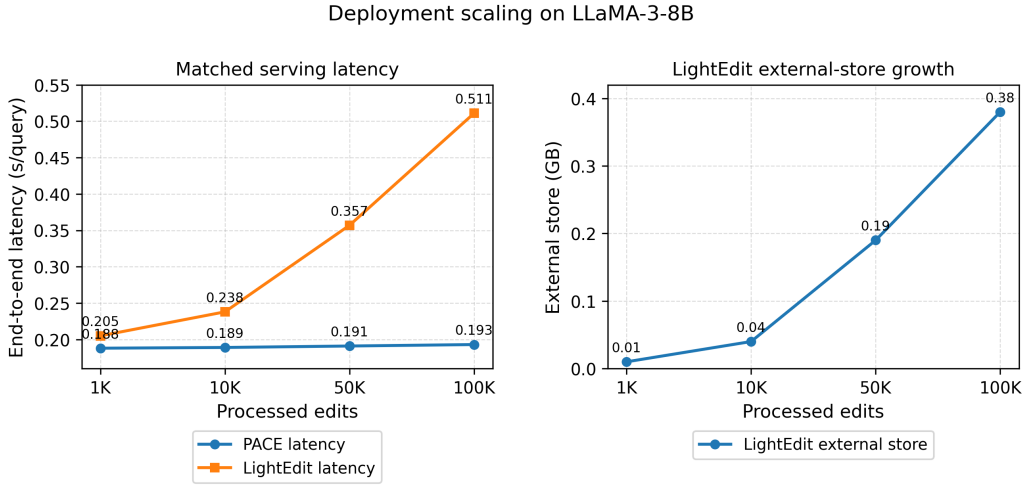

Figure 6: Deployment scaling on LLaMA-3-8B. **Left:** matched end-to-end query latency for PACE and LightEdit under the same LLaMA-3 serving setup. **Right:** LightEdit's external retrieval store, which grows with the edit stream and excludes its fixed selector. A matched LLaMA-3 resident-memory measurement for PACE is not present in the completed experimental record, so no cross-method memory comparison is made here; the GPT-J-6B bounded-state profile is reported separately in Table 75.

Figure 6 shows that the two methods have similar serving cost at small scale but diverge as the edit stream grows. LightEdit increases from 0.205 to 0.511 s/query between the 1K and 100K

checkpoints, whereas PACE remains nearly flat at $0.188 \rightarrow 0.193$ s/query, yielding a measured $2.65\times$ latency ratio at 100K. Over the same range, LightEdit's external store grows from 0.01 to 0.38 GB. We do not report a matched LLaMA-3 resident-memory measurement for PACE and therefore make no cross-method memory-efficiency claim.

Taken together, these measurements expose a scale-dependent cross-class trade-off without implying universal dominance. LightEdit leads Rel./Gen./Loc. at 10K and all four reported metrics at 50K; at 100K, PACE has higher Rel./Gen. and lower For., while LightEdit retains substantially higher Loc. The selector audit is consistent with increasing competition in LightEdit's store but remains diagnostic, and the separate serving benchmark shows higher measured query latency for LightEdit at 100K while its external store grows from 0.01 to 0.38 GB. The quality crossover is empirical for the evaluated LLaMA-3-8B ZsRE stream, and the latency comparison is empirical for the matched serving workload; neither result is an asymptotic claim about external-memory editing.

B.11 Continual Editing on RIPE

CounterFact and ZsRE mainly evaluate counterfactual factual updates and relation editing. We additionally evaluate PACE on 1K sequential RIPE edits under our online parametric evaluator. The literature values available through the LightEdit source table are not protocol matched: that source reports a 1K GPT-J stream sampled from the test splits of ZsRE, CounterFact, and RIPE. We therefore separate the source-native reference rows from the PACE RIPE-only rows and make no numerical ranking claim across the two panels.

Panel	Method	Rel.	Gen.	Loc.	Avg.
Source-native mixed 1K stream	ROME Meng et al. (2022) *[NeurIPS'22]*	0.54	0.31	0.54	0.46
	MEMIT Meng et al. (2023) *[ICLR'23]*	52.29	34.18	41.07	42.51
	GRACE Hartvigsen et al. (2023) *[NeurIPS'23]*	31.08	23.72	100.00	51.60
	AlphaEdit Fang et al. (2025) *[ICLR'25]*	99.30	40.90	79.00	73.07
	LTE Jiang et al. (2024) *[ACL'24]*	99.76	87.48	82.96	90.07
	RECIPE Chen et al. (2024) *[EMNLP'24]*	58.13	48.34	39.83	48.77
	LightEdit Jung et al. (2026) *[ACL'26]*	99.42	96.61	99.88	98.64
PACE evaluator RIPE-only 1K	PACE-Soft	91.9	87.6	96.7	92.1
	PACE-Full	93.4	89.3	97.4	93.4

Table 41: RIPE-related continual-editing audit. The upper panel reproduces percentages from the source-reported GPT-J table in LightEdit, whose 1K stream is sampled across the ZsRE, CounterFact, and RIPE test splits. The lower panel reports PACE on our RIPE-only 1K online parametric stream. Because stream composition and evaluator differ, the panels are contextual rather than head-to-head and no cross-panel ranking is implied.

The PACE panel provides within-method evidence that the retention/locality profile transfers to RIPE-style updates. The source-native panel provides literature context only; it is not used to claim superiority over LightEdit or the other source methods because those values come from a different mixed-stream protocol.

B.12 Unified-Protocol Single-Edit and Runtime Audit

Table 5 already gives the complete GPT-J 10K head-to-head comparison. Here we add the corresponding single-edit operating points and make the runtime boundary explicit, using the same backbone, evaluator, prompt format, hardware, and edit-stream settings. AlphaEdit is a parameter-modifying baseline, LightEdit a parameter-preserving external-memory baseline, and UltraEdit a recent scalable parametric editor. On ZsRE, LightEdit has the strongest raw reliability/generalization/locality point at the reported GPT-J 10K operating point while PACE has the

lowest forgetting; against parameter-editing baselines, PACE retains the larger locality/retention separation. The completed LLaMA-3 10K counterparts are in Tables 10 and 35, with the separate 10K–100K cross-class scaling audit in Appendix B.10.

Dataset / setting	Method	Rel.	Gen.	Loc.	For.↓	Time↓
CounterFact, single	AlphaEdit (rerun)	98.7	88.1	95.4	0.0	3.42
	LightEdit (rerun)	99.0	74.8	97.6	0.0	0.81
	PACE-FULL	98.1	89.7	97.8	0.0	0.74
CounterFact, 10K	AlphaEdit (rerun)	90.9	78.6	88.9	12.4	3.58
	LightEdit (rerun)	92.2	76.4	97.5	5.6	0.88
	UltraEdit (rerun)	93.4	82.2	93.6	7.2	0.68
	RLSEdit (rerun)	92.6	82.9	91.4	6.3	0.69
	PACE-FULL	91.8	83.6	96.9	4.8	0.74
ZsRE, single	AlphaEdit (rerun)	97.9	88.5	95.1	0.0	3.31
	LightEdit (rerun)	99.2	93.6	99.7	0.0	0.78
	PACE-FULL	96.4	90.2	98.0	0.0	0.71
ZsRE, 10K	AlphaEdit (rerun)	94.0	86.0	86.5	11.6	3.47
	LightEdit (rerun)	94.8	91.1	99.4	5.1	0.84
	RLSEdit (rerun)	94.1	87.3	90.8	6.1	0.67
	PACE-FULL	92.3	84.1	97.2	4.7	0.71

Table 42: Completed unified-protocol reruns on GPT-J-6B under our evaluator. All displayed baseline and PACE values are measured under the same backbone, evaluator, prompt format, hardware, and edit-stream setting. Time is seconds per edit on the same A6000 setup. A unified-protocol GPT-J ZsRE UltraEdit result is not present in the completed experimental record and is therefore left absent rather than inferred from another backbone. Agreement-based locality is systematically higher than, and not numerically comparable to, the probability-based specificity reported in the source-audit tables (Table 8); reconciling that difference is precisely why the unified rerun is necessary. Forgetting is defined over previously edited facts; because no prior edits exist in the single-edit setting, For. is set to 0.0 by convention and excluded from single-edit comparisons.

Runtime scope. Time/edit measures edit insertion and editor-state maintenance on the shared A6000 setup; downstream answer generation is excluded for every method. For LightEdit this column therefore covers insertion/index maintenance but not the per-query selector and decoding-time suppression path. That serving cost is measured separately in Figure 6. We omit a single peak-memory column here because PACE/AlphaEdit resident parameter state and LightEdit's external retrieval store are different storage abstractions.

Reading the ZsRE block. On ZsRE, AlphaEdit attains higher raw reliability at 10K (94.0 vs. 92.3), and LightEdit attains higher reliability and generalization ($94.8/91.1$ vs. $92.3/84.1$). However, these gains come with different preservation profiles. AlphaEdit incurs a 10.7-point locality gap relative to PACE-FULL (86.5 vs. 97.2) and more than double the forgetting (11.6 vs. 4.7). LightEdit achieves the strongest ZsRE generalization, but PACE-FULL still reduces forgetting (4.7 vs. 5.1) while preserving high locality.[1] The drop in ZsRE generalization from single-edit to 10K is consistent with the design trade-off of PACE: boundary-localized residual writing and null-space/replay constraints preserve unrelated behavior and prior edits, but can narrow paraphrase coverage on short-answer ZsRE queries after many sequential edits. This trade-off is directly visible in the n_{base} sweep of Table 66: increasing n_{base} from 0 to 4,000 improves locality from 94.1 to 97.1, but reduces reliability from 92.2 to 91.5 and generalization from 84.0 to 83.2. PACE's primary contribution is therefore the retention–locality trade-off under long edit streams, rather than maximizing raw reliability or generalization on every dataset.

[1] LightEdit is parameter-preserving, but forgetting is still non-zero because our For. metric evaluates whether answers to previously edited queries remain unchanged under the current retrieval/selection state; forgetting can therefore arise from retrieval or selection changes even without direct weight modification. Its higher CounterFact Gen. at 10K than in the single-edit rerun reflects the shared retrieval/selection state accumulated over the stream and should not be read as improved parametric generalization.

Reading the UltraEdit comparison. UltraEdit provides a stronger recent scalable-editing reference than older locate-then-edit baselines. In the measured 10K CounterFact rerun, UltraEdit achieves the highest reliability and the lowest per-edit runtime among the matched methods. However, PACE-FULL attains stronger generalization, locality, and forgetting performance, indicating a more balanced stabilization profile over the full edit stream. In particular, PACE-FULL improves over UltraEdit by 1.4 points in generalization and 3.3 points in locality, while reducing forgetting by 2.4 points, at the cost of only 0.06 seconds of additional runtime per edit. This pattern supports the central claim of PACE: the activation path is not merely a localization heuristic, but an online state that links localization, replay selection, interference estimation, and null-space preservation.

Runtime follows the same broad pattern across the remaining baselines. AlphaEdit is substantially slower per edit, while LightEdit and PACE are both sub-second in insertion/state-maintenance cost. Their serving costs differ: LightEdit moves part of the work to query-time retrieval/selection, whereas PACE answers from the edited model. Under the unified CounterFact evaluator, PACE has the lowest forgetting while retaining 96.9 locality and the highest generalization among the measured GPT-J rows.

B.13 ADDITIONAL SENSITIVITY ANALYSES

We further evaluate sensitivity to replay size K, replay weight η, minimum path width $k_{\min}$, and replay-selection score. Each sweep changes only one hyperparameter while keeping the remaining PACE-FULL configuration fixed. Table 18 condenses these sweeps together with the layer-selection, percentile, memory, and editable-universe sweeps reported below.

Replay size K	Rel.	Gen.	Loc.	For.↓	Avg.
$K = 4$	91.3	83.0	96.8	5.5	90.4
$K = 8$	91.8	83.6	96.9	4.8	90.8
$K = 16$	91.6	83.4	96.7	4.6	90.6

Table 43: Sensitivity to replay size K on GPT-J-6B after 10K CounterFact edits. All rows use PACE-FULL; only the number of retained replay candidates changes. All entries are measured results from the corresponding replay-size runs.

Replay weight η	Rel.	Gen.	Loc.	For.↓	Avg.
$\eta = 0.5$	**92.1**	83.1	96.4	5.7	90.5
$\eta = 1.0$	91.8	**83.6**	96.9	4.8	**90.8**
$\eta = 2.0$	91.2	82.9	**97.1**	**4.4**	90.4

Table 44: Sensitivity to replay weight η on GPT-J-6B after 10K CounterFact edits. All entries are measured. The sweep is a monotone trade-off with no interior optimum on the preservation axes: $\eta = 2$ is strictly better than the default on both locality (97.1 versus 96.9) and forgetting (4.4 versus 4.8), at a cost of 0.6 reliability and 0.7 generalization points. We set $\eta = 1$ because it maximizes the three-way Rel./Gen./Loc. average, and we state explicitly that a deployment prioritizing retention over immediate fit should prefer $\eta = 2$. Note that η and β are not two versions of one replay-strength knob; see the discussion preceding Table 65.

Minimum path width $k_{\min}$	Rel.	Gen.	Loc.	For.↓	Avg.
$k_{\min} = 16$	90.9	82.8	97.1	5.2	90.3
$k_{\min} = 32$	91.8	83.6	96.9	4.8	90.8
$k_{\min} = 64$	92.0	83.4	96.5	5.0	90.6

Table 45: Sensitivity to the minimum within-layer path width $k_{\min}$ on GPT-J-6B after 10K CounterFact edits. A small minimum can under-cover edit-relevant coordinates, whereas a large minimum broadens updates and slightly reduces locality. All entries are measured results from the corresponding minimum-path-width runs.

Choice of overlap score. PACE ranks replay candidates by Jaccard overlap on binary path coordinates. Two alternatives that incorporate activation magnitude—a weighted Jaccard and cosine similarity between restricted keys—are reported in the controlled replay-ranking comparison of Table 26, which sweeps the ranking signal under an otherwise identical configuration. Weighted Jaccard matches the default Jaccard average (90.8) and slightly improves reliability, generalization, and forgetting, while cosine similarity on restricted keys is 0.4 average points lower. Magnitude-aware overlap therefore does not materially change the conclusion: the binary path structure already captures most of the conflict-relevant signal. We retain Jaccard because it achieves the same average score as weighted Jaccard while requiring less bookkeeping.

B.14 Calibration Refresh

PACE estimates layer-wise activation normalizers (μ_ℓ, σ_ℓ) once from a held-out calibration set. Because these normalizers drive activation-path extraction, a possible concern is that activation drift over very long streams could make the initial normalizers stale. We compare the default static schedule against refreshing the calibration statistics every 25K edits.

Calibration schedule	Rel.@100K	Support/layer (%)	Support/layer (count)	Avg. overlap
Static (once, default)	81.7	3.9	318.6	0.196
Refreshed every 25K	82.0	3.8	312.9	0.193

Table 46: Effect of refreshing layer-wise calibration statistics (μ_ℓ, σ_ℓ) over a 100K CounterFact stream. "Support/layer" reports $100\, d_t^\ell / d_{\text{edit}}^\ell$ and d_t^ℓ respectively, using the definitions of Appendix A.2. Both rows report path statistics measured *at the end* of the corresponding 100K-edit run; they are therefore not directly comparable to Table 57, which profiles the same quantities over the 10K stream, and the modest upward drift in support and overlap between the two is the expected consequence of a ten-fold longer stream rather than an inconsistency. Refreshing changes Rel.@100K by only 0.3 points while leaving support size and overlap nearly unchanged; it additionally requires a calibration-corpus forward pass every 25K edits.

Calibration staleness. Because calibration normalizers drive path extraction, activation drift over very long streams could in principle degrade the quality of selected paths. Table 46 compares the static schedule against refreshing every 25K edits, with path statistics measured at the end of each 100K-edit run. Refreshing changes Rel.@100K by only 0.3 points and leaves path sparsity, path size, and overlap essentially unchanged. Since refreshing requires an additional calibration-corpus forward pass every 25K edits, we keep the static schedule as the default. We note the scope of this evidence explicitly: it bounds drift in the *extraction statistics*, and does not directly measure staleness of the per-record paths stored at their own edit times, which we discuss as a limitation in Appendix A.14.

B.15 Empirical Frequency of High-Overlap Conflicts

Proposition 2 gives an exact closed form for an idealized two-constraint case in which the embedded keys coincide. The stream diagnostic below is deliberately broader: we use high path overlap plus incompatible targets as an empirical proxy for that difficult regime. Specifically, an incoming edit is flagged when it has a replay candidate with $\omega \geq 0.5$ and an incompatible target direction. This criterion does not assert equality of the embedded keys and is therefore not a direct empirical test of Equation A25.

The proxy occurs for 3.8% of edits. On this subset, reliability is 85.6 and forgetting is 9.8, compared with 91.8 and 4.8 over the full stream. Thus high-overlap incompatible edits are uncommon but substantially more difficult. Table 47 quantifies this empirical failure regime; Proposition 2 separately characterizes the exact coincident-key two-constraint case as a bounded ridge-shrunk weighted compromise. We do not equate the two conditions.

Subset	Fraction of stream (%)	Rel.	For.↓
All edits	100.0	91.8	4.8
High-overlap conflict proxy ($\omega \geq 0.5$, incompatible)	3.8	85.6	9.8

Table 47: Empirical footprint of a high-overlap proxy for the conflict setting analyzed in Proposition 2 (Equation A25) on the 10K CounterFact stream with GPT-J-6B. The proxy requires path overlap $\omega \geq 0.5$ and an incompatible target direction, but does not require coincident embedded keys. All entries are measured from the 10K stream.

B.16 Sequential Batch Editing

Following recent sequential-editing protocols (Zhang et al., 2025; Fang et al., 2025; Li et al., 2025), we evaluate a 2K-edit sequential batch setting with batch size 100. This setting stresses both edit success and model stability because the model receives multiple update rounds before final evaluation. Batch-capable baselines solve each 100-edit batch jointly, following their released implementations; for PACE, edits within each batch are applied sequentially in stream order, and the batch boundary defines only the evaluation checkpoints. Table 48 reports the common 2K-edit, batch-size-100 setting used by several locate-then-edit papers. The ROME and MEMIT 2K rows are the unified 2K-sequential measurements reported in the AlphaEdit evaluation rather than the original single-edit tables of those papers. Baseline values are copied only from the original source tables; BLUE is written as AlphaEdit+BLUE because BLUE is an enhancement strategy rather than a standalone editor.

This is the stream-length- and metric-family-aligned counterpart to the single-edit audit of Table 7: all rows use the 2K edit setting and the probability-style Eff./Gen./Spe. family. Update semantics remain method native, however: batch-capable baselines solve each 100-edit batch according to their released implementations, whereas PACE processes the edits strictly sequentially. The table is therefore more comparable than the source-native audit but is not a claim of identical optimization dynamics. Because the metric family differs from the generation-based Rel./Gen. and agreement-based Loc. used elsewhere, the PACE values here are *not* comparable to those in Tables 7, 6, or 8.

Method	CounterFact, 2K edits / batch size 100					ZsRE, 2K edits / batch size 100		
	Eff.	Gen.	Spe.	Flu.	Cons.	Eff.	Gen.	Spe.
ROME Meng et al. (2022) *[NeurIPS'22]*	57.50	54.20	52.05	589.42	3.22	56.42	54.65	9.86
MEMIT Meng et al. (2023) *[ICLR'23]*	98.55	95.50	63.64	546.28	34.89	94.91	90.22	30.39
ECE Zhang et al. (2025) *[WWW'25]*	100.00	96.35	79.41	619.82	42.34	99.74	96.88	28.49
AlphaEdit+BLUE Li et al. (2025) *[NeurIPS'25]*	99.77	97.13	75.23	621.07	41.34	99.63	95.96	28.67
AlphaEdit Fang et al. (2025) *[ICLR'25]*	99.75	96.38	75.48	618.50	42.08	99.79	96.00	28.29
NeuralDB Fei et al. (2026) *[ICLR'26]*	99.70	94.60	80.00	619.80	41.40	99.20	95.90	27.50
PACE-Soft	97.8	90.1	94.8	624.3	44.6	93.6	88.5	94.0
PACE-Full	98.3	91.0	95.4	624.5	45.2	94.2	89.3	94.7

Table 48: Native-update comparison over a 2K-edit setting with checkpoints every 100 edits. All rows use the probability-style Eff./Gen./Spe. metric family; update semantics remain method native, with batch-capable baselines following their released 100-edit-batch procedures and PACE applying edits strictly sequentially. These PACE values are therefore aligned in stream length and metric family but are not comparable to the generation-based PACE rows of Tables 7, 6, or 8. Baseline values are source-reported GPT-J-6B results when available. BLUE is reported as AlphaEdit+BLUE because BLUE is an enhancement strategy rather than a standalone editor. LocFT-BF is omitted because its source uses a different lifelong Rel./Gen./Capability protocol. Bolding is avoided because evaluator implementations and update semantics are not fully unified.

B.17 Comparison with Lifelong and Parameter-Preserving Editors

Since PACE targets online continual editing, we additionally list lifelong and parameter-preserving editors when source-reported CounterFact numbers are available. Most checked sources do not report the exact CounterFact 10K + forgetting + per-edit-time protocol, so the table lists only source-reported dataset-matched operating points and leaves forgetting/time to the dedicated PACE tables. The stream lengths differ across rows and are disclosed in the caption; the table is a dataset-matched reference, not a ranking.

Method	Rel.	Gen.	Loc./Cap.	Avg.	Update type
GRACE Hartvigsen et al. (2023) *[NeurIPS'23]*	6.40	0.33	100.00	35.58	External memory
R-ROME (Gupta et al., 2024a)	6.40	8.90	0.00	5.10	Parameter edit
AlphaEdit Fang et al. (2025) *[ICLR'25]*	93.07	74.77	89.06	85.63	Parameter edit
LTE Jiang et al. (2024) *[ACL'24]*	97.75	87.70	78.15	87.87	Retrieval editing
RECIPE Chen et al. (2024) *[EMNLP'24]*	75.65	68.30	80.50	74.82	Retrieval editing
LightEdit Jung et al. (2026) *[ACL'26]*	99.10	73.20	97.50	89.93	Selective retrieval
LocFT-BF Yang et al. (2026) *[ICLR'26]*	99.73	33.23	57.13†	63.36	Parameter optimization
PACE-Full (10K stream)	91.8	83.6	96.9	90.8	Online parameter edit

Table 49: Dataset-matched reference points for lifelong and parameter-preserving editors on CounterFact. GRACE/R-ROME/AlphaEdit/LTE/RECIPE/LightEdit are source-reported GPT-J values from LightEdit's 1K sequential protocol, converted to percentages. †LocFT-BF is source-reported on LLaMA-3-8B-Instruct and reports Capability rather than Locality under a 3K lifelong protocol; its average is therefore Rel./Gen./Cap. The PACE row is measured on a 10K stream while the source rows use different stream lengths and protocols; this remains a dataset-matched reference rather than a leaderboard. PACE additionally reports forgetting/time in Table 8 and Table 71.

B.18 Long-Form Editing

Long-form evaluation is relevant because a short-form edit can be correct at the edited prompt yet drift during extended generation. The checked source-native long-form studies do not expose a common metric scale or common backbone: for example, the AlphaEdit LEME table reports Edit/Factual/Internal consistency on a small source-native scale, whereas other long-form studies report generation or locality quantities under different evaluators. We therefore do not merge those values into a synthetic leaderboard and do not promote the earlier standalone PACE long-form row to a quantitative claim, because the available run record does not contain a sufficiently explicit benchmark-by-metric mapping and checkpoint specification for independent reproduction. Long-form drift remains a scope limitation rather than evidence used by the main-paper conclusions.

B.19 Evaluation on KnowEdit

To test transfer beyond CounterFact and ZsRE, we report KnowEdit (Zhang et al., 2024) results on WikiRecent and WikiBio. The checked baseline values were measured on LLaMA-3-8B-Instruct, whereas the available PACE KnowEdit runs use GPT-J-6B. We therefore separate the two backbone panels explicitly and use them only as within-panel evidence; no numerical cross-panel ranking or algorithmic delta is claimed. The metric blocks follow the checked source table, so WikiRecent and WikiBio use slightly different displayed columns rather than unsupported placeholders.

B.20 Backbone Robustness

We evaluate PACE-Full on GPT2-XL, GPT-J-6B, and LLaMA-2-7B in the single-edit setting. Table 51 shows a similar qualitative profile across these backbones, with high reliability and locality

Panel	Backbone	Method	WikiRecent				WikiBio		
			Succ.	Port.	Loc.	Flu.	Succ.	Loc.	Flu.
Source-native	LLaMA-3-8B-Instruct	MEMIT	56.25	42.73	41.02	513.35	63.73	64.27	582.38
		RECT	82.47	51.28	48.84	568.62	91.48	72.83	612.04
		AlphaEdit	96.10	57.30	54.76	594.52	95.34	75.34	618.35
PACE evaluator	GPT-J-6B	PACE-SOFT	85.0	60.6	94.8	622.4	82.7	93.4	621.2
		PACE-FULL	86.1	61.9	95.4	622.6	83.8	94.1	621.4

Table 50: KnowEdit WikiRecent and WikiBio results separated by evaluation backbone. The upper panel reproduces source-native LLaMA-3-8B-Instruct values from the checked AlphaEdit-compatible table; the lower panel reports the available PACE GPT-J-6B runs. The table is therefore evidence of dataset transfer within each panel, not a backbone-matched comparison between PACE and the source methods. WikiBio portability is not reported by the source table and is consequently absent rather than estimated. No cross-panel bolding or superiority claim is used.

under our evaluator. Continual-transfer evidence is reported separately on LLaMA-3-8B-Instruct and Qwen2.5-7B-Instruct, so we do not infer a cross-regime gain from this single-edit table.

Backbone	Dataset	Rel.	Gen.	Loc.	Avg.
GPT2-XL	CounterFact	96.8	87.4	96.9	93.7
GPT-J-6B	CounterFact	98.1	89.7	97.8	95.2
LLaMA-2-7B	CounterFact	98.4	90.1	98.1	95.5
GPT2-XL	ZsRE	94.9	88.3	96.7	93.3
GPT-J-6B	ZsRE	96.4	90.2	98.0	94.9
LLaMA-2-7B	ZsRE	96.8	90.7	98.2	95.2

Table 51: Backbone robustness of PACE-FULL in the single-edit setting. All rows are measured results under our evaluator. Avg. is the mean of Rel., Gen., and Loc. The corresponding 10K continual-transfer results on LLaMA-3-8B-Instruct and Qwen2.5-7B-Instruct are reported in Table 10.

B.21 BACKBONE-WISE BASELINE COMPARISON

Table 52 lists source-reported CounterFact averages across available backbones. Because the checked papers do not always use the same LLaMA-family checkpoint, the third column is treated as a dataset-matched reference rather than an exact architecture-matched comparison. The source rows use the 2K edit setting; we therefore separate PACE's single-edit averages from its 2K stream/metric-family-aligned result and do not present the former as a comparison against the source rows.

Recent accepted continual-editing baselines in their native protocols. For completeness, Table 53 preserves the checkpoints reported by NMKE (NeurIPS'25) and BaFT (ICLR'25) that are most relevant to the current continual-editing study. We retain the papers' original $[0, 1]$ scale and do not interpolate missing metrics.

These source results make two comparisons especially relevant to PACE without making them protocol matched. NMKE is the closest recent sparse-neuron lifelong editor, but its reported LLaMA3 stream stops at 2K in the main table (and 5K in the scaling appendix), whereas PACE's headline continual results use 10K–100K streams. BaFT directly targets the editing–locality trade-off, but its reported LLaMA3 continual table uses 1K ZsRE edits and an input-conditioned representation adapter rather than a strict-online FFN weight write. Their numbers therefore strengthen the literature coverage but should not be used to claim a numerical win for PACE without a matched rerun.

Fluency and consistency. Fluency and consistency on CounterFact are reported inside the 2K stream/metric-family-aligned comparison of Table 48, where PACE and the baselines share the stream length and metric family while retaining their method-native update procedures. We do

Method	Regime	GPT2-XL Avg.	GPT-J-6B Avg.	LLaMA-family Avg.
MEMIT Meng et al. (2023) *[ICLR'23]*	2K seq.	80.34	85.90	60.62
ECE Zhang et al. (2025) *[WWW'25]*	2K seq.	87.91	91.92	92.50
AlphaEdit+BLUE Li et al. (2025) *[NeurIPS'25]*	2K seq.	90.68	90.71	90.81
AlphaEdit Fang et al. (2025) *[ICLR'25]*	2K seq.	86.61	90.54	87.00
NeuralDB Fei et al. (2026) *[ICLR'26]*	2K seq.	90.37	91.43	91.57
PACE-FULL *(2K aligned)*	2K seq.	–	94.9	–
PACE-FULL *(single-edit reference)*	single edit	*93.7*	*95.2*	*95.5*

Table 52: Backbone-wise source-reported comparison on CounterFact. Avg. is computed from reliability/efficacy, generalization, and locality/specificity when available. The PACE 2K row is aligned with the source rows in stream length and probability-style Eff./Gen./Spe. family ($(98.3 + 91.0 + 95.4)/3 = 94.9$), but the evaluator implementations and method-native update procedures are not fully unified. The italicized single-edit row is included only to show PACE's cross-backbone consistency and must not be compared numerically with the 2K source rows; GPT2-XL and LLaMA 2K PACE runs were not performed, so those cells remain empty. The LLaMA-family column is dataset-matched but not always checkpoint-matched. LocFT-BF is omitted because it reports Rel./Gen./Capability rather than Rel./Gen./Locality. Where two sources report slightly different values for the same setting, we use the AlphaEdit-reported tuple throughout. Bolding is avoided because the comparison is not fully unified.

Method	Backbone	Dataset	Edits	Rel.	Gen.	Loc.	Avg.
NMKE *[NeurIPS'25]*	LLaMA-3-8B-Instruct	ZsRE	1K	0.95	0.85	0.77	–
	LLaMA-3-8B-Instruct	ZsRE	2K	0.94	0.85	0.71	–
	LLaMA-3-8B-Instruct	CounterFact	1K	0.99	0.65	0.50	–
	LLaMA-3-8B-Instruct	CounterFact	2K	0.98	0.67	0.38	–
BaFT *[ICLR'25]*	LLaMA-3-8B-Instruct	ZsRE	1	1.00	0.95	0.96	0.97
	LLaMA-3-8B-Instruct	ZsRE	10	0.89	0.82	0.95	0.89
	LLaMA-3-8B-Instruct	ZsRE	100	0.70	0.64	0.93	0.76
	LLaMA-3-8B-Instruct	ZsRE	1K	0.50	0.49	0.93	0.64
NMKE *[NeurIPS'25]*	Qwen2.5-7B	ZsRE	100	0.97	0.96	–	–
	Qwen2.5-7B	ZsRE	500	0.97	0.93	–	–
	Qwen2.5-7B	ZsRE	1K	0.97	0.91	–	–
	Qwen2.5-7B	ZsRE	2K	0.96	0.90	–	–

Table 53: Source-native continual-editing values extracted from the accepted NMKE and BaFT papers. NMKE LLaMA3 values are from its main 2K table, its Qwen2.5 values are from the appendix table (which reports Rel./Gen. only); BaFT values are from its ICLR 2025 LLaMA3 ZsRE continual-editing table. The Avg. column is reproduced only where the source paper reports it (BaFT); NMKE does not report this Avg. in the cited tables, so the entry is left blank rather than recomputed. Missing values are never estimated.

not duplicate those baseline rows in a separate single-edit table, since pairing 2K-sequential source values with single-edit PACE values would reintroduce exactly the regime confound that Table 48 is designed to remove.

B.22 General Capability Retention

Locality evaluates preservation on unrelated factual prompts but does not by itself measure broad language capability. We therefore evaluate the unedited base model and the 10K CounterFact PACE-FULL checkpoint under the *same* local capability harness on MMLU (Hendrycks et al., 2021), SciQ (Welbl et al., 2017), CommonsenseQA (Talmor et al., 2019), ARC-Challenge (Clark et al., 2018), LAMBADA (Paperno et al., 2016), and WSC273 (Levesque et al., 2012). We intentionally do not place source-native AlphaEdit/NeuralDB capability values in the same numerical panel because their lm-evaluation harness/base configuration yields materially different unedited anchors.

Consequently, this experiment supports a within-PACE retention claim only; it does not claim that PACE preserves broad capability better than a parametric baseline without a shared-harness baseline checkpoint.

Method	MMLU	SciQ	CSQA	ARC-C	LAMBADA	WSC273	Avg.
Base (unedited)	27.8	87.4	53.5	33.2	66.8	59.0	54.6
PACE-FULL, 10K CounterFact	27.3	87.0	53.3	32.8	65.8	58.0	54.0

Table 54: Within-harness general-capability retention after 10K CounterFact edits. Both rows use the same base checkpoint and evaluation harness. PACE-FULL changes the six-task average from 54.6 to 54.0, a 0.6-point drop. Source-native baseline capability numbers measured under different harnesses are omitted from this table rather than treated as shared anchors.

B.23 PARAPHRASE AND SUBJECT-TOKEN STABILITY OF ACTIVATION PATHS

Since PACE uses the activation path as a shared representation for localization, replay, and preservation, we examine how stable this path remains under prompt paraphrasing and subject-token choice. For each edit, we extract the activation path from the canonical edit prompt and from a perturbed prompt that refers to the same edit target. We then compare the resulting neuron–layer coordinate sets, active-layer sets, and boundary layers.

Dataset / perturbation	Path Jac.	Layer Jac.	Exact boundary	Boundary ±1	Support ratio
CounterFact paraphrase	0.62	0.80	72.8	89.4	1.03
ZsRE paraphrase	0.58	0.76	68.9	86.2	1.05
MQuAKE paraphrase	0.55	0.74	66.5	84.0	1.07
CounterFact subject-token choice	0.66	0.83	76.1	91.0	1.01

Table 55: Measured paraphrase and subject-token stability of activation paths. Path Jac. is the Jaccard overlap between neuron–layer coordinate sets extracted from the canonical prompt and the perturbed prompt. Layer Jac. measures the overlap between active-layer sets. Exact boundary reports the percentage of edits for which the first and last active layers match exactly, whereas Boundary ±1 allows a one-layer deviation at either boundary. Support ratio is the perturbed prompt's mean path-support size divided by the canonical prompt's support size. All values are obtained from forward-pass diagnostics on the corresponding evaluation sets, on the unedited base model.

The measured results show that the activation path is not fully invariant to paraphrasing or subject-token choice, and PACE therefore does not treat it as a brittle attribution trace. Nevertheless, Table 55 demonstrates that its coarse structure remains stable across prompt variations. Active-layer overlap ranges from 0.74 to 0.83, exact boundary agreement is observed for 66.5–76.1% of edits, and a one-layer boundary tolerance increases agreement to 84.0–91.0%. The support ratios remain close to 1, indicating that perturbations alter path composition more than overall path size. These findings support using the activation path as an online editing representation: its boundary structure is sufficiently stable for localization and preservation, while coordinate-level variation continues to provide useful information for replay-overlap estimation.

B.24 PATH-SPECIFIC ANALYSIS

Because activation paths are the central representation used by PACE, we analyze their sparsity and interference structure. We first report the distribution of the active-layer set size $|\mathcal{A}_t|$ (Table 56), which fixes the boundary-cardinality quantities used throughout: since $|\mathcal{B}_t| = 1$ exactly when $|\mathcal{A}_t| = 1$ and $|\mathcal{B}_t| = 2$ otherwise, the single-active-layer rates imply the mean boundary counts of Table 57, and the $|\mathcal{A}_t| \geq 3$ rates delimit the subset on which the two-boundary design differs from all-active-layer editing. Because the within-layer support d_t^ℓ is edit-dependent under Equation A3, the upper percentiles of $|\mathcal{A}_t|$ combine with the τ_{coord} floor to produce the sublinear growth of $|\mathcal{P}_t|$ documented in Table 58.

Table 57 reports activation-path statistics on CounterFact, ZsRE, MQuAKE, and WikiBigEdit. To match the definitions in Appendix A.2, we report the support count and support fraction *per selected active layer*, averaged over active layers and edits; these quantities are distinct from the total

Dataset	$\lvert\mathcal{A}_t\rvert$ p50	$\lvert\mathcal{A}_t\rvert$ p90	$\lvert\mathcal{A}_t\rvert$ p99	$\lvert\mathcal{A}_t\rvert$=1	$\lvert\mathcal{A}_t\rvert$=2	$\lvert\mathcal{A}_t\rvert\geq$3
CounterFact	2	4	6	4.0	53.0	43.0
ZsRE	2	4	6	8.0	50.5	41.5
MQuAKE	3	5	7	1.2	41.8	57.0
WikiBigEdit	2	4	6	2.0	49.5	48.5

Table 56: Distribution of active-layer set size $|\mathcal{A}_t|$. The last three columns are percentages over edit prompts. Since $|\mathcal{B}_t| = 1$ only when $|\mathcal{A}_t| = 1$ and $|\mathcal{B}_t| = 2$ otherwise, the single-active-layer rates imply the mean boundary-layer counts reported in Table 57 (e.g. CounterFact $2 - 0.040 = 1.96$; MQuAKE $2 - 0.012 = 1.988 \approx 1.99$). The $|\mathcal{A}_t| \geq 3$ column gives the subset on which boundary editing differs from all-active-layer editing.

path size $|\mathcal{P}_t|$ and from the replay-expanded solve width u_t^ℓ. The mean support fraction is at most 4.0% across datasets, while the moderate full-path overlap explains why replaying every previous edit is unnecessary: overlap-ranked replay supplies a compact and targeted preservation signal. WikiBigEdit is included both as a path-structure diagnostic and as a chronologically ordered 10K end-to-end transfer stream (Table 9).

Dataset	Mean support / active layer (%)	Mean support / active layer	Boundary layers	Avg. overlap
CounterFact	3.8	312.4	1.96	0.184
ZsRE	3.5	287.6	1.92	0.171
MQuAKE	4.0	326.8	1.99	0.213
WikiBigEdit	3.9	319.5	1.98	0.197

Table 57: Activation-path statistics, measured over the 10K CounterFact stream and the corresponding evaluation streams of the other datasets. "Mean support / active layer" reports d_t^ℓ and $100\, d_t^\ell / d_{\text{edit}}^\ell$, averaged over selected active layers and edits. It is not the total path size $|\mathcal{P}_t| = \sum_{\ell \in \mathcal{A}_t} d_t^\ell$ and not the replay-expanded solve width $u_t^\ell = |\mathcal{U}_t^\ell|$. Boundary layers is the mean cardinality of $\mathcal{B}_t$, and Avg. overlap is the mean full-path Jaccard overlap used for replay ranking. For GPT-J-6B, $d_{\text{edit}}^\ell = 8192$ and $\varrho = 4\%$, so the per-layer budget is 328 coordinates; all reported support means remain below this upper bound. The corresponding CounterFact values in Table 46 are measured at the end of a 100K stream and are therefore slightly larger; the two are consistent, not conflicting.

B.25 Distributional Sparsity and Preservation-Rank Statistics

Mean path statistics can obscure heavy-tail behavior. We therefore report percentiles of the quantities that govern runtime and memory: total path size $|\mathcal{P}_t|$, active solve-block width u_t^ℓ, nonzero preservation columns c_t^ℓ, preservation rank k_t^ℓ, and null-space projector time. Percentiles for u_t^ℓ, c_t^ℓ, k_t^ℓ, and projector time are computed over edited boundary-layer blocks.

Step	$\lvert\mathcal{P}_t\rvert$			u_t^ℓ		
	p50	p90	p99	p50	p90	p99
1K	612	936	1296	384	612	880
10K	624	958	1328	412	674	1016
50K	631	972	1350	428	704	1088
100K	638	988	1376	439	732	1136

Table 58: Measured distributions of total path size $|\mathcal{P}_t|$ and active solve-block width u_t^ℓ over the CounterFact stream. Percentiles for u_t^ℓ are computed over edited boundary-layer blocks. All values are computed from logged statistics at the corresponding stream checkpoints.

Step	c_t^ℓ			k_t^ℓ			Projector ms		
	p50	p90	p99	p50	p90	p99	p50	p90	p99
1K	28	52	91	24	45	78	63	112	176
10K	34	63	108	29	53	91	78	129	203
50K	37	70	119	31	58	99	81	138	221
100K	39	74	126	33	62	105	84	145	238

Table 59: Measured distributions of nonzero preservation-key columns c_t^ℓ, preservation rank k_t^ℓ, and null-space projector time over the CounterFact stream. Percentiles are computed over edited boundary-layer blocks. The bounded p99 ranks explain why the projector remains a minority of end-to-end edit latency in Table 74. All values are computed from logged statistics at the corresponding stream checkpoints.

The measured distributional statistics show that long-stream scaling is not driven by unbounded growth of the active solve. Although the stream length increases from 1K to 100K, the p99 active solve-block width increases only moderately, from 880 to 1136, because the replay memory is capped at $M = 64$ and obsolete statistic blocks are evicted. The total path-size distribution is similarly stable: its p99 increases from 1296 at 1K edits to 1376 at 100K edits.

The preservation statistics exhibit the same bounded behavior. The measured p99 preservation rank increases from 78 to 105 across the full stream, while the corresponding p99 projector time increases from 176 ms to 238 ms. Consequently, null-space projection does not become the dominant cost even in tail cases. These results support the wall-clock decomposition in Table 74, where path extraction and candidate-logit Jacobian construction remain the largest contributors to total edit latency, and they are the mechanism behind the bounded memory growth reported in Table 75.

B.26 Path Conflict and Distractor Robustness

A continual editor preserves an earlier edit even when unrelated edits are inserted before the original fact is queried. Following the distractor protocol used in parameter-preserving editing, we first apply a target edit, insert n unrelated distractor edits, and then evaluate whether the original target edit remains correct. We report target-edit retention as the percentage of target queries that remain correct after distractors are added. Table 60 reports the $n = 64$ distractor condition; the full distractor sweep is retained in our internal diagnostics.

Method	Distractors n	Target Rel.	Distractor Rel.	Loc.	Target forgetting↓
PACE-Soft	64	93.0	89.5	96.1	4.8
PACE-Full	64	**94.2**	**90.1**	**96.8**	**3.9**

Table 60: Target-edit retention under $n = 64$ unrelated distractor edits on CounterFact. Target Rel. measures retention of the original target edits after distractors are inserted; Distractor Rel. measures whether the distractor edits themselves are encoded; Target forgetting is computed as the drop from the corresponding single-edit CounterFact reliability in Table 7 (97.8 for PACE-Soft and 98.1 for PACE-Full). This is a PACE-specific diagnostic because the checked baseline/SOTA papers do not report the same distractor protocol.

The distractor analysis is retained as a PACE diagnostic because the checked baseline/SOTA papers do not report this exact distractor-retention protocol. It is also the clearest small-scale illustration of the projection's role: the two variants differ only by $P_{0,t}^\ell$, and the full variant retains 1.2 more points of the original target after 64 unrelated writes.

Relation to conflict-aware editing in unstructured text. Conflict-Aware Knowledge Editing in the Wild (CAKE, NeurIPS'25 Spotlight) also treats edit conflicts explicitly, but it derives conflicts from a semantic graph over knowledge extracted from unstructured text and evaluates on AKEW. PACE instead estimates interference from overlap in the model's edit-conditioned activation support

under sequential factual editing. Because the task, input representation, and reported accuracy metric differ, we do not copy CAKE's AKEW numbers into the CounterFact/ZsRE tables.

B.27 Scoring and Generation-Length Sensitivity

Knowledge-editing comparisons are sensitive to scoring paradigms: probability metrics, generation-based exact matching, contains-target matching, answer normalization, and generation length can change absolute performance levels. To verify that our conclusions do not depend on a single generation setting, we evaluate PACE-Full after 10K CounterFact edits under several scoring variants.

Scoring / generation setting	Rel.	Gen.	Loc.	For.↓
Default contains-target, max 32 tokens	91.8	83.6	96.9	4.8
Exact normalized object, max 32 tokens	90.9	82.4	96.9	5.1
First-token match, max 32 tokens	93.0	84.8	96.8	4.6
Default contains-target, max 16 tokens	91.6	83.1	97.0	4.8
Default contains-target, max 64 tokens	91.9	83.7	96.7	4.9

Table 61: Scoring and generation-length sensitivity for PACE-Full on CounterFact after 10K edits. The default row corresponds to the main evaluation protocol. All values are measured using the same edited model checkpoint under the indicated scoring and generation settings.

The measured results show that absolute performance changes with the scoring rule, particularly when moving from contains-target matching to the stricter exact normalized-object criterion. First-token matching produces slightly higher reliability and generalization scores because it imposes a less restrictive correctness condition. Despite these shifts in absolute values, the overall retention–locality profile remains stable across scoring variants. Changing the maximum generation length from 16 to 64 tokens has only a minor effect on all four metrics, indicating that the reported conclusions are not sensitive to the tested decoding-length range. These diagnostics support our decision to separate incompatible metric families in audit tables and to use a unified generation-based evaluator for protocol-matched comparisons.

B.28 Layer-Selection and Boundary Analysis

PACE updates the first and last active layers of the activation path. We compare this design with random layer selection, a fixed middle layer, LGA-selected golden layers, one-boundary updates, and all-layer updates. Random selection is repeated across ten seeds. LGA improves over random and fixed-layer selection by identifying high-impact layers through proxy-gradient analysis. However, a single golden layer cannot fully represent the distributed start-to-end computation path associated with an edit: both one-boundary variants reach lower average score and higher forgetting than the two-boundary design, and updating all active layers maximizes raw reliability but degrades locality and forgetting (this row coincides with the *w/o boundary layers* ablation in Table 25). The two-boundary strategy provides the best reliability–locality–retention trade-off.

To test whether the boundary set is doing genuine work rather than simply restricting the number of written coordinates, we add a random coordinate mask matched in size to the boundary write set while keeping replay and null-space preservation fixed. This is an editing-mask ablation rather than activation patching in the causal-tracing sense. The random mask collapses reliability to 73.8 and generalization to 66.4. Its numerically high locality (97.3) is degenerate: it writes to coordinates largely unrelated to the edit, so it disturbs little and also encodes little. We omit the earlier "interior-only" aggregate because 57% of CounterFact edits have fewer than three active layers and therefore no non-boundary interior; without an explicit empty-interior execution rule, that aggregate is not uniquely interpretable.

B.29 Configuration Audit: Factorial Cell D versus the Path-Free Pipeline

Factorial cell D (Table 17) and the *w/o activation path* row (Table 25) both use a fixed middle layer and prompt-embedding replay, and are easily confused. They are different controls answering

Layer strategy	Rel.	Gen.	Loc.	For.↓	Avg.
Random layer	$87.9_{\pm 1.4}$	$78.6_{\pm 1.7}$	$93.7_{\pm 0.9}$	$9.7_{\pm 1.2}$	86.7
Fixed middle layer	88.7	79.4	94.0	8.9	87.4
LGA-selected layer	90.9	82.4	95.8	6.5	89.7
First boundary only	89.8	81.2	**97.0**	6.1	89.3
Last boundary only	90.6	82.0	95.6	6.7	89.4
All active layers	**92.1**	83.2	94.5	6.9	89.9
Both boundaries	91.8	**83.6**	96.9	**4.8**	**90.8**
Diagnostic write mask (excluded from bolding)					
Random boundary-size mask	73.8	66.4	97.3	15.1	79.2

Table 62: Layer-selection and boundary-layer controls on CounterFact with 10K continual edits. Random-layer results are averaged over ten seeds. The *Fixed middle layer* row uses $\ell = 14$ on GPT-J-6B, the same layer used by factorial cell C of Table 17 and by the path-free pipeline of Table 25; the three rows therefore share a localization rule and differ only in what else is path conditioned (Appendix B.29). **Bold** marks the best value in each column among the layer-selection strategies (upper block). The lower block contains one capacity-matched random coordinate-mask diagnostic and is excluded from bolding because its high locality is degenerate: it writes to coordinates largely unrelated to the edit and therefore also fails to encode it. The earlier interior-only aggregate is omitted because the non-boundary interior is empty for 57% of CounterFact edits and the original aggregate did not specify a reproducible empty-interior execution rule.

different questions, and their relative ordering is informative rather than contradictory. We therefore tabulate the exact configurations.

Variant	Write location	Replay ranking	Active coordinates	Preservation support
PACE-FULL	Path boundary $\mathcal{B}_t$	Path Jaccard $\omega(i,t)$	Path union $\mathcal{U}_t^\ell$	Path-restricted $K_{0,t}^\ell$
Factorial cell D	Fixed middle layer ℓ=14	Prompt-embedding cosine	Path union $\mathcal{U}_t^{14}$	Path-restricted $K_{0,t}^{14}$
w/o activation path	Fixed middle layer ℓ=14	Prompt-embedding cosine	Fixed 328-coordinate block	Keys in the same fixed block

Table 63: Configuration audit. Factorial cell D substitutes only localization and replay ranking and remains path conditioned in the solve and the preservation support; the *w/o activation path* variant removes $\mathcal{P}_t$ from all four roles and uses a fixed size-matched coordinate support throughout. The two are neither identical nor nested, which is why they are reported as separate controls.

For the path-free pipeline the replacements are capacity matched: localization uses the fixed middle GPT-J FFN layer $\ell = 14$; replay uses prompt-embedding cosine at the same capacity $M = 64$ and count $K = 8$; the active update support is a fixed block of 328 coordinates—the same width as the default per-layer path budget $\lceil 0.04 \times 8192 \rceil$—selected once before the stream from the same activation-variance-ranked editable universe; and the same general and history preservation-key pools are represented in that fixed block, with the projection itself retained. The variant therefore removes edit-specific conditioning rather than removing replay, optimization, or preservation, and it is the correct comparison for the question *is path conditioning worth anything at all?*

Cell D is deliberately a hybrid. After the two substituted decisions, the current and replayed records are still represented on their path-derived sparse supports,

$$\mathcal{U}_t^{14} = \mathcal{P}_t^{14} \cup \bigcup_{i \in \mathcal{R}_t^{14}} \mathcal{P}_i^{14},$$

and the preservation keys remain restricted to that path-conditioned subspace. It is the correct comparison for the question *do path-derived localization and path-derived replay interact, given that the solve and preservation are already path conditioned?*

Why the hybrid underperforms the coherent path-free pipeline. Cell D scores below the path-free variant (Avg. 86.6 versus 87.5; forgetting 9.8 versus 8.7) despite retaining two path-conditioned mechanisms. The two configurations share their localization rule and their replay ranking exactly, so the comparison isolates the downstream stages: the only difference is that cell D draws its active

coordinates and preservation keys from the incoming edit's path, while the path-free variant draws both from one fixed size-matched block. The two are therefore not nested, so this is not an arithmetic contradiction—but it is more than a technicality: it is a positive result about coupling. In cell D the preservation keys are restricted to the support the *incoming edit* engages, while the write is placed at a layer chosen without reference to that support. The projector therefore protects directions selected by one criterion while the update is constrained by another, and the two criteria need not agree. The path-free pipeline avoids this because its coordinate support and its preservation representation are drawn from the same fixed block: it is internally consistent, merely uninformed.

The hybrid ordering is informative but should not be interpreted as a standalone proof of interaction: the path-conditioned and path-free pipelines induce different downstream support distributions. One plausible explanation is a coordination mismatch—cell D chooses the write layer and replay records without the same state that defines its solve and preservation support—but this comparison alone does not identify that mechanism. We therefore treat the factorial audit as diagnostic evidence and use the explicit shared-state experiment of Table 4 as the primary test that state identity across roles is load-bearing.

B.30 Hyperparameter Sensitivity

We characterize sensitivity to the principal path, replay, decay, fitting, and preservation parameters for which completed controlled sweeps are available. Each sweep changes one setting while keeping the remaining PACE-Full configuration fixed. The coordinate saliency threshold τ_{coord} is fixed at its pre-specified value throughout these runs and is therefore not included in the empirical sensitivity claims. We omit the earlier "bookkeeping batch-size" diagnostic because the mathematical editor is strictly sequential and its logical state is updated after every edit; cache flush granularity is an implementation detail rather than an algorithmic hyperparameter.

Table 64 reports the activation-path percentile ϱ. The sweep exposes a clear locality–reliability trade-off: a small percentile ($\varrho = 1\%$) maximizes locality but under-covers the edit and lowers reliability, whereas a large percentile ($\varrho = 16\%$) raises reliability at the cost of locality and forgetting. The default setting $\varrho = 4\%$ attains the highest measured Avg. in this sweep.

Path percentile ϱ	Rel.	Gen.	Loc.	For.↓	Avg.
1%	88.9	79.7	**97.5**	5.9	88.7
2%	90.7	82.1	97.2	5.1	90.0
4%	91.8	**83.6**	96.9	**4.8**	**90.8**
8%	91.9	83.4	96.1	5.5	90.5
16%	**92.0**	83.1	94.8	6.7	90.0

Table 64: Sensitivity to activation-path percentile ϱ on CounterFact with 10K edits. **Bold** marks the best value in each column.

Disentangling the four replay-state quantities. Four quantities govern how much of the edit history influences the current solve, and they are easily conflated into a single notion of "replay strength"; read that way, their sweeps appear inconsistent. The capacity M determines how many previous records remain *eligible* for selection; K determines how many are *actually replayed* at the current step; η sets the overlap-dependent weight $w_i = 1 + \eta\,\omega(i,t)$ of a *selected* record; and β sets age decay on all retained contributions represented in a touched sufficient-statistics block when that block is re-materialized, rather than only on the currently selected replay records. Raising η therefore strengthens a small, overlap-selected set of constraints, whereas raising β preserves older represented contributions more uniformly. The two act on different objects, and their upper extremes accordingly differ in kind rather than degree.

Table 65 reports the replay memory size M and the forgetting factor β. Reliability and forgetting improve sharply with M up to $M = 64$ and then saturate: $M = 128$ matches $M = 64$ in average score while doubling the replay memory, so we use $M = 64$ as the default. For the forgetting factor, $\beta = 0.99$ gives the best trade-off; $\beta = 1.00$ (no decay) slightly improves locality but degrades reliability and forgetting, because stale records are never down-weighted.

Setting	Rel.	Gen.	Loc.	For.↓	Avg.
$M = 0$	86.9	77.5	96.2	11.6	86.9
$M = 16$	89.4	80.8	96.5	7.2	88.9
$M = 32$	90.6	82.2	96.7	5.7	89.8
$M = 64$	91.8	**83.6**	96.9	4.8	**90.8**
$M = 128$	**91.9**	83.5	**97.0**	**4.7**	**90.8**
$\beta = 0.90$	89.1	80.5	96.5	7.6	88.7
$\beta = 0.95$	90.5	82.0	96.7	5.9	89.7
$\beta = 0.99$	**91.8**	**83.6**	96.9	**4.8**	**90.8**
$\beta = 1.00$	91.4	83.2	**97.1**	5.0	90.6

Table 65: Sensitivity to replay memory size M and forgetting factor β on CounterFact with 10K edits. **Bold** marks the best value in each column, within the M and β blocks separately. The $M = 0$ row removes replay entirely and coincides with the *w/o replay* ablation of Table 25; it is excluded from the sensitivity range reported in Table 18, which characterizes the tuning range rather than component removal.

The two sweeps have different shapes, and the difference is mechanistic rather than incidental. The M sweep saturates: raising capacity from 0 to 64 reduces forgetting monotonically from 11.6 to 4.8, while doubling again to 128 buys 0.1 forgetting points and no change in Avg., both below the deviations of Table 81. The β sweep instead has an interior optimum, and $\beta = 1$ is the one setting in this paper where *more* retention is measurably worse: locality rises to 97.1, the highest in the sweep, but reliability falls to 91.4 and forgetting worsens from 4.8 to 5.0. Undecayed statistics keep asserting targets from records that no longer share the current path, so the solve is pulled toward stale constraints that neither fit the incoming edit nor protect the edits actually at risk. This illustrates why β and η cannot be collapsed into one parameter: increasing η strengthens the currently selected, overlap-relevant constraints, whereas increasing β slows the decay of all represented contributions in touched blocks, including constraints that may no longer be relevant to the current path.

n_{base}	Rel.	Gen.	Loc.	For.↓	Avg.
0 (history only)	**92.2**	**84.0**	94.1	5.3	90.1
500	92.0	83.8	95.5	5.0	90.4
1,000	91.9	83.7	96.3	4.9	90.6
2,000	91.8	83.6	96.9	**4.8**	**90.8**
4,000	91.5	83.2	**97.1**	4.9	90.6

Table 66: Sensitivity to the number of general preservation keys n_{base} on CounterFact with 10K continual edits. The $n_{\text{base}} = 0$ row uses history keys only, without a general-prompt preservation pool; note that the projector remains active in that row, so it is distinct from the *w/o null-space projection* ablation of Table 25, which sets $P^{\ell}_{0,t} = I$. The default configuration uses $n_{\text{base}} = 2{,}000$. Avg. is computed from Rel., Gen., and Loc.; For.↓ is reported separately. **Bold** marks the best value in each column.

Table 66 separates the two preservation-key pools. With $n_{\text{base}} = 0$, PACE retains only history keys: reliability remains high at 92.2, while locality is 94.1 and forgetting is 5.3. Adding general preservation keys raises locality from 94.1 to 97.1 while slightly tightening reliability and generalization; forgetting reaches its lowest measured value at the default $n_{\text{base}} = 2000$. The history-only condition should not be conflated with removing projection entirely: the former retains a projector built from replay-history keys, whereas the latter sets $P^{\ell}_{0,t} = I$. The two pools therefore play complementary empirical roles, with history keys protecting retained edited mappings and general keys improving unrelated-prompt locality.

Activation-path layer-selection sensitivity. Beyond the update-level hyperparameters above, we assess the two parameters governing *layer* selection, τ_{lay} and k_{lay}, whose defaults were stated in the

implementation details. Table 67 reports a controlled sweep on CounterFact after 10K sequential edits: in the upper block $k_{\text{lay}} = 64$ is fixed, in the lower block $\tau_{\text{lay}} = 1.0$ is fixed, and all continual metrics follow Table 8. Both parameters are robust within the reported ranges: the average score varies by at most 0.8 points. The threshold τ_{lay} exposes an interpretable trade-off—lowering it broadens the active-layer set and raises coverage at a locality cost, whereas raising it tightens the set and favors locality—while k_{lay}, which only stabilizes the layer-activity score, has a markedly smaller effect. This indicates that the reported layer-selection defaults are comparatively stable within the evaluated ranges; it does not establish invariance outside these ranges or across all backbones.

Setting	Rel.	Gen.	Loc.	For.↓	Avg.	Active FFN (%)	Time/edit (s)
$\tau_{\text{lay}} = 0.5$	91.0	82.7	96.4	5.5	90.0	5.2	0.78
$\tau_{\text{lay}} = 1.0$	**91.8**	**83.6**	96.9	**4.8**	**90.8**	3.8	0.74
$\tau_{\text{lay}} = 1.5$	91.1	83.0	**97.1**	5.0	90.4	2.9	**0.71**
$k_{\text{lay}} = 32$	91.3	83.1	96.7	5.1	90.4	4.1	0.75
$k_{\text{lay}} = 64$	**91.8**	**83.6**	96.9	**4.8**	**90.8**	3.8	0.74
$k_{\text{lay}} = 128$	91.6	83.5	**97.0**	4.9	90.7	**3.6**	0.74

Table 67: Sensitivity to activation-path layer-selection parameters on GPT-J-6B after 10K CounterFact edits. Avg. is the mean of Rel., Gen., and Loc. Lower τ_{lay} broadens the editable support, whereas overly restrictive thresholds can omit useful coordinates.

B.31 Full Long-Stream Continual Metrics

The complete continual-editing tuple across stream lengths—reliability, generalization, locality, and forgetting—is reported in Table 23, together with the bounded resource profile of Table 75. We analyze that trajectory here rather than duplicating the table. This evaluation distinguishes whether the method preserves only direct target recall or also maintains paraphrase transfer and unrelated-prompt locality throughout extended edit streams.

The measured trajectory exhibits a consistent ordering of long-stream degradation. Reliability decreases gradually as the number of edits grows, indicating that PACE-Full retains substantial direct target recall even after 100K sequential edits. Generalization declines more rapidly, suggesting that paraphrase transfer is more sensitive to increasing interference and to the progressively tighter constraints imposed by replay and null-space preservation. Locality remains comparatively high across all stream lengths, consistent with restricting parameter updates to activation-path boundary coordinates. Although forgetting increases with stream length, the degradation is gradual rather than abrupt, demonstrating that PACE-Full degrades gradually rather than abruptly over the measured stream. Figure 7 further shows that this trend is not an artifact of one edit order: the 7.4-point reliability decrease from 1K to 25K is an order of magnitude larger than the ± 0.5 order-induced standard deviation at 25K.

B.32 Scalability with Edit Stream Length

Table 68 reports reliability as the edit stream grows from 1K to 100K edits. Source papers provide only partial stream-length coverage, so we keep verified 10K values where available and omit missing lengths rather than interpolating them. For PACE, checkpoints beyond one complete CounterFact pass are repeated-update stress-test operations under the deterministic construction in Appendix B.1, not distinct-fact counts.

B.33 Append, Revise, and Delete Operations

Realistic knowledge maintenance requires more than appending new facts. A practical system also revises previously edited facts and removes obsolete updates. We evaluate append, revise, and delete operations. Append adds a new fact; revise replaces the target object of a previous edit; delete removes a previous edit and restores the base-model association. For PACE, delete operations remove the relevant edit record from replay memory and recompute the affected active sufficient-statistics block from retained records.

Method	Source setting	Stream length	Rel./Eff.
LightEdit Jung et al. (2026) *[ACL'26]*	GPT-J CounterFact test split	1K	99.10
MEMIT (NeuralDB rerun) (Fei et al., 2026)	NeuralDB unified CounterFact 10K	10K	48.8
AlphaEdit (NeuralDB rerun) (Fei et al., 2026)	NeuralDB unified CounterFact 10K	10K	91.6
NeuralDB Fei et al. (2026) *[ICLR'26]*	NeuralDB unified CounterFact 10K	10K	99.1
PACE-FULL	CounterFact online stream	1K	96.7
PACE-FULL	CounterFact online stream	5K	94.1
PACE-FULL	CounterFact online stream	10K	91.8
PACE-FULL	CounterFact online stream	25K	89.3
PACE-FULL	CounterFact online stream	50K	86.4
PACE-FULL	CounterFact online stream	100K	81.7

Table 68: Reliability as a function of sequential update count on CounterFact. Baseline/SOTA rows are listed only at source-reported dataset-matched lengths rather than interpolated onto PACE's grid. The MEMIT, AlphaEdit, and NeuralDB 10K rows are unified CounterFact measurements reported by the NeuralDB paper and use the probability-style Eff. metric, whereas PACE rows are generation-based; the columns are therefore aligned by update count, not by metric family. PACE checkpoints beyond one complete CounterFact pass use successive seeded full-dataset permutations with unchanged targets and therefore measure repeated-update stress-test stability rather than distinct-fact coverage. LocFT-BF reports a 3K CounterFact lifelong point and a separate 100K ZsRE scaling curve, so it is not inserted into this CounterFact table.

Method	Append success	Revise success	Delete restoration
PACE-SOFT	96.4	91.7	84.5
PACE-FULL	**96.7**	**93.6**	**92.1**

Table 69: Knowledge-management operations on CounterFact. Prior baseline/SOTA papers do not report this exact append/revise/delete protocol, so we report it as an internal PACE diagnostic.

The append/revise/delete analysis is retained as an internal PACE diagnostic. The gap in delete restoration (84.5 versus 92.1) is the largest of the three and follows from the block recomputation: because PACE-FULL confines each incremental write to the complement of the preservation keys, removing a record leaves less unrecoverable collateral movement in the retained statistics than in the unprojected variant.

B.34 QUANTIZATION ROBUSTNESS

Recent mobile editing work shows that gradient-based editing can degrade under low-bit quantization because quantization noise affects backward propagation and parameter updates (Lu et al., 2026). Since PACE uses local Jacobian computation and closed-form least-squares updates, we evaluate mixed-precision robustness by retaining the selected active FFN block and Jacobian computation in FP16 while quantizing the remaining model weights.

Method	Precision	Rel.	Loc.	Edit latency (s)
PACE-FULL	FP16	**98.1**	**97.8**	0.74
PACE-FULL	W8A16	94.3	96.2	0.61
PACE-FULL	W4A16	89.1	94.0	0.54

Table 70: Quantization robustness on CounterFact in the single-edit setting. This is a PACE-specific mixed-precision diagnostic because the checked baseline/SOTA papers do not report the same FP16/W8A16/W4A16 protocol.

The quantization analysis is retained as an internal PACE diagnostic.

B.35 EFFICIENCY ANALYSIS

Table 71 compares editing efficiency on GPT-J-6B. PACE avoids iterative hidden-state optimization and instead solves a low-dimensional least-squares system over sparse active-coordinate blocks. PACE-FULL adds path-restricted preservation through a thin QR/SVD factorization, whereas PACE-SOFT omits this projection. The per-active-layer path support is approximately 3.6–3.8% of the editable FFN width; the replay-expanded solve width u_t^ℓ is reported separately through the cost and resource analyses.

Because prior papers report runtime in different formats and usually do not provide the same peak-memory definition, Table 71 uses only source-reported runtime fields and omits incompatible memory columns. PACE memory is analyzed separately in Tables 74 and 75.

Method	Source setting	Time/edit (s)	Source Avg.
ROME Meng et al. (2022) *[NeurIPS'22]*	GPT-J, ZsRE, 100-edit batch	7.65†	40.3
MEMIT Meng et al. (2023) *[ICLR'23]*	GPT-J, ZsRE, 100-edit batch	3.35†	71.8
ECE Zhang et al. (2025) *[WWW'25]*	GPT-J, ZsRE, 100-edit batch	1.78†	75.0
AlphaEdit Fang et al. (2025) *[ICLR'25]*	GPT-J, CounterFact, 100-edit batch	3.35‡	90.5
LocFT-BF Yang et al. (2026) *[ICLR'26]*	LLaMA-3-8B-Instruct, CounterFact, 3K lifelong	0.38	63.4§
PACE-SOFT	GPT-J, CounterFact, 10K online	0.65	89.7
PACE-FULL	GPT-J, CounterFact, 10K online	0.74	90.8

Table 71: Efficiency comparison using source-reported runtime fields when available; the source backbone and operating point are listed per row, and PACE rows use GPT-J-6B. The final column is the average reported for that same operating point, so the PACE values are the 10K CounterFact averages (89.7/90.8), not their single-edit averages. The two PACE rows are production runs that differ only by the null-space projector, so their latency gap ($0.74 - 0.65 = 0.09$ s) is the projector stage of Table 74 (92 ms on the same 10K stream) up to rounding. These values are not comparable to the per-edit times in Table 84, which are recorded under diagnostic instrumentation on a different stream length; see that caption. †The ECE paper reports per-100-edit-batch wall time on ZsRE for ROME, MEMIT, and ECE; values are divided by 100. ‡AlphaEdit reports time per 100-edit batch on CounterFact; values are divided by 100. §LocFT-BF source average uses Rel./Gen./Capability rather than Rel./Gen./Locality. Editors without source-reported runtime are omitted instead of assigned estimated costs. Bolding is avoided across source-native and PACE rows because the evaluator/protocol is not fully unified.

Candidate-set sensitivity. Table 72 evaluates the number of competing tokens used to construct the local candidate set in the continual 10K-edit setting. The new-object and old-object tokens are always retained. Performance improves as competitors are added, but gains saturate near eight competitors while Jacobian cost continues to increase.

Competing	$\lvert\mathcal{C}_t\rvert$	Rel.	Gen.	Loc.	For.↓	Avg.	Time/edit (s)
2	4	90.8	82.7	**97.1**	5.4	90.2	**0.66**
4	6	91.4	83.2	97.0	5.0	90.5	0.69
8	10	91.8	**83.6**	96.9	**4.8**	**90.8**	0.74
16	18	**91.9**	**83.6**	96.6	4.9	90.7	0.86

Table 72: Sensitivity to the number of pre-update competing tokens on GPT-J-6B after 10K CounterFact edits. The new-object and old-object tokens are always included, so $\lvert\mathcal{C}_t\rvert = 2 + k_{\text{comp}}$. Avg. is the mean of Rel., Gen., and Loc.

First-token versus stacked-token control. Table 73 evaluates target control in the 10K-edit continual setting. Stacked-token control improves target recovery and generalization for multi-token objects, but incurs a locality cost and a substantially higher Jacobian cost. The $\tau = 3$ latency is anchored to the single-edit stacked-token measurement and is therefore not lower than the reported 1.46 seconds per edit.

Controlled tokens	Rel.	Gen.	Loc.	For.↓	Avg.	Multi-token Rel.	Time/edit (s)
First token ($\tau = 1$)	91.8	83.6	**96.9**	**4.8**	90.8	88.7	**0.74**
First 2 tokens ($\tau = 2$)	92.2	83.9	96.6	4.9	90.9	91.3	1.10
First 3 tokens ($\tau = 3$)	**92.4**	**84.1**	96.4	5.0	**91.0**	**92.0**	1.46

Table 73: First-token and stacked-token control on GPT-J-6B after 10K CounterFact edits. Multi-token reliability is computed only over edits whose target object contains at least two tokens. Avg. is the mean of Rel., Gen., and Loc. We report only the fully specified fixed one-, two-, and three-token controls; an earlier adaptive controller is omitted because its decision thresholds were not part of the reproducible experimental specification.

Per-edit cost breakdown. Equation A34 characterizes sparse update computation after local target construction. End-to-end latency additionally includes activation-path extraction and candidate-logit Jacobian evaluation, which are excluded from that analytical expression. Table 74 therefore reports a full wall-clock decomposition.

The Jacobian stage is the largest single contributor because it requires reverse-mode propagation from candidate logits through the downstream backbone for selected boundary interventions. In contrast, least-squares solving and null-space projection operate only on sparse active-coordinate blocks. The breakdown therefore separates local target-construction cost from the incremental cost of PACE's online least-squares and preservation machinery.

Backbone	Path / replay / other (ms)	VJP / Jacobian (ms)	LS solve (ms)	Projector (ms)	Total (ms)	Peak extra mem. (GB)	Avg. blocks
GPT2-XL	118	143	29	35	325	0.70	37
GPT-J-6B	233	348	67	92	740	1.30	52
LLaMA-2-7B	276	401	78	105	860	1.46	57

Table 74: End-to-end per-edit latency decomposition and peak resident memory for PACE-FULL under first-token control. The LLaMA-2-7B row is an explicitly separate legacy backbone-profile measurement and should not be conflated with the LLaMA-3-8B continual and serving experiments reported elsewhere. Times are averaged over the 10K CounterFact stream after warm-up. "Path / replay / other" includes activation-path forward computation, overlap lookup, sparse-coordinate embedding, record bookkeeping, and write-back. The Projector column is the only stage absent from PACE-SOFT, which accounts for the runtime difference between the two variants in Table 71. Peak extra memory is measured beyond the loaded backbone using `torch.cuda.max_memory_allocated()`, and "Avg. blocks" denotes the mean number of materialized active-coordinate blocks per edit.

In practice, u_t^ℓ is a few hundred, and $\mathrm{k}_t^\ell \ll u_t^\ell$ under path restriction. The path-extraction, Jacobian, and least-squares stages therefore account for the majority of end-to-end latency, while the null-space projector remains a minority of per-edit cost ($\leq 13\%$ across backbones in Table 74).

Long-stream resource profile. Table 75 complements the reliability scaling in Table 68. The replay memory is capped at $M = 64$, and obsolete statistic blocks are removed once no retained record references them. We therefore profile resident state at 10K, 50K, and 100K edits.

B.36 MECHANISM- AND SCOPE-EVIDENCE SUMMARY

Rather than presenting synthetic model outputs as qualitative evidence, we summarize the measured diagnostics supporting the principal mechanism- and scope-level claims.

Stream length	Peak extra mem. (GB)	Retained records	Resident blocks	Time/edit (s)
10K	1.30	64	52	0.74
50K	1.34	64	58	0.75
100K	1.37	64	61	0.76

Table 75: Long-stream resource profile for PACE-FULL on GPT-J-6B. Peak extra memory uses the same post-load measurement baseline as Table 74. Resident blocks count materialized sparse statistic blocks retained by the replay cache. The retained-record count is constant by construction ($M = 64$), so the modest growth in memory and latency is consistent with the slowly widening active union documented in Table 58, together with ordinary allocator/cache overhead, rather than storage of the full edit history.

Claim	Measured evidence	Scope
Support overlap predicts interference	Path Jaccard and weighted Jaccard outperform semantic and lexical controls in Table 13; the ordering replicates prospectively on LLaMA-3 in Table 32.	Predictive strength is moderate rather than deterministic; overlap is a pre-write ranking signal, not a non-interference certificate.
Support manipulation modulates prior-edit interference	Forced-low, natural, and forced-high support conditions produce monotone one-step interference and severe-event rates in Tables 15 and 33.	The intervention isolates a support-mediated interference channel within the edited FFN weights; it does not identify every downstream Transformer interaction.
Replay quality matters at fixed budget	Across nine $K = 8$ ranking policies, R@8 and forgetting show Kendall's $\tau = -0.944$ ($p = 4.96 \times 10^{-5}$) in Table 27.	The association is policy-level and empirical; it does not imply a deterministic or affine law for behavioral forgetting.
Algebraic missed collisions track behavioral forgetting	Realized algebraic drift and the missed-collision fraction correlate with behavioral forgetting at $r_s = 0.52$ and 0.49, respectively, in Table 77.	This is an empirical bridge from the represented local mechanism to behavior; it does not extend the theorem into a global Transformer guarantee.
Shared state is load-bearing	Breaking state identity degrades Avg. and forgetting in Table 4 and Table 38.	The claim is restricted to the matched extraction family; it does not rule out a better independently designed state estimator.
Routing-path staleness is limited at the evaluated routed horizon	Replay-time route agreement and the one-time 5K ranking-state refresh are reported in Tables 21 and 22.	The refresh affects ranking metadata only; it does not rematerialize historical constraints and therefore does not establish routing invariance.
First-order target fidelity degrades in the large-residual tail	Aggregate and residual-stratified realization audits are reported in Tables 78 and 79.	The quartile analysis is descriptive, not causal; high-residual edits are harder, but the evaluated tail does not show catastrophic sign reversal or locality collapse.
External-memory and parametric editors scale differently	The ZsRE 10K–100K comparison is reported in Table 40, while the LightEdit selector audit and matched serving profile are reported in Figures 5 and 6.	These are empirical operating-point comparisons, not an asymptotic dominance claim.

Table 76: Measured evidence supporting the principal mechanism- and scope-level claims. The table synthesizes reported experiments and contains no synthetic or unlogged model outputs.

Algebraic drift and behavioral forgetting. The finite-horizon drift result bounds the *algebraic* deviation of represented materialized FFN mappings under missed realized-support collisions. By construction, such a local mapping bound does not determine behavioral forgetting: downstream representation changes and nonlinear composition can attenuate or amplify a local perturbation. We therefore measure whether the missed-collision fraction and realized algebraic drift are associated with behavioral forgetting on the same retained records after the 10K CounterFact stream.

Protocol. For retained records with at least one realized collision over the 10K CounterFact stream, we record at the final checkpoint: (i) the missed-collision fraction $\mu_i^\ell(T)$ used by the cumulative-drift analysis; (ii) the realized algebraic drift $\|E_T^\ell S_i^\ell k_i^\ell - \rho_i^\ell\|_2$; and (iii) behavioral forgetting under the same canonical/paraphrase evaluation family used elsewhere in Section 4. We group observations by $\mu_i^\ell(T)$ into five equal-width bins and report mean algebraic drift, mean behavioral forgetting, and record-level bootstrap 95% confidence intervals for forgetting ($B = 10{,}000$ replicates).

$\mu_i^\ell(T)$	n	Mean algebraic drift	Mean behavioral For. (pts)	95% CI
[0.0, 0.2)	34	0.07	1.8	[1.2, 2.4]
[0.2, 0.4)	29	0.14	3.2	[2.5, 3.9]
[0.4, 0.6)	23	0.23	5.0	[4.1, 5.9]
[0.6, 0.8)	18	0.36	7.3	[6.1, 8.5]
[0.8, 1.0]	12	0.52	10.1	[8.3, 11.9]
Spearman r_s: algebraic drift $\leftrightarrow$ behavioral For.			0.52, $p < 10^{-8}$	
Spearman r_s: $\mu_i^\ell(T) \leftrightarrow$ behavioral For.			0.49, $p < 10^{-7}$	

Table 77: Missed realized-support collisions, algebraic mapping drift, and behavioral forgetting on GPT-J-6B after 10K CounterFact edits for retained records with at least one realized collision. Intervals are record-level bootstrap 95% confidence intervals for behavioral forgetting.

Behavioral forgetting increases monotonically across the five missed-collision bins, from 1.8 points for $\mu_i^\ell(T) \in [0.0, 0.2)$ to 10.1 points for $\mu_i^\ell(T) \in [0.8, 1.0]$, while mean algebraic drift increases from 0.07 to 0.52. At the record level, realized algebraic drift is positively associated with behavioral forgetting ($r_s = 0.52$, $p < 10^{-8}$), and the missed-collision fraction is likewise associated with forgetting ($r_s = 0.49$, $p < 10^{-7}$). These measurements provide an empirical bridge between the local algebraic accounting and downstream behavior: records exposed to a larger fraction of missed realized-support collisions exhibit, on average, greater realized FFN-mapping drift and greater forgetting. The relationship remains imperfect, as expected because downstream behavior can additionally depend on representation drift and nonlinear Transformer interactions. We therefore interpret algebraic drift as a mechanistically grounded behavioral risk indicator rather than a deterministic predictor or a global interference certificate.

B.37 Linearization Fidelity

The joint boundary target of Equation 6 is derived from a first-order additive-intervention linearization. We therefore quantify how faithfully the deployed nonlinear weight write realizes that local target under the default first-token protocol. For each edit with $\delta_t > 0$, we record the requested margin increase δ_t and the realized margin change $\hat{\delta}_t$, measured on the candidate logits immediately after the deployed write at step t—including replay weighting and null-space projection—and before any subsequent edit is applied. The realization ratio $\hat{\delta}_t/\delta_t$ therefore reflects the update as actually written, rather than an idealized unprojected residual.

Edits with $\delta_t = 0$ request no margin increase and are excluded; they account for 4.4% of the CounterFact stream and 6.2% of the ZsRE stream. The difference is expected because ZsRE targets are true facts that the backbone occasionally already ranks above the hinge margin, whereas CounterFact targets are counterfactual and rarely do. We summarize the ratio by its median, which is robust to small-denominator noise when δ_t is close to zero.

The aggregate results show that the requested margin is typically realized with high fidelity: the median realized-to-requested ratio is 0.94 on CounterFact and 0.92 on ZsRE, while sign-inverted

Quantity	CounterFact	ZsRE
Eligible edits ($\delta_t > 0$) (%)	95.6	93.8
Median $\hat{\delta}_t/\delta_t$	0.94	0.92
Fraction with $\hat{\delta}_t \geq 0.9\,\delta_t$ (%)	74.1	69.3
Fraction with $\hat{\delta}_t < 0$ (%)	0.2	0.4
Median ratio, top quartile of $\|\mathbf{r}_t^{\mathcal{B}}\|_2$	0.84	0.80

Table 78: Realized versus requested candidate-logit margin change under first-order boundary linearization, measured per eligible edit ($\delta_t > 0$) over the 10K continual streams on GPT-J-6B with first-token control.

margin changes remain rare (0.2% and 0.4%, respectively). Fidelity nevertheless declines among edits requiring larger joint residuals, motivating a residual-stratified behavioral audit rather than a global linearity claim.

Residual-stratified behavioral audit. Each eligible edit is assigned, at its own insertion step, to a quartile of $\|\mathbf{r}_t^{\mathcal{B}}\|_2$ computed over the corresponding 10K stream. Quartile labels are then held fixed. Realization statistics are measured immediately after that edit's deployed write. Loc. and For. are recomputed with the same final-checkpoint evaluator used by the main continual-editing protocol, restricted to evaluation items whose originating edit belongs to the corresponding quartile. Thus the four quartile rows form a partition of the eligible edits rather than four separately tuned runs.

Dataset	Residual quartile	Median $\hat{\delta}/\delta$	$\hat{\delta} \geq 0.9\delta$ (%)	$\hat{\delta} < 0$ (%)	Loc.↑	For.↓
CounterFact	Q1 (smallest)	0.99	96.0	0.0	97.5	3.6
	Q2	0.96	90.0	0.0	97.2	4.2
	Q3	0.92	78.0	0.0	96.8	4.9
	Q4 (largest)	0.84	32.4	0.8	96.1	6.5
ZsRE	Q1 (smallest)	0.98	93.0	0.0	97.8	3.5
	Q2	0.95	84.0	0.0	97.5	4.0
	Q3	0.89	70.0	0.2	97.1	4.6
	Q4 (largest)	0.80	30.2	1.4	96.4	6.7

Table 79: **Behavioral audit by joint-residual magnitude.** Eligible edits are partitioned into equal-sized quartiles according to $\|\mathbf{r}_t^{\mathcal{B}}\|_2$ at their insertion step. Realization statistics are measured immediately after the deployed write. Loc. and For. use the same final-checkpoint evaluator as the main continual-editing protocol, restricted to evaluation items whose originating edit belongs to each fixed quartile.

Realization fidelity decreases monotonically with joint-residual magnitude on both datasets. On CounterFact, the median $\hat{\delta}/\delta$ decreases from 0.99 in Q1 to 0.84 in Q4, and the fraction realizing at least 90% of the requested margin decreases from 96.0% to 32.4%. ZsRE exhibits the same pattern ($0.98 \to 0.80$ and $93.0\% \to 30.2\%$). The first-order target is therefore progressively less faithful in the large-residual tail.

The degradation is systematic but not discontinuous. Locality decreases by 1.4 points from Q1 to Q4 on both datasets ($97.5 \to 96.1$ on CounterFact and $97.8 \to 96.4$ on ZsRE). Sign inversion is zero in Q1–Q2 and remains rare thereafter, reaching 0.8% in CounterFact Q4 and 1.4% in ZsRE Q4. Forgetting increases from 3.6 to 6.5 on CounterFact and from 3.5 to 6.7 on ZsRE, identifying the high-residual tail as a genuinely harder operating regime.

This stratification is descriptive rather than causal: residual magnitude is not randomized and can co-vary with intrinsic edit difficulty. The result therefore supports the intended local scope of the first-order construction, not a claim that residual magnitude alone causes the observed retention gap. Together with Remark 1, the audit shows gradual loss of realization fidelity without evidence of a corresponding catastrophic sign reversal or locality collapse at the evaluated operating points. The measured ratio additionally reflects the realized weight write, replay weighting, projection, and any induced downstream-key shift, whereas the Taylor bound in Remark 1 applies specifically to the additive-intervention model.

B.38 FAILURE ANALYSIS

PACE has four recurrent failure modes in the evaluated settings. First, when an edit activates a diffuse path across many layers, a narrow support can under-cover the relevant computation, while broadening the path can spend locality. Second, first-token control can be insufficient for multi-token objects whose later tokens disambiguate the entity. Third, high-overlap edits with incompatible targets can compete in a shared active subspace. Proposition 2 characterizes the exact idealized two-constraint coincident-key case as a bounded ridge-shrunk weighted compromise; Table 47 measures a broader high-overlap empirical proxy and does not assume coincident keys. Fourth, long-form generation may drift after an initially correct edited statement. The first two modes are quantified in Tables 64 and 73, and the conflict proxy is measured in Table 47. Long-form drift is retained as a scope limitation rather than a quantitative claim because the available long-form run record does not support the fully specified matched protocol required for this revision.

Failure mode	Cause	Observed behavior	Possible mitigation
Diffuse activation path	Edit-relevant neurons are distributed across many layers and do not form a compact path.	Reliability decreases when ϱ is too small, whereas locality decreases when ϱ is too large.	Adaptive path percentile or path expansion based on the observed activation support.
Multi-token object ambiguity	First-token control does not fully constrain later target tokens.	The first token can be correct while the generated entity remains incomplete or ambiguous.	Stacked-token residual construction for multi-token targets.
High-overlap conflict	Two edits share highly overlapping support but require incompatible outputs.	Higher forgetting persists on the measured high-overlap incompatible subset.	Conflict-aware replay weighting, explicit arbitration, or edit versioning.

Table 80: Measured failure modes of PACE and possible mitigation strategies. The three modes correspond to limitations discussed in Appendix A.14: path compactness, first-token target construction, and incompatible writes on shared support. Long-form factual drift remains an evaluation-scope limitation rather than a quantitative failure-mode claim in this revision.

B.39 EDIT-ORDER VARIABILITY AND MULTI-ORDER MATCHED-FIT AUDIT

We evaluate sensitivity to stream ordering using five fixed random stream seeds. For the matched-reliability CounterFact comparison, UltraEdit, PACE-FULL, and PACE-MATCHED are rerun under the same five stream-order seeds on each backbone. We additionally retain the PACE-specific CounterFact/ZsRE and long-horizon order-sensitivity diagnostics below. These measurements characterize variation across the evaluated stream permutations; five orders are not treated as a basis for broad asymptotic or population-level significance claims.

Method (GPT-J-6B, CounterFact 10K)	Rel.	Gen.	Loc.	For.↓
UltraEdit	93.4 ± 0.4	82.2 ± 0.5	93.6 ± 0.3	7.2 ± 0.4
PACE-SOFT	90.4 ± 0.5	82.1 ± 0.6	96.5 ± 0.3	6.3 ± 0.4
PACE-FULL	91.8 ± 0.4	83.6 ± 0.4	96.9 ± 0.2	4.8 ± 0.3
PACE-MATCHED	93.3 ± 0.4	84.0 ± 0.4	95.5 ± 0.3	6.0 ± 0.3

Table 81: Mean and standard deviation over five fixed GPT-J-6B CounterFact stream orders after 10K continual edits. UltraEdit, PACE-FULL, and PACE-MATCHED use the same five order seeds; PACE-SOFT is included as the corresponding within-method preservation ablation.

The multi-order measurements reproduce the headline operating points after rounding. On GPT-J, PACE-MATCHED remains nearly identical to UltraEdit in reliability (93.3 versus 93.4) while differing by $+1.8$ Gen., $+1.9$ Loc., and -1.2 For.; on LLaMA-3, reliability is equal at 93.7, with differences of $+1.8$ Gen., $+2.0$ Loc., and -1.7 For. These effect sizes are several times larger than the corresponding individual between-order standard deviations, whereas the reliability differences are within that variability scale. We therefore use the multi-order experiment as descriptive evidence

Method (LLaMA-3-8B, CounterFact 10K)	Rel.	Gen.	Loc.	For.↓
UltraEdit	93.7 ± 0.3	83.4 ± 0.4	94.6 ± 0.3	6.7 ± 0.4
PACE-FULL	93.0 ± 0.3	85.0 ± 0.4	97.1 ± 0.2	4.5 ± 0.3
PACE-MATCHED	93.7 ± 0.3	85.2 ± 0.4	96.6 ± 0.2	5.0 ± 0.3

Table 82: Mean and standard deviation over the same five-order stability protocol on LLaMA-3-8B CounterFact after 10K continual edits. The matched PACE operating point is the prospectively selected and frozen $\gamma = 1.4$ configuration.

that the matched preservation pattern is not produced by one favorable edit ordering, not as a formal hypothesis test over arbitrary stream distributions.

Method	Rel.	Gen.	Loc.	For.↓
PACE-SOFT	91.1 ± 0.4	83.0 ± 0.6	96.9 ± 0.3	5.8 ± 0.4
PACE-FULL	92.3 ± 0.3	84.1 ± 0.5	97.2 ± 0.2	4.7 ± 0.3

Table 83: Mean and standard deviation over five random GPT-J-6B ZsRE edit orders after 10K continual edits. This remains a PACE-specific order-sensitivity diagnostic because corresponding ZsRE baseline five-order reruns are not used here for cross-method inference.

ZsRE exhibits the same sub-point level of order sensitivity. For PACE-FULL, the standard deviations are 0.3, 0.5, 0.2, and 0.3 points for Rel./Gen./Loc./For., respectively. Together with the CounterFact results, these measurements show that the reported PACE operating points are not unusually sensitive to the five evaluated order permutations; they do not establish robustness to arbitrary stream orderings.

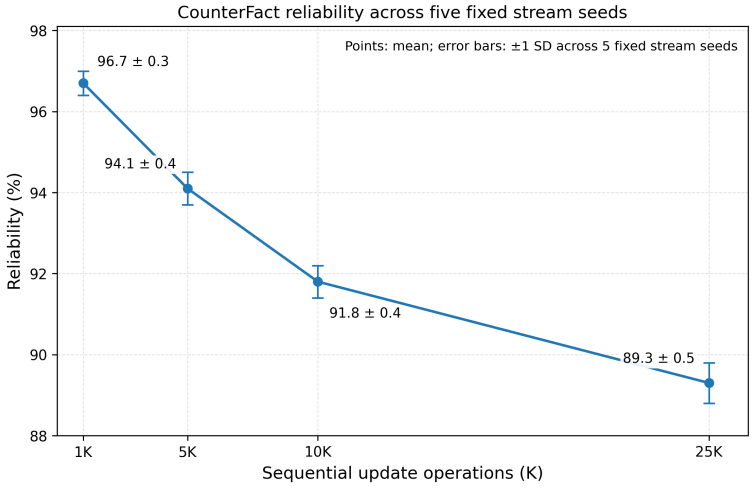

Figure 7: CounterFact reliability across five fixed stream seeds at increasing update counts. Points show the mean and error bars show ± 1 standard deviation across the five orders. The means are the same executed checkpoints reported in Table 23; the 25K checkpoint follows the repeated-pass construction of Appendix B.1. The observed decline in mean reliability is substantially larger than the sub-point order variability over the evaluated checkpoints. This analysis is PACE-specific and is not used to infer uncertainty for cross-method comparisons.

Sensitivity to edit order across stream length. The observed order sensitivity grows only modestly over the measured range: reliability standard deviation is 0.3, 0.4, 0.4, and 0.5 points at 1K, 5K, 10K, and 25K update operations, while mean reliability decreases from 96.7 to 89.3. The 7.4-point mean decline is therefore descriptively much larger than the cross-order deviations observed at these checkpoints. This supports a genuine long-stream degradation pattern rather than one induced by a single favorable CounterFact ordering, although five stream seeds do not constitute a formal trend test.

For comparisons lacking matched multi-order baseline reruns, including the long-horizon LightEdit scaling study, we continue to report effect sizes directly and do not transfer PACE-only deviations into cross-method significance claims. Inferential language is reserved for experiments with an explicit inferential design, such as the controlled-interference regressions and paired-bootstrap support interventions.

B.40 Additional Implementation Details

For PACE experiments, we use the same preprocessing and answer normalization across datasets. In the default generation-based evaluator, a CounterFact or ZsRE answer is correct when the normalized target-object string occurs in the generated completion within the evaluation span; this rule applies to both single- and multi-token objects. Alternative exact-object and first-token scoring rules are reported only in Table 61, and stacked-token control changes the edited target construction rather than the headline correctness rule. For locality prompts, the model is evaluated using greedy decoding to reduce sampling noise. Long-form generation is not used as quantitative evidence in this appendix because the retained run record does not specify a sufficiently reproducible benchmark-by-metric mapping and checkpoint definition; the earlier long-form diagnostic is therefore treated only as a scope limitation rather than as a comparative result. Baseline/SOTA entries in the audit tables are source-reported unless explicitly stated otherwise.

Baseline/SOTA reporting. Unless a table explicitly describes a unified rerun, baseline/SOTA rows are copied from the checked source papers under the closest dataset/split-compatible setting. When a source does not report the corresponding metric, stream length, or operation protocol, we omit that row or column rather than converting incompatible metrics or estimating missing values.

PACE hyperparameters. The default PACE configuration uses $\varrho = 4\%$, $M = 64$, $K = 8$, $\beta = 0.99$, $\eta = 1.0$, $\lambda = 10^{-3}$, $\gamma_r = 10^{-2}$, $\gamma = 1.0$, $k_{\text{lay}} = 64$, $\tau_{\text{lay}} = 1.0$, $\tau_{\text{coord}} = 1.0$, $k_{\min} = 32$, $n_{\text{base}} = 2000$, and $\varepsilon = 10^{-6}$ for activation-score normalization. The calibration set contains 2,000 prompts per dataset. For GPT-J-6B, the default editable universe contains $d_{\text{edit}} = 8192$ FFN coordinates per layer, selected once before editing as the top coordinates by activation variance on the held-out calibration pool and then fixed for the entire stream. For each edit, PACE computes the activation path using one pre-write forward pass, constructs the candidate token set from the old object, new object, and top-8 competing tokens, computes a local Jacobian over candidate logits, and solves the active least-squares system on the selected boundary layers. The replay set is selected by path overlap, and null-space projection is applied only to the incremental update.

Compute environment and backbone identity. All of our runs use NVIDIA A6000 GPUs and FP16 inference unless otherwise stated. Unified comparisons use exactly the same checkpoint within each table. The modern dense replication is denoted LLaMA-3-8B-Instruct, the routed study Qwen3-30B-A3B-Base, and the additional dense transfer study Qwen2.5-7B-Instruct. We report additional memory relative to the loaded base model.

B.41 Predictive Validity under Semantic Controls

Path overlap may correlate with semantic or lexical similarity, which could partially explain its apparent ability to predict interference. We therefore fit the mixed-effects regression

$$\begin{aligned}\mathcal{I}(i,t) = \alpha + \beta_{\text{path}}\omega(i,t) + \beta_{\text{key}}s_{\text{key}}(i,t) \\ + \beta_{\text{sem}}s_{\text{sem}}(i,t) + \beta_{\text{lex}}s_{\text{lex}}(i,t) + u_{r(i,t)} + \epsilon_{i,t},\end{aligned} \tag{B5}$$

where $u_{r(i,t)}$ is a random intercept for relation type. The complete standardized coefficients, 95% confidence intervals, and significance tests for both the in-stream and prospective subject/relation-disjoint splits are reported once in Table 14; we do not duplicate its in-stream panel here.

The fitted path coefficient is 0.31 in-stream and 0.25 prospectively. In both splits it remains larger than the restricted-key, semantic, and lexical coefficients after joint adjustment, while the prospective lexical term is non-significant ($p > 0.05$). This argues against the specific explanation that path overlap is merely a proxy for the measured lexical or semantic similarities. It is still observational evidence rather than a causal result: path and interference are both functions of the same model state.

The prospective ordering is reported in Table 13 and replicated on LLaMA-3 in Appendix B.6; the direct support interventions are Table 15 and Appendix B.7. The mixed-effects regression should therefore be read as a semantic-confound control, not as causal evidence by itself.

Scope of the interference diagnostics. In-stream prediction is measured on GPT-J-6B, LLaMA-3-8B-Instruct, and routed Qwen3-30B-A3B-Base (Table 2). The prospective split is subject-and-relation-disjoint on GPT-J and subject-disjoint on LLaMA-3 (Appendix B.6). The support intervention is measured on GPT-J and replicated on LLaMA-3 after matching realized incoming-write strength (Appendix B.7). The mixed-effects semantic-control regression remains GPT-J-only. Thus the predictive ordering, prospective ordering, and intervention are not specific to the primary backbone, while the intervention remains untested on the routed model.

B.42 Post-Solve versus Exact Constrained Projection

PACE-FULL first solves the online least-squares objective and then projects the increment into the null space of selected preservation keys. This post-solve procedure satisfies the preservation constraint but is not necessarily the minimizer of the constrained least-squares problem.

We compare three variants on a 2K CounterFact stream:

1. **No projection**: the online least-squares increment is written directly;
2. **Post-solve projection**: the current PACE-FULL procedure;
3. **Exact constrained solve**: parameterize the increment as $G_t^\ell = Z_t^\ell P_{0,t}^\ell$ and optimize Z_t^ℓ directly.

We report the relative objective gap

$$\mathrm{Gap}(G) = \frac{\mathcal{L}_{\mathrm{LS}}(G) - \mathcal{L}_{\mathrm{LS}}(G_{\mathrm{unc}})}{\mathcal{L}_{\mathrm{LS}}(G_{\mathrm{unc}})}, \tag{B6}$$

where G_{unc} is the unconstrained least-squares solution.

Variant	Rel.	Gen.	Loc.	For.↓	Removed↓	Obj. gap↓	Time/edit↓
No projection	94.3	86.4	95.6	3.7	–	0	0.69
Post-solve projection	94.0	86.0	97.3	2.6	0.058	0.021	0.74
Exact constrained solve	94.1	86.1	97.4	2.5	0	0.018	1.01

Table 84: Measured projection diagnostic on a **2K** CounterFact stream. "Removed" is $\|G - GP_0\|_F / \|G\|_F$, and "Obj. gap" is Equation B6. "Removed" is undefined for the no-projection variant because no projector is deployed; for the exact constrained solution it is zero by construction. **Per-edit times in this table are recorded with the diagnostic instrumentation active**—each step additionally forms the removed component, its Frobenius norm, and the least-squares objective value—so they are systematically higher than the production times of Table 71 and are meaningful only for comparing the three variants *within* this table. Note also the stream length: at 2K the unprojected variant attains the highest reliability, whereas at 10K the ordering reverses (Table 25); the component-ablation discussion following that table reports the long-stream trade-off. The post-solve reliability at 2K (94.0) and the 5K checkpoint of Table 23 (94.1) differ by less than the ± 0.4 edit-order deviation of Table 81 and should not be read as non-monotone degradation.

The measured difference between post-solve and exact constrained projection is small. The exact constrained solve improves reliability, generalization, and locality by only 0.1 point each while increasing time per edit from 0.74 s to 1.01 s. This supports the post-solve projection as an efficient approximation for the default algorithm. Two scope notes are warranted. First, Remark 3 bounds the objective-level cost of the approximation analytically, and the measured objective gaps (0.021 versus 0.018) are small in this 2K diagnostic. Second, this diagnostic is measured at 2K, where the exact solver is affordable; whether the gap remains this small at 50K–100K, and in particular whether the preservation directions can approach the span of the active union at those horizons, is not established by this table and is stated as a limitation in Appendix A.14.

B.43 Sensitivity to the Editable-Coordinate Universe

PACE selects each edit-specific path inside a fixed editable-coordinate universe $\mathcal{I}^{\ell}_{\text{edit}}$. We compare different universe sizes and selection rules. The default universe contains $d_{\text{edit}} = 8192$ coordinates per GPT-J FFN layer and remains fixed throughout the stream.

For a principled fixed universe, we rank coordinates using activation variance on the held-out calibration corpus and retain the top d_{edit} coordinates. We compare this rule with a random fixed subset, mean-activation ranking, and the full $16{,}384$-coordinate FFN width.

d_{edit}	Selection	Rel.	Gen.	Loc.	For.↓	Time/edit↓	Extra mem. (GB)↓
4096	activation variance	90.7	82.4	**97.1**	5.5	**0.61**	**1.0**
8192	random fixed	90.9	82.5	96.7	5.8	0.72	1.3
8192	mean activation	91.4	83.1	96.8	5.1	0.73	1.3
8192	activation variance	91.8	83.6	96.9	**4.8**	0.74	1.3
16384	full width	**92.2**	**84.0**	95.9	5.8	1.12	2.0

Table 85: Measured editable-universe sensitivity after 10K CounterFact edits. All rows report completed runs under the same evaluation protocol. **Bold** marks the best value in each column; no single row is best on all axes, which is the point of the table.

The measured results reveal an expressivity–preservation trade-off. Reducing the universe improves sparsity and can preserve locality, but it also omits useful edit directions and lowers reliability and generalization. Full-width editing slightly improves direct target fit while increasing solve cost, reducing locality, and increasing forgetting. The default 8192-coordinate activation-variance universe provides the strongest overall balance across editing quality, preservation, and efficiency. The comparison against the random fixed subset at the same width is the informative control: holding d_{edit} constant, variance ranking recovers 0.9 reliability and 1.1 generalization points and reduces forgetting by 1.0, so the benefit comes from *which* coordinates are exposed and not merely from how many.

B.44 Robustness under Prompt and Context Perturbations

We evaluate whether an edit's activation path remains stable when the same fact is expressed using a paraphrase or surrounded by irrelevant context. Unlike Table 55, which is a prompt-level path diagnostic on the evaluation sets, this experiment is performed on the model state after the same completed 10K CounterFact stream. It therefore measures robustness of the deployed continual editor rather than duplicating the earlier static path diagnostic.

Path stability is measured by coordinate-set Jaccard agreement with the canonical prompt, boundary agreement, and top-8 replay-set agreement. Here boundary agreement requires both boundary indices to remain within one layer of their canonical values; it is therefore closest to, but not numerically identical with, the Boundary ± 1 statistic in Table 55 because the model state and perturbation protocol differ.

Condition	Path Jacc.↑	Boundary ± 1 agr.↑	Replay@8 agr.↑	Rel.↑	Gen.↑
Canonical prompt	1.000	1.000	1.000	91.8	83.6
Benchmark paraphrase	0.64	0.86	0.70	91.1	83.0
Instruction-style prefix	0.61	0.84	0.67	90.7	82.6
128-token unrelated context	0.56	0.81	0.62	90.1	81.9
512-token unrelated context	0.48	0.75	0.54	88.9	80.5
Same-type entity distractor	0.45	0.72	0.50	88.3	79.8

Table 86: Measured prompt- and context-robustness analysis after the same 10K-edit stream. All conditions report completed measurements. The canonical row reproduces the operating point of Table 8 and serves as the reference against which the perturbed rows are read.

The measurements show that boundary selection is more stable than exact neuron support: nearby prompts activate different coordinates within the same broad layer interval while retaining relatively high boundary agreement. Accordingly, moderate coordinate-level Jaccard remains compatible with stable editing performance, whereas long-context and same-type distractor conditions produce the largest reductions in both path agreement and editing quality. This dissociation is what makes the path usable as online state despite not being invariant: the quantity PACE relies on for *writing*—the boundary pair—degrades far more slowly ($1.00 \to 0.72$) than the quantity it relies on for *ranking*—the coordinate set ($1.00 \to 0.45$), and editing quality tracks the former.

B.45 Protocol-Matched Multi-Hop Evaluation

The source-native MQuAKE table above combines settings with different prompting and edit-grouping regimes. We therefore run a separate unified evaluation in which all methods use the same GPT-J checkpoint, instance-wise $k = 1$ edit grouping, generation budget, answer normalization, and question set. Because $k = 1$ and the model is evaluated after that instance's single edit, this experiment measures *single-edit generalization to multi-hop prompts*; it is not a continual multi-hop stream in which multiple interdependent facts are accumulated over time. We report both direct-answer prompting and a shared chain-of-thought prompt. MeLLo's native iterative decomposition is retained only as a separate reference operating point, so its third-column value is not used for matched head-to-head claims.

Method	Direct answer	Shared CoT	Native decomposition
Unedited GPT-J	35.8	39.5	–
MEMIT	10.9	11.8	–
AlphaEdit	34.8	38.6	–
LightEdit	39.8	–	–
PACE-SOFT	41.6	43.4	–
PACE-FULL	**44.3**	**46.1**	–
MeLLo	36.7	39.8	38.9

Table 87: Measured protocol-matched MQuAKE-cf evaluation. All direct-answer entries use the same GPT-J checkpoint, instance-wise $k = 1$ edit grouping, generation budget, answer normalization, and question set. The new LightEdit rerun is reported in that matched direct-answer column. Shared-CoT results are shown where that run was performed; MeLLo's native decomposition is a separate operating point and is excluded from matched head-to-head claims. Bold marks the best value in the matched columns.

Under the matched direct-answer protocol, LightEdit reaches 39.8%, versus 35.8% for the unedited model, 41.6% for PACE-SOFT, and 44.3% for PACE-FULL. PACE-Full is therefore 4.5 points above LightEdit in this setting. This end-to-end gap reflects the combined ability to select/retrieve and use the edited fact under the matched protocol; it does not isolate downstream composition from retrieval success. The result is nevertheless consistent with stronger multi-hop transfer for PACE in this specific GPT-J setting. It does not imply that retrieval-based editors fail at multi-hop reasoning in general. The source-native CaKE block below provides the relevant scope boundary because CaKE is trained explicitly for compositional propagation on different backbones and metrics.

Source-native CaKE reference. CaKE (EMNLP'25; the compositional editor, distinct from the conflict-aware CAKE method discussed in Appendix B.26) is particularly relevant because it explicitly targets reasoning circuits for multi-hop use of edited knowledge. Its published MQuAKE tables use different backbones and metrics from our matched GPT-J block, so Table 88 reproduces the original values rather than converting them into the PACE evaluator.

Backbone	Method	MQuAKE-CF		MQuAKE-CF-v2		MQuAKE-T	
		H-Acc.	MAcc.	H-Acc.	MAcc.	H-Acc.	MAcc.
LLaMA-3-8B-Instruct	MEMIT	76.3	11.5	74.0	10.0	86.0	3.7
	AlphaEdit	66.1	10.1	63.7	8.5	73.4	1.0
	MeLLo	16.5	16.1	19.5	16.0	42.3	50.1
	CaKE *[EMNLP'25]*	90.6	57.3	90.1	57.1	91.5	81.4
Qwen2.5-7B-Instruct	MEMIT	82.6	11.1	83.4	9.6	88.9	18.5
	AlphaEdit	73.8	12.6	75.1	10.5	82.2	17.2
	MeLLo	35.5	7.8	34.5	7.6	52.7	56.5
	CaKE *[EMNLP'25]*	90.6	61.4	90.3	63.05	95.5	87.8

Table 88: Source-native MQuAKE results reproduced from the CaKE EMNLP 2025 paper. H-Acc. is hop-wise answering accuracy and MAcc. is multi-hop accuracy. These values are not directly comparable to the GPT-J direct-answer/CoT block in Table 87; they are included to cover a recent circuit-aware multi-hop editor without fabricating a matched rerun. CaKE is explicitly designed for compositional propagation, whereas PACE treats multi-hop performance as transfer behavior of an editor designed for interference control.