BINDING VALIDATION EVIDENCE TO RECOVERABLE LEARNING STATE

Anonymous authors
Paper under double-blind review

ABSTRACT

A continually adapting model can restore its weights while retaining the replay state that caused a failure. Adaptive Plasticity Architecture (APA) treats the complete learning state as the object of validation and recovery. A publication binds one frozen candidate to its captured parent and recorded evidence. Recovery restores ancestor content under a fresh activation version while preserving operational history and spent audit allocations. We give conditional consistency guarantees and test the individual checks with trained digit adapters. Removing the parent check permits a stale sibling to erase previously acquired classes. Content hashes alone permit evidence reuse after rollback. Matched sequential checkpoint controls follow identical learning trajectories. An independent 256-edit CounterFact extension compares screened and unscreened memory under shared retrieval rules and includes a compatible GRACE baseline with eight optimization steps per edit. Controlled memory insertions test incorrect targets and conflicting aliases. A durable reader race found during review is repaired. Reusing immutable snapshot references reduces measured publication cost without removing history growth. Bounded auditing provides a separate lifetime acceptance guarantee under explicit sampling assumptions. These results distinguish the consistency of a learning transition from the quality of the learning policy that proposes it.

1 INTRODUCTION

A replay buffer can survive the update that made it harmful. Restoring weights then leaves the next optimizer step exposed to the same data. Retrieval records and retained optimizer state create similar dependencies. Low-rank adaptation (Hu et al., 2022) and retrieval (Lewis et al., 2020) supply useful update mechanisms. Their outputs still need a precise boundary between a state that was evaluated and a state that becomes active.

Concurrent construction makes that boundary harder to maintain. Two workers may improve different tasks from the same parent. Publishing the slower candidate after the first has committed can erase the first task. Recovery creates another ambiguity. A proposal prepared at version 7 must remain stale after version 8 fails and version 9 restores version 7's content. The bytes match but the activation does not. This is the ABA problem.

APA makes these dependencies explicit. The worker captures a parent before construction. Validation records evidence for that parent and the frozen candidate. Publication rechecks both identities. Recovery restores the complete declared learning state under a fresh version. Figure 1 shows the boundary. The mechanism assumes a trusted construction path and cannot infer validator accuracy from a hash. It also cannot retract outputs or external actions already exposed.

The contribution is a contract for binding recovery to learning state and validation evidence with executable tests of its assumptions. The underlying locks and checkpoints are established techniques. Our experiments isolate the failures each check prevents using the same trained candidates. They also expose limits. A complete checkpoint with matched decisions follows the same learning trajectory. Longer language-model streams and controlled memory faults assess the policies carried by the contract. Appendix A states the implemented state boundary. Learned routing and semantic consolidation are outside the empirical contribution.

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.

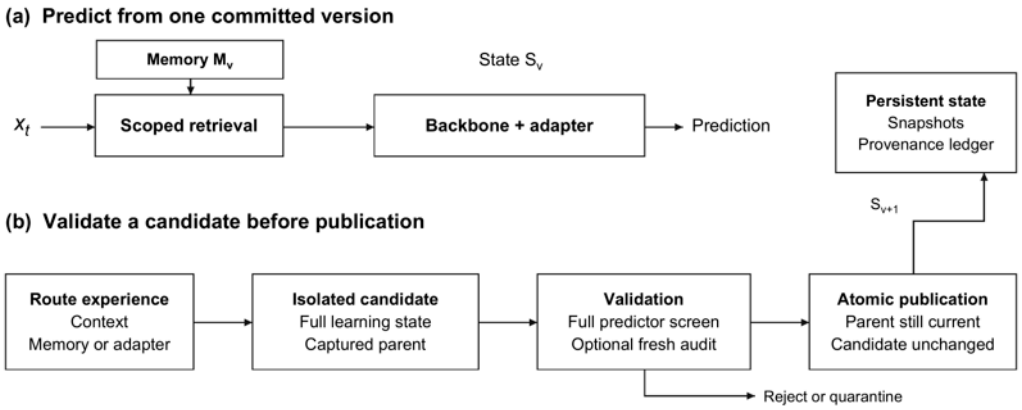


Figure 1: APA separates inference from persistent adaptation. Validation and publication refer to the same complete candidate and captured parent.

2 Related work

Optimistic concurrency control separates speculative computation from validation and commit (Kung & Robinson, 1981). Recovery systems for machine learning already retain state beyond weights. CheckFreq records data-iteration progress (Mohan et al., 2021). Check-N-Run captures model and optimizer state together with metric and reader state (Eisenman et al., 2022). APA applies these established ideas to evidence-bearing adaptive transitions. Its scope is the binding of a constructed candidate to its parent and validation evidence while distinguishing restorable content from monotonic operational history. We do not claim a new concurrency-control or checkpointing primitive.

Continual learning addresses interference between new and prior knowledge through mechanisms such as complementary learning systems and elastic weight consolidation (McClelland et al., 1995; Kirkpatrick et al., 2017). Nested Learning and Titans study nested optimization and persistent memory (Behrouz et al., 2025a;b). Model editing supplies additional proposal mechanisms. ROME and MEMIT modify factual associations in weights (Meng et al., 2022; 2023). SERAC and GRACE use edit memories while WISE separates knowledge across memories (Mitchell et al., 2022; Hartvigsen et al., 2023; Wang et al., 2024a). AlphaEdit constrains parameter updates. NeuralDB and HiEdit address scalable and lifelong editing (Fang et al., 2025; Fei et al., 2026; Wang et al., 2026). EasyEdit supports implementation and WikiBigEdit supports long-stream evaluation (Wang et al., 2024b; Thede et al., 2025). Successful factual edits can still harm general behavior (Gu et al., 2024). APA governs how proposals are published and recovered. We compare a compatible GRACE implementation on the same new stream. This restricted comparison does not establish a general ranking of editing methods.

Reusing a holdout for adaptive decisions can invalidate ordinary generalization arguments (Dwork et al., 2015). Our fresh-audit path combines existing concentration bounds (Hoeffding, 1963; Maurer & Pontil, 2009) with a lifetime error allocation. Safe policy improvement with baseline bootstrapping also makes baseline-relative decisions under uncertainty (Laroche et al., 2019). It studies a different decision object. The audit here concerns a fixed candidate and declared bounded losses. Its guarantee is separate from the reused empirical screens. A neural-loss integration test targets an explicitly finite benchmark distribution.

3 Architecture and state contract

3.1 Prediction state and operational history

Let c be the request context and b_v an immutable backbone identifier. The predictor at activation v is

$$f_v(x, c) = f(x, c, W_{b_v}, \theta_v, \mathcal{R}(x, c, M_v^{\mathrm{mem}}, I_v, C_v)) . \tag{1}$$

The fast trainable parameters are θ_v and persistent retrieval memory is M_v^{mem}. State I_v governs retrieval and indexing while C_v governs routing and control. Time and other external inputs to expiration or scope rules are explicit request inputs. Replay memory M_v^{rep} influences future adaptation even when it does not participate in inference. The rollbackable learning state is

$$S_v = (b_v, \theta_v, M_v^{\text{rep}}, M_v^{\text{mem}}, I_v, O_v, R_v, C_v). \tag{2}$$

The remaining components O_v and R_v hold retained optimizer and random-number-generator state. Completeness is relative to a declared computation boundary. Every mutable component that can affect prediction or a future learning transition must be retained or uniquely reconstructible. Snapshots pin immutable backbone and configuration identifiers. Recovery must not silently substitute different dependencies.

APA keeps monotonic operational state H_v separately. It contains activation and audit counters together with append-only publication and recovery history. Execution depends on (S_v, H_v). Restoring S_u does not recreate (S_u, H_u). Spent audit allocations remain spent and restored content receives a fresh version. Write a computation as $T(S, H, e, \xi, K) \to (o, S^+, H^+)$. Here e is an external event and ξ is the supplied or restored randomness. The pinned dependencies are K. Equal learning states reproduce predictions under equal request inputs and deterministic execution. Equal subsequent transitions additionally require all applicable operational inputs to agree.

Consider a replay target admitted during a harmful update. Weight-only recovery may restore the immediate prediction while leaving that target available to the next optimization step. Restoring the replay buffer removes this mismatch. Retained optimizer momentum and stateful routing can cause the same problem. Appendix C.2 gives explicit counterexamples.

Working context remains request-scoped unless promoted through a persistent proposal. Retrieval filters entries by scope and expiration and by membership in the surviving content lineage. This lineage must be retained or reconstructible from the restored state and its pinned manifest. It differs from append-only publication history. Recovery invalidates descendant content without erasing its activation record. Conflicting entries are quarantined or reviewed. Provenance and confidence metadata do not establish correctness. The language-model experiments use supplied CounterFact subjects and deterministic lexical matching.

3.2 Parent capture and evidence binding

A worker atomically captures active identity (v_p, h_p) and its immutable snapshot before construction. The digest is $h_p = h(\text{enc}(S_{v_p}))$ for canonical stored representation enc. Construction uses a detached copy of that snapshot together with declared update data and immutable configuration. Its randomization policy is also declared. It cannot read mutable learning state activated after capture. The complete candidate C is frozen before validation and has digest $h_C = h(\text{enc}(C))$. Its proposal package is

$$P = (v_p, h_p, C, h_C, \pi, \kappa, E). \tag{3}$$

Provenance π identifies permitted observations and the construction policy. Configuration identifier κ pins immutable dependencies. Evidence E records the parent and candidate identities together with the comparator and evaluation specification. It also records the measured results and decision.

Validation cannot change the persistent candidate representation. Evaluation-local tensors and caches are permitted under the declared randomness policy. Approval of one candidate cannot authorize a modified candidate. The authoritative publication check runs under a single-writer lock. The active identity must still equal (v_p, h_p). Evidence must approve that parent and h_C. The recomputed candidate digest must match. Assuming no collisions among encountered digests these checks publish the exact evaluated bytes against the current parent.

The construction restriction is a separate premise. A hash check cannot prove which state an arbitrary worker read while building a candidate. Lemma 1 therefore assumes a trusted construction path that enforces the captured-parent rule. Without that premise the checks establish identity binding but not computational lineage. Capturing a parent only after optimization is insufficient.

Activation versions distinguish repeated occurrences of identical content. Recovery may restore an earlier digest but must advance the version. An outstanding proposal from the earlier activation remains stale. A new proposal must be built from the restored activation and receive corresponding evidence. Versions must remain monotonic across restart within the stated persistence model.

Reader atomicity requires more than atomic pointer replacement. In-process readers must share the publication lock until the new manifest is authoritative or failure recovery has finished. A reader then sees one complete published activation. The reference store uses content-addressed snapshots and complete manifests under a process lease. These guarantees are local. Power loss and storage-controller failure require stronger durability mechanisms. Network filesystems and replicated publication require separate analysis. The contract does not provide multi-writer consensus or cryptographic authentication.

3.3 Candidate construction and publication

The adapter proposer trains on the current update and retained replay examples. APA-BT screens a finite sequence of effective adapter candidates. APA-M proposes scoped memory without changing the adapter. APA-H proposes both. Each route packages its persistent changes into one complete candidate. Screening evaluates the predictor that would become active.

The protocol captures a parent and constructs detached candidates. It freezes and screens each candidate under a declared selection rule. A formal audit reserves its lifetime allocation before exposing fresh observations and evaluates only the fixed selected candidate. Publication rechecks parent identity and evidence binding under the lock. Delayed monitoring may restore a selected ancestor under a fresh version. Superseded descendants and outstanding proposals then become inactive or stale. Operational history remains intact. Appendix B states the algorithm and failure assumptions.

Complete transactions do not validate every future consequence. Replay and optimizer changes can affect later learning without changing the predictor tested by the current screen. Claims about those effects require losses that exercise the relevant transitions. Likewise a smaller adapter step does not attenuate an accompanying memory insertion. Reused screening data provide empirical approval of a candidate. Fresh auditing requires the separate conditions in Section 4.

3.4 Backtracking and complete recovery

Let $D_v = B_v A_v$ be the effective rank-r adapter and $D^\star = B^\star A^\star$ a proposed adapter. APA-BT tests

$$D_\alpha = \Pi_r((1-\alpha)D_v + \alpha D^\star), \qquad \alpha \in \{1, \tfrac{1}{2}, \tfrac{1}{4}, \tfrac{1}{8}, \tfrac{1}{16}\}. \tag{4}$$

Here Π_r is a best rank-r projection in Frobenius norm. The first passing complete candidate in the displayed order is selected. Smaller scales need not improve loss or preservation. Each candidate is evaluated independently. Nonparameter state follows a declared policy across scales. Parameter-coupled state must be transformed consistently or reconstructed or reset. The experiments initialize proposal optimizers afresh and derive event randomness deterministically.

Equation (4) interpolates effective adapters rather than factor coordinates. Equivalent factorizations yield the same projected map when the best projection is unique. Later optimization can still depend on the factors. APA therefore stores the exact factor pair and uses deterministic conventions for ties. The full-scale endpoint retains proposed factors. A zero-update endpoint retains the unchanged state directly. Appendix C.1 gives the low-rank construction.

Theorem 1 establishes serial publication under the construction and atomicity assumptions. Corollary 1 establishes ABA immunity. Proposition 1 specifies what complete restoration reproduces. A matched complete-state checkpoint with identical proposals and decisions follows the same sequential trajectory by Proposition 2. Accuracy gains therefore belong to the learning policy. Under concurrency or faults the transaction checks can change the active state by rejecting invalid transitions. Recovery can still discard useful descendants because monitoring and ancestor selection remain policy choices.

4 Fresh auditing with a lifetime budget

Let $\mathcal{H}_t$ be the history after reserving lifetime audit index t but before revealing its observations. The frozen candidate S'_t and captured parent S^p_t are fixed given $\mathcal{H}_t$. So are protected comparators $Q_{t,1}, \ldots, Q_{t,K_t}$ and their distributions. Panel sizes and tolerances must also be fixed. The loss and evaluation-randomness policy are part of that specification. The concentration rule is chosen before

inspection. Earlier screens and audits may inform these choices. Current audit observations may not. Every reserved index remains spent even when the audit fails or the candidate is abandoned. Recovery does not refund it.

Conditional on $\mathcal{H}_t$ each protected panel is IID from $P_{t,k}$ and the gain panel is IID from P_t^g. Define paired differences

$$d_{tki} = \ell(S'_t; Z_{tki}) - \ell(Q_{t,k}; Z_{tki}), \qquad g_{ti} = \ell(S_t^p; X_{ti}) - \ell(S'_t; X_{ti}).$$

Positive d denotes degradation and positive g denotes improvement. We assume $\ell \in [0,1]$ so $d, g \in [-1,1]$. Evaluation leaves the candidate unchanged. Evaluation randomness is fixed before inspection or included in the observations in a way that preserves conditional IID sampling. Protected panels need not be mutually independent. For $\delta \in (0,1)$ allocate to each of the $K_t + 1$ one-sided statements

$$a_t = \frac{\delta}{t(t+1)(K_t+1)}. \tag{5}$$

The total allocation is at most δ because $\sum_{t\geq 1} 1/[t(t+1)] = 1$. The counter must persist across restart as well as rollback. For $n \geq 2$ observations in $[-1,1]$ Hoeffding's one-sided radius is

$$r_H(n,a) = \sqrt{\frac{2\log(1/a)}{n}}. \tag{6}$$

Let $s^2 = (n-1)^{-1}\sum_i (Y_i - \bar{Y})^2$. Rescaling the empirical Bernstein bound from $[0,1]$ gives

$$r_B(n,a,s^2) = \sqrt{\frac{2s^2\log(2/a)}{n}} + \frac{14\log(2/a)}{3(n-1)}. \tag{7}$$

Choosing the smaller radius after observation requires additional valid confidence allocation. Let $\mu_{t,k}^d = \mathbb{E}[d_{tki} \mid \mathcal{H}_t]$ and $\mu_t^g = \mathbb{E}[g_{ti} \mid \mathcal{H}_t]$. For maximum regression $\epsilon_{t,k}$ and required gain γ_t the audit returns PASS only if

$$\bar{d}_{t,k} + r_{t,k} \leq \epsilon_{t,k} \quad \text{for all } k, \qquad \bar{g}_t - r_t^g \geq \gamma_t. \tag{8}$$

Proposition 3 (Lifetime false-audit-acceptance control). Under the conditional sampling and fixed-specification assumptions the probability of any PASS for which $\mu_{t,k}^d > \epsilon_{t,k}$ for some k or $\mu_t^g < \gamma_t$ is at most δ.

Proof. Each one-sided confidence statement fails with conditional probability at most a_t. If all statements hold then Equation (8) implies every population constraint. A false PASS requires at least one failed statement. Its conditional probability is at most $(K_t+1)a_t = \delta/[t(t+1)]$. Averaging over histories and applying a union bound gives

$$\Pr(\text{at least one false PASS}) \leq \sum_{t=1}^{\infty} \frac{\delta}{t(t+1)} = \delta. \tag{9}$$

No independence across attempts or constraints is needed beyond conditional panel validity. The result covers the declared bounded losses. Undeclared behavior and distribution shift are outside its scope. Resampling a fixed holdout can target its finite empirical distribution. It does not provide this guarantee for an external data-generating distribution. Raw cross-entropy requires a bounded transformation or a different analysis. Current audit data cannot guide candidate search. Data-dependent stopping or sample-size enlargement would require a different argument. A stale candidate cannot be published even after PASS. Audited publication is a subset of audit acceptance only if the evidence-bound publication checks hold. Future effects of persistent state are covered only when the audited loss exercises them.

A Hoeffding radius at most margin m requires $n \geq 2\log(1/a)/m^2$ observations per panel. With one protected group and $\delta = 0.05$ a margin of 0.05 requires at least 3506 observations at attempt 1 and 9516 at attempt 60. These are radius requirements rather than acceptance probabilities. Variance adaptation can improve power but cannot replace fresh trusted data. Appendix C.4 gives the derivation.

5 EVALUATION

5.1 DESIGN AND MEASUREMENTS

Prediction quality and publication consistency are separate outcomes. The original digit study measures learning under reusable screens. Controlled faults reuse identical trained candidates across publication rules. The new language-model study uses a longer stream with disjoint prompt roles. Synthetic auditing and a finite-benchmark neural-loss exercise examine a separate statistical protocol.

The optical-digit dataset contains 3823 training cases and 1797 official test cases (Alpaydin & Kaynak, 1998). A seeded within-class split assigns 60% to optimization and 20% each to development and monitoring. A 64-unit ReLU classifier learns digits 0 through 4 before its backbone is frozen. A rank-four adapter learns groups $(5, 6)$ and $(7, 8)$ and (9). Seeds 20 through 29 cover eight conditions and six methods for 480 runs. Each proposal uses 120 SGD steps. Replay retains up to 32 examples per class. Every fourth event replaces each current training label with probability 0.8 by a uniformly drawn old-class label. Validation labels stay clean. The controller receives no corruption indicator. Appendix D specifies all thresholds and sensitivity settings.

The new CounterFact study selects 256 records by text hash before model scoring. Their subjects and prompts are disjoint from the original eight development and 64 evaluation records. Normalized prompt sets are also disjoint across selected records. Rewrite prompts train candidates. The first paraphrase screens them. Separate neighborhood prompts provide screening and periodic monitoring. Final evaluation uses the second paraphrase and eight other neighbors per case. Three orders use seeds 2031 through 2033 on SmolLM2-135M-Instruct (Ben Allal et al., 2025). Canonical final batching removes order-dependent grouping of evaluation inputs. Appendix J records the selection and implementation.

GRACE (Hartvigsen et al., 2023) and APA-BT each receive eight optimization steps per edit. GRACE trains an activation codebook while APA-BT trains rank-eight query and value adapters with replay. Steps are matched; tokens and operations are not. GRACE settings are selected on the original development cases. A 100-step development check is reported separately. Both memory routes use the same whole-subject retriever and capacity of 256 assertions. All routes share the 64-token input limit and 12-token target limit. Supplied subjects simplify retrieval.

The empirical gate requires target NLL gain of 0.01. Protected NLL increases are limited to 0.20 against the parent and 0.40 against the frozen model. Recent-edit regression may not exceed 0.25. An eight-event monitor restores the preceding boundary when frozen-relative NLL increases by more than 0.55. Rejected and recovered edits remain in every final denominator. These engineering thresholds do not invoke Proposition 3. The original two-model results and their post hoc overlap check appear in Appendix H.

5.2 DIGIT RETENTION AND ACQUISITION

Table 1 reports mean accuracy and sample SD over ten seeds. Under corruption APA-BT improves both retention and acquisition relative to replay. Clean-stream differences are smaller. Hard gating preserves old classes largely by rejecting updates. Rank two yields only 45.3% new-class accuracy. Reducing label replacement to 50% yields 60.5%. These outcomes show that retention and acquisition must be assessed together.

Under corruption paired-seed bootstrap intervals estimate an old-class gain over replay of 28.2 percentage points with a 95% interval of 25.3 to 31.0. The new-class gain is 42.2 points with an interval of 38.4 to 45.7. The clean-stream new-class difference is -0.4 points with an interval of -2.8 to 2.1. These descriptive intervals concern this benchmark and protocol. They are not adjusted for multiple comparisons.

All 80 paired condition-seed trajectories for APA-BT and the checkpoint match in their state hashes at every event. Their decisions and final metrics also agree. Transaction bookkeeping provides no predictive gain in this matched sequential setting. Relative to backtracking without rollback APA-BT adds only 0.6 points of old-class accuracy under corruption with an interval spanning approximately zero. It loses 4.7 points of new-class accuracy. The evidence supports backtracking under the tested

Table 1: Digit accuracy in percent. Mean ± sample SD across seeds 20-29.

	Clean		Corrupted	
Method	Old	New	Old	New
Frozen	98.3 ± 0.4	0.0 ± 0.0	98.3 ± 0.4	0.0 ± 0.0
Replay	89.4 ± 1.4	85.2 ± 2.3	63.9 ± 4.8	39.0 ± 4.8
Hard gate	96.3 ± 2.6	14.1 ± 18.4	94.0 ± 1.1	20.5 ± 17.9
BT no rollback	90.7 ± 1.3	87.6 ± 2.8	91.5 ± 1.0	85.9 ± 4.1
APA-BT	91.3 ± 1.0	84.8 ± 4.4	92.1 ± 1.5	81.2 ± 6.0
Checkpoint	91.3 ± 1.0	84.8 ± 4.4	92.1 ± 1.5	81.2 ± 6.0

corruption process. It does not establish the chosen recovery policy as a better retention-acquisition tradeoff.

5.3 Which publication checks matter?

Ten fresh digit seeds produce two candidates that each improve their assigned new-class development loss from the same parent. The publication rules receive identical candidate bytes and evidence. Three schedules test a delayed sibling commit and an ABA recovery and substitution of candidate bytes after validation. The last intervention deliberately bypasses the public immutable interface. It models a storage or integration fault.

Table 2: Invalid publications under three controlled faults. Ten trained digit seeds per cell.

Publication rule	Stale sibling	ABA	Substitution
Sequential checkpoint	10/10	10/10	10/10
No parent check	10/10	10/10	0/10
Content check only	0/10	10/10	0/10
No candidate binding	0/10	0/10	10/10
APA	0/10	0/10	0/10

Table 2 reports invalid publications under these constructed schedules. Omitting the parent check permits the delayed sibling to overwrite the first update. Prior-task accuracy falls by 89.4 percentage points on average. A content-only check blocks that overwrite but accepts the old activation after recovery. Candidate binding detects substituted bytes. These are mechanism tests rather than estimates of natural failure rates. A checkpoint equipped with the same checks implements the same contract. A deliberately false construction attestation also passes APA in all ten trials. This counterexample explains the trusted-construction premise rather than weakening it silently.

5.4 Longer editing and matched memory

Table 3: CounterFact after 256 edits on SmolLM2. Percent mean and sample SD across three orders. GRACE and APA-BT use eight optimization steps per edit.

Method	Rewrite	Paraphrase	Neighbor	Consistency
Frozen	27.7 ± 0.0	26.6 ± 0.0	70.2 ± 0.0	100.0 ± 0.0
GRACE	100.0 ± 0.0	26.6 ± 0.0	69.5 ± 0.0	97.0 ± 0.1
APA-BT	68.1 ± 4.5	51.6 ± 4.9	48.7 ± 2.5	3.0 ± 1.6
Ungated memory	96.9 ± 0.0	88.3 ± 0.0	70.1 ± 0.0	99.7 ± 0.0
APA-M	96.5 ± 0.0	88.3 ± 0.0	70.1 ± 0.0	99.7 ± 0.0

Table 3 compares acquisition with preservation after 256 edits. Rewrite and paraphrase preference favor the supplied counterfactual answer. Neighbor preference favors the original answer. Consistency measures agreement with frozen next-token predictions on those same neighbors. None is a

measure of broad language competence. The retained per-case scores support paired comparisons without treating prompts as independent runs.

GRACE reaches 100.0% rewrite preference but 26.6% paraphrase preference. The frozen paraphrase score is 26.6%. The selected codebook configuration therefore shows little paraphrase transfer on this checkpoint. APA-BT reaches 51.6% paraphrase preference while neighbor preservation falls to 48.7%. It accepts 234.0 proposals per stream and retains 126.0 after recovery.

Both memory routes reach 88.3% paraphrase preference. The empirical gate rejects 3 of 768 clean-stream proposals. It does not improve paraphrase acquisition or neighborhood consistency in this matched comparison. Supplied entity names give both routes a direct retrieval cue. Their outcome is not evidence that the publication contract improves a learning policy.

The clean stream does not by itself test rejection of harmful memory. A separate matched experiment applies six insertion conditions to 32 records. Every policy receives the same candidate and uses the same scope-aware retriever. An independently retained intended answer supports validation after the proposal target is corrupted. The strengthened gate requires positive contrastive preference as well as NLL improvement. It also checks scope and conflicting assertions or aliases. The NLL gate accepts 14 of 32 wrong-target insertions. The strengthened gate accepts none but rejects seven clean insertions. Clean-case paraphrase preference falls from 87.5% to 75.0%. Appendix K reports both sides of this tradeoff. Synthetic aliases test declared routing behavior rather than discovery of natural aliases. The gate cannot establish truth when the source of its intended answer is itself compromised.

5.5 Recovery and storage cost

A scheduled notification experiment injects one corrupted replay state and then admits clean descendants. Notification after one or four or eight events discards zero or three or seven descendants with positive development-loss gain. This is a measured cost of complete recovery. The notification schedule is supplied by the experiment and does not measure detector quality. After the next identical clean update complete recovery reaches 93.1% old-class accuracy. Weight-only recovery reaches 7.3% with immediate notification and about 54% at the later boundaries. The retained replay state changes future learning. Appendix I reports every seed and the acquisition cost.

The durable prototype originally allowed an in-process read between an inner commit and manifest publication. A reader could observe state that a failed publication would later withdraw. Public reads now hold the publication mutex. Concurrent reader tests cover both successful publication and injected failure. Another change reuses the addresses of immutable snapshots instead of serializing all historical states at every event. At event 64 median end-to-end storage time falls from 4.67 seconds to 0.055 seconds over three local repetitions. The payload contains 1.76 MiB of adapter tensors. Retained files still reach 162.7 MiB and retained snapshot arrays reach 228.5 MiB. These costs exclude learning and neural monitoring. They demonstrate a prototype improvement rather than a scalable retention policy.

5.6 Audit power and verification

The original synthetic study uses 1000 independent runs of 60 attempts per panel size. Beneficial proposals have protected regression 0. Harmful proposals have regression 0.06 against tolerance 0.05. Both have gain 0.20 against a required 0.01. With 2048 observations per panel empirical Bernstein accepts 76.4% of beneficial attempts. Hoeffding accepts one of 30000. Neither lifetime-budgeted rule produces an observed false-acceptance run. Zero events in 1000 runs imply a one-sided 95% upper limit of about 0.30% for that simulation. Appendix F gives the distributions and results.

The neural integration exercise fixes a digit adapter before auditing the bounded loss $\min\{\mathrm{NLL}/10, 1\}$. Each attempt reserves its allocation on disk before sampling. IID draws with replacement target the uniform distribution over the fixed official test cases within old or new class groups. At 2048 draws per panel empirical Bernstein certifies all ten candidates. Hoeffding certifies none even at 8192 draws. All ten candidates satisfy the finite-population constraints so this test measures power and protocol integration. Repeated draws add no new labels. They do not certify performance outside this benchmark. Appendix L records the exact populations and sample indices.

All 480 digit runs were reexecuted during revision. Their final accuracies and decision counts match the retained results exactly. The verifier reproduces selection identities and all 80 split files. It also checks 1200 paired digit events and 1920 original language-model shadow events. The revised implementation passes 49 unit and fault tests. A bounded explorer checks 19531 schedules and 112305 attempted transitions through depth six. These checks cover their stated inputs and local failure model.

6 Discussion and limitations

APA addresses which state an approval authorizes. It does not make an inaccurate validator reliable. The stale-sibling experiment shows why that distinction matters for a real learner. The matched checkpoint shows where it provides no predictive advantage. The reader race also illustrates the gap between an atomic on-disk pointer and an implementation that presents one authoritative state to every caller.

Recovery has a cost even when every byte is restored correctly. Under digit corruption the original rollback policy loses 4.7 percentage points of new-class accuracy for only a small old-class improvement. Scheduled fault notification makes discarded useful descendants measurable but does not solve detection or ancestor selection. Selective recovery would require revalidation of retained descendants.

The new language-model comparison covers one small checkpoint and three orders from one benchmark. GRACE is adapted to that architecture under a limited development grid. Supplied subject names and bounded prompt lengths shape the memory result. The insertion tests use controlled faults and declared synthetic aliases. Stronger models and natural update streams remain necessary before making claims about lifelong editing or general factual reliability.

Local screens can accumulate regression or overfit reused data. The fresh-audit theorem applies only to declared bounded losses with valid conditional sampling. Its finite-benchmark implementation adds no guarantee about future populations. Cached snapshot references reduce repeated serialization but complete states and manifests still grow with history. Retention and compaction need their own recovery argument.

7 Conclusion

Validation evidence belongs to a particular candidate built from a particular activation. APA preserves that relationship across publication and complete-state recovery under explicit assumptions. Real-data fault ablations isolate the checks that enforce it. Longer editing streams and matched memory controls separate those guarantees from acquisition and preservation. The revised implementation closes a durable reader race and removes repeated historical serialization. Reliable proposal generation and monitoring remain separate requirements. A correct publication boundary makes those requirements testable.

Reproducibility statement

Appendices B and C give the contract and its assumptions and proofs. Appendices D through F document the original studies. Appendices I through N give the revision protocols and raw-result checks. The separate research artifact retains executable code with dataset identifiers and per-event records and adapter checkpoints. All 480 digit outcomes were reexecuted. The targeted saved-adapter and frozen-batching checks are reported explicitly.

AI use statement

Generative AI assisted with language editing and mathematical arguments. They also supported literature checks and reference formatting. During revision they were used for critical review of the methodology and interpretation of results and for clarification of mathematical assumptions and proof arguments. They were not used in this revision to generate experimental data or rerun the reported experiments. The authors take responsibility for the manuscript and its technical claims. This responsibility includes the cited literature and the reported software and experimental results..

REFERENCES

E. Alpaydin and C. Kaynak. Optical recognition of handwritten digits. UCI Machine Learning Repository, 1998. doi: 10.24432/C50P49.

Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, and Vahab Mirrokni. Nested learning: The illusion of deep learning architectures. In *Advances in Neural Information Processing Systems*, volume 38, 2025a. doi: 10.52202/085713-1565.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. doi: 10.52202/085713-3786.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Agustín Piqueres Lajarín, Hynek Kydlíček, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Ben Burtenshaw, Clémentine Fourrier, Haojun Zhao, Hugo Larcher, Mathieu Morlon, Cyril Zakka, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM2: When smol goes big: Data-centric training of a fully open small language model. In *Conference on Language Modeling*, 2025.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. In *Advances in Neural Information Processing Systems*, volume 28, pp. 2350 to 2358, 2015.

Assaf Eisenman, Kiran Kumar Matam, Steven Ingram, Dheevatsa Mudigere, Raghuraman Krishnamoorthi, Krishnakumar Nair, Misha Smelyanskiy, and Murali Annavaram. Check-N-Run: A checkpointing system for training deep learning recommendation models. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, pp. 929 to 943, 2022.

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.

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 to 16819, 2024. doi: 10.18653/v1/2024.emnlp-main.934.

Thomas 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*, volume 36, 2023.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13 to 30, 1963. doi: 10.1080/01621459.1963.10500830.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521 to 3526, 2017. doi: 10.1073/pnas.1611835114.

H. T. Kung and John T. Robinson. On optimistic methods for concurrency control. *ACM Transactions on Database Systems*, 6(2):213 to 226, 1981. doi: 10.1145/319566.319567.

Romain Laroche, Paul Trichelair, and Rémi Tachet des Combes. Safe policy improvement with baseline bootstrapping. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3652 to 3661, 2019.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample variance penalization. In *Proceedings of the 22nd Annual Conference on Learning Theory*, 2009.

James L. McClelland, Bruce L. McNaughton, and Randall C. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3):419 to 457, 1995. doi: 10.1037/0033-295X.102.3.419.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1262.

Kevin Meng, Arnab Sen Sharma, Alex J. 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, 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 to 15831, 2022.

Jayashree Mohan, Amar Phanishayee, and Vijay Chidambaram. CheckFreq: Frequent, fine-grained DNN checkpointing. In *19th USENIX Conference on File and Storage Technologies (FAST 21)*, pp. 203 to 216, 2021.

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 to 59354, 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, 2024a. doi: 10.52202/079017-1703.

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

Yangfan Wang, Tianyang Sun, Chen Tang, Jie Liu, Wei Cai, and Jingchi Jiang. HiEdit: Lifelong model editing with hierarchical reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 39924 to 39942, 2026. doi: 10.18653/v1/2026.acl-long.1855.

A STATE INVENTORY AND IMPLEMENTATION BOUNDARIES

The contract covers the state needed by the declared predictor and its later update rule. Table 4 identifies that boundary in the implemented experiments. Immutable dependencies may be referenced by a pinned identifier. Mutable values must be retained or uniquely reconstructible. A separate operational history records which content was active and which audit allocations were spent.

Table 4: State inventory for the implemented reference.

Component	Retained value or reconstruction rule
Backbone	Pinned public checkpoint and file hashes. The digit backbone is reconstructible from the recorded data and seed and training configuration.
Adapter	Exact low-rank factor arrays. Effective interpolation is reevaluated before acceptance.
Replay	Admitted examples and targets with deterministic capacity policy.
Semantic memory	Assertions with subject and target and provenance fields. Controlled alias tests also retain explicit alias lists.
Optimizer	Fresh SGD or AdamW per proposal. No momentum is carried between proposals. The protocol records this reset policy.
Randomness	Initial seed and deterministic event seed. The digit state retains generator state after proposal construction.
Retrieval and control	Immutable run configuration plus active memory. No learned router or approximate retrieval index is evaluated.
Operational history	Monotonic activation and audit counters with the publication ledger. Recovery never refunds an audit.

Scope is an explicit request input in the controlled memory experiment. Retrieval filters by that scope before ranking whole-name matches. Conflicting aliases are quarantined. These are lexical and metadata checks. They do not infer entailment or establish source reliability.

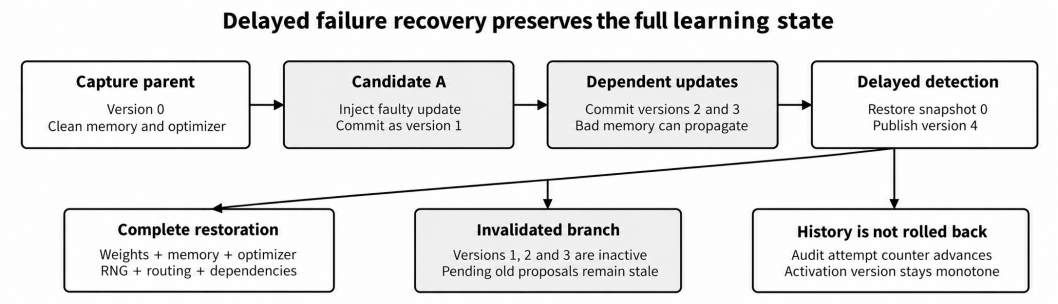


Figure 2: Complete recovery restores ancestor content and removes dependent descendants. The corresponding operational records remain in history.

B TRANSACTION ALGORITHM AND FAILURE MODEL

Construction must begin from an explicitly captured parent. The protocol is:

1. Atomically capture the active version and content digest with its complete immutable snapshot.
2. Construct detached candidates from that snapshot using only permitted update data and pinned configuration. Include memory and other persistent changes with any parameter update.
3. Freeze each candidate before screening it. When backtracking is enabled select the first passing candidate in the declared order. Reject or quarantine the proposal if none passes.
4. Bind evidence to the captured parent and the frozen candidate digest. If a formal audit is requested reserve its lifetime allocation durably before revealing fresh observations. Audit only the selected candidate. Preserve the allocation after any subsequent failure.

5. Acquire the publication lock. Recheck the current parent identity and evidence approval for that parent and candidate. Recompute the candidate digest. A failed check leaves the active state unchanged.
6. Write the complete immutable snapshot and manifest. Publish it atomically under a fresh activation version and retain the corresponding activation record.
7. Apply the declared monitoring policy at its boundaries. On recovery restore the selected ancestor under a fresh version. Invalidate superseded content and preserve operational history.

Steps 1 and 2 are construction requirements. The checks in step 5 cannot retrospectively prove that a worker complied with them. The serial-publication theorem therefore assumes the construction path is trusted. It also assumes evidence comes from the declared validator and cannot be substituted. These are authority and isolation assumptions rather than consequences of content hashing.

The proposal store copies state on insertion and retrieval. Mutating caller-owned arrays therefore cannot change the stored bytes associated with validation evidence. An activation record retains its sequence number and event type. It identifies the active version and previous record digest along with the state digest and evidence for its parent. A trusted prior digest makes inconsistent changes to this logical hash chain detectable. A writer with unrestricted storage access can still replace the entire chain. Authentication and external anchoring require separate mechanisms.

The local store uses a single-writer process lease backed by an operating-system lock. The lock is released when the process terminates or its handle closes. Snapshots and manifests are content-addressed. Deserialization rejects unsupported state types and object arrays. It also rejects malformed addresses and digest mismatches. A small atomically replaced pointer identifies the authoritative manifest. Unreachable files left by failed preparation are not active state. Automatic garbage collection is not implemented.

Publication takes effect at replacement of the authoritative pointer. An interruption before that point leaves the previous complete activation authoritative under the local storage assumptions. The reported fault tests cover that path and a valid retry of the same already-validated candidate while its parent remains current. A retry using unchanged evidence does not constitute a new audit. Revealing a new audit panel does require a new allocation. The activation and audit counters must not regress across restart.

Audit reservations must be independent of rollbackable learning snapshots. Reservation precedes panel exposure so a crash after exposure cannot make the same allocation available again. This durability condition is required by the lifetime theorem. Neither that theorem nor a test of a particular interruption point proves durability under arbitrary storage failure.

The implementation assumes documented local atomic-replacement and flushing semantics. Process interruption differs from sudden power loss or controller failure. Network filesystems and distributed caches are outside the tested failure model. Multiple publication writers and replicated consensus are also outside scope. A deployment must establish the persistence assumptions required by its own failure model.

C MATHEMATICAL DETAILS AND CONDITIONAL GUARANTEES

FORMAL STATEMENTS REFERENCED IN THE MAIN TEXT

Definition 1 (Evidence-bound candidate). Let $X_v = (v, h(\mathrm{enc}(S_v)), S_v)$ be an active state with strictly increasing activation version v. The relevant proposal fields are $P = (v_p, h_p, h_C, C, E)$. Candidate C is constructed from the captured snapshot identified by (v_p, h_p) under the declared construction policy. It is frozen before validation. Its digest is $h_C = h(\mathrm{enc}(C))$. Evidence E identifies both this digest and the captured parent.

Lemma 1 (Commit binding). Assume a trusted construction path that enforces Definition 1. Assume a single activation writer and no collisions among encountered digests. Validation evidence must refer to immutable candidate bytes and the captured parent. If commit checks approval and both identities under the publication lock then each successful commit publishes the exact validated candidate constructed from the state current at its publication point.

The identity checks establish which bytes and parent the evidence authorizes. The construction premise establishes where those bytes came from. The latter conclusion would not follow from a matching digest alone. Appendix C.2 gives the invariant argument.

Theorem 1 (Serial activation safety). Under Lemma 1 and atomic reads of complete active versions successful activations admit the serial order defined by their publication points. Each committed proposal names the activation immediately preceding it. Recovery publishes a complete ancestor snapshot as a new activation. Later proposals must name that new activation. Induction over publication points proves the claim. This is a local serial-publication result. It does not guarantee progress for a worker whose proposals repeatedly become stale. It also does not provide distributed consensus or Byzantine-fault tolerance.

Corollary 1 (ABA immunity). Suppose every commit and recovery receives a fresh strictly increasing activation version. Restoring content whose digest matches an earlier state cannot revive a proposal bound to that earlier version. The version check fails even when the content check succeeds. This assumes versions remain monotonic across restart and do not wrap within the execution considered.

Proposition 1 (Complete-state restoration). Suppose a snapshot covers the complete declared learning state and required immutable dependencies remain available. Under deterministic execution restoration reproduces predictions for equal request inputs. Equal later transitions require the applicable operational state to agree as well. External events and random choices must also match. This follows from equality of the inputs to $T(S, H, e, \xi, K)$. Recovery retains current operational history. It does not recreate (S_u, H_u) or refund audits. Byte equality does not establish determinism for nondeterministic CUDA kernels. Recovery also cannot retract external outputs or implement selective unlearning.

Theorem 2 (Necessity of causal-state coverage). Suppose two admissible states have identical retained components but differ in an omitted component c. Suppose their next transitions differ under equal external inputs and random choices. No recovery procedure that observes only the retained components can guarantee the correct transition for both states. The procedure receives identical information in the two cases. It cannot determine which distinct transition to reproduce.

Proposition 2 (Matched-checkpoint equivalence). Consider two deterministic sequential controllers with identical initial learning state and identical applicable operational inputs. Suppose they construct the same candidates and make the same acceptance and monitoring decisions. Suppose they also restore the same snapshots. Additional bookkeeping that never changes construction or a decision leaves their predictive trajectories equal. The claim follows by induction. It does not apply when operational counters alter randomness or policy inputs differently across the controllers.

C.1 Efficient effective-weight interpolation

For two rank-r updates define

$$L_\alpha = \begin{bmatrix}\sqrt{1-\alpha}B_v & \sqrt{\alpha}B_\star\end{bmatrix}, \qquad R_\alpha = \begin{bmatrix}\sqrt{1-\alpha}A_v \\ \sqrt{\alpha}A_\star\end{bmatrix} \tag{10}$$

Then $L_\alpha R_\alpha = (1-\alpha)D_v + \alpha D_\star$. Thin QR decompositions reduce the projection to a small core:

$$L_\alpha = Q_L T_L, \qquad R_\alpha^\top = Q_R T_R, \qquad T_L T_R^\top = U\Sigma V^\top \tag{11}$$

Orthonormal outer factors preserve the nonzero singular values. Truncating the core SVD gives a best rank-r approximation in Frobenius norm:

$$B_\alpha = Q_L U_r \Sigma_r^{1/2}, \qquad A_\alpha = \Sigma_r^{1/2} V_r^\top Q_R^\top \tag{12}$$

The core has dimension at most $2r \times 2r$. For input dimension d_{in} and output dimension d_{out} the arithmetic cost is $O((d_{\text{in}} + d_{\text{out}})r^2 + r^3)$. A dense effective matrix need not be materialized. This standard construction reduces projection cost. It does not remove the forward passes needed for screening.

The implementation checks reconstructed updates against dense truncated SVD and checks invariance under independent factor rescalings. At a singular-value tie across the truncation boundary the optimum need not be unique. The invariance tests use nondegenerate cases up to numerical precision. Full and zero endpoints retain their original factors. Intermediate candidates are reevaluated because neither interpolation nor projection guarantees monotonic loss or locality.

C.2 What complete restoration proves

Proof of Lemma 1 and Theorem 1. Let $A_j = (v_j, h_j, S_j)$ be the active state after activation j. Assume A_j is complete. The construction premise makes C a candidate from the captured immutable parent. Under the writer lock commit checks $(v_p, h_p) = (v_j, h(\mathrm{enc}(S_j)))$. It checks that evidence approves this parent and h_C. It also checks $h(\mathrm{enc}(C)) = h_C$. Failure leaves active state unchanged. Success publishes

$$A_{j+1} = (v_j + 1, h\left(\mathrm{enc}(C)\right), C) \tag{13}$$

No intervening publication can occur while the lock is held. The parent active at that point is therefore the captured parent. Digest equality and immutable evidence bind publication to the validated bytes. Ordering commits by their pointer replacements gives the claimed serial history. Recovery is another activation that publishes complete ancestor content under a fresh version. The same reasoning then applies to subsequent proposals.

Atomic pointer replacement identifies the published activation. In-process readers share the publication mutex until replacement or failure recovery completes. Under these read semantics they observe either A_j or A_{j+1} without mixing their components. In the persistent implementation the content-addressed manifest is written before the CURRENT pointer is replaced. This argument assumes trustworthy local storage and the stated publication semantics.

Proof of Corollary 1. Recovery from v_j publishes the earlier content at $v_j + 1$. Every outstanding proposal from before recovery carries a version no greater than v_j. Its parent version cannot equal the new active version. This remains true if the restored content digest matches its old parent exactly. The bounded explorer includes such ABA traces and probes retained stale proposal descriptors.

Proof of Proposition 1. Equal inputs to a deterministic computation $T(S, H, x, \xi, K)$ produce equal outputs and next states. This proves the one-step statement. Repeating the argument proves trajectory equality when all applicable inputs continue to agree. Restoring learning content alone does not force operational inputs to agree.

A replay example shows why weight restoration is insufficient. Let $w^+ = w - \eta(w - m)$ where m is a stored replay target. Restoring $w = 0$ while retaining failed target $m = 1$ gives $w^+ = \eta$. Restoring both $w = 0$ and $m = 0$ gives $w^+ = 0$. The immediate weight is identical but the next step is different.

Proof of Theorem 2. Let R_{-c} be the retained representation after component c is omitted. By assumption

$$R_{-c}(S) = R_{-c}\left(\widetilde{S}\right) \quad \text{but} \quad T(S, H, x, \xi, K) \neq T\left(\widetilde{S}, H, x, \xi, K\right) \tag{14}$$

A deterministic recovery procedure sees the same retained input in both cases and returns the same reconstruction. It cannot reproduce both distinct correct transitions. Randomization cannot give a probability-one guarantee for both either because its output distribution is the same for both retained inputs. The executable counterexample uses

$$g = w - m, \qquad q^+ = \frac{1}{2}q + g, \qquad w^+ = w - 0.1\rho q^+ + 0.01(-1)^r \tag{15}$$

The replay target is m. Momentum is q and r represents random state. The controller scale is ρ. Holding the other quantities fixed while varying one of these components can change the next state. The theorem does not require storing fields that are uniquely reconstructible from retained state and immutable configuration.

These results establish computational sufficiency and necessity within the declared boundary. They do not identify the best recovery ancestor or show that every discarded descendant was harmful. External outputs and actions remain outside the restored local state.

C.3 Matched control equivalence

Let S_t^A and S_t^C be the APA and checkpoint states with $S_0^A = S_0^C$. Assume applicable operational inputs also agree. Equal parent states with the same event and randomness produce equal candidates. The matched decision rules select the same candidate or reject it. Matched monitoring and recovery then produce $S_{t+1}^A = S_{t+1}^C$. Induction proves Proposition 2.

The digit study independently executes both controllers. The language-model shadow control instead sends fixed candidates and decisions through both state-management paths. Hash equality in the shadow study is a software observation. It is not an independent estimate of learning improvement. Neither control implies equal cost because state copying and persistence can add overhead without changing predictions. Hashing has a cost too.

C.4 VARIANCE-ADAPTIVE LIFETIME AUDITING

For paired observations $d_1, \ldots, d_n \in [-1, 1]$ define

$$\overline{d} = \frac{1}{n}\sum_{i=1}^{n} d_i, \qquad s_d^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(d_i - \overline{d}\right)^2 \tag{16}$$

Apply the empirical Bernstein inequality to $Z_i = (d_i + 1)/2$. Its sample variance is $s_d^2/4$. Rescaling the upper confidence limit gives Equation (7). Applying the same argument to $-g_i$ gives the lower confidence limit on gain.

The candidate and all audit specifications are measurable with respect to pre-observation history $\mathcal{H}_t$. Conditional on this history each panel is IID under its declared distribution. Let F_t denote a false PASS at allocated attempt t. Set $F_t = \varnothing$ when the reserved attempt is abandoned without acceptance. The one-sided bounds imply

$$\begin{aligned}\Pr(\text{at least one false acceptance}) &\le \sum_{t=1}^{\infty}(K_t + 1)a_t \\ &= \delta\sum_{t=1}^{\infty}\left(\frac{1}{t} - \frac{1}{t+1}\right) = \delta.\end{aligned} \tag{17}$$

Here the left-hand side is $\Pr(\bigcup_t F_t)$. Each term $(K_t+1)a_t$ equals the deterministic value $\delta/[t(t+1)]$ even if K_t depends on history. Formally the union bound is applied after averaging each conditional failure probability over histories. Independence across protected groups is unnecessary.

The sampling requirement concerns each fresh panel conditional on history. Reusing observations to select a revised candidate invalidates that panel for the revised candidate. Empirical candidate search on reusable development data remains permissible before fresh auditing. Increasing sample size until a favorable decision appears is outside this fixed-sample proof. A time-uniform procedure needs a different guarantee. Abandoned and stale attempts consume their allocations just as rejected attempts do. Recovery and restart must not refund them.

A Hoeffding radius at most m requires

$$n \ge \frac{2\log(1/a)}{m^2} \tag{18}$$

For one protected group with $\delta = 0.05$ and $m = 0.05$ this requires 3506 observations at attempt 1 and 9516 at attempt 60. These are requirements on the confidence radius. They do not imply a given probability of acceptance. Empirical Bernstein can use low paired-loss variance but its additive term may dominate for small panels. The concentration rule is fixed before seeing the data.

C.5 LOCAL AND CUMULATIVE PROTECTION

Suppose each of T accepted transitions satisfies $R(S_{j+1}) - R(S_j) \le \epsilon$ for one fixed risk functional. Telescoping gives only

$$R\left(S_T\right) - R\left(S_0\right) \le T\epsilon \tag{19}$$

The cumulative allowance grows with the number of transitions. A fixed-reference test instead checks $R(S') - R(S_0) \le \epsilon_{\text{total}}$ for each candidate. It requires a preserved reference and a stable protected distribution. Changing the reference or risk definition breaks a direct comparison across time unless that change is made explicit. The empirical protocols use both comparisons but reused panels provide no population-level cumulative bound. Recovery steps must be accounted for separately unless they satisfy the same risk premise.

C.6 THEORY AND EXECUTABLE EVIDENCE

Table 5 connects the claims to their reported checks. The checks can reveal implementation violations. They do not establish the assumptions or replace proofs.

Table 5: Formal claims with executable checks and assumptions. Tests detect implementation violations; they do not replace proofs or assumptions.

Claim	Executable evidence	Assumptions
Serial activation safety	Bounded schedule exploration through depth six with stale-parent and stale-sibling tests	Requires trusted parent-isolated construction and a single writer with atomic complete-state reads
ABA immunity	Rollback traces that restore previously observed content and probe retained stale proposals	Requires the activation counter to remain persistent and strictly monotonic across rollback
Complete-state restoration	Delayed-failure recovery and deterministic replay across restart	Requires complete learning state and equal computational inputs with pinned dependencies and deterministic execution
Necessity of causal-state coverage	Constructed one-step counterexamples that independently vary omitted state components	A state that is uniquely reconstructible from a retained state and an immutable configuration need not be stored explicitly
Matched-checkpoint equivalence	Independently paired digit trajectories and 1920 language-model shadow events	Requires identical proposals and decisions with matched recovery targets and random choices
Lifetime false-acceptance control	Audit-allocation tests and paired-loss fresh-panel simulations	Requires bounded losses and fresh conditional IID panels under a fixed candidate and audit specification

D DIGIT STUDY AND SENSITIVITY RESULTS

The UCI Optical Recognition of Handwritten Digits data contain 64 values per image on an 8×8 grid. Values are divided by 16. The official training and test files are retained without modification. A seeded within-class permutation assigns 60% of training examples to optimization and 20% to development. The remainder is used for monitoring. The research artifact records the split indices and dataset archive hash. The learning controller never sees official test labels. The separate finite-benchmark audit in Appendix L uses them only after its candidate is fixed.

Table 6: Default digit configuration. Sensitivity conditions change one factor at a time.

Setting	Value
Hidden units	64
Base epochs and learning rate	70 and 0.08
Adapter rank and mutable parameters	4 and 300
Proposal SGD steps and learning rate	120 and 0.06
Current and replay minibatch sizes	64 and 64
Replay examples per class	32
Update events	12
Target cross-entropy gain minimum	0.005
Parent-relative old accuracy tolerance	3 percentage points
Frozen-reference old accuracy tolerance	6 percentage points
Recent-class accuracy tolerance	8 percentage points
Per-old-class monitor tolerance	12 percentage points
Corrupted event period	Every fourth event
Label replacement probability	0.8
Evaluation seeds	20 through 29

The base classifier has one 64-unit ReLU hidden layer and ten output coordinates. Initial training uses only digits 0 through 4. Backbone parameters are then frozen. New classes arrive as $(5, 6)$

followed by $(7, 8)$ and then (9). Four cycles give twelve update events. Every fourth event carries corruption in the default stress test. At each of those events a current-example label is independently replaced with probability 0.8 by a uniformly drawn old-class label. The controller receives no corruption indicator. Development and monitoring labels remain clean by construction.

Replay is class-balanced up to its capacity. Current examples enter candidate replay state and become active only if the corresponding candidate is accepted. The digit implementation uses the transpose-equivalent convention $D = AB$. The main equations and language-model implementation use $D = BA$. Both interpolate the effective matrix. Each digit proposal uses fresh SGD state with gradient clipping at norm 5 and event-specific randomness.

The eight evaluation conditions are clean data and default corruption plus six one-factor variations. The stream variations increase the horizon to 36 events or shuffle task order. Capacity variations reduce adapter rank to two or replay to eight examples per class. The remaining settings double the proposal learning rate or reduce label replacement to 50%. The same ten seeds are paired across methods within each condition. This is a sensitivity study on one dataset rather than a factorial experiment or independent evidence across eight learning problems.

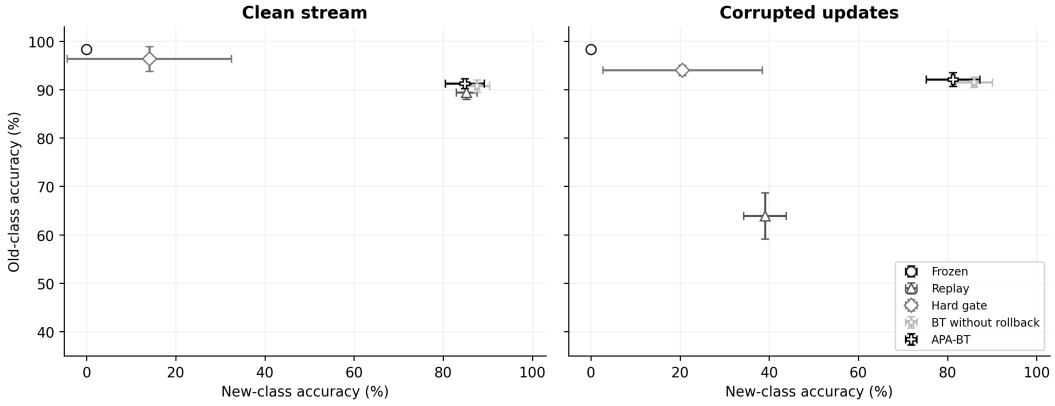


Figure 3: Digit retention and acquisition. Error bars show one sample SD across seeds 20-29. Old-class retention and new-class acquisition should be interpreted jointly.



Figure 4: One-factor digit sensitivity across eight settings. Values are mean accuracies over ten seeds. Removing rollback often increases acquisition; lower rank limits acquisition despite high retention.

The largest gains over replay occur where replay is most vulnerable to corrupted updates. These include the default and longer streams as well as the task-order and small-memory settings. Rank two gives a different outcome. APA-BT retains 93.6% old-class accuracy but reaches only 45.3% new-class accuracy. Screening cannot compensate for insufficient adaptation capacity.

Table 7: Digit sensitivity results. Mean ± sample SD across seeds 20-29. Complete ablations and per-seed outcomes are retained in the research release.

	Replay		APA-BT	
Condition	Old	New	Old	New
Clean	89.4 ± 1.4	85.2 ± 2.3	91.3 ± 1.0	84.8 ± 4.4
Corrupted	63.9 ± 4.8	39.0 ± 4.8	92.1 ± 1.5	81.2 ± 6.0
Long Stream	53.6 ± 7.5	46.4 ± 4.1	90.9 ± 1.4	82.8 ± 6.7
Task Order	68.6 ± 8.1	35.8 ± 7.2	92.0 ± 0.9	81.1 ± 5.6
Rank Two	84.5 ± 10.7	23.1 ± 6.5	93.6 ± 1.4	45.3 ± 13.9
Larger Step	64.1 ± 6.6	41.7 ± 8.8	90.7 ± 1.3	83.4 ± 12.0
Small Memory	43.3 ± 13.5	40.2 ± 7.5	91.9 ± 1.4	77.8 ± 6.6
Half Corruption	74.4 ± 5.7	48.7 ± 4.4	93.0 ± 1.6	60.5 ± 10.5

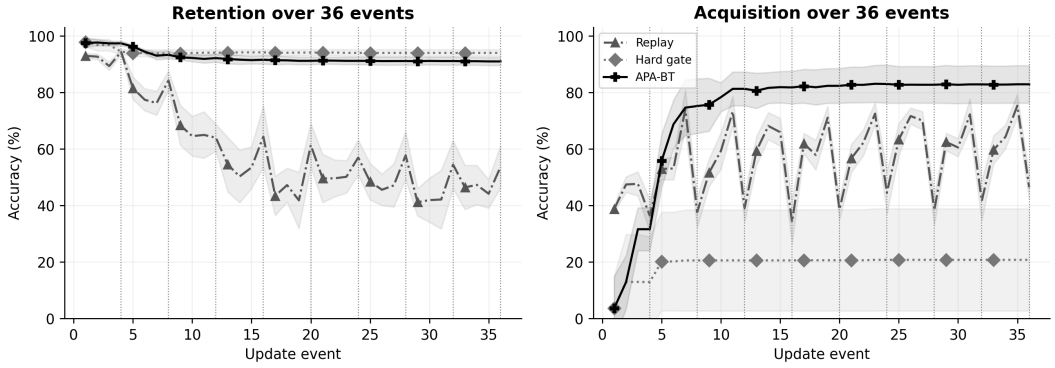


Figure 5: Performance over the 36-event digit stream. Curves show means and shaded bands one sample SD across ten seeds. Dotted lines identify corrupted events for visualization only; the controller never receives corruption indicators.

With 50% label replacement APA-BT retains 93.0% old-class accuracy and reaches 60.5% on new classes. Acquisition is lower than under 80% replacement. The corruption schedule is unchanged in this comparison. The nonmonotonic outcome may involve which proposals pass and which descendants monitoring removes. The reported aggregates do not isolate that mechanism. They should not be read as evidence that more corruption improves learning in general.

The independently executed checkpoint uses the same initialization and proposal randomness as APA-BT. Replay capacity and candidate scales are matched. So are acceptance and monitoring rules. All 80 paired condition-seed runs have identical active-state hashes after corresponding events. This supports Proposition 2 in the deterministic NumPy implementation. A separate regression check reports that requiring explicit parent capture leaves these sequential outcomes unchanged.

Digit runtime covers proposal optimization through activation and immediate monitoring. It includes state copying and candidate screening. It excludes base-model training and cached feature extraction. Final test evaluation and serialization are also excluded. Snapshot-array counts cover stored numerical arrays. They omit object overhead. Rejected proposals and ledger text are also excluded. They are not peak-memory bounds. CPU timing with cached features should not be extrapolated to transformer fine-tuning.

D.1 Paired uncertainty and the cost of rollback

Table 8 reports APA-BT minus each comparator using ten paired seeds. The analysis resamples seed indices with replacement 100000 times under bootstrap seed 20260915. Pairing is preserved. The interval endpoints are the 2.5th and 97.5th percentiles of the mean paired difference. Resampling events or test examples would not measure variability across complete runs. The intervals are descriptive and unadjusted for multiple comparisons.

Table 8: APA-BT minus comparator in percentage points with descriptive 95% paired-seed bootstrap intervals. Zero checkpoint contrasts reflect matched trajectories.

Stream	Comparator	Old accuracy	New accuracy
Clean	Replay	$+1.9$ $[+1.1, +2.6]$	-0.4 $[-2.8, +2.1]$
Clean	Hard gate	-5.1 $[-6.5, -3.4]$	$+70.8$ $[+60.9, +80.1]$
Clean	BT no rollback	$+0.6$ $[+0.2, +0.9]$	-2.8 $[-5.2, -0.5]$
Clean	Checkpoint	$+0.0$ $[+0.0, +0.0]$	$+0.0$ $[+0.0, +0.0]$
Corrupted	Replay	$+28.2$ $[+25.3, +31.0]$	$+42.2$ $[+38.4, +45.7]$
Corrupted	Hard gate	-1.9 $[-3.0, -0.9]$	$+60.7$ $[+51.0, +70.8]$
Corrupted	BT no rollback	$+0.6$ $[-0.0, +1.3]$	-4.7 $[-8.8, -1.3]$
Corrupted	Checkpoint	$+0.0$ $[+0.0, +0.0]$	$+0.0$ $[+0.0, +0.0]$

Compared with hard gating APA-BT acquires substantially more new classes while retaining slightly less old-class accuracy. The no-rollback comparison isolates the monitoring and recovery policy. Under corruption rollback changes old-class accuracy only modestly with an interval spanning approximately zero. New-class accuracy falls by 4.7 percentage points. These results support complete recovery as an implemented capability. They do not establish this monitoring policy as an optimal learning strategy. The exact zero checkpoint contrasts reflect matched trajectories.

All six methods were rerun separately at seed 20 across eight conditions. All 48 final outcomes and decision counts matched the retained primary runs. The audit rerun also reproduced its counts. These repeat existing configurations and add no independent samples. The separate research artifact includes `revision_analysis.py` for those comparisons and Table 8.

E Original language-model protocol and detailed outcomes

E.1 Checkpoints and record selection

SmolLM2-135M-Instruct uses revision `12fd25f77366fa6b3b4b768ec3050bf629380bac`. Qwen2.5-0.5B uses revision `060db6499f32faf8b98477b0a26969ef7d8b9987`. The checkpoint cards specify Apache 2.0. Models load locally without remote model code. Downloaded

files are hash-tracked. Pretrained base weights are downloaded separately rather than duplicated in the research artifact.

CounterFact contains 21919 records. A record supplies a rewrite template and subject with new and original targets. It also supplies paraphrase and neighborhood prompts. Eligibility requires at least two paraphrases and three neighbors plus nonempty targets and a subject not already selected. Eligible records are ordered by a SHA-256 key derived from a fixed selection seed and case identifier. Selection makes no model queries. The first eight records form development data and the next 64 form evaluation data. Subjects are disjoint between these subsets. Training and screening text are checked against final paraphrase text for normalized duplicates.

The experimental unit is an ordered 64-edit stream. Prompts from a case are correlated observations. Each record's first paraphrase is reserved for gain screening. Remaining paraphrases support final measurement. Neighborhood prompt 0 enters the reusable screen and prompt 1 enters monitoring. Prompts from index 2 onward support final locality evaluation. Sixteen records within each shuffled stream supply the reusable screening and monitoring panels. All 64 assigned edits remain in final evaluation even if rejected or later removed.

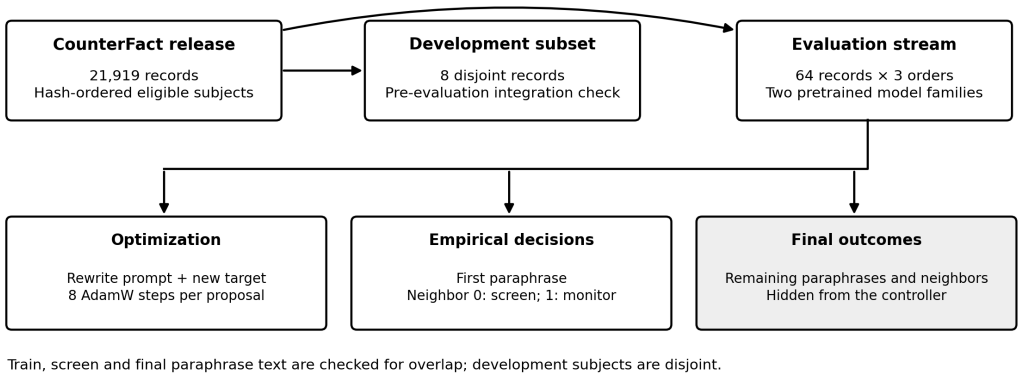


Figure 6: Intended data roles in the original CounterFact protocol. Three shared neighborhood strings were found during revision. Appendix H reports that limitation. The new stream enforces disjoint roles globally.

E.2 Tokenization and adaptation thresholds

Backbones operate in bfloat16 and adapters in float32. Rank-eight adapters are inserted into query and value projections. SmolLM2 has 460800 adapter parameters out of 134975808 total parameters. Qwen2.5 has 540672 out of 494573440. Backbone parameters remain frozen. Evaluation mode disables dropout while adapter gradients remain enabled for optimization.

Each proposal receives eight AdamW steps at learning rate 0.001 with zero weight decay and gradient clipping at norm 1. A minibatch contains the current rewrite plus up to three uniformly sampled active replay examples. A new optimizer is initialized for each proposal. No moments carry across events. Replay holds the 16 most recent admitted rewrites. Rejected examples and descendants removed by recovery do not remain active. Event randomness is derived from stream seed and event index.

Prompts are tokenized without automatically added special tokens. Only the last 64 prompt tokens are retained. Targets are tokenized separately with a leading space and capped at 12 tokens. Loss is evaluated only on target positions after the causal shift. Later target tokens condition on earlier target tokens through teacher forcing. This protocol may differ from other CounterFact implementations. The reported scores should not be substituted directly into external leaderboard comparisons.

The empirical gate requires target-screen NLL improvement of at least 0.01. Mean protected NLL may rise by at most 0.20 against the parent and 0.40 against the frozen base. The allowance for recent edits is 0.25. That panel contains the last eight active edit identifiers. Monitoring runs every

eight events with frozen-reference allowance 0.55. All are fixed reused engineering thresholds. Raw NLL is unbounded and does not meet the loss assumption of Proposition 3.

E.3 Memory and hybrid routes

Ungated retrieval stores the 64 supplied assertions and leaves model parameters frozen. A subject match in the prompt triggers a short prepended counterfactual update statement. The longest subject is selected when several match. The benchmark supplies both the subject and target. The system does not infer that either assertion is true world knowledge.

APA-M proposes one semantic entry per event without changing backbone or adapters. Its screen evaluates the retrieval-conditioned candidate. APA-H combines the entry with replay-assisted adapter training and effective-weight backtracking. Parent and candidate evaluation use the same case-insensitive whole-subject matching rule within these APA routes. Monitoring and final inference use that rule as well. Entries retain source and scope with creation-event metadata. They are committed and restored with the complete state.

Semantic memory can hold all 64 facts. Replay holds 16 examples. Retrieval consumes prompt tokens during inference. These are different capacities and serving costs. The memory advantage is not an equal-resource comparison. All three trained gated routes share the same proposal-training step budget. Their screening work can differ. The checkpoint control matches policy and isolates state-management behavior. It does not establish equal end-to-end computation.

APA-M and APA-H were added after limited locality was observed in the initial SmolLM2 parameter-update study. They are exploratory extensions evaluated on the same selected dataset. Every route is reported. Independent confirmation would require a new evaluation stream with the route and matching policy fixed beforehand.

E.4 Metric definitions

For prompt x and nonempty target sequence $y_{1:L}$ define normalized target NLL as

$$N_S(x, y) = -\frac{1}{L} \sum_{i=1}^{L} \log p_S\left(y_i \mid x, y_{<i}\right) \tag{20}$$

New-answer preference is

$$e_S(x) = \mathbf{1}\{N_S\left(x, y_{new}\right) < N_S\left(x, y_{original}\right)\} \tag{21}$$

Rewrite preference uses the supplied rewrite prompt. Paraphrase preference averages held-out paraphrases within each case and then across cases. Neighborhood original-answer preference reverses the inequality on neighborhood prompts. Ties satisfy neither strict preference. The metrics compare two supplied answers. They do not measure unconstrained generation.

For a final neighborhood prompt next-token consistency is

$$c_S(x) = \mathbf{1}\{\arg\max_y p_S(y \mid x) = \arg\max_y p_{S_0}(y \mid x)\} \tag{22}$$

Use a fixed tie-breaking convention for the argmax. The comparison is to frozen checkpoint S_0 and preserves its mistakes as well as its correct predictions. The neighborhood set probes locality around edited facts. Calling these prompts unrelated means they are not the direct rewrite target. It does not make them a broad general-capability benchmark.

Original-answer NLL change is retained as another measure. A low mean screening NLL regression does not imply high final next-token consistency. The metrics differ and the samples differ. Their disagreement does not contradict Proposition 3 because these experiments do not invoke it.

Accepted coverage counts edits whose proposals were committed at least once. Active coverage counts identifiers in the final surviving lineage. Recovery can make active coverage much lower. The early-edit measure is final rewrite preference on the first quarter of the assigned stream. It is not a conventional backward-transfer score or maximum-to-final forgetting measure.

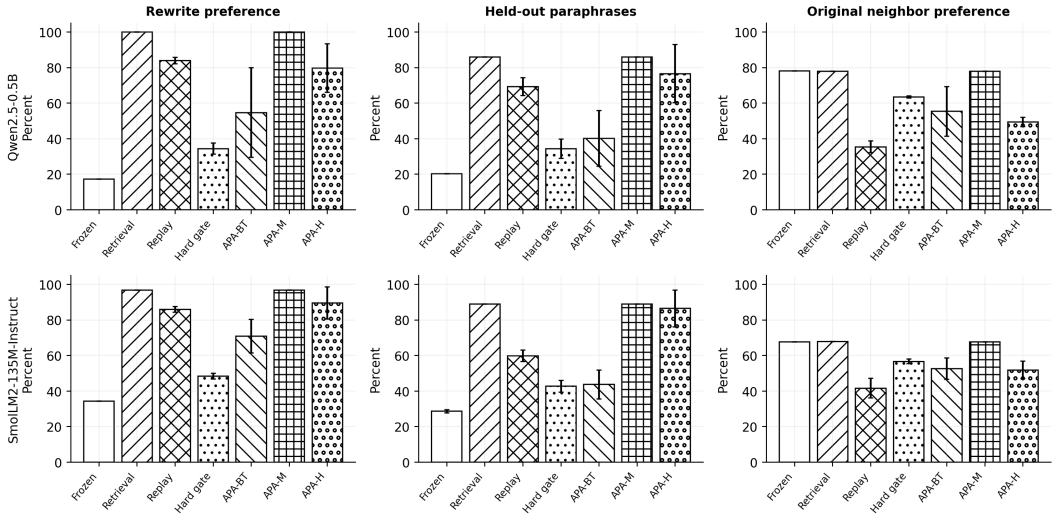


Figure 7: Language-model outcomes. Error bars show one sample SD across three update orders. Acquisition and neighborhood preservation measure distinct behaviors. These original aggregates precede the overlap check in Appendix H.

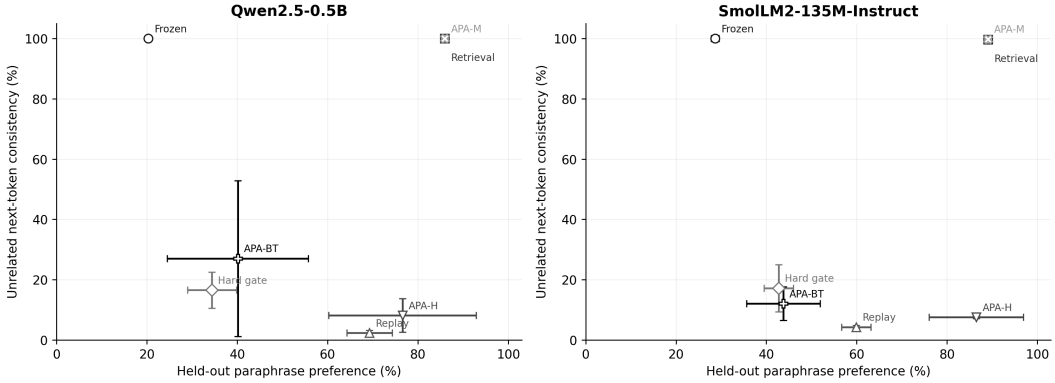


Figure 8: Original paraphrase preference and neighborhood next-token consistency averaged across three orders. Acquisition of requested edits does not establish preservation of unrelated behavior.

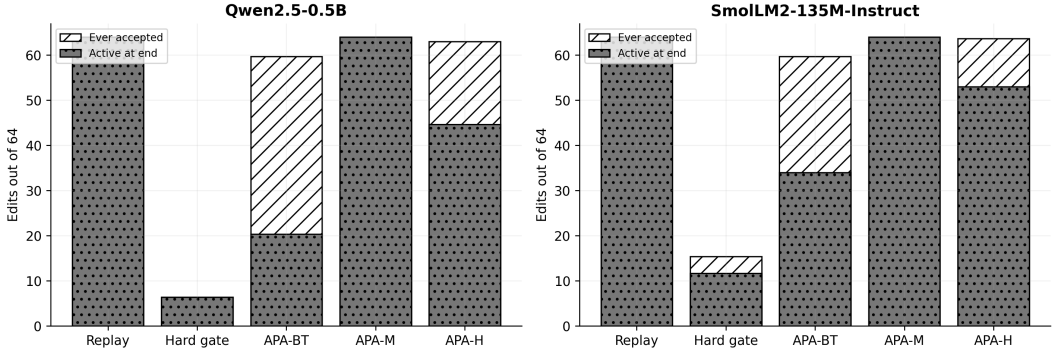


Figure 9: Ever-accepted and finally active updates after 64 events. The difference measures updates discarded by recovery. Rejected and discarded edits remain in the final evaluation denominator.

Table 9: Coverage and resource accounting per 64-event stream averaged across three orders.

Method	Accepted	Active	Rollbacks	Train	Screen	VRAM
SmolLM2						
Replay	64.0	64.0	0.0	50.0	0.0	336.1
Hard gate	15.3	11.7	1.0	49.4	20.4	336.1
APA-BT	59.7	34.0	3.3	48.1	34.2	336.1
APA-M	64.0	64.0	0.0	0.0	20.4	325.4
APA-H	63.7	53.0	1.3	49.6	31.1	410.9
Qwen2.5						
Replay	64.0	64.0	0.0	40.9	0.0	1084.4
Hard gate	6.3	6.3	0.0	41.1	17.7	1084.4
APA-BT	59.7	20.3	5.3	41.2	35.5	1084.4
APA-M	64.0	64.0	0.0	0.0	18.0	1111.5
APA-H	63.0	44.7	2.3	40.8	36.6	1228.5

E.5 Complete results and resource accounting

Training and screening times are in seconds. VRAM is peak PyTorch allocation in MiB. GPU timing synchronizes the device before elapsed time is read. Training covers proposal optimization. Screening covers candidate search and gate evaluation. The timers exclude monitoring and final evaluation as well as loading and serialization. Transaction and checkpoint-copy costs are recorded separately in the research artifact. The displayed timings are not complete adaptation latency.

Peak PyTorch allocation includes loaded model tensors managed by the process. It excludes other processes and driver-managed memory. CPU snapshots and disk storage are also excluded. A small trainable-parameter fraction is neither a FLOP ratio nor a serving-latency bound. The research artifact retains per-prompt NLL with case identifiers and frozen next-token predictions. Event records identify the decision and selected scale. Retained adapter states support numerical checks.

The three order seeds describe these records and checkpoints only. Sample SD across orders does not establish population uncertainty over models or datasets. It also combines order changes with any numerical variability in the execution. Small frozen-score differences may reflect reduced-precision batch effects but the retained aggregates do not isolate their cause. A fixed-batch deterministic rerun would be needed to do so. We retain the reported values rather than replace them with artificial zero dispersion.

F Audit simulation and software verification

Each synthetic run has 60 attempts alternating between 30 beneficial and 30 harmful proposals. Protected paired differences have support $\{-1, 0, +1\}$. Beneficial probabilities are $(0.02, 0.96, 0.02)$ and harmful probabilities are $(0.02, 0.90, 0.08)$. Protected mean regressions are therefore 0 and 0.06. Gain observations use $(0.05, 0.70, 0.25)$ on the same support for both proposal types. Mean gain is 0.20.

The protected tolerance is 0.05 and required gain is 0.01. The lifetime budget is $\delta = 0.05$. Each panel size has 1000 independent runs generated from multinomial counts. Within a run all three rules receive the same sampled observations. The seed is $90000 + n$ for panel size n.

False-commit frequency is the fraction of runs with at least one harmful acceptance. Beneficial acceptance is the fraction of the 30000 beneficial attempts accepted. Panel sizes define separate conditions and are not pooled into one binomial experiment.

Unrestricted empirical screening has high beneficial acceptance at small panel sizes but often accepts a harmful proposal within a run. Both lifetime-budgeted procedures show no false-commit runs under these conditions. They are conservative at small panel sizes. Empirical Bernstein has greater power than Hoeffding for these low-variance paired differences.

Zero events among 1000 independent runs give exact one-sided 95% upper confidence limit $1 - 0.05^{1/1000} \approx 0.002991$. This is about 0.30% for the simulated configuration. It is not zero risk. The

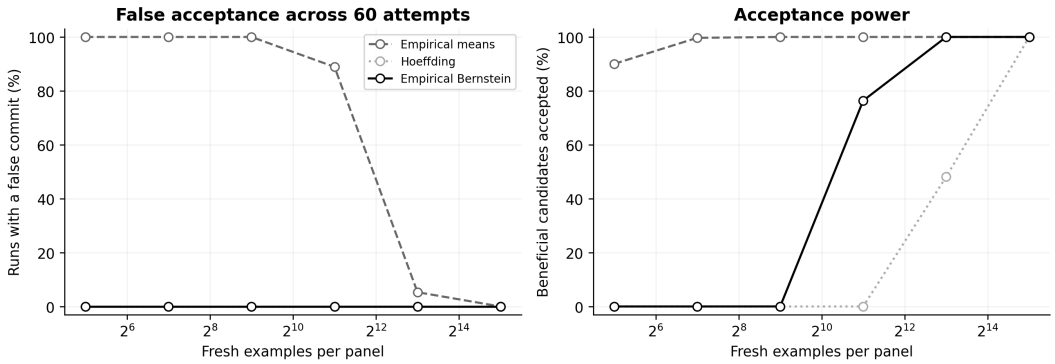


Figure 10: Fresh-panel audit simulation. A false commit is at least one harmful acceptance within 60 attempts. Empirical Bernstein bounds have greater power for these low-variance paired losses; both lifetime-budgeted procedures are conservative for small panels.

Table 10: Synthetic audit results in percent. False is the fraction of 1000 runs with a harmful acceptance in 60 attempts; beneficial is the fraction of 30000 beneficial proposals accepted.

	Empirical		Hoeffding		Bernstein	
Panel	False	Beneficial	False	Beneficial	False	Beneficial
32	100.00	90.02	0.00	0.00	0.00	0.00
128	100.00	99.67	0.00	0.00	0.00	0.00
512	100.00	100.00	0.00	0.00	0.00	0.01
2048	88.90	100.00	0.00	0.00	0.00	76.38
8192	5.30	100.00	0.00	48.14	0.00	100.00
32768	0.00	100.00	0.00	100.00	0.00	100.00

analytical lifetime bound remains $\delta = 0.05$ under Proposition 3. The simulation does not certify the language-model editing streams.

Table 11: Representative executed invariants. Complete test names and reports are retained with the research source.

Injected condition	Required observation
Candidate prepared before another activation	Old captured parent rejected before proposal registration
Commit attempted without validation	Activation rejected and active state unchanged
Caller mutates candidate or active-state copy	Stored candidate and active bytes remain unchanged
Two siblings share an old parent	One commit makes the other stale
Commit followed by rollback to equal values	Old proposal remains stale because the version advanced
Fault discovered after dependent commits	All declared learning-state components restore together
Restart after rollback	Active version and spent audit counter remain advanced
Interruption before active-pointer publication	The previously published complete state remains authoritative
Corrupted content-addressed manifest	Recovery fails closed instead of loading altered state
Second live writer	Operating-system lease rejects concurrent ownership
Rank-factor rescaling	Interpolated effective matrix matches the dense reference
Malformed or unbounded audit input	No acceptance and no silent confidence claim
Scope expiry or semantic contradiction	Ineligible retrieval excluded and conflicting proposal quarantined
Consolidation candidate constructed	Parent remains unchanged and linear effective map is preserved

These tests exercise particular inputs and failure traces. They are not exhaustive verification. The verification suite comprises 49 unit and fault tests plus a bounded explorer covering 19531 schedules and 112305 attempted transitions through depth six. A correct recovery implementation can coexist with an inaccurate monitor or an overfit gate.

The earlier parent-capture correction was checked with 48 seed-20 reruns. The present revision re-executes all 480 primary digit runs and reproduces every final accuracy and decision count. The release verifier also compares 1200 paired digit event states and 1920 language-model shadow events. These are consistency checks rather than additional experimental samples.

Targeted GPU tests use one development edit per model. They check unchanged frozen-backbone hashes and expected adapter changes. They also check replay and semantic-memory restoration together with bitwise restoration of same-batch target NLL on the tested path. This does not establish cross-hardware CUDA determinism.

The release retains additional routing and consolidation utilities. They are outside the empirical contribution of this paper. Stored confidence is not evidence that an assertion is true.

G Reproduction and priorities for further evaluation

The manuscript package contains editable LaTeX and figures with the conference style. The experimental research artifact is separate. It contains implementation modules and tests alongside configurations and provenance records. It also retains per-seed and per-event outputs with analysis scripts and trained adapter states. Aggregate tables are generated from those records. Public model weights are downloaded separately at pinned revisions.

CPU experiments use Python and NumPy. GPU experiments additionally require CUDA-capable PyTorch with Transformers 4.57.1 plus tokenizers and safetensors. The reported GPU environment uses Python 3.14 with PyTorch 2.11.0 and CUDA 12.8. Matplotlib generates the scientific figures. The research artifact records experimental commands and dependency versions. Manuscript compilation uses a separate environment.

The revision adds the experiments in the following appendices. Remaining priorities include larger language models and broader development grids for editing baselines. Natural alias discovery and independently collected audit labels are outside the present evidence. A data-driven detector should be assessed by time to detection and harmful predictions exposed before recovery. Selective recovery and storage compaction would need their own state-consistency checks. Learned routing and semantic consolidation remain unevaluated design extensions.

H Original 64-edit language-model results

The following results preserve the original two-model study for comparison. The revision found three normalized strings shared by potential screening anchors and final neighborhood prompts. Their presence invalidates an unconditional claim that every original neighborhood query was held out. Paraphrase roles remain disjoint. These unfiltered aggregates are retained as historical results. The new 256-record study excludes all cross-role and cross-record prompt overlap. The artifact also provides a post hoc overlap-filtered reanalysis of the original scores.

Table 12 separates acquisition from preservation. Rewrite and paraphrase preference concern the supplied counterfactual target. Neighbor preference concerns original answers. Consistency measures agreement with frozen next-token predictions on held-out neighborhood prompts. Those prompts test local effects around the edited facts rather than broad general ability.

APA-M reaches 89.1% paraphrase preference and 99.8% consistency on SmolLM2. The corresponding Qwen2.5 scores are 85.9% and 100.0%. Ungated retrieval matches both paraphrase scores. Every APA-M proposal passes screening across all six model-order streams. The experiment demonstrates that memory can be integrated into the transaction protocol. It does not measure a screening benefit for acquisition or show that the memory gate rejects harmful facts. The two routes also use different lexical matching rules. Their small locality differences cannot be attributed to gating alone.

Replay reaches 59.9% and 69.3% paraphrase preference but only 4.3% and 2.3% consistency. APA-BT raises consistency to 12.0% and 27.0% while reducing paraphrase preference to 43.8% and 40.1%. On Qwen2.5 an average of 59.7 proposals are accepted but only 20.3 remain active after recovery. Order variability is substantial. Protected NLL on the small reused screen is a weak proxy for held-out next-token preservation. APA-H retains high acquisition without resolving this problem. Its paraphrase scores are 86.5% and 76.6% while consistency is 7.6% and 8.1%.

Table 12: CounterFact outcomes in percent after 64 edits. Mean ± sample SD across three orders.

Method	Rewrite	Paraphrase	Neighbor	Consistency
SmolLM2				
Frozen	34.4 ± 0.0	28.6 ± 0.9	67.8 ± 0.0	100.0 ± 0.0
Retrieval	96.9 ± 0.0	89.1 ± 0.0	68.0 ± 0.0	99.6 ± 0.0
Replay	85.9 ± 1.6	59.9 ± 3.3	41.6 ± 5.5	4.3 ± 0.5
Hard gate	48.4 ± 1.6	42.7 ± 3.3	56.7 ± 1.3	17.2 ± 7.8
APA-BT	70.8 ± 9.4	43.8 ± 8.1	52.6 ± 6.0	12.0 ± 5.6
APA-M	96.9 ± 0.0	89.1 ± 0.0	67.8 ± 0.0	99.8 ± 0.0
APA-H	89.6 ± 9.0	86.5 ± 10.4	51.9 ± 4.9	7.6 ± 0.5
Qwen2.5				
Frozen	17.2 ± 0.0	20.3 ± 0.0	78.1 ± 0.0	100.0 ± 0.0
Retrieval	100.0 ± 0.0	85.9 ± 0.0	77.9 ± 0.0	100.0 ± 0.0
Replay	83.9 ± 1.8	69.3 ± 5.0	35.4 ± 3.4	2.3 ± 0.8
Hard gate	34.4 ± 3.1	34.4 ± 5.4	63.5 ± 0.7	16.5 ± 6.0
APA-BT	54.7 ± 25.1	40.1 ± 15.7	55.3 ± 14.0	27.0 ± 25.8
APA-M	100.0 ± 0.0	85.9 ± 0.0	77.9 ± 0.0	100.0 ± 0.0
APA-H	79.7 ± 13.6	76.6 ± 16.3	49.4 ± 2.6	8.1 ± 5.6

The routes use unequal resources. Semantic memory can retain all 64 facts while replay holds 16 examples. Retrieval also consumes inference tokens. Backtracking shares the proposal-training budget with hard gating but adds screening passes. Mean screening time per SmolLM2 stream is 34.2 seconds for APA-BT and 20.4 for hard gating. Qwen2.5 times are 35.5 and 17.7 seconds. These timers omit monitoring and final evaluation. They also exclude model loading. The matched checkpoint isolates state-management behavior under matched policy. It is not evidence of equal end-to-end cost. Appendix E.5 gives coverage and resource accounting.

I Real-data fault ablations and recovery cost

The mechanism study uses optdigits seeds 0 through 9. These runs are separate from the primary learning comparison. Backbone training and the rank-four proposer use the original defaults. Each seed produces one candidate for classes 5 and 6 and another for classes 7 and 8. Both start from the same captured parent. Each has positive target cross-entropy gain on its assigned development classes. This diagnostic evidence certifies that measured gain only. It is not a full preservation certificate.

All publication variants receive the same two candidates and evidence. The delayed-sibling schedule validates both before committing the first. It then attempts the second commit. The ABA schedule inserts a complete recovery to the initial content before that attempt. The substitution schedule changes the stored candidate bytes after validation while retaining the approved descriptor. This intervention deliberately bypasses the public immutable interface. It tests a defensive check against an integration or storage fault.

The sequential checkpoint is the control released with the original study. It copies complete state but has no concurrent parent check. The first APA ablation disables the parent check at validation and commit. The second retains parent-content checks while ignoring activation versions. The third retains parent checks while omitting candidate-digest verification at commit. The preparation API remains unchanged. Table 2 reports ten repetitions per schedule and variant. Every blocked attempt leaves the active state unchanged.

The stale sibling loses 89.4 percentage points of accuracy on the first candidate's classes when publication is allowed. An ABA violation need not immediately change predictive accuracy. Its failure is reuse of evidence from a superseded activation. The substitution diagnostic likewise tests authorization rather than requiring a particular accuracy loss.

A further intervention constructs a candidate from the initial parent but falsely attaches the current parent identity after another commit. All ten such proposals publish when the caller supplies

passing evidence. The contract cannot establish computational provenance against that caller. The implementation and Lemma 1 therefore require a trusted construction path. This negative control is retained in the raw results.

For delayed recovery the initial replay labels are cyclically shifted within old classes. The adapter is initially unchanged. One corrupted state is published. Subsequent clean updates cycle through the three new-class groups until a scheduled notification boundary at event 1 or 4 or 8. The notification is supplied by the experiment. No claim is made that the ordinary monitor discovers this corruption at those times.

Table 13: Accuracy after the next identical clean update. Mean and sample SD across ten seeds. Removed descendants had positive immediate development gain.

Notification event	Clean removed	Old full	Old weights	New full	New weights
1	0	93.1 ± 1.1	7.3 ± 6.1	37.1 ± 5.7	35.9 ± 1.6
4	3	93.1 ± 1.1	53.7 ± 18.0	37.1 ± 5.7	37.9 ± 1.2
8	7	93.1 ± 1.1	54.3 ± 19.1	37.1 ± 5.7	38.0 ± 1.1

Every clean descendant in these schedules has positive immediate development-loss gain on its target classes. Full recovery discards those descendants along with the corrupted state. A weight-only control restores the same adapter factors while retaining the current replay state. Both then receive the same clean update with identical event randomness. Table 13 measures the resulting divergence. The term beneficial descendant refers to its immediate measured gain. It does not imply that the descendant would remain useful on every later task.

J Independent 256-edit language-model extension

The dataset file is the same pinned CounterFact release as the original study. A new selection seed of 20260920 orders records by the SHA-256 digest of the seed and case identifier. Selection excludes all original development and evaluation subjects and normalized prompts. It also excludes any candidate record whose prompts overlap a previously selected record or whose own final prompts overlap its training or screening or monitoring roles. No model score enters these filters. The retained 256 cases supply 256 final paraphrases and 2048 final neighborhood prompts. The artifact stores exact identifiers and text.

The new selection is an exploratory extension of the original study. The initial revision pilot was stopped after a text-only audit found cross-role duplication. Its results are not pooled into the reported study. A separate batch-one pilot was used to investigate evaluation grouping. The reported extension fixes canonical case ordering and batch size before comparing completed method streams.

SmolLM2 and its revision match Appendix E.1. Update orders use NumPy permutations with seeds 2031 through 2033. APA-BT uses the original rank-eight query and value adapters. A training batch contains the current example and up to three replay examples. AdamW uses learning rate 0.001 and eight steps. The empirical screen and monitor retain the original thresholds. The policy is unchanged by inspection of extension test scores.

The two memory routes share a case-insensitive whole-subject retriever. The longest matching subject selects a prepended update statement. Both have capacity for 256 assertions. Their adapter tensors stay frozen. The ungated route accepts each proposed insertion. APA-M evaluates that same complete candidate with the empirical gate. Retrieval applies during screening and final inference. The 64-token input cap includes prepended memory content. These controls remove the lexical-matching and capacity differences in the original memory comparison.

J.1 GRACE implementation and development selection

The baseline loads the official GRACE editor at commit `f674183f17a995d109e10ee6140d4c3e6d016115` from the authors' repository.[1] The source SHA-256 is retained with the research

[1] https://github.com/Thartvigsen/GRACE/tree/f674183f17a995d109e10ee6140d4c3e6d016115

artifact. A fetch script retrieves that exact file for reproduction. Experiment-tracking utilities are excluded from the import path. The editor operates on `model.layers[22].mlp.up_proj`. Its query is the last prompt-token activation. The upstream `replace_prompt` rule replaces earlier prompt positions with the selected value.

GRACE keeps the backbone frozen. Its codebook uses Euclidean distance with cold value initialization and coverage-based radius expansion. Development tests combine radii 1 and 5 with Adam learning rates 0.1 and 1. Both eight-step and 100-step settings are evaluated on the original eight development cases. Selection maximizes the mean of rewrite preference and paraphrase preference and original-neighbor preference. Ties retain the first configuration in the declared grid. The main comparison uses the best eight-step configuration to match the number of optimization steps per edit.

Table 14: GRACE development selection on eight cases. Scores are percentages. These cases are excluded from the new stream.

Steps	Radius	Rate	Rewrite	Paraphrase	Neighbor	Mean
8	1	0.1	75.0	25.0	70.3	56.8
8	1	1.0	100.0	25.0	68.8	64.6
8	5	0.1	75.0	25.0	67.2	55.7
8	5	1.0	100.0	25.0	56.2	60.4
100	1	0.1	100.0	25.0	68.8	64.6
100	1	1.0	100.0	25.0	68.8	64.6
100	5	0.1	100.0	25.0	56.2	60.4
100	5	1.0	100.0	25.0	56.2	60.4

The selected eight-step configuration uses radius 1 and learning rate 1. The best 100-step development score is the same. This does not establish that more steps can never help or that the grid finds the best possible GRACE configuration. The comparison concerns this adaptation of the method to SmolLM2.

Two compatibility edits are explicit. Distance evaluation is promoted to float32 because CUDA distance computation does not support the bfloat16 key representation used here. During inference examples are grouped by prompt-token count so they share the query position. The replacement mask and values are broadcast over the token dimension explicitly. Training retains batch size one. A float32 linear-layer test reproduces upstream batch-one training exactly over eight steps. Batched inference differs from concatenated single-example inference by less than 6×10^{-8} in that fixture. This does not prove numerical identity for every bfloat16 model input.

GRACE and APA-BT receive the same rewrite labels and token limits. Both take eight optimizer steps. GRACE updates codebook values without replay while APA-BT processes replay examples and may screen several interpolated candidates. Their parameter counts and operation counts therefore differ. The memory routes do no parameter optimization. Table 15 reports measured work instead of treating step count as equal compute.

Table 15: Measured work per 256-edit stream averaged across three orders. Active is the active edit count except for GRACE where it is codebook size. Full time includes final evaluation and excludes model loading.

Method	Accepted	Active	Train s	Screen s	Full s
Frozen	0.0	0.0	0.0	0.0	22.4
GRACE	256.0	255.0	102.0	0.0	128.9
APA-BT	234.0	126.0	183.0	148.6	367.0
Ungated memory	256.0	256.0	0.0	0.0	35.1
APA-M	255.0	255.0	0.0	80.6	119.8

Final evaluation uses canonical case order. GRACE retains a matching frozen reference with the same query-position grouping for next-token consistency. The other routes use the shared fixed-

batch frozen reference. All scores are case-macro averages. Standard deviations describe three update orders. They do not quantify uncertainty over all possible benchmarks or checkpoints.

K MATCHED MEMORY INSERTION STRESS TEST

The stress test resets state between interventions. It uses the first 32 text-selected extension records and separate anchors from records 33 through 48. All policies receive identical parent and candidate bytes. They share scope filtering and whole-name matching over canonical subjects and explicit alias lists. The memory capacity is 256 and token limits match the longer study.

The six interventions are clean insertion and replacement of the requested target by the original answer and a scope mismatch and a contradictory insertion and an alias collision and a declared alias. A contradictory insertion starts from the accepted requested assertion for that subject. An alias collision assigns an existing subject name to a different entity. The declared-alias condition supplies a synthetic alias identifier. Its final query explicitly refers to that alias. These fixtures test declared lexical routing. They do not test discovery of natural aliases.

Wrong targets are wrong relative to the supplied counterfactual editing objective. The validator retains that intended answer independently of the injected proposal. This is an information assumption. If the same compromised source supplies both the proposed assertion and its validation target then the gate has no independent basis for detecting the corruption.

The NLL comparator requires intended-target gain of 0.01 and local anchor regression at most 0.20. It isolates these empirical checks rather than rerunning the full long-stream controller. The strengthened gate additionally requires positive preference for the intended answer over the competing answer and a preference-margin gain of 0.01. Metadata checks require one isolated insertion within capacity and the declared scope. Conflicting assertions and cross-entity alias collisions are quarantined. The gate thresholds are fixed before this stress test. No statistical confidence certificate is attached to these reused screens.

Table 16: Matched insertion tests. A is the number admitted out of 32. P is final intended-answer preference in percent. The same retriever is used throughout.

Insertion	Ungated A	P	NLL A	P	New gate A	P
Clean	32/32	87.5	32/32	87.5	25/32	75.0
Wrong target	32/32	3.1	14/32	15.6	0/32	25.0
Scope mismatch	32/32	25.0	0/32	25.0	0/32	25.0
Contradiction	32/32	87.5	0/32	87.5	0/32	87.5
Alias collision	32/32	87.5	32/32	87.5	0/32	25.0
Declared alias	32/32	81.2	31/32	81.2	23/32	78.1

Both acceptance and held-out preference are reported in Table 16. A rejected clean insertion is a cost. Scope filtering is common to every policy so exclusion of an out-of-scope retrieval cannot be credited to the gate alone. The metadata check also prevents ambiguous content from entering persistent state. The final paraphrase is separate from the screen. The declared-alias fixture adds an explicit alias reference to that final paraphrase.

L BOUNDED NEURAL-LOSS AUDIT OVER A FINITE BENCHMARK

This exercise tests integration of the reservation and concentration rules with a trained neural predictor. It does not transfer the fresh-audit theorem to the reused language-model screens. For each optdigits seed from 0 through 9 one replay-assisted candidate learns classes 5 through 9 from the optimization split. The candidate uses the original proposer defaults. No official-test score selects the candidate or its clipping rule.

The loss is $\ell = \min\{\mathrm{NLL}/10, 1\}$. The protected population is uniform over the 901 official test cases from classes 0 through 4. The gain population is uniform over the remaining 896 cases. These are finite empirical distributions. The maximum allowed mean regression is 0.05 and required mean gain is 0.01. Each method receives its own lifetime budget of 0.05. Its four attempts use panel sizes

128 and 512 and 2048 and 8192 in that order. A method is selected before each panel. The two rules are compared separately rather than choosing the smaller observed bound.

The durable store persists the attempt and per-claim allocation before invoking the sampler. Operating-system randomness is requested only after that reservation. Indices are sampled independently with replacement within each group. Exact index arrays are retained for replay. Candidate and comparator digests are checked again after sampling. A changed parent prevents publication. Sampling failure still consumes the reserved attempt.

Table 17: Finite-benchmark neural audit. PASS counts out of ten models. The last column counts distinct existing cases in both Bernstein panels on average.

Draws per panel	Hoeffding	Bernstein	Distinct cases
128	0/10	0/10	238.1
512	0/10	0/10	779.9
2048	0/10	10/10	1611.3
8192	0/10	10/10	1796.7

All ten candidates satisfy the exact finite-population constraints. Mean protected regression ranges from 0.0213 to 0.0263. Mean gain ranges from 0.7003 to 0.7225. Thus Table 17 measures power for acceptable candidates. It contains no estimate of neural false-acceptance frequency.

Each fixed model pair is evaluated on all 1797 test cases to form the finite populations. Subsequent index draws require no additional model passes. The experiment uses existing benchmark labels and collects zero new labels. Full enumeration already determines the finite-population means. Sampling here exercises the audit implementation and its bounds rather than offering a cheaper benchmark evaluation. A guarantee for future observations requires a trusted sampling protocol for that external population.

M DURABLE PUBLICATION SCALING AND READER VERIFICATION

The storage benchmark uses the released SmolLM2 adapter and replay state from the original APA-BT run at seed 2031. Its 460800 float32 adapter values occupy 1.76 MiB. JSON serialization with base64 arrays and metadata occupies 2.35 MiB. Each event changes one adapter scalar and the event identifier to create a distinct snapshot. This is a storage workload using the actual state format. It is not an additional learning result.

Each complete event captures the parent and constructs a detached candidate. It publishes the proposal and validation record and commit. A final capture and digest check observes the result. All flushes performed by the durable implementation are included. Learning and neural monitoring and initial model loading are excluded. The experiment retains every snapshot and candidate and full manifest.

The uncached control serializes all retained states for every manifest. The revised path caches the immutable snapshot and candidate addresses. Both write identical content and maintain the same authoritative state. Deserialization still verifies every referenced file during recovery. Cached addresses do not imply continuous background detection of newly corrupted disk files. The corruption guarantee applies when those files are read.

Table 18 reports medians across three repetitions on the local Windows filesystem with warm caches. Runtime timing is descriptive for the shared workstation. It is not a claim about another filesystem or power-loss durability. Both variants retain the same blob set after the tested sequence and rollback. The source artifact records each timing component and process-memory measurement.

At 64 commits the store retains 228.5 MiB of NumPy arrays across its internal states. This figure excludes the Python runtime and temporary caller copies. Process working-set measurements include those other allocations and are retained separately. Snapshot storage grows with retained versions. Full manifests include increasingly long histories so their cumulative metadata can grow quadratically. Garbage collection and compact journals remain future work.

Table 18: Complete durable event cost and retained state. Medians of three repetitions. State arrays exclude runtime and temporary copies.

Event	Uncached seconds	Cached seconds	Files MiB	Arrays MiB
1	0.141	0.047	4.7	7.0
8	0.629	0.050	21.4	31.6
16	1.290	0.050	40.7	59.8
32	2.308	0.051	80.1	116.0
64	4.670	0.055	162.7	228.5

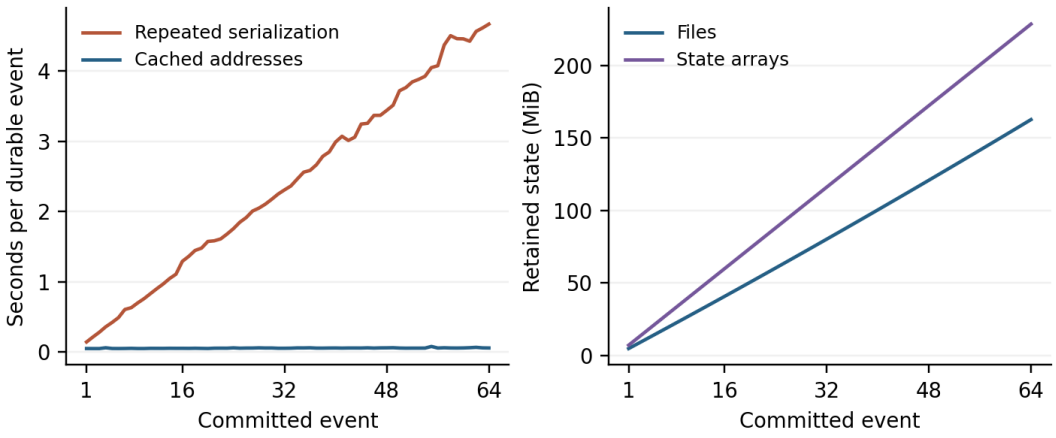


Figure 11: Measured durable event latency and retained storage. Curves use medians over three repetitions. Reference caching removes repeated historical serialization. It does not remove retained snapshots or full manifests.

Reader tests pause publication immediately before replacement of the authoritative pointer. Concurrent readers request state and version and ledger and captured parent and candidate and ledger verification. Every read waits until publication succeeds or the old published state has been reloaded after an injected failure. Reopening the store reproduces the same result. A separate test corrupts a cached snapshot file and verifies that recovery refuses it. These tests cover the identified race and stated fault paths. They do not exhaust every possible execution.

N Reproducibility findings from the revision

The supplied archive made independent checks possible. Selection regeneration reproduces the original eight development and 64 evaluation records exactly. All 80 primary digit split files match the declared within-class seed rule. Reexecuting all 480 primary digit runs reproduces final accuracies and decision counts exactly. The raw retained checkpoint comparisons cover 1200 digit event pairs and 1920 original language-model shadow events.

The original neighborhood split shares three normalized query strings with potential screens. Removing those three instances leaves 509 neighborhood prompts across all 64 cases. Table 19 reaggregates their retained scores without retraining. It is a post hoc sensitivity check. Original figures and Table 12 retain the unfiltered numbers. The new extension enforces global role separation before scoring.

Table 19: Original neighborhood scores after excluding three overlapping query instances. Mean and sample SD across three orders. S denotes SmolLM2 and Q denotes Qwen2.5.

Method	S neighbor	S consistency	Q neighbor	Q consistency
Frozen	67.7 ± 0.0	100.0 ± 0.0	78.0 ± 0.0	100.0 ± 0.0
Retrieval	67.9 ± 0.0	99.6 ± 0.0	77.8 ± 0.0	100.0 ± 0.0
Replay	41.6 ± 5.6	4.1 ± 0.5	35.4 ± 3.4	2.4 ± 0.8
Hard gate	56.7 ± 1.3	17.3 ± 7.8	63.4 ± 0.6	16.6 ± 6.0
APA-BT	52.5 ± 6.1	12.0 ± 5.5	55.2 ± 13.7	27.1 ± 25.8
APA-M	67.7 ± 0.0	99.8 ± 0.0	77.8 ± 0.0	100.0 ± 0.0
APA-H	51.8 ± 4.8	7.6 ± 0.4	49.4 ± 2.6	8.2 ± 5.6

The original frozen scores also vary slightly with evaluation order. A controlled rerun holds one frozen model fixed while changing only the input grouping. It reproduces the retained order-dependent differences. Repeating the same grouping is exact on the tested hardware. Canonical ordering is therefore used in the new extension. This check does not establish cross-hardware determinism.

Pinned model files are verified by SHA-256. The artifact retains same-batch frozen-score checks and saved-adapter reload checks. Separate development-edit probes verify unchanged backbone hashes and complete recovery of adapter and replay and semantic memory. These are targeted checks of declared behavior. They are not evidence that every harmful update will be detected.

The revised CPU suite contains 49 unit and fault tests. Additional GRACE compatibility and model-level checks are reported separately. Reproduction scripts save protocol files and per-event records and final per-case scores. Base model weights and downloaded third-party source are fetched separately at pinned revisions. The manuscript source and the research artifact are supplied as separate replacement packages.