A PASSING PATCH CAN BREAK A LATER TASK: REVOCABLE LEARNING FROM VERIFIED AGENT TRACES

Anonymous authors
Paper under double-blind review

ABSTRACT

A software-repair episode can pass repository tests and compilation, train a LoRA child, and reveal its harm only when a different task that passed under the parent fails under the child. Both observations can be correct: the verifier establishes source-issue success, whereas the later regression concerns the policy produced from that success. StatePermit records this cross-task transition through a versioned parent–child lineage. Every eligible verified episode first becomes a redacted, provenance-bound memory record; promotion to model state additionally requires stable training loss and bounded parent-to-child divergence; and fixed, identity-disjoint canaries can restore a passing ancestor and quarantine an implicated trace. Across five matched 300-issue SWE-bench Lite trajectories, verified-trace memory averages 21.4% resolve with 0.9 replay regressions per 100 evaluator identities. Ungated online LoRA reaches 25.4% but produces 18.3 regressions, whereas StatePermit reaches 27.5% with 4.6. Of 1,186 verifier-passed candidates, loss or divergence screening leaves 83 memory-only; monitoring operationally attributes and revokes 43 of 1,000 committed sources. Strong fixed-weight and continual-learning alternatives finish within 0.9 resolve points yet separate sharply in downstream burden. On a controller-hidden 2,000-check evaluator, StatePermit records 44 new failures, compared with 58 for a frozen cross-encoder, 110 for MIR-LoRA, and 134 for Replay LoRA. Qwen2.5-72B and constraint-based SQL comparisons preserve the same qualitative utility–burden pattern. Persistent learning should therefore distinguish evidence that licenses inspectable reuse from evidence that licenses a shared weight change, and evaluate the latter by its future-task consequences rather than source-task success alone.

1 INTRODUCTION

One logged persistent update begins with an unambiguous success: a software-repair episode passes the repository tests and compilation that judge its source issue. The episode trains a bounded LoRA child. At a later replay observation, an evaluator identity that passed under the parent fails under that child. Restoring the parent removes the failure while leaving the source verifier's verdict unchanged. Across the primary streams, monitoring attributes this sequence to 43 committed source traces. The failure is not a contradiction in source-task verification; it is a cross-task consequence that does not exist until the trained child serves later requests.

The storage mechanism determines how widely that consequence propagates. As an external memory record, the episode can influence a request only when retrieval places an inspectable, removable record in context. As a weight update, it changes the policy used for every later request, including requests that never retrieve the episode. The consequential decision is therefore whether evidence about one completed task is sufficient to move a trace from attributable memory into shared model state. A useful trace can remain available without receiving that broader authority.

StatePermit implements this distinction as a trace-specific state fork. A verifier-passed episode is first redacted, bound to its provenance and tool versions, and retained in external memory. Weight promotion separately constructs a LoRA child and tests training-loss stability and parent-to-child token divergence. Every accepted child records its parent and source trace. Scheduled canaries test the active lineage; deterioration restores the last passing ancestor, discards provisional descendants, and quarantines the implicated source from retrieval and future training. A stale governance version sus-

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.

pends promotion without deleting an otherwise eligible memory record. The resulting *persistent-state authorization* lifecycle makes retention, promotion, and continued deployment separate executable decisions.

The evaluation reconstructs this lifecycle under common candidate exposure. A frozen producer supplies the same ordered episodes to every policy, and a candidate becomes available only after its source issue has been completed and scored. Each policy therefore solves an issue from state built exclusively from earlier candidates. Fixed replay identities compare adjacent persistent-state snapshots throughout the stream; a separate 2,000-identity evaluator compares pre-stream and final states without controlling admission or restoration. Candidate dispositions, opened regression incidents, harmful dwell, endpoint utility, and complete-run cost expose different moments of the same persistent update.

Endpoint resolve alone masks distinct operating regions. Verified-trace memory averages 21.4% resolve with 0.9 replay regressions per 100 identities; ungated LoRA reaches 25.4% with 18.3, and StatePermit reaches 27.5% with 4.6. On the hidden evaluator, StatePermit records 44 new failures, versus 58 for the cross-encoder and 110 for MIR-LoRA.

We treat the persistent update—a source trace, its memory record, its candidate child, and later attributable transitions—as an empirical object for live-learning agents. The controller binds pre-write evidence to a state mutation and post-write evidence to a reversible lineage; the matched protocol compares utility, later failures, exposure before restoration, and complete-run cost. Admission, eligibility, cadence, and rollback interventions, a second 70B-class backbone, constraint-based SQL, controller-hidden evaluation, and stress tests probe this lifecycle.

Online memory and fast-weight systems already demonstrate evolving case banks, chronological coding streams, self-rewriting memory, stability-gated updates, and joint parametric/non-parametric state (Zhou et al., 2025; Joshi et al., 2025; Wei et al., 2025; Zhang et al., 2026; Ren et al., 2026; Hoy & Celik, 2025; Fallah et al., 2026; Wang et al., 2026). StatePermit connects those ingredients through provenance before promotion, memory retention when promotion is withheld, lineage-aware observation, and restoration. Source-task success can thus license inspectable reuse without licensing an irreversible policy change.

2 Related Work

Online adaptation and chronological evaluation. Test-time adaptation changes behavior under shift (Sun et al., 2020; Wang et al., 2021; 2022; Niu et al., 2023; Zhang et al., 2022), while continual learning controls forgetting through regularization and replay (Lange et al., 2022; Kirkpatrick et al., 2017; Zenke et al., 2017; Aljundi et al., 2019). SWE-Bench-CL, CLaaS, and aTTT study chronological transfer, replay-backed online LoRA, and repetition-aware updates (Joshi et al., 2025; Fallah et al., 2026; Wang et al., 2026). StatePermit controls and evaluates one persistent update through its later parent-to-child regressions.

Evolving external memory. Retrieval and memory agents carry experience through context (Guu et al., 2020; Lewis et al., 2020; Yao et al., 2023; Shinn et al., 2023); AgentFly, Evo-Memory, JitRL, and Live-Evo study case selection, streaming evolution, retrieved-trajectory optimization, and memory revision (Zhou et al., 2025; Wei et al., 2025; Li et al., 2026; Zhang et al., 2026). StatePermit retains an attributable memory fallback for the same source episode whose parameter branch may be withheld or restored, so failures actuate record quarantine or lineage restoration.

Fast weights and state control. Adapters and low-rank updates provide economical parameter changes (Houlsby et al., 2019; Hu et al., 2022; Dettmers et al., 2023). SEAL, STABLE, and TMEM respectively study persistent self-edits, stability-bounded LoRA edits, and joint explicit/parametric state (Zweiger et al., 2025; Hoy & Celik, 2025; Ren et al., 2026). StatePermit combines executable provenance, loss and divergence admission, memory-only retention, monitored dwell, and parent restoration into one source-trace lifecycle under common candidate exposure.

3 STATE-WRITE AUTHORIZATION

3.1 AUTHORIZATION STATE AND DECISIONS

An episode E contains prompts, actions, tool calls, tool outputs, verifier outcomes, and task success. The persistent state at event t is $S_t = (\theta_t, \mathcal{M}_t, G_t)$: the active adapter θ_t, retrieval memory $\mathcal{M}_t$, and versioned governance state G_t. A policy is evaluated by

$$A(\pi) = (u_{\text{local}}, b_{\text{admit}}, b_{\text{dwell}}, c_{\text{ops}}),$$

where u_{local} is benchmark-native utility, b_{admit} counts candidate dispositions at admission, b_{dwell} measures newly introduced failures and their exposure after persistence, and c_{ops} is complete-run cost. The tuple does not force memory and weights to share a denominator. It requires each burden to name its candidate unit, observation time, and state-changing response.

At event t, the controller exposes one of four authority decisions:

$$D_t(E, S_t) \in \{\text{commit}, \text{retain}, \text{revoke}, \text{suspend}\},$$

with no decision emitted when admission fails before either store. These decisions enact object-typed transitions: commit creates an adapter child, retain inserts or preserves a memory record, revoke restores an adapter parent or quarantines an attributed record, and suspend disables a write path. Eviction removes quarantined or out-of-index memory without granting new authority. A weight-governance mismatch can therefore suspend promotion while retaining an episode under the independently versioned memory policy.

Only verifier-passed episodes enter either persistent store. Timeouts, indeterminate outcomes, and verifier failures remain in transient run logs and cannot be retrieved or trained on. Memory admission additionally requires allowed provenance, successful redaction, a current memory schema, and a unique content hash. Weight promotion applies the stricter conjunction of current tool and verifier versions, finite and non-increasing warmup loss, and candidate-to-parent mean token KL below $\epsilon = 0.12$. Tool versions, schemas, verifier logic, base-model identity, and redaction policy are independently versioned so that a stale weight dependency can suspend commits while verified, redacted memory remains usable.

3.2 BOUNDED COMMITS AND DWELL MONITORING

Within the last 16 transformer blocks, the candidate update applies LoRA to `q_proj` and `v_proj` using rank 16, $\alpha = 32$, and dropout 0.05; all other parameters stay frozen. AdamW uses learning rate 5×10^{-5}, cosine decay, 10 warmup steps, gradient clipping at 1.0, bf16, and batch size 4. Each accepted trace is traversed once in 1,024–2,048-token chunks cut only between complete tool-call records; the one-pass schedule is clipped to 10–50 optimizer steps. The loss combines cross-entropy over schema, action, and planning tokens with a KL penalty to the deployed parent. The admission KL is recomputed after training on the same non-padding assistant-token positions under parent and child, then averaged per token. Loss stability rejects non-finite runs or a final five-step median above the warmup median.

Every $M = 5$ committed updates, a frozen 100-task canary suite evaluates the active lineage. The ring buffer of $N = 8$ passing checkpoints distinguishes a passing ancestor from provisional children. If the score falls below 90/100, the controller restores the last checkpoint that passed the full suite and discards all provisional descendants. It then replays the five-source batch against that ancestor and operationally attributes the incident to the earliest child whose removal restores the passing outcome; implicated source traces are quarantined from both memory retrieval and later training. The reported 43/1,000 revocations count these attributed source commits, whereas discarded descendants are lineage cleanup rather than additional incidents. Harmful dwell starts at the attributed commit and ends at restoration; an unrestored incident is right-censored at the earlier of the 200-commit cap and the end of the 300-issue stream. The same identity-disjoint canary panel is reused for at most 40 scheduled looks per seed. Only its aggregate score reaches the controller; its task contents remain hidden from the adapting policy.

Algorithm 1 StatePermit authorization and monitored dwell

Require: episode E, deployed adapter θ, governance state G, memory $\mathcal{M}$
if task verifier fails, times out, or is indeterminate **then**
  make no persistent write; return θ
end if
redact E and bind its provenance, versions, and content hash
if memory provenance, redaction, or schema eligibility fails **then**
  make no persistent write; return θ
end if
retain E in $\mathcal{M}$
if weight-governance state is stale **then**
  suspend promotion; return θ
end if
construct bounded candidate θ' from E
if warmup-loss or KL eligibility fails **then**
  keep E memory-only; return θ
end if
commit θ'; record its parent, source trace, and lineage
if scheduled canaries fall below the rollback threshold **then**
  restore the last passing ancestor
  quarantine implicated source traces from retrieval and training
end if
if on-demand attribution replay assigns a new failure to E **then**
  quarantine E and evict it from the active index
end if
return current deployed adapter

3.3 Persistent-memory actions

The memory record stores the redacted trace, executable outcome, tool and schema versions, repository or database identity, and a content hash. Active retrieval filters on compatible versions and provenance, removes quarantined hashes and near-duplicates, retrieves under the shared token cap, and logs the records used. After a retrieval-conditioned downstream failure, a distinct on-demand *attribution replay* reruns that query while removing each retrieved record in turn from the same state. A record whose removal restores the prior passing outcome is quarantined; duplicate hashes and records outside the bounded active index are evicted. This attribution replay is separate from the fixed measurement replay panel.

4 Experimental Design

4.1 Workloads, models, and compute

The primary benchmark is the 300-issue SWE-bench Lite v1.1 release in agentic mode (Jimenez et al., 2024). Repository tests and compilation determine resolve; static analysis is an auxiliary admission screen and cannot turn a failing task into a pass. MultiSQL-Personalize uses a 200-event adaptation stream per seed and a separate 180-task endpoint evaluator generated from declarative schemas, integrity constraints, and multi-session preference templates. An endpoint task is satisfied only when all declared hard constraints hold. Adaptation time is the first event whose gain of at least 2.0 points over the pre-update running mean persists for five evaluations. Schema and template families, rather than individual utterances, are partitioned between online, calibration, canary, replay, and external pools.

The full study uses Llama-3.1-70B-Instruct (Grattafiori et al., 2024); a second-backbone comparison repeats the matched stream with Qwen2.5-72B-Instruct (Yang et al., 2024). Each run uses one 4$\times$ NVIDIA A100 80GB node, Ubuntu 22.04, CUDA 12.4, PyTorch 2.3.1, and bf16 adapter training. Five global RNG seeds $\{13, 37, 61, 71, 97\}$ determine trace generation, retrieval ties, and optimizer order. Both backbones retain their published chat templates; greedy decoding, tool schemas, per-issue

prompt and tool ceilings, issue order, retrieval corpus, and verifier versions are identical across policies. The reported Llama primary sweep consumes 320 A100 GPU-hours, including all online decisions and scheduled monitors.

4.2 MATCHED ONLINE TIMELINE

A matched unit is a seed–issue pair and its logged candidate episode. Before online policy evaluation, a frozen offline producer constructs one exogenous candidate stream per seed and records its complete prompts, tool transactions, outputs, and verifier outcomes for the 300 issues in their common order. The producer is an exposure-construction pipeline, not one of the evaluated online policies. Policy π first solves issue i from state S_{i-1}^{π}; only that policy-generated solution contributes to resolve. After issue i is completed and scored, producer episode E_i becomes available for retention or promotion and may affect only $i+1$ and later issues. No current-issue or future-issue candidate is exposed before evaluation. Across five seeds, 1,186 of the producer's 1,500 candidates pass verification; this measures candidate availability in the exogenous stream, whereas the 21.4–27.5% resolve rates measure policy-generated solutions. Policy actions affect their own endpoint and burden measurements but never another row's candidate input.

All rows process all 300 issue events. Weight-changing rows may commit at most 200 children per seed, then continue inference and measurement without further writes. The fixed replay evaluator contains 300 held-out check identities per seed, for 1,500 pooled identities. Each identity is observed at prespecified state transitions on a schedule frozen before comparison and identical across policies; executions are not the denominator. The external evaluator compares pre-stream and final states once for 400 disjoint identities per seed, totaling 2,000. Neither fixed measurement suite actuates state. The 100-task canary panel recurs every five commits and alone can restore an adapter; on-demand attribution replay can quarantine memory. Online, calibration, canary, replay, and external pools are separated by issue/schema identity, revision, and hash.

4.3 MATCHED POLICIES

Every row shares an $8,192$-token retrieval cap and at most 4 records per request. Memory-only ranks complete verified episodes by frozen embedding similarity; Prompt-only places recent verified summaries in the same cap. Naive TTT applies the LoRA recipe after every completed episode. Per-session LoRA admits verifier-passed episodes but resets after each SWE issue/tool dialogue. Eligibility-only applies the full verifier, provenance, redaction, loss, and KL conjunction without a dwell monitor.

The verifier-conditioned router adds verifier type, repository/schema identity, and version compatibility to the embedding score. The frozen cross-encoder is a DeBERTa-v3-base pairwise reranker fitted on the pre-stream calibration partition; it scores the query jointly with each trace summary and verifier fields. Structured memory replaces free text with an execution graph whose nodes are actions and validated constraints. Replay LoRA samples uniformly from all currently eligible verified traces. MIR-LoRA uses the same buffer and selects records with the largest predicted loss increase on the current candidate (Aljundi et al., 2019). EWC-LoRA uses a diagonal Fisher estimate over that buffer (Kirkpatrick et al., 2017). The cross-encoder fit, EWC coefficient, LoRA rank, KL threshold, canary cadence, and restoration threshold are frozen after calibration and before the five reported online streams. Canaries measure every row, but only StatePermit and the monitored ablations use them to change state. Complete-run overhead includes generation, tools, retrieval, verification, training, monitoring, and restoration.

4.4 METRICS AND INFERENCE

Resolve and SQL satisfaction are task-level endpoints. At each scheduled replay observation, a check's outcomes under the immediately preceding and resulting persistent-state snapshots are compared. A pass-to-fail transition opens one regression incident; subsequent scheduled failures for that identity remain in the same incident until a passing snapshot closes it. Replay reg./100 is 100 times opened incidents divided by the 1,500 fixed identities; executions, snapshots, and commits are not denominator units. External burden is the pre-stream-to-endpoint pass-to-fail rate on 2,000 unseen identities. A harmful commit opens at least one replay incident; a monitor miss is not flagged by

Table 1: Observable transitions for each persistent object. Evidence is tied to the object and time at which it can change state.

Protected state	Admission evidence	Dwell evidence	Authorized action
Live LoRA adapter	Verifier, provenance, redaction, loss, KL	Scheduled canary score	Commit or restore parent
Retrieval memory	Verified trace and bounded context policy	On-demand attribution replay	Store, quarantine, or evict
Governance state	Version and integrity match	Tool/schema drift observations	Enable or suspend each write path

Table 2: Primary Llama-3.1-70B result (300 SWE-bench Lite issues, five seeds, 200-commit cap). Overhead covers the complete run.

Method	Resolve (%)	Replay reg./100	Overhead
Memory-only	21.4 ± 1.1	0.9 ± 0.4	—
Prompt-only	22.8 ± 1.0	1.1 ± 0.5	+1.2%
Naive TTT	25.4 ± 1.6	18.3 ± 3.4	+19%
Per-session LoRA	24.6 ± 1.2	7.8 ± 2.1	+14%
Eligibility-only	26.3 ± 1.2	6.1 ± 1.7	+16%
StatePermit	27.5 ± 1.3	4.6 ± 1.4	+18%

the next canary. Harmful dwell counts update intervals through restoration, censoring an unrestored incident at the 200-commit cap or stream end. Memory insertion can also open a regression by changing context.

Tables report mean $\pm$ standard deviation across the five complete seeded trajectories; pooled numerators accompany the 1,500 replay and 2,000 external identity denominators. Outcomes within a seed share a chronological persistent-state history, so issues are not treated as exchangeable resampling units. With five complete trajectories, all cross-policy comparisons are descriptive: endpoint differences are differences of trajectory means, and burden contrasts are also reported as pooled incident-count differences on the common identity panels. The primary endpoint is resolve over the full 300-issue stream under the 200-commit cap.

4.5 Connecting measurements to state changes

Table 1 maps each observation to the state it can change. Adapter admission requires provenance and executable verification; only scheduled canaries can restore an active lineage. Memory uses the same provenance boundary but attributes downstream failures through retrieval identifiers and on-demand leave-one-record-out replay. Fixed replay and external suites measure burden without actuating state. Governance objects control which write branch remains enabled.

Admission burden is counted over candidate episodes; replay and external regressions over held-out transitions after persistent state affects generation; harmful dwell over update intervals; and cost over complete runs. "Not applicable" is used when a policy has no weight-write boundary, while its retrieval-conditioned burden remains measured. This preserves the meaning of each denominator when comparing memory insertion with parameter mutation.

5 Results

5.1 Source verification alone does not eliminate downstream burden

Across the five matched trajectories, StatePermit averages 6.1 points higher resolve than Memory-only. Relative to naive TTT, it records approximately 75% fewer replay regressions at slightly lower complete-run cost. The comparison characterizes the consequence of controlling the weight-write

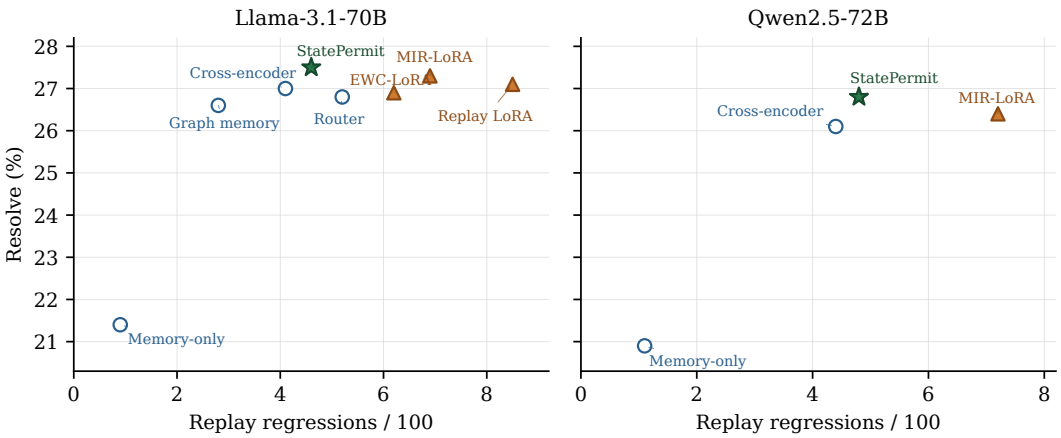


Figure 1: Descriptive utility–burden operating points on two backbones. Circles use fixed weights, triangles are continual learners, and stars are StatePermit. Each point is the mean 300-issue endpoint; weight-changing methods have a 200-commit cap. Upper-left is preferable.

Table 3: Evaluated comparator set. Replay and external columns are regressions per 100 identities. Panel A uses Llama-3.1-70B; Panel B repeats the matched comparison with Qwen2.5-72B.

Method	Resolve	Replay	External	Overhead
Panel A: Llama-3.1-70B-Instruct				
Memory-only	21.4 ± 1.1	0.9 ± 0.4	1.0 ± 0.4	—
Verifier-conditioned router	26.8 ± 1.1	5.2 ± 1.3	3.3 ± 0.8	+11%
Frozen cross-encoder	27.0 ± 1.2	4.1 ± 1.0	2.9 ± 0.7	+15%
Structured execution memory	26.6 ± 1.4	2.8 ± 0.9	2.4 ± 0.6	+13%
Replay LoRA	27.1 ± 1.2	8.5 ± 2.2	6.7 ± 1.4	+23%
MIR-LoRA	27.3 ± 1.3	6.9 ± 1.8	5.5 ± 1.2	+25%
EWC-LoRA	26.9 ± 1.2	6.2 ± 1.6	4.8 ± 1.0	+24%
StatePermit	27.5 ± 1.3	4.6 ± 1.4	2.2 ± 0.6	+18%
Panel B: Qwen2.5-72B-Instruct				
Memory-only	20.9 ± 1.0	1.1 ± 0.4	0.9 ± 0.3	—
Frozen cross-encoder	26.1 ± 1.1	4.4 ± 1.1	3.1 ± 0.8	+16%
MIR-LoRA	26.4 ± 1.3	7.2 ± 1.9	5.8 ± 1.3	+26%
StatePermit	26.8 ± 1.2	4.8 ± 1.3	2.5 ± 0.7	+19%

path: updating on every completed episode reaches 25.4% resolve but creates four times StatePermit's replay burden, whereas full eligibility without dwell control reaches 26.3% with 6.1 regressions.

5.2 Strong memory and continual learners occupy different operating regions

On Llama, the cross-encoder, structured memory, MIR, and EWC are within -0.5, -0.9, -0.2, and -0.6 resolve points of StatePermit, but their burdens separate. Relative to StatePermit, they respectively have 7 fewer/14 more, 27 fewer/4 more, 35 more/66 more, and 24 more/52 more replay/external failures. Appendix Table A10 gives every count beside its endpoint difference.

Qwen preserves the ordering: StatePermit reaches 26.8% resolve versus 26.1% for the cross-encoder and 26.4% for MIR, with the lowest external burden among these high-utility rows. Figure 1 contrasts lower-burden structured memory, cross-encoder retrieval, and authorized updates with lower burden than the continual-learning controls.

Table 4: Component interventions on SWE-bench Lite. "Eligibility" is verifier+provenance+redaction+loss+KL; "restore" actuates canary-triggered parent restoration.

Gate configuration	Resolve (%)	Replay reg./100
Full eligibility + monitor + restore	27.5 ± 1.3	4.6 ± 1.4
Task-verifier admission only	26.8 ± 1.4	7.1 ± 2.0
Monitor + restore only	25.9 ± 1.5	8.4 ± 2.3
Full eligibility only	26.3 ± 1.2	6.1 ± 1.7
Ungated online LoRA	25.4 ± 1.6	18.3 ± 3.4

Table 5: Post-write validation on a disjoint 2,000-check suite. The suite is never used for admission, canary selection, rollback, or threshold tuning. Persistent memory methods remain included because memory can change later outputs.

Method	External reg./100	New failures / 2,000	Held-out pass (%)
Memory-only	1.0 ± 0.4	20/2000	64.9 ± 2.0
Verifier-conditioned router	3.3 ± 0.8	66/2000	67.2 ± 1.8
Frozen cross-encoder	2.9 ± 0.7	58/2000	67.7 ± 1.8
Structured execution memory	2.4 ± 0.6	48/2000	67.4 ± 1.7
Replay LoRA	6.7 ± 1.4	134/2000	66.6 ± 1.9
MIR-LoRA	5.5 ± 1.2	110/2000	67.0 ± 1.7
EWC-LoRA	4.8 ± 1.0	96/2000	66.8 ± 1.8
StatePermit	2.2 ± 0.6	44/2000	68.1 ± 1.7

5.3 Admission and revocation act on different events

Task-verifier admission produces the largest aggregate burden reduction in the intervention ladder, from 18.3 to 7.1 replay regressions per 100 identities. Applying all pre-write predicates lowers the rate further to 6.1. Adding canary monitoring and restoration to that same eligibility policy reaches 4.6 and improves resolve by 1.2 points, whereas monitoring and restoration without eligibility leave 8.4 regressions. Admission controls which child exists; revocation controls how long an attributed harmful child remains active. Appendix Tables A2 and A3 vary KL, cadence, and restoration thresholds; Table A4 summarizes the default disposition flow.

5.4 An evaluator unseen by the controller preserves the burden contrast

The external evaluator preserves the broad burden ordering without sharing controller triggers. StatePermit records 44 new failures, compared with 58 for the cross-encoder, 110 for MIR-LoRA, and 134 for Replay LoRA, while attaining the highest mean held-out pass rate. Replay and external burden correlate strongly but imperfectly (Appendix Figure A1), so neither can substitute for the other. Memory-only supplies the minimum-burden endpoint; StatePermit supplies the highest resolve mean near a two-failure-per-100 external operating point.

5.5 Storage and control mechanisms yield distinct operating points

Under the common context ceiling, a frozen cross-encoder raises resolve from 26.8% to 27.0% and lowers both burdens relative to the heuristic router; execution-graph memory lowers replay burden further at 26.6% resolve. MIR reduces Replay LoRA's replay/external burden from 8.5/6.7 to 6.9/5.5 while reaching 27.3% resolve, whereas EWC reaches 26.9% with 6.2/4.8. These rows share candidate exposure, LoRA targets, schedules, and write cap; only StatePermit and monitored ablations actuate canary restoration.

The disjoint and cross-backbone checks test distinct consequences. On Qwen, StatePermit gains 5.9 resolve points over Memory-only and uses less than half MIR-LoRA's excess external burden above the memory floor. On the disjoint Llama suite, it has 24%, 60%, and 67% fewer external failures than the cross-encoder, MIR, and Replay LoRA. Appendix Tables A10 and A11 give descriptive endpoint differences and counts.

5.6 The effect spans issue strata and constraint-based sessions

The aggregate gain appears across repository families, verifier-agreement strata, patch sizes, initial failure counts, and both halves of the stream (Appendix Tables A12–A13). The hardest large-patch and mixed-verifier slices favor fixed-weight alternatives in point estimates, while smaller patches and unanimous verifiers favor StatePermit; the win/tie/loss profiles identify heterogeneity rather than universal subgroup wins. MultiSQL-Personalize then changes both task semantics and verifier type. StatePermit reaches 86.3$\pm$1.5% constraint satisfaction versus 77.8$\pm$1.8% for Memory-only and 84.9 $\pm$ 1.6 for the structured ledger (Appendix Table A14), with 2.1$\pm$0.6 external regressions per 100. The code and SQL streams consequently agree on the value of separating persistence decisions, even though their best fixed-weight representations differ.

5.7 The complete accounting changes the policy choice

StatePermit, MIR-LoRA, the cross-encoder, and EWC-LoRA finish within 0.6 resolve points but have different burdens. Above the memory floor, MIR adds 6.0 replay and 4.5 external regressions per 100; StatePermit adds 3.7 and 1.2. Structured memory has the smallest fixed-weight increments at a 0.9-point resolve tradeoff.

Of 1,500 candidates, 1,186 pass verification; provenance and redaction reject 103, loss and KL retain 83 in memory, and 1,000 become commits. Monitoring attributes and revokes 43 sources with median [IQR] dwell 5 [5,10] intervals. By 100 commits, StatePermit gains 4.9 resolve points with 4.2 replay burden; the final 100 add 1.2 and 0.4 (Appendix Table A5). A tighter KL gate lowers admission and burden, while two- and ten-commit monitoring give median dwell of two and ten intervals.

The +18% overhead covers the complete run; accepted updates add 11.4% at p50 and 19.7% at p95 (Appendix Table A7), compared with +19% for unscreened Naive TTT.

6 Discussion

Source-task verification and persistent-update evidence are complementary rather than competing signals. Verification removes the largest aggregate share of burden before training, while the 6.1-to-4.6 regression reduction from restoration shows that screened children can still reveal consequential behavior only after deployment. The 43 attributed revocations instantiate this temporal gap: a valid source episode and a harmful parent-to-child transition can coexist.

The frontier maps directly to system choices. Structured memory offers the lowest replay burden among high-utility rows; the cross-encoder trades lower replay for higher external burden than StatePermit; and MIR reaches endpoint-close utility with greater burden. Cadence interventions sharpen the mechanism: moving from ten-commit to two-commit monitoring shortens median harmful dwell from ten to two intervals while increasing whole-run overhead from 16% to 21%. Persistent-state evaluation should therefore report where utility is stored, which later failures it introduces, and how quickly the responsible state can be removed.

The evidence characterizes policy-level regions for an exogenous stream with executable verifiers, bounded adapters, and restorable parents, not potential outcomes for each rejected trace. The fixed canary provides adaptive feedback despite hidden task contents. Within the tested code and SQL regimes, pre-write screening and post-write restoration reduce different observed burdens while preserving competitive endpoint utility.

7 Conclusion

A passing patch can train a child that breaks a later task. Across five matched SWE-bench Lite trajectories, StatePermit adds 6.1 resolve points over verified-trace memory while recording approximately three quarters fewer replay regressions than ungated online LoRA. Similar utility but different future-task failures persist on a second backbone, a controller-hidden evaluator, and constraint-based SQL. By separating verified memory from reversible promotion, StatePermit keeps useful experience inspectable without making every successful episode an unconditional change to the policy serving future tasks.

Reproducibility Statement

Sections 4 and 5 specify the workloads, models, shared comparison controls, estimands, and trajectory-level aggregation. The appendices document the controller configuration, comparator policy definitions, raw counts, sensitivity analyses, and result-to-configuration map supporting the reported policy-level comparisons.

AI Use Statement

AI tools were used to assist manuscript preparation, including language editing, formatting, and consistency checks.

References

Rahaf Aljundi, Lucas Caccia, Massimo Caccia, Eugene Belilovsky, Min Lin, Laurent Charlin, and Tinne Tuytelaars. Online continual learning with maximally interfered retrieval. In *Advances in Neural Information Processing Systems*, volume 32, pp. 11849–11860, 2019. URL https://papers.nips.cc/paper/2019/hash/15825aee15eb335cc13f9b559f166ee8-Abstract.html.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Srndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Machine Learning and Knowledge Discovery in Databases*, volume 8190 of *Lecture Notes in Computer Science*, pp. 387–402, 2013. doi: 10.1007/978-3-642-40994-3_25.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium*, pp. 2633–2650, 2021. URL https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, pp. 10088–10115, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/1feb87871436031bdc0f2beaa62a049b-Abstract-Conference.html.

Kion Fallah, Silen Naihin, Barak Widawsky, and Qingqing Mao. CLaaS: Continual learning as a service for sample-efficient online learning. *arXiv preprint arXiv:2606.05559*, 2026. URL https://arxiv.org/abs/2606.05559.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. REALM: Retrieval-augmented language model pre-training. In *International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3929–3938, 2020. URL https://proceedings.mlr.press/v119/guu20a.html.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799, 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

William Hoy and Nurcin Celik. STABLE: Gated continual learning for large language models. *arXiv preprint arXiv:2510.16089*, 2025. doi: 10.48550/arXiv.2510.16089. URL https://arxiv.org/abs/2510.16089.

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. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R. Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Thomas Joshi, Shayan Chowdhury, and Fatih Uysal. SWE-Bench-CL: Continual learning for coding agents. *arXiv preprint arXiv:2507.00014*, 2025. URL https://arxiv.org/abs/2507.00014.

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):3521–3526, 2017. doi: 10.1073/pnas.1611835114.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2022. doi: 10.1109/TPAMI.2021.3057446.

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, pp. 9459–9474, 2020. URL https://papers.nips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Yibo Li, Zijie Lin, Ailin Deng, Xuan Zhang, Yufei He, Shuo Ji, Tri Cao, and Bryan Hooi. Just-in-time reinforcement learning: Continual learning in LLM agents without gradient updates. In *International Conference on Learning Representations*, 2026. doi: 10.48550/arXiv.2601.18510. URL https://openreview.net/forum?id=us2YPNouOm.

Shaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofe Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2302.12400.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. Mind the style of text! adversarial and backdoor attacks based on text style transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4569–4580, 2021. doi: 10.18653/v1/2021.emnlp-main.374. URL https://aclanthology.org/2021.emnlp-main.374/.

Tao Ren, Weiyao Luo, Hui Yang, Rongzhi Zhu, Xiang Huang, Yuchuan Wu, Bingxue Chou, Jieping Ye, Jiafeng Liang, Yongbin Li, and Yijie Peng. Scaling self-evolving agents via parametric memory. *arXiv preprint arXiv:2606.04536*, 2026. URL https://arxiv.org/abs/2606.04536.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9229–9248, 2020. URL https://proceedings.mlr.press/v119/sun20b.html.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uXl3bZLkr3c.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7201–7211, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Wang_Continual_Test-Time_Domain_Adaptation_CVPR_2022_paper.html.

Yanbo Wang, Jinhua Hao, Yuze Shi, Kun Yuan, and Ming Sun. No time like the present: Agentic test-time training for LLM agents. *arXiv preprint arXiv:2607.03441*, 2026. doi: 10.48550/arXiv.2607.03441. URL https://arxiv.org/abs/2607.03441.

Tianxin Wei, Noveen Sachdeva, Benjamin Coleman, Zhankui He, Yuanchen Bei, Xuying Ning, Mengting Ai, Yunzhe Li, Jingrui He, Ed Huai hsin Chi, Chi Wang, Shuo Chen, Fernando Pereira, Wang-Cheng Kang, and Derek Zhiyuan Cheng. Evo-memory: Benchmarking LLM agent test-time learning with self-evolving memory. *arXiv preprint arXiv:2511.20857*, 2025. URL https://arxiv.org/abs/2511.20857.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3987–3995, 2017. URL https://proceedings.mlr.press/v70/zenke17a.html.

Marvin Zhang, Sergey Levine, and Chelsea Finn. MEMO: Test time robustness via adaptation and augmentation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 38629–38642, 2022. URL https://papers.nips.cc/paper_files/paper/2022/hash/fc28053a08f59fccb48b11f2e31e81c7-Abstract-Conference.html.

Yaolun Zhang, Yiran Wu, Yijiong Yu, Qingyun Wu, and Huazheng Wang. Live-evo: Online evolution of agentic memory from continuous feedback. *arXiv preprint arXiv:2602.02369*, 2026. URL https://arxiv.org/abs/2602.02369.

Huichi Zhou, Yihang Chen, Siyuan Guo, Xue Yan, Kin Hei Lee, Zihan Wang, Ka Yiu Lee, Guchun Zhang, Kun Shao, Linyi Yang, and Jun Wang. Agentfly: Fine-tuning LLM agents without fine-tuning LLMs. *arXiv preprint arXiv:2508.16153*, 2025. URL https://arxiv.org/abs/2508.16153.

Adam Zweiger, Jyo Pari, Han Guo, Yoon Kim, and Pulkit Agrawal. Self-adapting language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2483. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/6b41e04c41726e2a60e456d0a2b961ab-Abstract-Conference.html.

A Controller, Verifiers, and Operating Points

A.1 Verifier provenance and uncertainty

The verifier audit contains 500 labeled traces, 100 from each family, stratified over pass, fail, timeout, and indeterminate outcomes. Labels follow underlying task truth rather than verifier output.

Executable unit, compile, and database checks supply primary admission evidence; static analysis and NLI are auxiliary conjunctive screens and never authorize alone.

Table A1: Verifier audit. FP and FN give count/decision-denominator with 95% Wilson intervals. Timeout and indeterminate annotations are included in the negative-decision denominator and are always ineligible.

Family (role)	FP count [CI]	FN count [CI]	Rate bin	Timeout/indet.
Unit tests (primary)	1/52 [0.3,10.1]%	2/48 [1.2,14.0]%	0–2/2–5%	3/100
Compilation (primary)	0/51 [0.0,7.0]%	0/49 [0.0,7.3]%	0–1/0–1%	1/100
Static analysis (aux.)	3/54 [1.9,15.1]%	4/46 [3.4,20.3]%	4–6/7–9%	0/100
DB constraints (primary)	0/52 [0.0,6.9]%	1/48 [0.4,10.9]%	0–1/1–3%	2/100
NLI consistency (aux.)	3/46 [2.2,17.5]%	5/54 [4.0,19.9]%	5–7/9–11%	0/100

The FP denominator is the set of positive verifier decisions and the FN denominator is the complementary negative-decision set; the two therefore sum to 100 within each family. Timeout and indeterminate events are reported again in the last column to expose their operational frequency, not as additional traces. The primary executable checks have the smallest observed error counts, motivating their required role, while the wider auxiliary intervals motivate conjunction rather than independent authorization.

A.2 Threshold sensitivity and gate flow

Table A2: Utility and burden under one-factor changes to the default ($\epsilon = .12, M = 5, R = 90$).

Setting	Resolve	Replay	External
Default	27.5 ± 1.3	4.6 ± 1.4	2.2 ± 0.6
KL $\epsilon = .08$	26.9 ± 1.2	3.7 ± 1.1	1.8 ± 0.5
Cadence $M = 2$	27.2 ± 1.2	3.9 ± 1.2	1.9 ± 0.5
Cadence $M = 10$	27.7 ± 1.4	5.7 ± 1.6	2.9 ± 0.8
Restore $R = 88$	27.6 ± 1.3	5.3 ± 1.5	2.6 ± 0.7
Restore $R = 92$	27.1 ± 1.2	4.0 ± 1.2	1.9 ± 0.6

Table A3: Write flow, exposure, and cost for the same sweep. Counts pool five seeds; dwell is median [IQR] update intervals.

Setting	Weight-eligible/verifier-pass	Revoked/commits	Harmful dwell	Overhead
Default	1000/1186	43/1000	5 [5,10]	+18%
KL $\epsilon = .08$	921/1186	34/921	5 [5,5]	+17%
Cadence $M = 2$	1000/1186	47/1000	2 [2,4]	+21%
Cadence $M = 10$	1000/1186	39/1000	10 [10,20]	+16%
Restore $R = 88$	1000/1186	35/1000	5 [5,10]	+17.8%
Restore $R = 92$	1000/1186	51/1000	5 [5,10]	+18.7%

Tightening ϵ lowers both admission and burden, whereas changing M leaves admission fixed and moves dwell and cost. Changing R alters attributed revocations without changing the candidate set. These orthogonal responses are the empirical signature of separate admission and dwell stages.

Table A4: Default gate flow pooled across five SWE seeds. Rejection categories are mutually exclusive at their ordered decision points.

Stage	Count	Stage	Count
Episodes	1500	Verifier passes	1186
Provenance rejects	46	Redaction rejects	57
Loss-sanity rejects	34	KL rejects	49
Committed updates	1000	Revoked updates	43

The nearby operating points form a broad utility–burden region. Tightening KL or restoration lowers replay and external burden with a modest endpoint tradeoff, while extending monitor cadence increases burden and sometimes utility. The nonmonotonic resolve changes and differing overhead profiles show that the lifecycle exposes a configurable operating region rather than a single threshold-specific optimum.

A.3 Update, capacity, latency, and drift diagnostics

Table A5: StatePermit by committed-update count. Gains concentrate in the first 100 updates and then taper.

Updates	Resolve (%)	Replay reg./100	p95 overhead
0 (memory)	21.4 ± 1.1	0.9 ± 0.4	—
50	24.9 ± 1.2	3.8 ± 1.3	+14.6%
100	26.3 ± 1.3	4.2 ± 1.4	+17.1%
200	27.5 ± 1.3	4.6 ± 1.4	+18.3%

Most utility arrives before the commit cap: the first 100 commits add 4.9 resolve points over the initial memory state, while the next 100 add 1.2. Replay burden grows by only 0.4 over that latter interval, supporting a bounded horizon without implying that later commits are risk-free.

Table A6: LoRA-rank sensitivity and per-update operating cost. Higher rank adds little resolve while increasing regressions and latency.

Rank	Resolve (%)	Replay reg./100	p95 overhead
8	26.2 ± 1.2	3.9 ± 1.2	+14.1%
16	27.5 ± 1.3	4.6 ± 1.4	+18.3%
32	27.8 ± 1.4	6.7 ± 1.9	+27.4%

Rank 16 is the calibrated operating point: rank 32 adds 0.3 resolve points while increasing replay burden by 2.1 and p95 overhead by 9.1 points, whereas rank 8 provides a lower-cost option with 1.3 fewer resolve points.

Table A7: Per-accepted-update latency by token chunk and optimization steps. These percentiles differ from whole-run overhead in the main tables.

Chunk / steps	p50	p95	p99
1024 / 20	+7.8%	+14.3%	+21.6%
2048 / 30	+11.4%	+19.7%	+28.3%
2048 / 50	+16.2%	+24.1%	+34.7%

The latency percentiles condition on an accepted update and therefore answer a different question from complete-run overhead. The 2,048/30 setting used for the default exposes a 28.3% p99 tail even though candidates that do not train reduce the amortized full-run cost to +18%.

Table A8: Tool-drift diagnostic. Stronger drift lowers utility and increases rollback and tail latency, motivating the governance-stale memory fallback.

Drift level	Resolve (%)	Rollback (%)	p95 overhead
Low (minor bumps)	27.1 ± 1.3	2.8 ± 1.1	+19.2%
Medium (API changes)	25.8 ± 1.5	4.9 ± 1.6	+22.4%
High (schema swap)	24.1 ± 1.8	7.3 ± 2.1	+26.8%

Tool drift tests the governance branch rather than adaptation quality alone. As version changes become more severe, more candidate lineages are restored and p95 cost rises; observed version mismatch invokes suspension before another weight candidate is constructed.

A.4 Comparator protocol

Here, matched denotes identical candidate episodes and order, evaluator identities and observation schedules, retrieval caps, calibration procedure, decoding controls, and complete-run cost boundaries. Weight-changing rows additionally share the base LoRA targets and write cap. Internal state construction, sample-selection rules, replay composition, and regularization objectives remain method-defining differences, summarized in Table A9. The rows estimate complete policy-level operating points under common exposure. They target the mechanism families emphasized by STABLE, TMEM, Live-Evo, and CLaaS—pre-write stability, joint explicit/parametric state, memory revision, and replay-backed updates—while preserving the common controller and evaluation contract needed for the matched comparison.

Table A9: State construction in the strongest comparators. The eligible pool contains all observed, currently eligible traces; quarantined traces are excluded.

Policy	State construction or selection	Shared control
Memory-only	Frozen embedding rank over complete traces	Top 4, 8,192-token cap
Frozen cross-encoder	DeBERTa-v3 pairwise reranker fitted on calibration pairs	Same eligible pool and context cap
Structured memory	Deterministic action–constraint execution graph	Same source traces and context cap
Replay LoRA	Uniform selection from the eligible pool	Common LoRA recipe and write cap
MIR-LoRA	Largest predicted loss increase in that pool	Same buffer and LoRA recipe
EWC-LoRA	Diagonal Fisher over that pool; calibrated coefficient	Same LoRA targets and norm budget
StatePermit	Current eligible trace with monitored lineage	Same LoRA recipe and write cap

B Comparator Counts and Distributional Evidence

Replay and external burden share an ordering but disagree enough to change method comparisons. Figure A1 shows this relation at the method-by-seed level before the pooled counts and descriptive endpoint differences.

Memory-only has the largest descriptive endpoint difference from StatePermit. The burden counts provide the complementary evidence used in the main-text operating-point comparison; they are pooled only after per-seed rates are computed.

Qwen reproduces the large gain over memory-only and the endpoint overlap with the nearest fixed-weight and continual-learning alternatives. Its pooled burden counts follow the Llama pattern: the

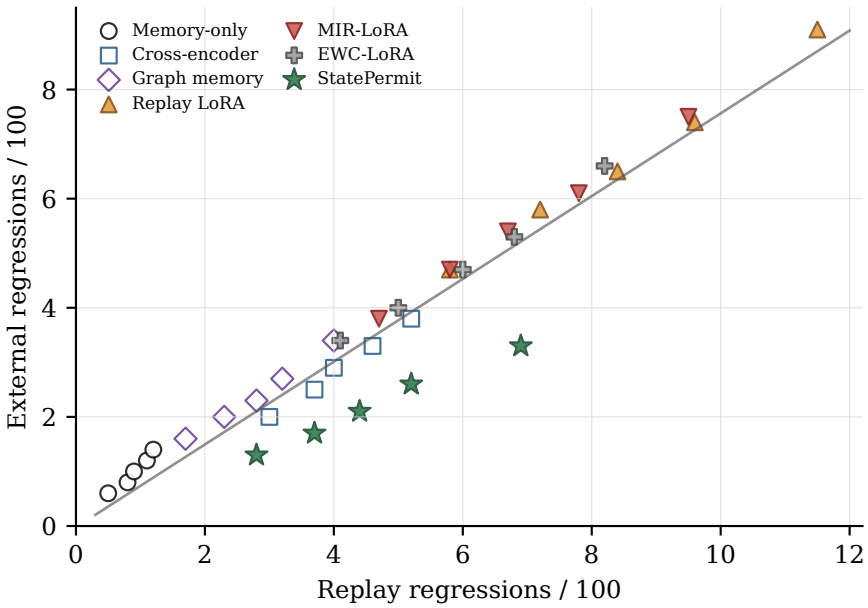


Figure A1: Method-by-seed replay and external burden. Their descriptive Spearman association is 0.86, and pairwise method-rank concordance is 0.82. Scatter around the trend shows that the evaluators are related but nonredundant.

Table A10: Llama comparator set with pooled burden counts and descriptive resolve differences versus StatePermit.

Method	Replay incidents / 1,500	External failures / 2,000	Resolve Δ (points)
Memory-only	14	20	−6.1
Verifier-conditioned router	78	66	−0.7
Frozen cross-encoder	62	58	−0.5
Structured execution memory	42	48	−0.9
Replay LoRA	128	134	−0.4
MIR-LoRA	104	110	−0.2
EWC-LoRA	93	96	−0.6
StatePermit	69	44	reference

Table A11: Qwen comparator counts and descriptive resolve differences versus StatePermit.

Method	Replay incidents / 1,500	External failures / 2,000	Resolve Δ (points)
Memory-only	17	18	−5.9
Frozen cross-encoder	66	62	−0.7
MIR-LoRA	108	116	−0.4
StatePermit	72	50	reference

cross-encoder has fewer replay failures than StatePermit but more external failures, while MIR has more of both.

B.1 Issue-level slices

Each slice reruns the four decision-relevant policies with the same issue assignments and seed averaging. Repository and stream partitions are disjoint; the other partitions provide alternative views of the same 300 issues. The final two columns report StatePermit burden.

Table A12: Resolve and burden by issue slice. M, X, and MIR denote Memory-only, frozen cross-encoder, and MIR-LoRA.

Slice	n	M	X	MIR	StatePermit	Replay / external
Django-family	120	22.0	27.2	27.6	28.3	4.4 / 2.0
Other repositories	180	21.0	26.9	27.1	27.0	4.8 / 2.3
≤ 3 initial failures	174	23.1	28.4	28.6	29.1	4.1 / 1.9
> 3 initial failures	126	19.0	25.1	25.5	25.3	5.3 / 2.6
Patch ≤ 30 lines	192	22.7	27.5	27.8	28.4	4.0 / 1.8
Patch > 30 lines	108	19.1	26.1	26.4	25.9	5.7 / 2.9
Verifier unanimous	214	22.3	27.8	28.0	28.7	4.0 / 1.8
Verifier mixed	86	19.2	25.0	25.6	24.6	6.1 / 3.2
Issues 1–150	150	20.7	26.3	26.6	26.9	4.2 / 2.0
Issues 151–300	150	22.1	27.7	28.0	28.1	5.0 / 2.4

Burden is highest in the large-patch and mixed-verifier slices, the two strata in which StatePermit also does not have the highest point-estimate resolve. These slices identify where stricter admission or fixed-weight state is most valuable, while the repository and temporal partitions show that the aggregate result is not confined to one half of the stream.

Table A13: Paired issue outcomes for StatePermit against the two closest alternatives. W/T/L are wins, ties, and losses after seed averaging.

Slice	W/T/L vs X	Δ vs X	W/T/L vs MIR	Δ vs MIR
Django-family	16/89/15	1.1	14/93/13	0.7
Other repositories	21/137/22	0.1	19/140/21	-0.1
≤ 3 failures	21/136/17	0.7	18/140/16	0.5
> 3 failures	14/97/15	0.2	12/99/15	-0.2
Patch ≤ 30 lines	24/149/19	0.9	21/153/18	0.6
Patch > 30 lines	11/85/12	-0.2	9/85/14	-0.5
Verifier unanimous	27/169/18	0.9	23/174/17	0.7
Verifier mixed	8/68/10	-0.4	6/68/12	-1.0
Issues 1–150	18/116/16	0.6	16/119/15	0.3
Issues 151–300	20/114/16	0.4	17/117/16	0.1

Most paired outcomes are ties after seed averaging, and the signed differences vary across slices. The slice analysis consequently supports distributional coverage and heterogeneity rather than a claim of subgroup-wide superiority.

C Cross-Domain Corroboration

MultiSQL-Personalize separates a 200-event adaptation stream per seed from a 180-task held-out endpoint evaluator. Each state event closes one interaction and yields at most one candidate persistence transition; each endpoint task binds a database schema, a sequence of session-level preference updates, and executable hard constraints. The SQL controller inherits the SWE admission thresholds, five-commit canary cadence, and evaluator roles. Schema/template groups are disjoint across adaptation, canary, fixed replay, and external pools, and all policies receive identical pool

identities and schedules. Satisfaction is the percentage of the 180 endpoint tasks whose final query and stored preferences meet every hard constraint. A sustained improvement is the first event at which satisfaction exceeds the pre-update running mean by at least 2.0 points for five consecutive evaluations. Adaptation events report median [IQR].

Table A14: MultiSQL-Personalize endpoint and adaptation speed. Successes are mean per-seed counts; adaptation episodes are median [IQR], and — means the sustained-gain criterion is not reached within 200 events.

Method	Satisfaction	Successes/180	Adaptation episodes
Memory-only	77.8 ± 1.8	140.0 ± 3.2	—
Structured constraint ledger	84.9 ± 1.6	152.8 ± 2.9	9 [6,14]
Verifier-conditioned router	84.2 ± 1.7	151.6 ± 3.1	8 [5,13]
Replay LoRA	85.7 ± 1.6	154.2 ± 2.9	6 [4,10]
StatePermit	86.3 ± 1.5	155.4 ± 2.7	7 [4,11]

The structured ledger is the closest fixed-weight alternative at 84.9% satisfaction, compared with 86.3% for StatePermit. Replay LoRA adapts one episode earlier in the median, so endpoint and adaptation speed alone do not determine the preferred policy.

Table A15: MultiSQL burden and whole-run cost under the same closure protocol.

Method	Replay reg./100	External reg./100	Overhead
Memory-only	0.8 ± 0.3	0.7 ± 0.3	—
Structured constraint ledger	2.6 ± 0.7	1.9 ± 0.5	+12%
Verifier-conditioned router	3.4 ± 0.9	2.4 ± 0.6	+11%
Replay LoRA	5.8 ± 1.4	4.5 ± 1.1	+22%
StatePermit	3.9 ± 1.0	2.1 ± 0.6	+18.5%

Adding burden and cost separates those endpoints. StatePermit improves satisfaction by 1.4 points over the ledger while recording 2.1 rather than 1.9 external regressions per 100. Replay LoRA adapts fastest but incurs more than twice StatePermit's external burden. The SQL stream thus reproduces the utility–persistence tradeoff with declarative constraints instead of code-patch semantics.

D Adaptive Stress Tests and Protected-State Evidence

D.1 Verified-poison lifecycle

Table A16: Benign-stream verified-poison diagnostic. Harmful is measured at commit time; next-flag is counterfactual detection at the next replay without rollback; rollback is later revocation when enabled. The columns have different denominators and are not additive.

Setting	Harmful (%)	Next-flag (%)	Rollback (%)	Δ resolve
Ungated TTT	18.7 ± 3.2	0.4 ± 0.3	—	-4.8
Verifier-only	6.9 ± 1.8	1.2 ± 0.7	—	-2.1
Full controller	4.9 ± 1.4	—	4.3 ± 1.2	-1.1

This benign-stream intervention is separate from the four 100-attempt adaptive families below. It distinguishes three moments: whether an attempted write is harmful under the independent replay oracle, whether the next scheduled observation would flag it, and whether an enabled controller eventually restores its parent. Their denominators differ, so the columns are interpreted as a sequence rather than summed.

The adaptive protocol holds the benign stream and five seeds fixed and assigns 20 of each attack family's 100 held-out attempts to each seed, combining test-time evasion and textual poisoning patterns (Biggio et al., 2013; Qi et al., 2021). Attackers observe acceptance and end-to-end latency

but cannot inspect canary contents or exact restoration timing. Direct gaming targets executable-verifier blind spots; slow poisoning distributes behavior change below the per-update KL threshold; canary-targeted flips optimize small-KL changes against surrogate tasks; duplicate evasion mutates quarantined traces while preserving intent. A persistent-state write is harmful when the independent replay oracle records a new pass-to-fail transition from the pre-write to post-write state: parent-to-child for adapters and pre-insertion to post-insertion for retrieval memory. Never-restored adapter writes are censored at the 200-commit cap or stream end.

Table A17: Adaptive attacks: harmful persistent-state writes/attempts, monitor misses/harmful writes, and harmful dwell in median/p95 update intervals.

Attack	Method	Harmful/100	Misses/harmful	Dwell median/p95
Direct gaming	Ungated TTT	19/100	15/19	20/44
	Verifier-only	8/100	5/8	15/31
	Frozen cross-encoder	6/100	4/6	10/24
	StatePermit	4/100	1/4	5/10
Slow multi-episode	Ungated TTT	23/100	19/23	25/53
	Verifier-only	12/100	9/12	20/42
	Frozen cross-encoder	9/100	7/9	15/36
	StatePermit	7/100	3/7	10/21
Canary-targeted flip	Ungated TTT	21/100	18/21	25/49
	Verifier-only	11/100	9/11	20/39
	Frozen cross-encoder	8/100	6/8	15/33
	StatePermit	6/100	3/6	10/23
Duplicate evasion	Ungated TTT	17/100	13/17	20/41
	Verifier-only	9/100	6/9	15/34
	Frozen cross-encoder	7/100	5/7	10/27
	StatePermit	5/100	2/5	5/14

Slow multi-episode and canary-targeted attacks are the hardest cases for every policy. Relative to verifier-only admission, StatePermit reduces both harmful writes and monitor misses, and its largest p95 dwell is 23 rather than 42 update intervals.

StatePermit retains benign resolve within 0.4 points of the main 27.5% endpoint in every attack family. The fixed cross-encoder avoids weight mutation but still loses 1.5–2.1 resolve points under poisoned retrieval state, demonstrating why external memory remains inside the downstream-burden measurement.

D.2 Protected-state evidence matrix

The table separates the state transition from the experiment that probes it. In particular, adapter restoration and memory quarantine respond to different attribution units even when both are triggered by a future-task failure.

The privacy probe is an exact-string extraction audit (Carlini et al., 2021): redaction rejects 57/1,186 (4.8%) verifier-passed episodes as secret-bearing, and post-adaptation exact-string extraction is 0.3% (2/648). This evidence motivates redaction as an admission screen while preserving paraphrase and side-channel leakage as explicit residuals.

E Experimental Manifest

The identifiers below bind each result family to one frozen split and governance bundle. Only the canary bundle can actuate adapter restoration. Fixed measurement replay and external suites are measurement-only, whereas on-demand attribution replay may quarantine memory records.

Table A18: Adaptive attacks: attack-induced resolve change, benign-stream resolve, rollback count/commits, and whole-run overhead.

Attack	Method	Attack Δ	Benign resolve	Rollback/commits	Overhead
Direct gaming	Ungated TTT	-4.7	25.2	0/200	+19%
	Verifier-only	-2.2	26.7	0/200	+16%
	Frozen cross-encoder	-1.5	26.9	N/A	+15%
	StatePermit	-0.8	27.4	4/200	+18.4%
Slow multi-episode	Ungated TTT	-5.4	25.0	0/200	+19.3%
	Verifier-only	-3.0	26.5	0/200	+16.2%
	Frozen cross-encoder	-2.1	26.8	N/A	+15.2%
	StatePermit	-1.4	27.2	7/200	+18.8%
Canary-targeted flip	Ungated TTT	-5.1	25.1	0/200	+19.1%
	Verifier-only	-2.7	26.6	0/200	+16.1%
	Frozen cross-encoder	-1.9	26.8	N/A	+15.1%
	StatePermit	-1.2	27.1	6/200	+18.7%
Duplicate evasion	Ungated TTT	-4.2	25.3	0/200	+19.0%
	Verifier-only	-2.4	26.7	0/200	+16.0%
	Frozen cross-encoder	-1.6	26.9	N/A	+15.0%
	StatePermit	-0.9	27.3	5/200	+18.5%

Table A19: Protected state, tested threat, and enforced transition under the stated trusted component.

Protected object	Threat / trusted component	Enforced transition
Live adapter	Verifier-passed poison; executable verifiers and checkpoints	Conjunctive admission, bounded dwell, parent restoration
Retrieval memory	Poisoned or duplicate episodes; provenance index	Verified insertion, quarantine, bounded retrieval
Verifier/provenance	Stale tools, schemas, or allowlists; signed version map	Suspend promotion; retain only memory-eligible traces
Canary/restore state	Small-KL and monitor-targeted changes; hidden canaries	Fixed-panel observation and restore to a passing parent
Tool/schema state	API or schema change; trusted version signal	Version-bound eligibility and memory-only fallback
Secret-bearing episode	Direct secret strings; redaction rules	Redact before either persistent write

Table A20: Evidence attached to each protected object. Residuals state the observed failure mode rather than a universal guarantee.

Protected object	Comparator or probe	Cost, residual, and evidence
Live adapter	MIR-LoRA; adaptive attacks	+18% whole-run; slow poison; Tables A17–A18
Retrieval memory	Frozen cross-encoder; duplicates	+15% whole-run; semantic duplicates; Tables 3, A17
Verifier/provenance	Verifier-only; tool drift	Revalidation; semantic drift; Table A8
Canary/restore state	Eligibility-only; targeted flip	Scheduled latency; monitor misses; Tables 4, A17
Tool/schema state	Naive TTT; high drift	+26.8% p95; delayed version signal; Table A8
Secret-bearing episode	Redaction gate; extraction probe	57/1,186 rejected; 2/648 extracted; paraphrase residual

Table A21: Result-to-configuration map. All rows use seeds $\{13, 37, 61, 71, 97\}$; identifiers bind the reported policies to fixed splits, canaries, and evaluators.

Result family	Config ID	Split / governance bundle	Paper evidence
Primary SWE frontier	`swe.main`	SWE Lite 300; canary bundle v3	Table 2
Expanded closure	`swe.closure.xb`	same stream / two backbones	Table 3; Appendix B
Independent burden	`swe.extreg`	`ext-2000`, disjoint from controls	Table 5; Figure A1
Gate attribution	`swe.gates.sweep`	default plus one-factor sweeps	Table 4; Tables A2–A4
SQL corroboration	`sql.closure`	`sql-180`, same governance bundle	Tables A14–A15
Adaptive poison	`swe.poison.adaptive`	four held-out families, 100 attempts each	Tables A17–A18