WHICH EXPERIENCES SHOULD BE REPLAYED? CAUSAL SELECTION FOR LATENT-MEMORY AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

The latent-memory paradigm in self-evolving agents, exemplified by MemGen, augments reasoning with trainable Weaver latent vectors; however, it still exhibits substantial forgetting in the tested four-task staged No-replay sequential-learning setting. Replaying historical experience can mitigate this forgetting; the key is to identify examples worth replaying. To this end, we diagnose Weaver event-slot trajectories and find that static saliency signals can nominate candidates but cannot independently provide the bilateral helpful/non-harmful evidence required for replay admission; a slot's intervention effect also depends on its specific injection event. We therefore propose Causal Experience Selection (Causal replay), which uses static signals to narrow the event-slot candidate set, independently masks each candidate at the incoming checkpoint, and determines sample admission from bilateral teacher-forced evidence before constructing a fixed-budget replay set. In four-stage continual learning with Qwen2.5-1.5B-Instruct and a MemGen Weaver, Causal replay improves final four-task mean accuracy (Macro) from No replay's 25.41 ± 0.38 to 45.88 ± 0.71, indicating that replay mitigates forgetting in the tested MemGen setting. Under strictly matched actual per-source replay counts and training volume, differing only in historical-example identity, Causal replay obtains a mean paired gain of $+2.19$ percentage points (pp) over quantity-matched Random across three seeds (95% paired-t interval $[+1.13, +3.25]$). This supports selection based on event-slot evidence over random selection in this controlled comparison; task-order permutation and Qwen3-4B also show same-direction results.

1 INTRODUCTION

Self-evolving agents accumulate capabilities along multiple pathways (Fang et al., 2025): verbal reflection externalizes experience as retrievable text, skill curation organizes reusable module libraries, and weight-space self-training and code-space methods directly rewrite parameters or programs. The latent-memory route instead transforms "experience" from human-readable natural language into machine-native continuous representations, enabling cross-sample information transfer (Zhang et al., 2026a; Wu et al., 2026). MemGen (Zhang et al., 2026a) is representative: its trainable Weaver generates multiple event-slot latent vectors per example and injects them into a frozen reasoner. Its event-slot structure changes the memory unit from "one experience corresponds to one text segment" to "one experience corresponds to a set of latent vectors tagged with injection-time markers," thereby supporting agents' experience reuse and current reasoning through continuous memory representations.

However, in four-stage sequential learning with the MemGen Weaver on GSM8K $\rightarrow$ KodCode (Xu et al., 2025) $\rightarrow$ GPQA (Rein et al., 2023) $\rightarrow$ AQuA8000, GSM8K accuracy falls from 55.04 after the first stage to 23.65 at the end, revealing forgetting of previously learned tasks. A common way to mitigate such forgetting is to replay historical examples during subsequent training, but a limited replay budget requires selecting among them. We therefore focus on a central question: which historical examples are most useful to replay for mitigating forgetting? To answer this question, we diagnose event-slot trajectories at the current checkpoint and find that static flow and norm signals can nominate candidates but cannot serve as replay credit: masking omitted candidates can still repair answers, whereas masking top-ranked candidates can cause damage. The same slot index can also have different effects across injection events.

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.

This identification gap motivates us to re-examine conventional experience-replay and data-selection rules. Existing buffer management, influence estimation, and data selection typically use sample-level proxies such as loss, gradients, similarity, or parameter changes (Rebuffi et al., 2017; Aljundi et al., 2019a; Koh & Liang, 2017; Pruthi et al., 2020). These signals can organize or rank historical examples, but do not directly measure whether a specific event-slot in the current MemGen checkpoint supports or undermines prediction of the example's reference completion. Causal analyses of internal representations have primarily supported post-hoc interpretation or targeted editing of pretrained models (Vig et al., 2020; Meng et al., 2022; Geiger et al., 2021), rather than providing a replay-admission rule during training. For latent-memory agents, the question is thus not only "which examples appear important," but "which event-slot evidence in the current model supports replaying this example."

For multi-task staged replay, we propose **Causal Experience Selection**, with the full configuration denoted Causal replay (Full). At each incoming checkpoint after the first stage, static norms nominate a small set of event-slot candidates; we independently mask each $c = (e, s)$ and use bilateral teacher-forced loss changes for helpful/non-harmful admission. Identical slot indices at different events are treated as distinct candidates. Admitted examples form a replay set under pattern deduplication, source quotas, and a fixed budget, while held-out multi-task evaluation determines downstream value.

In four-stage continual learning with Qwen2.5-1.5B-Instruct and a MemGen Weaver, Causal replay (Full) attains a final Macro of 45.88 ± 0.71, above 25.41 ± 0.38 for No replay, indicating that replay mitigates forgetting in the tested MemGen setting. To isolate selector quality, Full and quantity-matched Random share the Stage-1 initialization, training volume, and actual per-source replay counts, differing only in historical sample selection. Full improves over Random by a mean $+2.19$ pp across three paired seeds (45.88 ± 0.71 versus 43.69 ± 0.50; 95% paired-t interval $[+1.13, +3.25]$). Component ablations isolate individual design choices. In task-order permutation and Qwen3-4B (Yang et al., 2025) checks, Causal replay shows gains in the same direction over the respective quantity-matched Random controls. Our contributions are as follows:

1. **Diagnosis of latent-memory replay.** In the tested MemGen sequential-learning setting, we characterize the identification gap between static candidate rankings and per-candidate intervention effects: static signals can nominate candidates but cannot jointly determine repairability and damage, and intervention effects depend on the event.

2. **Event-conditioned causal admission.** We propose Causal Experience Selection, which narrows the event-slot candidate set with static signals, applies per-candidate slot masking at the current checkpoint, and admits examples using bilateral teacher-forced helpful/non-harmful evidence; set-level constraints control replay-set size and diversity.

3. **Two-layer controlled validation.** No replay reveals forgetting in the tested task chain, while Causal replay demonstrates that replay can mitigate it. Under strict per-source matching, the causal selector yields positive paired gains over quantity-matched Random across three seeds; ablations attribute key choices, and task-order and Qwen3-4B checks show gains in the same direction.

2 Related Work

2.1 Self-Evolving Agents and Latent Memory

Self-evolving agents reuse experience through reflection, skill libraries, parameter updates, or code evolution (Fang et al., 2025; Shinn et al., 2023; Xia et al., 2026; Zelikman et al., 2022; Zhang et al., 2026c), while explicit-memory systems store, retrieve, or manage readable experience (Packer et al., 2023; Park et al., 2023; Xu et al., 2026; Yan et al., 2026; Zhong et al., 2024). Continuous-prompt methods optimize or compress learned vectors (Li & Liang, 2021; Lester et al., 2021; Mu et al., 2023); TokMem and MemGen instead generate procedural or experience vectors that are injected into a reasoner, while MEM1 maintains a compact evolving internal state (Wu et al., 2026; Zhou et al., 2026; Zhang et al., 2026a).

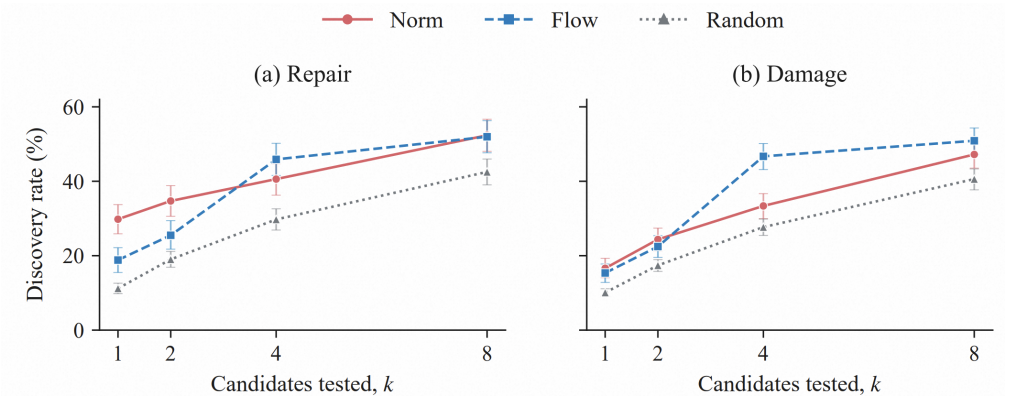


Figure 1: **What candidate rankings find under equal budgets.** We independently test k candidates per question. (a) The fraction of originally wrong questions with a candidate whose removal repairs the answer. (b) The fraction of originally correct questions with a candidate whose removal breaks the answer.

2.2 Forgetting Mitigation and Experience Replay

Continual fine-tuning also suffers catastrophic forgetting (McCloskey & Cohen, 1989; French, 1999; Luo et al., 2025); replay and buffer management, alongside regularization and distillation methods, are common remedies (Kirkpatrick et al., 2017; Li & Hoiem, 2017; Rusu et al., 2016; Lopez-Paz & Ranzato, 2017; Rebuffi et al., 2017; Rolnick et al., 2019; Buzzega et al., 2020; Aljundi et al., 2019a;b; Huang et al., 2024). Existing selection typically uses loss, gradients, or parameter changes rather than a Weaver's checkpoint-specific latent state.

2.3 Sample Selection and Causal Intervention

Data selection estimates sample value (Koh & Liang, 2017; Pruthi et al., 2020; Ghorbani & Zou, 2019; Paul et al., 2021; Mindermann et al., 2022; Xia et al., 2024), whereas causal interventions test internal-state effects (Vig et al., 2020; Meng et al., 2022; Wang et al., 2022; Geiger et al., 2021; 2022). The former rarely measures memory states, and the latter is usually post-hoc; we use slot masking for replay admission at the current checkpoint.

3 Diagnostic Study: The Latent-Memory Identification Gap

In MemGen, the Weaver generates and injects multiple latent slots at injection events throughout the prompt and reasoning process; replay must therefore determine not only which example to select but also which event-slot within it provides useful memory evidence. To characterize this mechanism, we conduct visual behavioral diagnostics of event-slot trajectories and interventions. Figures 1 and 2 visualize Qwen2.5-1.5B-Instruct on GSM8K. We also verify the same diagnostic conclusion across GSM8K, KodCode, GPQA, and AQuA with Qwen2.5-1.5B-Instruct and Qwen3-4B. The central finding of this section is that **static latent signals can only narrow the candidate search space and cannot serve as replay credit; reliable admission must preserve event conditioning and be completed by per-candidate intervention at event-slot granularity.** These behavioral diagnostics establish this identification gap; §4 consequently constructs a conservative selection pipeline, whose downstream value is tested only by the controlled comparison in §5.

3.1 Static Signals Nominate Candidates Only

In the diagnostics, each example has one prompt event and up to five inference events, with 8 latent vectors injected at each event, yielding up to 48 event-slot candidates. We rank candidates separately by latent norm or target-conditioned flow and select the top-k under each ranking; Random selects the same number at random. We test the selected candidates independently, one at a time: each time, we zero only one event-slot, keep all other inputs and the model unchanged, and greedily decode

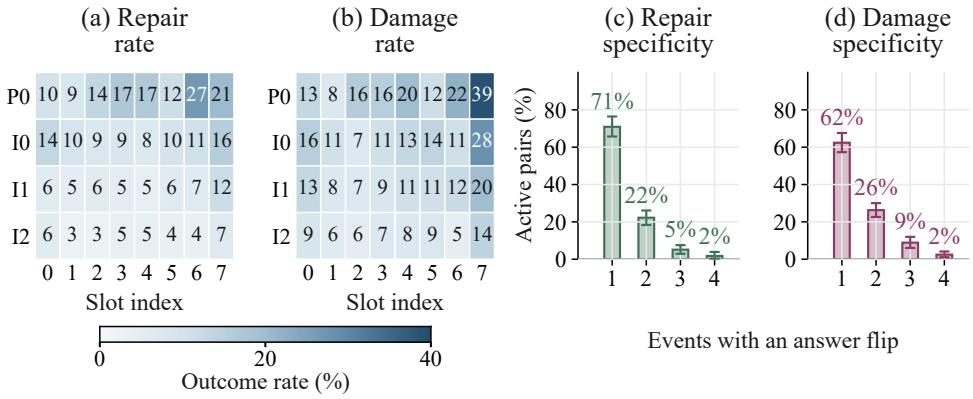


Figure 2: **Event-conditioned answer changes.** (a,b) Repair and damage rates for each event-slot. (c,d) For question-slot pairs with an answer change at least once, the number of events at which it occurs.

a new complete answer. An answer changing from wrong to correct is repair, while one changing from correct to wrong is damage.

Figure 1 asks a simple question: if we can inspect only k candidates, does a ranking find answer-changing candidates more often than random choice? The left panel reports the fraction of originally wrong questions for which at least one tested candidate repairs the answer when masked; the right panel reports the corresponding fraction of originally correct questions for which masking at least one tested candidate breaks the answer. Across budgets, Norm and Flow find both repairs and damage more often than Random. Rankings help identify candidates with substantial answer-changing effects, but these candidates can either repair or damage answers and therefore require further screening.

3.2 Admission Must Preserve Event Conditioning

We further find that per-candidate interventions need to distinguish injection events: even for the same question and slot index, masking the slot at different events can yield different changes in the answer. In Figure 2(a,b), repair and damage rates vary across event-slot pairs, showing that slot index alone does not determine an outcome. Here, an answer change means that masking the slot at an event changes the greedy answer: wrong-to-correct for repair or correct-to-wrong for damage. A question-slot pair is active for repair (damage) if this change occurs at least once for repair (damage). In Figure 2(c,d), 70.9% of active repair pairs and 62.4% of active damage pairs show this change at only one event. Thus, the same slot is effective at one event but not the others; indexing a candidate by slot alone would merge the effective occurrence with ineffective ones. The intervention unit must therefore retain its event-slot identity. Appendix Figure 4 provides an additional Qwen3-4B single-slot masking distribution that shows the same event-conditioned variation.

4 Method

The greedy-generation diagnostics in §3 reveal an identification gap: candidate rankings based on static signals can narrow the search space but are insufficient to jointly determine an example's repairability and risk of damage, and intervention effects depend on the injection event. We therefore propose **Causal Experience Selection**. At each subsequent stage's incoming checkpoint M_t, static latent signals first nominate a small set of event-slot candidates. We then independently mask each candidate and use the resulting teacher-forced loss changes as conservative evidence to determine whether the complete example passes bilateral admission. Finally, random greedy sampling constructs the replay set from the admitted pool under pattern-cap, source-quota, and total-budget constraints.

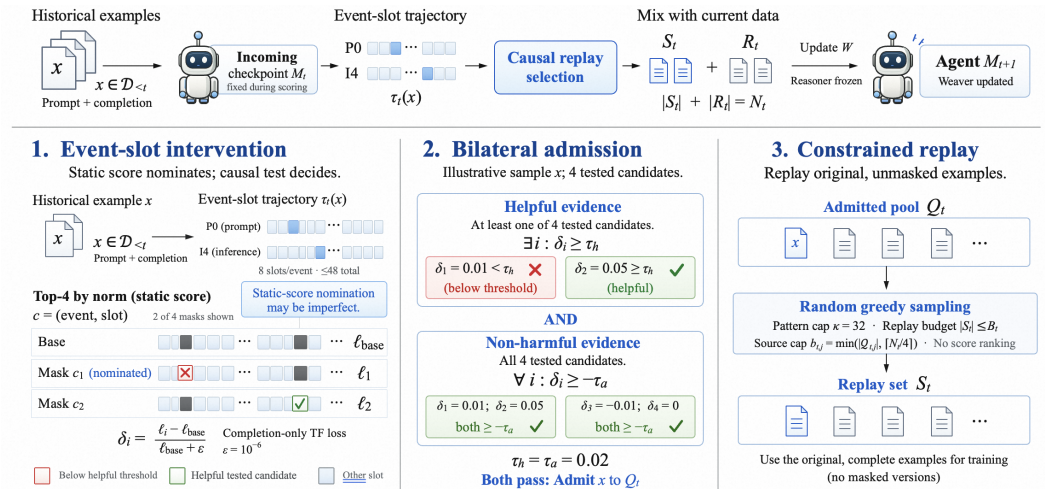


Figure 3: **Method overview.** At each subsequent stage's incoming checkpoint M_t, static norm nominates top-4 event-slot candidates $c = (e, s)$. Per-candidate teacher-forced masks provide sample-level bilateral admission evidence: at least one candidate must be helpful and all tested candidates must be non-harmful. The admitted pool is sampled under pattern, source, and budget constraints; the original, unmasked examples form the replay set.

4.1 Fixed-Size Replay Protocol

To isolate selection quality in controlled main comparisons from training volume and source composition, we construct each stage's mixed set under a fixed total budget. In a sequential task stream $\mathcal{T}_1, \ldots, \mathcal{T}_T$, let D_t denote the current-task candidate pool at stage t after deduplication and availability filtering, N_t the target mixed-set size pre-specified by the data protocol, $B_t \leq N_t$ the stage replay budget, $D_{<t} = \biguplus_{i<t} D_i$ the historical pool, and M_t the incoming checkpoint. All intervention effects and admission decisions are computed under fixed M_t without using held-out labels. At $t = 1$, we train the Weaver only on D_1; for $t > 1$ we select $S_t \subseteq D_{<t}$ and $R_t \subseteq D_t$ such that

$$\widetilde{D}_t = R_t \uplus S_t, \qquad |S_t| \leq B_t, \qquad |R_t| = N_t - |S_t| \implies |\widetilde{D}_t| = N_t, \tag{1}$$

The protocol requires $N_t - |S_t| \leq |D_t|$. Training on $\widetilde{D}_t$ updates the Weaver's trainable parameters W while keeping the reasoner frozen, producing M_{t+1}. Main comparisons vary only S_t, fixing the public MemGen Weaver (Zhang et al., 2026a), candidate pools, training configuration, and evaluation sets, and match actual per-source replay counts; alternative ranking signals are reported under an independent full-pool protocol (§5.1). We retain complete $D_{<t}$ to standardize candidate availability, so an example may be first introduced after its source stage; this proportion is logged.

4.2 Latent-Memory Trajectory and Candidate Nomination

Within this fixed-size protocol, we next construct the sample-level evidence required for historical examples to become eligible for S_t. During a teacher-forced forward pass, the Weaver injects 8 learned query latents at the prompt-to-completion boundary and at up to five positions immediately following delimiters within the continuation, obtaining contextualized hidden states that are mapped to the reasoner embedding space. These positions form the injection-event set $E(x)$ with $|E(x)| = 1 + \min(K(x), 5)$, where $K(x)$ is the number of eligible continuation delimiters; thus the event-slot state sequence $\tau_t(x)$ contains at most 48 vectors. The event specifies the injection moment and the slot the latent vector within it: for $c = (e, s)$, $e \in E(x)$ and $s \in \{1, \ldots, 8\}$, identical slot indices at different events are distinct candidates, while admission always includes or excludes the complete example.

To retain event conditioning while reducing per-candidate intervention cost, we rank all (e, s) by their ℓ_2 norm and nominate the top $k = 4$ as $C_t(x)$, breaking ties by event then slot. This step provides no replay credit; it only reduces the subsequent intervention search space from at most 48

candidates to k (§3.1), where both norm and flow suffice for nomination, so we use the ℓ_2 norm, which is cheaper and needs no target gradient.

4.3 Bilateral Causal Admission

To ground replay credit in behavioral evidence for an example's own continuation rather than static saliency, we measure each candidate's intervention effect at the incoming checkpoint M_t of each subsequent stage. Let $\ell_{\text{base}}(x)$ denote reference-completion cross-entropy under teacher forcing: each token is conditioned on the prompt and preceding reference tokens; prompt tokens incur no loss. For each candidate $c = (e, s) \in C_t(x)$, we intervene **independently**: keeping input, labels, and parameters fixed, we zero out the latent hidden state of slot s at event e after Weaver contextualization and before the Weaver-to-reasoner projection, then re-run the forward pass to obtain $\ell_t(x; c)$. We define the scale-free relative effect and its two directional extrema:

$$\delta_t(x, c) = \frac{\ell_t(x; c) - \ell_{\text{base}}(x)}{\ell_{\text{base}}(x) + \varepsilon}, \quad h_t(x) = \max_{c \in C_t(x)} \big(\delta_t(x, c)\big)_+, \quad a_t(x) = \max_{c \in C_t(x)} \big(-\delta_t(x, c)\big)_+, \tag{2}$$

with $\varepsilon = 10^{-6}$, ensuring a positive denominator for nonnegative cross-entropy. Thus δ_t is well-defined, and the pipeline uses only this scale. A positive $\delta_t(x, c)$ means that removing the candidate raises the reference-completion loss, providing helpful evidence; a negative value means that removal lowers the loss, indicating potential harm. These two directions are the continuous surrogate for the repair and damage of §3, obtained without re-decoding each candidate. The admission set is

$$\mathcal{Q}_{t,j} = \big\{x \in D_j : \; h_t(x) \geq \tau_h \; \wedge \; a_t(x) \leq \tau_a\big\}, \tag{3}$$

Let $\mathcal{Q}_t = \biguplus_{j<t} \mathcal{Q}_{t,j}$ denote the aggregate admitted pool shown in Figure 3. Admission requires two conditions: at least one tested candidate provides helpful evidence of at least τ_h, and the potential harm from every tested candidate is at most τ_a. h_t and a_t determine eligibility only, not cross-sample utility ranking. Both conditions require independent per-candidate measurements: aggregating candidates into one mask would obscure the potential harm of individual candidates and invalidate the harm constraint. This local teacher-forced test measures only whether latent evidence supports the example's own annotation; it does not predict post-optimization transfer and therefore serves only as a conservative admission threshold.

4.4 Set-Level Sampling

Under a fixed replay budget, we apply source, pattern, and total-budget constraints only within the admitted pool and use non-ranking sampling to avoid treating local effects as cross-sample utility.

Pattern deduplication. We define the signed pattern $p_t(x) = \big((c_{t,i}, \operatorname{sign} \delta_t(x, c_{t,i}))\big)_{i=1}^{k}$, where $c_{t,i} = (e_{t,i}, s_{t,i})$ are event-slot identifiers ordered by decreasing nomination norm and $\operatorname{sign} \delta_t(x, c_{t,i}) \in \{-1, 0, +1\}$. Each source retains at most κ examples per pattern.

Budget and source allocation. We pre-specify the stage replay budget B_t and per-source caps $b_{t,j} = \min\big(|\mathcal{Q}_{t,j}|, \lceil N_t/4 \rceil\big)$. Sources are processed in ascending $|\mathcal{Q}_{t,j}|$ order, breaking ties by source index; unused historical budget backfills only the current task.

Non-ranking sampling. The magnitudes of h_t, δ_t, and static norms depend on example difficulty, length, and context and cannot provide a calibrated benefit ranking across eligible examples. We therefore apply greedy sampling to a fixed-seed random permutation of the admitted pool:

$$S_{t,j} = \operatorname{GreedySample}\big(\pi_{\text{seed}}(\mathcal{Q}_{t,j}),\; b_{t,j},\; \kappa\big), \tag{4}$$

Let $S_t = \biguplus_{j<t} S_{t,j}$. An example is included only while its source quota, pattern cap, and total budget remain available; R_t is sampled uniformly from D_t to fill the remaining $N_t - |S_t|$ examples. Pool size or pattern capacity can make the actual replay count $m_{t,j} < b_{t,j}$. Comparisons therefore match the actual per-source replay-count vector $\mathbf{m}_t$, not the nominal cap (§5.1).

Protocol integrity. Each replay stage re-scores historical examples at its incoming checkpoint to avoid stale intervention estimates. Cached records are reused only when the checkpoint, target task, and latent configuration match; all splits use a fixed seed.

4.5 Algorithm and Cost

Algorithm 1 in Appendix A gives detailed pseudocode covering cache validation, score records, selection logs, and constrained sampling. With $k = 4$, scoring requires 5 gradient-free forward passes per historical example, rather than 49 for exhaustive testing of 48 event-slots, totaling $5|D_{<t}|$ passes per stage. Parameter and budget settings are given in §5.1.

5 Experiments

5.1 Setup and Baselines

Tasks, models, and statistics. We evaluate on GSM8K → KodCode → GPQA → AQuA8000 using a frozen Qwen2.5-1.5B-Instruct reasoner and a MemGen-style Weaver trained for two supervised fine-tuning epochs. AQuA8000 is a fixed-size pool from the training side of AQuA that controls final-stage mixed-set size and replay budget; a held-out AQuA test set remains separate. Held-out GSM8K, KodCode, GPQA, and AQuA sets contain 1,319, 2,000, 198, and 254 examples. Macro is the equal-weight average of the four task accuracies; scores use percent and differences percentage points (pp). Values are mean ± sample standard deviation (SD) across seeds 42/43/44. Additional checks are single-run and are not pooled with these results. The reported paired-t interval is two-sided, uses the three seed-wise differences, and has $df = 2$.

Replay and training composition. The full method uses fixed-τ mode with $k = 4$, $\tau_h = \tau_a = 0.02$, and $\kappa = 32$; replay-budget and per-source-cap ratios are 0.5 and 0.25, respectively. These choices use a held-in validation split only. Causal scoring uses only training-side candidates at each stage's incoming checkpoint. Table 1 gives the stage composition.

Table 1: Per-stage training composition (example counts).

Stage	$\lvert D_t \rvert$	Admitted $\mathbf{q}_t$ (GSM/Kod/GPQA)	Replay $\mathbf{m}_t$ (GSM/Kod/GPQA)	Current backfill
S1: GSM8K	6,726	$(0, 0, 0)$	$(0, 0, 0)$	6,726
S2: KodCode	6,995	$(1702, 0, 0)$	$(1577, 0, 0)$	5,418
S3: GPQA	223	$(53, 54, 0)$	$(48, 51, 0)$	124
S4: AQuA8000	8,000	$(2000, 1834, 119)$	$(1765, 1651, 104)$	4,480

$\mathbf{q}_t$ and $\mathbf{m}_t$ count historical examples per source after bilateral admission (Section 4.3) and constrained sampling (Section 4.4), respectively; their difference reflects set-level constraints (Section 4.4), specifically Pattern Cap here.

Comparison protocol. No replay is the forgetting baseline: after Stage 1, it trains only on current-task examples, whereas replay methods mix current and historical examples. All methods share the Stage-1 checkpoint. Full and quantity-matched Random (Q-matched Random in tables) form the primary per-source-matched selector comparison: they share pools, actual per-source replay counts, training-example counts, optimization, and evaluation, differing only in historical-example identity. Each branch re-scores at its incoming checkpoint after divergence; token counts are not matched. Proportional Random provides a source-allocation reference, while Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017) and Learning without Forgetting (LwF) (Li & Hoiem, 2017) provide classical continual-learning references (Appendix Table 6). Full-pool Random, Norm-proxy, TF-delta, and Similarity instead follow an independent full-pool protocol; their different source compositions preclude direct comparison with the main results.

5.2 Controlled Comparisons on the Main Sequence

Table 2(A) shows that Full improves final Macro from 25.41 ± 0.38 for No replay to 45.88 ± 0.71. Under the strictly per-source-matched selector comparison, Full gains 1.92, 2.68, and 1.97 pp over quantity-matched Random across paired seeds, for a mean $+2.19$ pp (95% paired-t interval $[+1.13, +3.25]$). In the seed-43 per-task breakdown in Table 2(B), Full exceeds quantity-matched Random on all four final tasks. In Table 2(C), Full exceeds quantity-matched Random in learned-task average by 0.64 and 1.53 pp at the two early stages, respectively; both replay strategies also exceed No replay. Overall, under the fixed-budget, per-source-matched comparison, Causal replay

selects historical examples more effectively than random replay and improves both final performance and early retention.

Table 2: **Controlled main-sequence results.** Accuracy (%). Panel A reports seeds 42–44 (mean ± sample SD); panels B and C use seed 43. Teal and bold mark the highest accuracy in each column (within stage in panel C); gray marks a non-winning Causal replay score.

(A) Final Macro

Role	Method	S42	S43	S44	Mean ± SD
Platform	No replay	25.03	25.41	25.79	25.41 ± 0.38
Control	Q-matched Random	43.16	43.75	44.16	43.69 ± 0.50
Ours	**Causal replay**	**45.08**	**46.43**	**46.13**	$\mathbf{45.88 \pm 0.71}$

(B) Final per-task accuracy (seed 43)

Role	Method	GSM8K	KodCode	GPQA	AQuA	Macro
Platform	No replay	23.65	9.15	24.75	44.09	25.41
Control	Q-matched Random	53.53	54.60	24.75	42.13	43.75
Ours	**Causal replay**	**54.59**	**55.50**	**28.79**	**46.85**	**46.43**
	Δ vs. Random	+1.06	+0.90	+4.04	+4.72	+2.68

(C) Stagewise accuracy (seed 43)

After stage	Method	GSM8K	KodCode	GPQA	Learned avg.
GSM8K	Shared start	55.04	—	—	55.04
KodCode	No replay	27.45	58.65	—	43.05
	Q-matched Random	55.57	59.40	—	57.49
	Causal replay	**56.25**	**60.00**	—	**58.13**
GPQA	No replay	9.02	**57.40**	22.73	29.72
	Q-matched Random	54.21	55.55	23.23	44.33
	Causal replay	**55.12**	57.20	**25.25**	**45.86**

5.3 Component Attribution

Bilateral admission and set-level constraints. Table 3 fixes the full method's actual per-source replay counts, current-task backfill, total training budget, and evaluation protocol, and compares different admission and set-construction rules. Causal gate only retains bilateral admission but removes Pattern Cap and draws uniformly from each source's admitted pool; w/o harm avoidance removes only the $a_t \leq \tau_a$ condition; gate + rank replaces the random-permutation scan with a descending-h_t scan. Source quota and checkpoint re-scoring (§4.4) are held constant as protocol controls.

Table 3: **Component ablations.** Final Macro accuracy (%, mean ± sample SD over seeds 42–44). Δ is Full minus the variant (pp). Teal bold marks the best Macro; gray marks the Full method label.

Variant	Selection rule	Macro	Δ
Quantity-matched Random	Same per-source replay counts, random	43.69 ± 0.50	+2.19
Causal gate only	Bilateral gate, no Pattern Cap	45.10 ± 0.48	+0.78
w/o harm avoid.	Helpful gate only ($\tau_a = \infty$)	44.79 ± 0.52	+1.09
gate + rank	Full pipeline, h_t-sorted scan	45.42 ± 0.51	+0.46
Causal replay (Full)	Bilateral gate + Pattern Cap	$\mathbf{45.88 \pm 0.71}$	—

In the three-seed means, Causal gate only is 1.41 pp above quantity-matched Random, showing that bilateral event-slot admission outperforms random selection under the same per-source quantities. Removing harm avoidance costs 1.09 pp, indicating a contribution from excluding potentially harmful evidence; replacing the random-permutation scan with h_t ranking costs 0.46 pp, supporting the use of local admission evidence without turning it into a cross-example ranking. Full's Pattern-Cap-constrained sampling yields a 0.78 pp gain over the uniform sampling without Pattern Cap used by Causal gate only.

As a supplement, Appendix Table 7 reports a single seed-43 check under an independent full-pool protocol: Norm-proxy (40.28) and TF-delta (41.63) both fall below full-pool Random (42.99), indicating that static latent strength and local loss changes cannot alone serve as reliable replay-ranking criteria in this check; Similarity (44.10) only shows that retrieval-style selection has signal in this protocol. Because source composition differs, this check is not directly compared with the controlled main or ablation results; the direct controlled conclusion for "gating rather than ranking" remains gate + rank in Table 3.

5.4 Robustness

Replay scale. With $\tau_h{=}\tau_a{=}0.02$ and $\kappa{=}32$ fixed, we apply nested subsampling by source to the complete S4 replay set (counts $\mathbf{m}_4{=}(1765, 1651, 104)$). Quantity-matched Random uses the same replay-count vectors. Causal replay exceeds quantity-matched Random at all three tested scales—1,376, 2,641, and 3,520 replayed examples—showing the same direction across these checks. Complete correct counts and Macro values appear in Appendix Table 5.

Table 4: **Single-run robustness checks.** Accuracy (%). Teal bold marks the best score per column; gray marks the Causal replay label and its non-winning scores.

(A) Task-order permutation

Method	GPQA	KodCode	GSM8K	AQuA	Macro
No replay	**29.80**	10.15	10.61	33.46	21.01
Q-matched Random	27.27	49.45	50.57	38.98	41.57
Causal replay	26.77	**50.95**	**51.71**	**43.31**	**43.18**

(B) Cross-model check (Qwen3-4B)

Method	GSM8K	KodCode	GPQA	AQuA	Macro
No replay	63.38	17.15	27.78	62.99	42.83
Q-matched Random	63.84	53.80	**28.28**	58.66	51.15
Causal replay	**65.66**	**63.30**	27.78	**65.35**	**55.52**

Panel (A) permutes the task order to GPQA → KodCode → GSM8K → AQuA8000: Causal replay attains a Macro of 43.18, above 41.57 for quantity-matched Random ($+1.62$ pp). Panel (B) changes only the backbone to Qwen3-4B while retaining the default GSM8K → KodCode → GPQA → AQuA8000 order, with 55.52 for Causal replay versus 51.15 for quantity-matched Random ($+4.38$ pp). Both single-run checks align in direction with the selection gain on the main sequence.

6 Discussion and Limitations

Static signals only nominate candidates: in the tested task chain, Causal replay mitigates forgetting relative to No replay and yields positive paired gains over quantity-matched Random under strict per-source matching; ablations attribute these gains to key design choices.

Teacher-forced admission measures local effects at each incoming checkpoint rather than post-optimization transfer. The main comparison covers three seeds and one task order; task-order permutation and Qwen3-4B are single-run same-direction checks. Per-candidate re-scoring incurs cost, and example-level matching does not imply token-level compute matching.

7 Conclusion

We formulate latent-memory replay as a checkpoint-specific selection problem: static signals nominate event-slot candidates, while bilateral helpful/non-harmful evidence from per-candidate intervention determines admission. Causal replay mitigates forgetting relative to No replay in the tested task chain and yields positive paired gains over quantity-matched Random across three seeds under strict per-source matching.

AI USE STATEMENT

We used generative AI tools to assist with literature organization, paper outlining, bilingual drafting and editing, code-editing suggestions, and experimental-log summarization. All AI-assisted text, claims, code changes, and experimental interpretations were manually reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

This paper studies training-time selection of benchmark examples and does not involve human-subject data collection or end-user deployment. Selective replay may still amplify bias or over-represent specific task styles if the source pool is imbalanced. We accordingly impose explicit source-task budgets, report per-task results rather than aggregated scores alone, and require separation between training-side data and held-out evaluation.

REFERENCES

Ahmad Al-Tawaha, Shangding Gu, Peizhi Niu, Ruoxi Jia, and Ming Jin. Remembering more, risking more: Longitudinal safety risks in memory-equipped llm agents. *arXiv preprint arXiv:2605.17830*, 2026.

Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. *Advances in neural information processing systems*, 32, 2019a.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32, 2019b.

Yoshua Bengio, Yann LeCun, et al. Scaling learning algorithms towards ai. *Large-scale kernel machines*, 34(5):1–41, 2007.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

Hanxiang Chao, Yihan Bai, Rui Sheng, Tianle Li, and Yushi Sun. Stale: Can llm agents know when their memories are no longer valid? *arXiv preprint arXiv:2605.06527*, 2026.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352, 2023.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 2978–2988, 2019.

Jinyuan Fang, Yanwen Peng, Xi Zhang, Yingxu Wang, Xinhao Yi, Guibin Zhang, Yi Xu, Bin Wu, Siwei Liu, Zihao Li, et al. A comprehensive survey of self-evolving ai agents: A new paradigm bridging foundation models and lifelong agentic systems. *arXiv preprint arXiv:2508.07407*, 2025.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. *Advances in neural information processing systems*, 34:9574–9586, 2021.

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. Inducing causal structure for interpretable neural networks. In *International Conference on Machine Learning*, pp. 7324–7338. PMLR, 2022.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pp. 2242–2251. PMLR, 2019.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*, volume 1. MIT press Cambridge, 2016.

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538 (7626):471–476, 2016.

Yuxian Gu, Li Dong, Hongning Wang, Yaru Hao, Qingxiu Dong, Furu Wei, and Minlie Huang. Data selection via optimal control for language models. In *International Conference on Learning Representations*, volume 2025, pp. 61932–61959, 2025.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1416–1428, 2024.

Yiding Jiang, Allan Zhou, Zhili Feng, Sadhika Malladi, and Zico Kolter. Adaptive data optimization: Dynamic sample selection with scaling laws. In *International Conference on Learning Representations*, volume 2025, pp. 58014–58037, 2025.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. Pmlr, 2017.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 3045–3059, 2021.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)*, pp. 4582–4597, 2021.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, 2024.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, et al. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning*, pp. 15630–15649. PMLR, 2022.

Jesse Mu, Xiang Li, and Noah Goodman. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36:19327–19352, 2023.

Siru Ouyang, Jun Yan, Yanfei Chen, Rujun Han, Zifeng Wang, Bhavana Dalvi Mishra, Rui Meng, Chun-Liang Li, Yizhu Jiao, Kaiwen Zha, et al. Skillos: Learning skill curation for self-evolving agents. *arXiv preprint arXiv:2605.06614*, 2026a.

Siru Ouyang, Jun Yan, I Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long Le, Samira Daruki, Xiangru Tang, et al. Reasoningbank: Scaling agent self-evolving with reasoning memory. In *International Conference on Learning Representations*, volume 2026, pp. 94327–94354, 2026b.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in neural information processing systems*, 34:20596–20607, 2021.

Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *European conference on computer vision*, pp. 524–540. Springer, 2020.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in neural information processing systems*, 33: 19920–19930, 2020.

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, and Timothy P Lillicrap. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*, 2019.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 5533–5542. IEEE, 2017.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Shuai Shao, Qihan Ren, Dongrui Liu, Chen Qian, Boyi Wei, Dadi Guo, Jingyi Yang, Xinhao Song, Linfeng Zhang, Weinan Zhang, et al. Your agent may misevolve: Emergent risks in self-evolving llm agents. In *International Conference on Learning Representations*, volume 2026, pp. 99728–99793, 2026.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. *Advances in neural information processing systems*, 28, 2015.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401, 2020.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. *arXiv preprint arXiv:2409.07429*, 2024.

Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.

Yuhuai Wu, Markus N Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. *arXiv preprint arXiv:2203.08913*, 2022.

Zijun Wu, Yongchang Hao, and Lili Mou. Tokmem: One-token procedural memory for large language models. In *International Conference on Learning Representations*, volume 2026, pp. 32460–32480, 2026.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*, 2024.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, et al. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*, 2026.

Zidi Xiong, Yuping Lin, Wenya Xie, Pengfei He, Zirui Liu, Jiliang Tang, Himabindu Lakkaraju, and Zhen Xiang. How memory management impacts llm agents: An empirical study of experience-following behavior. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 623–645, 2026.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604, 2026.

Zhangchen Xu, Yang Liu, Yueqin Yin, Mingyuan Zhou, and Radha Poovendran. Kodcode: A diverse, challenging, and verifiable synthetic dataset for coding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6980–7008, 2025.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z Pan, et al. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12805–12825, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Ye Yu, Xiaopeng Yuan, Haibo Jin, Heming Liu, Yaoning Yu, and Haohan Wang. Do self-evolving agents forget? capability degradation and preservation in lifelong llm agent adaptation. *arXiv preprint arXiv:2605.09315*, 2026.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, volume 2024, pp. 1651–1678, 2024.

Guibin Zhang, Muxin Fu, and Shuicheng Yan. Memgen: Weaving generative latent memory for self-evolving agents. In *International Conference on Learning Representations*, volume 2026, pp. 22555–22588, 2026a.

Haozhen Zhang, Quanyu Long, Jianzhu Bao, Tao Feng, Weizhi Zhang, Haodong Yue, and Wenya Wang. Memskill: Learning and evolving memory skills for self-evolving agents. *arXiv preprint arXiv:2602.02474*, 2026b.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin gödel machine: open-ended evolution of self-improving agents. In *International Conference on Learning Representations*, volume 2026, pp. 104223–104294, 2026c.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, et al. Agentic context engineering: Evolving contexts for self-improving language models. In *International Conference on Learning Representations*, volume 2026, pp. 86069–86100, 2026d.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 19724–19731, 2024.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Bryan Kian Hsiang Low, and Paul Liang. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents. In *International Conference on Learning Representations*, volume 2026, pp. 58413–58438, 2026.

Algorithm 1 Detailed Causal Experience Selection

Require: Incoming checkpoint M_t, current pool D_t, historical pools $\{D_j\}_{j<t}$, and $(N_t, k, \tau_h, \tau_a, \kappa, \text{seed}, B_t)$
Ensure: Fixed-size mixed set $\widetilde{D}_t$
Initialize $S_t \leftarrow \varnothing$ and admitted pools $\mathcal{Q}_{t,j} \leftarrow \varnothing$ for all $j < t$
for each source $j < t$ and example $x \in D_j$ **do**
  if a cached record matches M_t, target task, and latent configuration **then**
    Load $C_t(x)$, $\{\delta_t(x,c)\}_{c \in C_t(x)}$, $h_t(x)$, $a_t(x)$, and $p_t(x)$
  else
    Compute the teacher-forced baseline loss $\ell_{\text{base}}(x)$ on the completion
    Extract event-slot states and set $C_t(x) \leftarrow \text{TopK}_{(e,s)} \|\text{hidden}(x,e,s)\|_2$
    for each $c = (e,s) \in C_t(x)$ **do**
      Zero slot s at event e after contextualization; compute $\ell_t(x;c)$ and $\delta_t(x,c)$
    end for
    Derive $h_t(x)$, $a_t(x)$, and signed pattern $p_t(x)$; persist the score record
  end if
  if $h_t(x) \geq \tau_h \ \wedge \ a_t(x) \leq \tau_a$ **then**
    Add $(x, p_t(x))$ to $\mathcal{Q}_{t,j}$
  end if
end for
for each $j < t$ **do**
  $b_{t,j} \leftarrow \min\big(|\mathcal{Q}_{t,j}|, \lceil N_t/4 \rceil\big)$
end for
for sources $j < t$ in ascending $|\mathcal{Q}_{t,j}|$ order, breaking ties by j **do**
  for each $(x, p_t(x)) \in \pi_{\text{seed}}(\mathcal{Q}_{t,j})$ **do**
    if source quota, pattern cap κ, and total budget B_t remain available **then**
      Add x to S_t and update the source, pattern, and total counters
    end if
  end for
end for
$R_t \leftarrow \text{Unif}(D_t, N_t - |S_t|)$; persist the selection log; **return** $\widetilde{D}_t \leftarrow R_t \uplus S_t$

A Scoring and Replay Details

Each score record is keyed by $(t, j, \text{SHA256}(\text{prompt}\|\text{completion})[:16])$ and stores the incoming checkpoint, target task, and latent configuration. Its candidate fields comprise $\ell_{\text{base}}(x)$, $C_t(x)$, the masked loss and $\delta_t(x,c)$ for every $c \in C_t(x)$, together with $h_t(x)$, $a_t(x)$, the signed $p_t(x)$, and the admission indicator. Each stage-level selection log stores per-source admitted-pool sizes $|\mathcal{Q}_{t,j}|$, caps $b_{t,j}$, actual replay counts $m_{t,j}$, and the total budget B_t, pattern histogram, active constraints, and current-task backfill count $N_t - |S_t|$. The admission, sampling, and matching rules are defined in §4.3–§4.4.

B Classical Continual-Learning Baseline Configuration

EWC trains the Weaver on current-task data only, adding $\lambda \sum_i F_i(\theta_i - \theta_i^*)^2$ over its trainable parameters to the objective; F is the previous stage's final diagonal Fisher estimate, and λ is validation-selected from $\{1, 10, 100\}$. LwF distills from the frozen previous-stage Weaver, adding $\alpha \mathcal{L}_{\text{KD}}$ to the objective; α is validation-selected from $\{0.5, 1.0, 2.0\}$. Neither replays historical examples; both match No replay's training-step count.

C Supplementary Diagnostic Figure

This section supplements the main-text event-conditioning diagnostic with a Qwen3-4B single-slot masking distribution. This figure does not compare replay training methods and does not extend the main text's causal or training-gain claims.

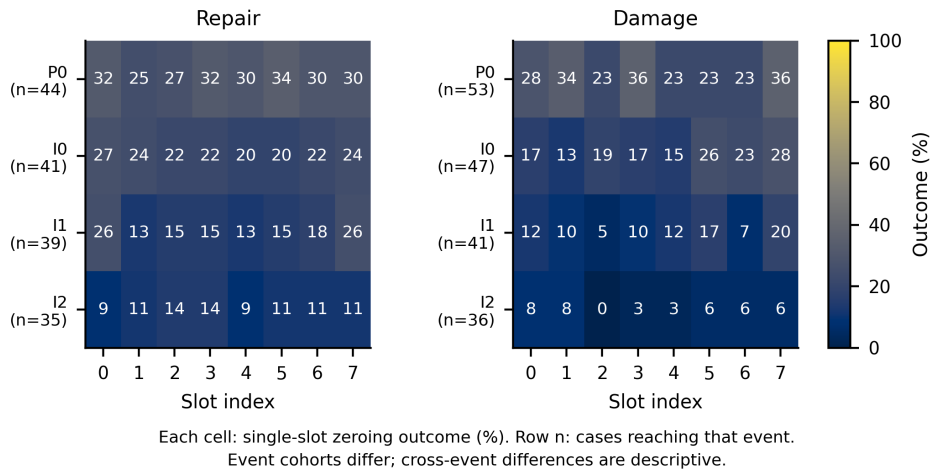


Figure 4: **Qwen3-4B event-slot single-slot masking diagnostics.** Each cell records the repair or damage proportion after zeroing out a fixed event-slot, with row-side n indicating the number of cases reaching that event. The distribution varies across event-slot pairs; different events have different sample cohorts, so cross-event values are interpreted descriptively only.

D Supplementary Experiment Tables

This appendix retains integer counts and per-task details for non-primary controls. The three-seed main results and early-stage trajectories are reported in Table 2.

Table 5: Replay-scale check (seed 43), corresponding to §5.4. Correct counts are ordered as GSM8K / KodCode / GPQA / AQuA. Macro is computed as $\frac{1}{4}(100c_G/1319 + 100c_K/2000 + 100c_P/198 + 100c_A/254)$ and displayed to two decimal places.

S4 replay count	Method	Correct (G/K/P/A)	Macro
0	No replay ref.	$(312, 183, 49, 112)$	25.41
1,376	Q-matched Random	$(600, 850, 50, 108)$	38.94
1,376	Causal replay	$(620, 880, 52, 110)$	**40.14**
2,641	Q-matched Random	$(670, 1000, 50, 107)$	42.04
2,641	Causal replay	$(685, 1040, 54, 111)$	**43.73**
3,520	Q-matched Random	$(706, 1092, 49, 107)$	43.75
3,520	Causal replay	$(720, 1110, 57, 119)$	**46.43**

The zero-replay row is a training-chain reference, not an independent zero-replay experiment sharing the S4 incoming checkpoint.

Table 6: Source-allocation and classical continual-learning references. Panel (A) reports the single seed-43 per-task result for Proportional Random using correct counts; its Macro is computed as $\frac{1}{4}(100c_G/1319 + 100c_K/2000 + 100c_P/198 + 100c_A/254)$. Panel (B) reports three-seed mean per-task scores for EWC and LwF, with Macro mean $\pm$ sample SD; it is not pooled with the seed-43 results in Table 2(B).

(A) Proportional Random (seed 43)

Method	Correct (G/K/P/A)	Macro
No replay	$(312, 183, 49, 112)$	25.41
Proportional Random (by source size)	$(716, 1091, 51, 99)$	43.39

(B) Classical continual-learning baselines

Method	GSM8K	KodCode	GPQA	AQuA	Macro
EWC	35.33	30.57	24.75	44.49	$33.79_{\pm 0.71}$
LwF	37.68	33.15	25.42	44.88	$35.29_{\pm 0.63}$

Table 7: Alternative ranking rules under the independent full-pool protocol (seed 43). Each method replays about 4,000 examples from the merged historical pool at S4, with full-pool Random as the control. This protocol changes source composition and is not directly compared with the main-text per-source-matched results.

Method	GSM8K	KodCode	GPQA	AQuA	Macro
full-pool Random (merged pool)	53.15	53.75	27.27	37.80	42.99
Norm-proxy replay	47.99	45.70	23.74	43.70	40.28
TF-delta replay	49.73	51.65	25.76	39.37	41.63
Similarity replay	51.02	52.90	28.79	43.70	44.10