# WHEN MEMORY HURTS: CAUSAL MEASUREMENT AND JOINT READ–WRITE CONTROL IN LONG-HORIZON AGENTS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Persistent memory can improve a language agent on average while causing it to fail on individual decisions. We measure such failures with coupled interventions on memory exposure and write-back, followed by retain–delete tests of the records that exposure induces. Our benchmark, NMR-LOOP, covers embodied control, web workflows, and evolving personal state. On organic streams without injected records, retrieved memory turns a success into a failure—*negative memory retrieval* (NMR)—on 4.5% [4.2, 4.8] of paired probes despite a mean benefit of 2.6 percentage points [2.2, 2.9]; brackets denote 95% stream-clustered bootstrap confidence intervals. We then introduce LOOPGUARD, a joint read–write controller trained on intervention labels for immediate utility and for harmful records induced within a finite follow-up window. At comparable positive-transfer rates near 12%, LOOPGUARD attains a controlled-evaluation NMR of 3.3%, compared with 9.5% for dense retrieval, 5.8% for Causal Memory Intervention, 5.3% for MeClear, and 4.7% for its read-only variant. Joint control also moves write-back interaction estimates toward zero in all three domains over eight future episodes. At deployment, LOOPGUARD executes each task once, with no additional counterfactual reader calls and a 6.2% token overhead over dense retrieval.

## 1 INTRODUCTION

Long-running language agents store conversations, tool outcomes, plans, and procedures so that they can reuse experience across sessions (Park et al., 2023; Shinn et al., 2023; Wang et al., 2023; Packer et al., 2023; Sumers et al., 2024). Work on agent memory accordingly designs and evaluates mechanisms for writing, retrieval, consolidation, and forgetting (Xu et al., 2025; Wu et al., 2025; Hu et al., 2025). Such evaluations typically report average task performance, which captures the net value of memory but can conceal individual decisions on which retrieved memory causes a failure.

Consider a web agent that retrieves a previously successful reimbursement workflow after the interface has changed. Following the old workflow sends the payment to the wrong account; without that memory, the agent would inspect the current page and succeed. We call such a paired success-to-failure transition *negative memory retrieval* (NMR). The retrieved record is historically accurate and semantically relevant; it is harmful because the conditions that made the experience useful no longer hold. In the average, benefits on other requests can mask these harmful reversals.

Memory writing extends the problem beyond the current decision. An agent may summarize an unsuccessful trajectory as a reusable lesson, drop the uncertainty about its outcome, and later retrieve the resulting record. Reading shapes behavior, and behavior supplies the material for future memory. Evaluating only the currently retrieved block leaves this feedback loop unmeasured; tracing provenance alone reveals where a record came from, but not whether it harms a later decision.

We measure these effects with three coupled interventions. First, exposing versus removing the retrieved block at a fixed checkpoint measures its effect on the current task. Second, crossing exposure to a designated *seed* record with write-on and write-off follow-up measures how memory updates change the seed's longitudinal effect. Third, retain–delete probes at later checkpoints test whether individual records induced by the seed—its *descendants*—harm subsequent decisions (Figure 1). The distinction matters: an adverse write-back interaction is defined on task outcomes, whereas a harmful



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.

descendant requires a specific induced record that worsens a later decision. All paired quantities are defined under a declared replay coupling and a finite observation window.

Recent work addresses evolving state, decision-aware memory selection, abstention, and persistent contamination (Chao et al., 2026; Fan et al., 2026; Srivastava, 2026; Yang et al., 2026; Iscan, 2026; Wang et al., 2026; Liu, 2026). We bring immediate exposure effects, write-back effects, and record-level harm into a single evaluation protocol and make three contributions:

- **A longitudinal causal formulation** of memory harm that separates immediate reversals (NMR), write-back interaction, and deletion-verified harmful descendants, each with an explicit coupling, denominator, and observation window (Section 3).
- **NMR-LOOP**, which instantiates the protocol in embodied control, web workflows, and evolving personal state (Section 4). Controlled streams with injected state changes support matched seed interventions; organic streams measure harmful reversals during ordinary memory use.
- **LOOPGUARD**, a controller that uses offline intervention labels to govern both retrieval and memory formation (Section 5). Its read model combines predicted immediate utility with predicted descendant burden, and a write model trained on store-versus-withhold interventions routes proposed records. At positive-transfer rates near 12%, it attains 3.3% NMR, compared with 5.8% for Causal Memory Intervention (CMI), 5.3% for MeClear, and 4.7% for its read-only variant, without additional counterfactual reader calls at deployment (Section 6).

## 2 RELATED WORK

**Memory and changing state.** Memory architectures range from reflection and reusable skills to multi-tier stores and dynamically linked records (Shinn et al., 2023; Wang et al., 2023; Packer et al., 2023; Xu et al., 2025). LoCoMo, LongMemEval, MemoryAgentBench, and LongMemEval-V2 test long-range recall and experience reuse (Maharana et al., 2024; Wu et al., 2025; Hu et al., 2025; Wu et al., 2026). MINTEval studies multi-target interference, STALE evaluates implicit invalidation and introduces CUPMem, and StateMemBench and StateMem target evolving relational state (Lee et al., 2026; Chao et al., 2026; Fan et al., 2026). These benchmarks motivate our state changes; our paired estimand asks whether exposing memory worsens a matched decision.

**Decision-aware selection and clearance.** Causal Memory Intervention (CMI) compares outcomes under memory interventions (Srivastava, 2026). Decision-Aware Memory Cards use counterfactual-inspired utility signals (Guan et al., 2026); MeClear combines leave-one-out screening, sampled Shapley attribution, and recovery verification (Yang et al., 2026); and RSCB-MC treats withholding memory as a risk-sensitive action (Iscan, 2026). These methods make immediate utility and selective exposure important points of comparison. LOOPGUARD extends the prediction target from immediate utility to the harmful future records that a decision produces, and uses this target for both read and write control.

**Persistent influence and provenance.** Memory poisoning attacks show that adversarial records can redirect later behavior (Chen et al., 2024; Dong et al., 2025). State Contamination studies influence that persists in memory after a toxic episode, and Memory Contagion investigates cross-temporal evaluator bias (Wang et al., 2026; Liu, 2026). PPMF preserves provenance and constrains the authority granted to consolidated memory (Xu et al., 2026). We instead target ordinary task reliability under state change and fallible writing. Our distinguishing measurement is a seed-induced record whose effect on a later task is verified by deletion. Because multiple sources can contribute to such a record, we restrict unique-parent attribution to controlled lineages.

## 3 LONGITUDINAL CAUSAL FORMULATION

### 3.1 INTERVENTION UNITS AND COUPLING

At checkpoint $i$, the agent observes $o_i$, maintains working state $w_i$, and retrieves an ordered block $R_i$ from store $M_i$. Before the *reader* (the language model that acts on retrieved memory) sees this block, we checkpoint the observation, working state, environment, store, model configuration, and available random state. The reader configuration, utility definition, and evaluation horizon are fixed

across arms. Task utility $Y_i \in [0, 1]$, with larger values indicating better performance, is measured before the subsequent memory-writing step.

We distinguish two interventions. A *block intervention* compares $Y_i(R_i)$ with $Y_i(\varnothing)$ and thus measures the effect of the current retrieved block at a fixed checkpoint. The empty-block arm retains the same pre-exposure history and working state, including the effects of earlier memory use. An *item intervention* holds an ordered background block $B_i$ fixed and varies a designated seed $m_i$:

$$R_i^1 = B_i \oplus m_i, \qquad R_i^0 = B_i, \qquad R_i^{\mathrm{plc}} = B_i \oplus p_i, \tag{1}$$

where $\oplus$ denotes insertion at a prespecified slot and $p_i$ is a length-matched placebo intended to carry no task-relevant information. All non-target content and its relative order are fixed at exposure, and nothing is refilled or re-truncated after deletion. Because the item control equals an empty block only when $B_i = \varnothing$, we report block-level harm rates and item-level effects separately.

For replayable pairs, both arms restore the same pre-exposure checkpoint and share component-indexed random tapes for reader decoding, environment transitions, retrieval tie-breaking, and writing (Section A). For either intervention, we write $Y_i^a(U_i)$ for the outcome under arm $a \in \{0, 1\}$ and shared random tape $U_i$, suppressing $U_i$ when the coupling is clear. Paired NMR is defined with respect to this declared coupling. For non-replayable APIs, repeated calls per arm instead estimate $\mu_i^a = \mathbb{E}[Y_i^a \mid i]$ and the expected contrast $\mu_i^1 - \mu_i^0$. The marginal contrast characterizes expected utility, whereas the coupling-specific NMR characterizes paired reversals; marginal success rates alone only bound the paired harm rate (Section A).

### 3.2 Immediate harm and positive transfer

Our diagnostic evaluation considers memory-optional tasks, for which the current evidence suffices without retrieved memory. Let $\Delta_i^{\mathrm{block}} = Y_i(R_i) - Y_i(\varnothing)$ be the paired block effect and define, for a tolerance $\epsilon \geq 0$, the rates of negative and positive memory retrieval (NMR and PMR)

$$\mathrm{NMR}_\epsilon = \mathbb{P}(\Delta_i^{\mathrm{block}} < -\epsilon), \qquad \mathrm{PMR}_\epsilon = \mathbb{P}(\Delta_i^{\mathrm{block}} > \epsilon). \tag{2}$$

Probabilities are taken over evaluation units and any randomness in the declared coupling, so for stochastic outcomes these paired rates depend on the chosen coupling.

For binary success and $\epsilon = 0$, NMR is the rate of transitions from success with empty memory to failure with the retrieved block, and PMR, which we also call the positive-transfer rate, is the rate of the reverse transition. All reported experiments use binary success with $\epsilon = 0$, and we write NMR and PMR for $\mathrm{NMR}_0$ and $\mathrm{PMR}_0$. Conditional harm, $h_{\mathrm{cond}} = \mathbb{P}(Y_i(R_i) = 0 \mid Y_i(\varnothing) = 1)$, measures harm among units that succeed with empty memory; whenever $\mathbb{P}(Y_i(\varnothing) = 1) > 0$, it satisfies $h_{\mathrm{cond}} = \mathrm{NMR}_0 / \mathbb{P}(Y_i(\varnothing) = 1)$. Over the same evaluation units and weights, the mean effect decomposes as

$$\mathbb{E}[Y_i(R_i)] - \mathbb{E}[Y_i(\varnothing)] = \mathrm{PMR}_0 - \mathrm{NMR}_0. \tag{3}$$

Positive average transfer can therefore coexist with harmful reversals, which is why we report the mean effect alongside NMR and conditional harm.

Our primary estimand is the total effect of the retrieved block relative to empty memory, which includes both memory content and changes in context occupancy. A length-matched placebo block provides a complementary diagnostic of sensitivity to added context.

### 3.3 Longitudinal effects and write-back amplification

For the item intervention, we cross seed exposure $A \in \{0, 1\}$ with post-outcome writing $W \in \{0, 1\}$. Write-on enables the prespecified update policy of the evaluated condition, including LoopGuard's write routing when applicable. Write-off freezes all mutations of external memory, including append, replacement, consolidation, and deletion, while still allowing the environment and working state to evolve. Let $Y_{i+\ell}^{a,w}$ denote the endpoint of the seed episode ($\ell = 0$) or of the $\ell$-th future episode ($1 \leq \ell \leq H$). We define the discounted longitudinal treatment effect (LTE) and the write-back amplification contrast (WAC) as

$$\mathrm{LTE}_H^w(i) = \sum_{\ell=0}^{H} \gamma^\ell \big(Y_{i+\ell}^{1,w} - Y_{i+\ell}^{0,w}\big), \quad 0 < \gamma \leq 1, \tag{4}$$

$$\mathrm{WAC}_H(i) = \mathrm{LTE}_H^1(i) - \mathrm{LTE}_H^0(i). \tag{5}$$

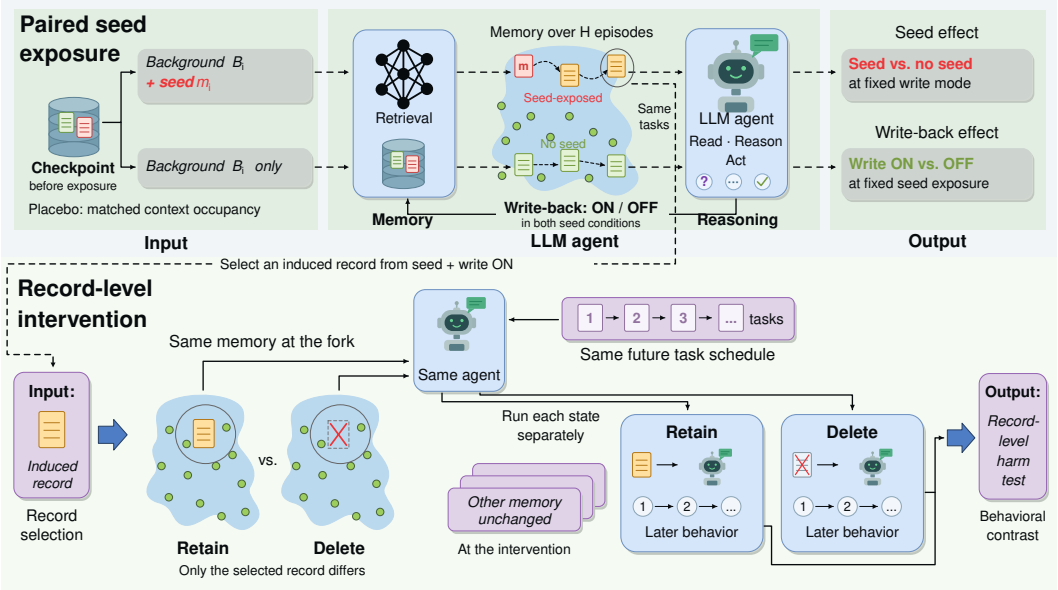


Figure 1: **Two levels of causal intervention.** Seed exposure and write-back are crossed at a shared checkpoint, and outcomes are followed through the seed episode and $H$ future episodes. Write-back amplification is the difference between the seed's effects with writing on and off. Candidate descendants are treatment-induced record versions; their harm is verified by retaining versus deleting a version from the retrieved block at a prespecified later checkpoint, with other evidence held fixed. The task schedule is shared across branches, whereas branch states may diverge.

Because writing starts after the seed endpoint, $Y_i^{a,1} = Y_i^{a,0}$, and the $\ell = 0$ terms cancel in $\mathrm{WAC}_H$. A negative WAC means that enabling memory updates makes the seed's longitudinal effect more adverse. WAC is an interaction between assigned writing policies and can reflect either harmful additions or lost beneficial adaptations. The descendant intervention below isolates the later contextual harm of a particular induced record.

### 3.4 Intervention-verified harmful descendants

A *candidate descendant* is a new or revised record version that is produced in the exposed branch and is either absent from the coupled control or semantically different from its control counterpart under a fixed matching rule; logged ancestry must connect it to the seed. At a later checkpoint $t$ at which a candidate $d$ is retrieved, we delete that version from the exposed block while keeping the remaining evidence fixed. Its local effect is

$$I_{d,t} = Y_t(R_t) - Y_t(R_t \setminus \{d\}). \tag{6}$$

A record is a *pathwise verified harmful descendant* if its production is treatment-induced and $I_{d,t} < -\epsilon$ at a prespecified exposure under the declared coupling. When the control branch contains a revised counterpart of the record, the audit additionally compares retaining the exposed version with replacing it by that counterpart. Both comparisons measure the record's contextual effect.

**Finite-window accounting.** An append or content-changing revision creates a record version, and each retrieval of that version is a separate activation; verified versions are counted once. Candidates whose test is not completed before follow-up ends remain unresolved. Each quantity has its own denominator: NMR uses paired decision probes, WAC uses seed interventions, and descendant counts use treated streams. A per-stream descendant rate therefore includes seed exposures without an immediate error, whereas a rate conditional on immediate seed NMR describes a different, selected population. Lineage generations count parent–child writes, whereas episode indices track elapsed follow-up; unique-parent analyses rely on separately resolved parent attribution (Section A).

# 4 NMR-LOOP: CLOSED-LOOP EVALUATION

**Controlled streams.** Each *stream* is a sequence of episodes in which an agent reads from and writes to a persistent memory store. In controlled streams, a seed lineage begins with a historical experience whose use can become harmful after a state change. The protocol covers four mechanisms: stale state, mismatched preconditions, unreliable outcome provenance, and write-back inversion, in which an uncertain outcome is recorded as a successful lesson. Target tasks contain sufficient current evidence to succeed without the seed memory. Randomized seed exposure compares the same background block with and without the seed, testing whether historical experience remains applicable.

**Organic streams.** In organic streams, agents write memory as usual and no candidates are injected. Natural retrieval events are sampled on a fixed schedule, including events with no apparent error. Each sampled checkpoint receives block and empty-memory probes, and length-matched placebo probes diagnose context occupancy. The probes run on shadow branches and do not alter the deployment stream. Organic prevalence therefore describes the sampled retrieval workload, whereas controlled seed interventions support the longitudinal contrasts; we report the two populations separately.

**Domains and sampling units.** The protocol covers household tasks derived from ALFWorld (Shridhar et al., 2021), sandboxed web workflows derived from WebArena and WorkArena (Zhou et al., 2024; Drouin et al., 2024), and evolving personal-state tasks informed by LongMemEval, STALE, and StateMem (Wu et al., 2025; Chao et al., 2026; Fan et al., 2026). Splits are formed by latent rule and entity family, so related histories cannot serve as independent evidence across splits. The stream is the resampling unit: checkpoints and counterfactual branches within a stream remain dependent. Each reported rate states its sampling unit and denominator.

**Setup.** The study uses 3,600 controlled streams, each with 24 paired target checkpoints (86,400 checkpoint pairs in total), and 2,700 organic streams. Controller training uses 24,000 paired interventions from separate training streams; development, calibration, and test splits are disjoint at the stream level. Downstream effects are tracked over $H = 8$ future episodes with discount $\gamma = 0.90$. Unless stated otherwise, results use the full corresponding test split; operating points and uncertainty estimation are described in Section 6, and Table 5 summarizes the remaining settings.

**From a seed to a harmful descendant.** The descendant audit applies three checks. Restoring the same pre-exposure checkpoint, with background order and the declared random tapes fixed, isolates the seed intervention; paired follow-up identifies a record version produced or revised because of exposure and connected to the seed by a logged ancestry path; and a retain–delete probe at a later retrieval checkpoint tests whether that particular version worsens the task outcome. Only records that pass all three checks enter the verified harmful set.

**Measurement validity.** The protocol requires environment state, task success, version creation, and pre-treatment equality across arms to be recorded. Blinded annotation assesses semantic record matching, applicability, and mechanism labels, and disagreements are adjudicated. Exclusions are limited to corrupted checkpoints or invalid exogenous coupling and never result from ordinary post-treatment divergence. Fixed checkpoints isolate read effects, whereas policy-specific future stores measure closed-loop outcomes; write-only interventions affect only decisions after the fixed seed endpoint.

# 5 JOINT READ–WRITE CONTROL WITH LOOPGUARD

## 5.1 LEARNING IMMEDIATE UTILITY AND DESCENDANT BURDEN

For candidate $m_j$ at an offline supervision checkpoint $i$, an item-deletion intervention yields the immediate utility $u_{ij} = Y_i(R_i) - Y_i(R_i \setminus \{m_j\})$, with harm and benefit labels $h_{ij} = \mathbf{1}[u_{ij} < -\epsilon]$ and $b_{ij} = \mathbf{1}[u_{ij} > \epsilon]$. Following the resulting branches supplies a descendant-burden target

$$g_{ij} = \sum_{d \in \mathcal{D}^{\text{harm}}_{ij,H}} \gamma^{e(d)}, \tag{7}$$

where $\mathcal{D}_{ij,H}^{\text{harm}}$ contains descendant versions whose harmful exposure is verified within $H$ future episodes, and $e(d) \in \{1, \ldots, H\}$ is the future episode of $d$'s first verified harmful exposure. Each version contributes once. Discounting by episode rather than by lineage generation ties the target to elapsed time, and the target measures verified burden within this window. Candidates with incomplete audits retain an unresolved label. Input features include semantic rank, age, version distance, current-state compatibility, recorded outcome, source authority, precondition coverage, conflicts, and parent summaries; gold future outcomes and descendant labels serve only as training targets.

The read model uses a shared encoder to predict $(\hat{u}_j, \hat{p}_j^h, \hat{p}_j^b, \hat{g}_j)$ for each candidate $m_j$: sigmoid heads predict the harm and benefit probabilities, and a softplus head keeps the predicted burden nonnegative. An order-sensitive block head directly estimates the effect of the assembled retrieval block, accounting for interactions among items. The read loss is

$$\begin{aligned}\mathcal{L}_R = \; & \text{Huber}(\hat{u}, u) + \lambda_1 \,\text{BCE}(\hat{p}^h, h) + \lambda_2 \,\text{BCE}(\hat{p}^b, b) \\ & + \lambda_3 \,\text{Huber}(\log(1+\hat{g}), \log(1+g)) + \lambda_4 \mathcal{L}_{\text{set}},\end{aligned} \tag{8}$$

where the set loss $\mathcal{L}_{\text{set}}$ trains the block head on actual contrasts between assembled blocks and empty memory rather than on sums of item scores, because item utilities are context-dependent. Unresolved descendant targets are masked, and examples are weighted by stream.

The write model is evaluated after task execution and before the proposed memory update, using only information available at that point. It is trained on its own store-versus-withhold interventions on proposed record versions, revision comparisons, and subsequent descendant tests, with the immediate task outcome held fixed (Section B).

### 5.2 Read selection and write routing

The controller greedily assembles candidates under a token budget using the score

$$s_R(m_j \mid x, R) = \hat{u}_j - \lambda_h \hat{p}_j^h - \lambda_g \hat{g}_j - \lambda_r \,\text{Redundancy}(m_j, R), \tag{9}$$

where $x$ is the current decision context and $R$ is the partially assembled block. Candidate-head outputs are fixed at the checkpoint; only the redundancy penalty depends on $R$. After greedy selection, a frozen set-level gate evaluates the assembled block with the block head and exposes it only if the estimate exceeds $\tau_R$; otherwise, the agent acts without memory.

After task execution, the write gate routes each proposed record to one of four actions (Figure 2). A record is *stored* only if its predicted utility exceeds a fixed development threshold and its predicted reuse and descendant risk falls below $\tau_W$. A verified supersession triggers *revision* and re-scoring, and the revised record retains the original provenance. Unresolved but potentially useful evidence is *quarantined*, keeping its provenance while being excluded from task retrieval, and unsupported or dominated records are *discarded*.

At deployment, the controller scores candidates and routes writes around a single task execution. Table 4 reports the resulting online overhead; offline supervision is accounted for separately.

## 6 Experiments

**Comparators and operating points.** We compare dense top-$k$ retrieval, CMI (Srivastava, 2026), MeClear (Yang et al., 2026), read-only and joint LoopGuard (hereafter read-only and joint control), and an oracle that retrieves only valid memories. CMI evaluates the influence of candidate memories; MeClear combines attribution with query-scoped clearance while retaining the persistent bank. Read-only LoopGuard applies the learned read gate with ordinary writing. The oracle uses privileged validity information and serves as a diagnostic reference. For each method, we select an operating point on the calibration split that targets a PMR of 12% and freeze it before test evaluation. We report test NMR with the attained test PMR and count additional intervention calls separately from ordinary task execution.

**Uncertainty.** Brackets denote pointwise 95% stream-clustered bootstrap confidence intervals from 4,000 resamples. For between-method differences, each resample uses the same stream IDs across methods and retains every checkpoint within a sampled stream. Calibrated policies remain fixed during resampling. Rate differences are reported in percentage points (pp).

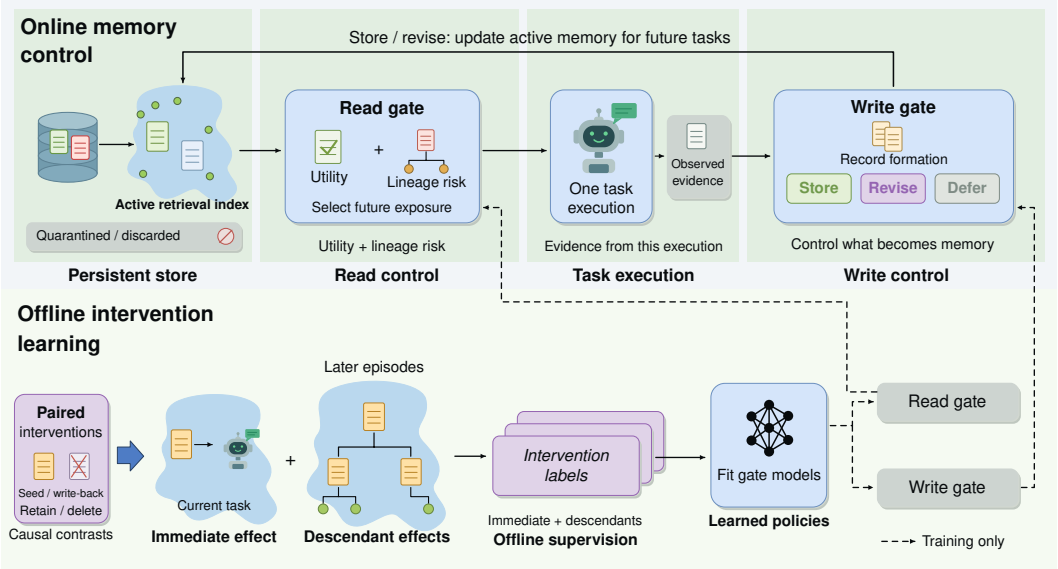


Figure 2: **LOOPGUARD controls both retrieval and memory formation.** The read gate selects what to expose using estimated utility and descendant burden. After a single task execution, the write gate routes proposed records; revisions are re-scored before activation, and quarantined or discarded records stay outside retrieval. Offline item-deletion and store-versus-withhold interventions provide separate read and write supervision, including finite-horizon descendant targets. Dashed arrows denote training-only paths; deployment requires no counterfactual reader calls.

Table 1: **Controlled comparison at similar positive transfer.** NMR and PMR are computed over all eligible paired block probes within each domain and then averaged with equal weight across domains. Brackets denote 95% stream-clustered bootstrap intervals.

| Method | NMR (%) ↓ | PMR (%) |
|---|---|---|
| Dense top-$k$ | 9.5 [9.1, 9.9] | 12.0 [11.8, 12.2] |
| CMI (Srivastava, 2026) | 5.8 [5.5, 6.1] | 12.1 [11.9, 12.3] |
| MeClear (Yang et al., 2026) | 5.3 [5.0, 5.6] | 11.9 [11.7, 12.1] |
| LOOPGUARD read-only | 4.7 [4.4, 4.9] | 12.1 [11.9, 12.3] |
| LOOPGUARD joint | **3.3** [3.1, 3.5] | 12.2 [12.0, 12.4] |
| Oracle valid-only | 0.8 [0.7, 0.9] | 12.3 [12.1, 12.5] |

PMR values are attained test rates. Empty retrieved blocks remain in the denominator and contribute no transitions under the replay coupling. The oracle is a privileged reference at the displayed operating point.

### 6.1 METHOD COMPARISON AT SIMILAR POSITIVE TRANSFER

The controlled comparison uses 2,160 test streams, each with 24 paired target checkpoints. Rates are computed within each domain and then averaged with equal weight across domains. Joint control attains an NMR of 3.3%, compared with 5.8% for CMI, 5.3% for MeClear, and 9.5% for dense retrieval (Table 1). Its paired NMR differences from CMI and MeClear are $-2.5$ pp $[-2.8, -2.1]$ and $-2.0$ pp $[-2.4, -1.7]$, respectively, while the attained test PMRs remain close: 12.2% for joint control, 12.1% for CMI, and 11.9% for MeClear. Joint control therefore reduces harmful reversals at comparable positive transfer.

### 6.2 ORGANIC HARM WITHOUT INJECTED CANDIDATES

The organic evaluation uses dense top-$k$ retrieval with ordinary writing at its native operating point and injects no candidate memories. It contains 1,620 test streams and 32,400 eligible paired natural-retrieval probes, balanced across the three domains (Table 2).

Table 2: **Negative memory retrieval on organic streams.** Evaluation uses dense top-$k$ retrieval with ordinary writing and no injected candidates. NMR is the percentage of eligible paired probes that succeed without the current retrieved block and fail with it; probes receive equal weight within each row. Brackets denote 95% stream-clustered bootstrap intervals.

| Domain | Test streams | Paired probes | NMR (%) ↓ |
|---|---|---|---|
| All domains | 1,620 | 32,400 | 4.5 [4.2, 4.8] |
| Embodied control | 540 | 10,800 | 3.7 [3.2, 4.1] |
| Web workflows | 540 | 10,800 | 5.3 [4.8, 5.8] |
| Evolving personal state | 540 | 10,800 | 4.5 [4.0, 5.0] |

Table 3: **Controlled write-back interaction by domain.** WAC is the write-on minus write-off contrast in the seed's longitudinal effect, averaged with equal weight over the 720 seed interventions in each domain. Units are discounted success counts per intervention over $H = 8$ future episodes with $\gamma = 0.90$. Brackets denote 95% stream-clustered bootstrap intervals.

| Controlled domain | Dense WAC | Joint WAC |
|---|---|---|
| Embodied control | $-0.28$ [$-0.32$, $-0.23$] | $-0.02$ [$-0.06$, 0.02] |
| Web workflows | $-0.43$ [$-0.47$, $-0.39$] | $-0.06$ [$-0.10$, $-0.02$] |
| Evolving personal state | $-0.28$ [$-0.32$, $-0.24$] | $-0.06$ [$-0.10$, $-0.02$] |

The mean paired block effect is 2.6 pp [2.2, 2.9], whereas NMR is 4.5% [4.2, 4.8]. Among probes that succeed without the current retrieved block, conditional harm is 6.9% [6.5, 7.4]. Average benefit thus coexists with memory-induced failures on individual paired probes. Web workflows have the highest domain-level NMR point estimate (5.3%).

The length-matched placebo probe has an NMR of 1.2% [1.1, 1.3] and a mean effect of 0.0 pp [$-0.1$, 0.2]. Placebo probes show fewer harmful reversals than retrieved-memory probes.

### 6.3 Write-back effects and harmful descendants

The longitudinal evaluation follows one prespecified eligible seed intervention from each controlled test stream, yielding 720 interventions per domain. Seed checkpoints are selected before intervention outcomes are observed. Dense retrieval and joint control share the same checkpoint cohort and follow-up window ($H = 8$, $\gamma = 0.90$).

Under dense retrieval, WAC is negative in all three domains and largest in magnitude for web workflows (Table 3): enabling memory updates makes the seed's longitudinal effect more adverse than freezing external memory. Joint control moves all three point estimates toward zero. Its interval includes zero for embodied control but remains negative for web workflows and evolving personal state. These estimates describe the write-back interaction over the eight-episode window.

**Deletion-verified descendants.** Because WAC combines the effects of harmful additions and of changes in useful adaptation, a record-level deletion audit tests whether a particular induced version harms later behavior. Before examining descendant outcomes, we randomly select 1,000 treated controlled test streams and follow the prespecified seed intervention in each. Under dense retrieval with ordinary writing, the audit identifies 136 branch-differential candidate records. All 136 undergo deletion verification, and 37 are harmful, a verification fraction of 27.2%. Over the seed episode and eight future episodes, this amounts to 136 candidates and 37 verified harmful descendants per 1,000 treated streams. The denominator includes audited streams without immediate seed NMR.

### 6.4 Read-only versus joint control

Joint control lowers NMR from 4.7% for the read-only variant to 3.3%, a paired difference of $-1.4$ pp [$-1.7$, $-1.1$], at attained PMRs of 12.1% and 12.2%, respectively. Both variants are evaluated on the same test streams with the paired bootstrap procedure above and differ in whether the learned controller also governs writing. NMR and PMR measure the current block's effect relative to each policy's own matched no-block reference; for binary outcomes, their difference is the

Table 4: **Deployment cost in the evaluated implementations.** Extra reader calls are additional counterfactual invocations per request. Token ratios cover online reader, writer, and revision input and output tokens, relative to dense retrieval. Ordinary task execution is included in every row; offline labeling and training are excluded.

| Method | Extra reader calls ↓ | Token ratio ↓ |
|---|---|---|
| Dense top-$k$ | 0 | 1.00× |
| CMI (Srivastava, 2026) | 3.0 | 3.12× |
| MeClear (Yang et al., 2026) | 7.4 | 6.83× |
| LOOPGUARD joint | 0 | 1.062× |

within-policy block effect in Equation (3). Because each policy is evaluated under the store it induces, the comparison measures the contribution of write control as a whole, in addition to retrieval control.

### 6.5 DEPLOYMENT COST AND OFFLINE SUPERVISION

Joint control adds no counterfactual reader calls at deployment and incurs a 6.2% token overhead relative to dense retrieval, versus token ratios of 3.12× for CMI and 6.83× for MeClear (Table 4). Token accounting sums online input and output tokens for reader, writer, and revision calls, including counterfactual calls where used; controller-encoder and retrieval computation fall outside this metric.

Offline supervision uses 24,000 paired interventions. Immediate labels require arm evaluations, descendant labels require follow-up trajectories and deletion audits, and write labels require store-versus-withhold branches after the current task outcome. The intervention count thus reflects the scale of offline supervision, including follow-up trajectories and record-level audits; these labeling and training costs are separate from the deployment ratios.

## 7 DISCUSSION AND CONCLUSION

Memory that helps an agent on average can still cause it to fail on individual decisions. Coupled exposure, write-on/write-off follow-up, and record deletion make these effects separately measurable. In NMR-LOOP, joint read–write control lowers NMR relative to dense retrieval, CMI, MeClear, and its read-only variant at positive-transfer rates near 12%, and it moves estimated write-back interactions toward zero over eight future episodes without counterfactual reader calls at deployment.

NMR complements the mean block effect by identifying harmful reversals at individual checkpoints. The evaluation covers memory-optional tasks under the specified replay coupling with an eight-episode follow-up window. Longer horizons and interventions on interacting record sets are natural extensions. More broadly, evaluating retrieval together with memory formation offers a practical path toward more reliable long-horizon agents.

## AI USE STATEMENT

In this work, we used generative AI tools (large language models) solely for language polishing, including improving grammar, clarity, and readability. These tools were not used to generate substantive content or to perform any research tasks, including developing methods or theoretical frameworks, formulating claims or hypotheses, constructing proofs, designing experiments, generating or processing data, analyzing or interpreting results, writing code, searching the literature, translating text, or preparing figures. All AI-assisted revisions were reviewed by the authors to ensure that they preserved the intended meaning and technical accuracy. We take full responsibility for the final content of this work, including all claims, results, and references.

## REFERENCES

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. URL https://arxiv.org/abs/2605.06527.

Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. AgentPoison: Red-teaming LLM agents via poisoning memory or knowledge bases. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Shen Dong, Shaochen Xu, Pengfei He, Yige Li, Jiliang Tang, Tianming Liu, Hui Liu, and Zhen Xiang. Memory injection attacks on LLM agents via query-only interaction. *arXiv preprint arXiv:2503.03704*, 2025. URL https://arxiv.org/abs/2503.03704.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, David Vazquez, Nicolas Chapados, and Alexandre Lacoste. WorkArena: How capable are web agents at solving common knowledge work tasks? In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Xinyi Fan, Miri Liu, Ruozhen Yang, Siru Ouyang, and Jiawei Han. Can agent memory systems track evolving state? *arXiv preprint arXiv:2608.19652*, 2026. URL https://arxiv.org/abs/2608.19652.

Xinyu Guan, Qianyang Zhao, and Yuming Deng. Decision-aware memory cards: Counterfactual-inspired context selection and compression for tool-using LLM agents. *arXiv preprint arXiv:2606.08151*, 2026. URL https://arxiv.org/abs/2606.08151.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in LLM agents via incremental multi-turn interactions. *arXiv preprint arXiv:2507.05257*, 2025. URL https://arxiv.org/abs/2507.05257.

Mehmet Iscan. Learning when to remember: Risk-sensitive contextual bandits for abstention-aware memory retrieval in LLM-based coding agents. *arXiv preprint arXiv:2604.27283*, 2026. URL https://arxiv.org/abs/2604.27283.

Hyunji Lee, Justin Chih-Yao Chen, Joykirat Singh, Zaid Khan, Elias Stengel-Eskin, and Mohit Bansal. MINTEval: Evaluating memory under multi-target interference in long-horizon agent systems. *arXiv preprint arXiv:2605.18565*, 2026. URL https://arxiv.org/abs/2605.18565.

Zewen Liu. Memory contagion: Cross-temporal propagation of evaluator bias via agent memory. *arXiv preprint arXiv:2606.23195*, 2026. URL https://arxiv.org/abs/2606.23195v2.

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*, 2024.

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

Joon Sung Park, Joseph C. O'Brien, Carrie J. 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*, 2023.

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, 2023.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021.

Saksham Sahai Srivastava. Causal intervention-based memory selection for long-horizon LLM agents. *arXiv preprint arXiv:2605.17641*, 2026. URL https://arxiv.org/abs/2605.17641.

Theodore R. Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. Cognitive architectures for language agents. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2309.02427.

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. *Transactions on Machine Learning Research*, 2023.

Yian Wang, Agam Goyal, Yuen Chen, and Hari Sundaram. State contamination in memory-augmented LLM agents. *arXiv preprint arXiv:2605.16746*, 2026. URL https://arxiv.org/abs/2605.16746.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations*, 2025.

Di Wu, Zixiang Ji, Asmi Kawatkar, Bryan Kwan, Jia-Chen Gu, Nanyun Peng, and Kai-Wei Chang. LongMemEval-V2: Evaluating long-term agent memory toward experienced colleagues. *arXiv preprint arXiv:2605.12493*, 2026. URL https://arxiv.org/abs/2605.12493.

Jinghan Xu, Yiyong Xiao, Wanru Shao, Hankai Liu, and Xinjin Li. Memory provenance laundering in LLM agents: A non-amplification firewall for persistent memory. *arXiv preprint arXiv:2607.29167*, 2026. URL https://arxiv.org/abs/2607.29167.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Boyu Yang, Jiazheng Sun, Zilong Lu, Zhi Qiu, Xin Peng, and Jun Zheng. MeClear: Cooperative game-theoretic attribution and risk-aware memory clearance for long-horizon LLM agents. *arXiv preprint arXiv:2609.09115*, 2026. URL https://arxiv.org/abs/2609.09115v1.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. *International Conference on Learning Representations*, 2024.

# A Evaluation Settings and Protocol Details

Table 5: Evaluation scale and analysis settings. Program totals and held-out test cohorts are listed separately.

| Setting | Value or scope |
|---|---|
| Controlled program | 3,600 streams; 24 paired target checkpoints per stream; 86,400 pairs across the full program |
| Controlled test | 2,160 streams; 51,840 paired target checkpoints; 720 test streams per domain |
| Organic program | 2,700 unseeded streams |
| Organic test | 1,620 streams; 32,400 paired probes; 540 streams and 10,800 probes per domain |
| Domains | Embodied control, sandboxed web workflows, evolving personal state |
| Controlled mechanisms | Stale state, mismatched preconditions, unreliable outcome provenance, write-back inversion |
| Follow-up | $H = 8$ future episodes; discount $\gamma = 0.90$ |
| Comparison regime | PMR approximately 0.12 |
| Uncertainty | 4,000 stream-clustered bootstrap resamples; 95% intervals |
| Training supervision | 24,000 paired interventions; a pair can require multiple model calls |

**Replay requirements.** A replay checkpoint must specify the observation, goal, environment and working state, store snapshot, ordered candidates, retriever state, model and prompt configuration, and exogenous random state. Component-indexed random tapes prevent extra reader calls in one branch from shifting the randomness of the writer or the environment; a suitable index combines stream, component, episode, call, and token or transition indices. Under the specified replay protocol, shared uniforms must be mapped through each branch's own token distribution, while tools respond to that branch's own actions. Replay from equal states must reproduce the same result while preserving each arm's marginal sampling law.

For binary outcomes with marginal success probabilities $p_0$ and $p_1$ under the two arms, the pathwise harm rate under any coupling satisfies the Fréchet–Hoeffding bounds

$$\max(0, p_0 - p_1) \le \mathbb{P}(Y^0 = 1, Y^1 = 0) \le \min(p_0, 1 - p_1). \tag{10}$$

The width of this range shows why the coupling matters: repeated non-replayable API calls estimate marginal mean effects, whereas paired NMR relies on the joint replay law defined above.

**Seed and descendant interventions.** The protocol selects eligible checkpoints independently of observed harm, fixes background order and formatting, and constructs the assigned exposure blocks. It executes the seed endpoint before branching into write-on and write-off arms; write-off blocks append, revision, consolidation, and deletion of external memory. Future exogenous tasks remain coupled for $H$ episodes, while post-treatment states may diverge. Retain–delete descendant tests run on shadow branches and do not mutate the evaluated store. Unconditional seed-level counts include assignments with no immediate seed harm.

Every append or content-changing revision requires an immutable version ID; each retrieval requires an activation ID. A candidate descendant requires treatment-induced production under a frozen cross-branch matching rule and a provenance path to the seed. Unique-parent analysis additionally requires a resolved parent intervention at the child-producing write; ambiguous multi-parent paths are excluded from that tree analysis. Generations count parent–child writes, whereas the episode index records elapsed follow-up. Nonharmful intermediaries remain in the lineage graph.

**Expected-effect audits.** Expected-effect verification uses repeated trials that are independent of discovery. For independent paired differences $D_r \in [-1, 1]$, $r = 1, \ldots, S$, Hoeffding's inequality gives the one-sided upper confidence bound $U^I = \min\{1, \bar{D} + \sqrt{2\log(1/\delta_d)/S}\}$, and the audit supports negative expected influence only if $U^I < -\epsilon$. A fixed family of $N_{\text{audit}}$ tests can use the Bonferroni allocation $\delta_d = \delta_{\text{audit}}/N_{\text{audit}}$; adaptive testing requires valid sequential control. Untested and censored versions remain unresolved. These repeated-trial audits target expected influence, whereas the reported pathwise audits evaluate single-record effects under the specified coupling; set interventions would extend both to redundant or synergistic records.

## B CONTROLLER SUPERVISION AND COMPUTE

**Write-action supervision.** A store-versus-withhold intervention starts after the current task outcome is fixed and holds the available execution evidence constant. Follow-up then supplies utility, harmful-reuse, and descendant targets. Revision compares a replacement against the superseded version. Quarantine retains provenance while excluding the content from task retrieval; release from quarantine requires corroboration and re-scoring. Gold future outcomes serve only as supervision targets.

The read loss in Equation (8) combines utility regression, harm and benefit classification, descendant regression, and an assembled-block target. Unresolved descendant targets are masked during fitting and kept distinct from verified zero-burden targets. Features, loss weights, and routing rules are fixed before test evaluation. Because item utilities are context-dependent, the set head is trained on actual assembled-block contrasts rather than on sums of item scores.

**Compute accounting.** The offline workload comprises seed pairs, follow-up trajectories, descendant audits, write-action comparisons, and model fitting. The deployment ratios in Table 4 cover online reader, writer, and revision tokens. The 24,000-pair supervision count reports intervention units, each of which can involve multiple model calls.