VeriMem 2.0: Verification-Gated Declarative Memory for LLM Agents — Prevention at Write, Not Detection at Use

Anonymous authors
Paper under double-blind review

Abstract

LLM agents persist declarative memory—facts, rules, preferences—across episodes, but existing write policies admit entries on the model's own judgment (LLM-judge scoring; success filters), and both write hallucinations: under matched extractors and retrievers, ungated writes contaminate 12.8% and LLM-judged writes 17.9% of audited location claims on ScienceWorld (pooled over three seeds), while the best LLM agent answers memory-validity probes correctly only 55.2% of the time. We present VeriMem 2.0, a memory admission gate that requires each declarative write to carry an executable evidence predicate—a decidable check over a logged evidence corpus—and to abstain when no predicate is expressible. In a controlled comparison where only the gate varies, VeriMem reduces contamination to 1.5% on ScienceWorld (0.0% on ALFWorld) and sustains this under adversarial injection of up to 80 forged entries (dose 0.4), landing zero of them on both environments while ungated and success-filtered stores absorb 100%. It holds success rate under attack while ungated memory degrades on ALFWorld (0.35 vs. 0.15). Ablations attribute the entire defense to the executable predicate: removing it collapses VeriMem to no-gate and admits 32/40 forgeries, while removing evidence-log retraction changes nothing (0/40). The gate costs clean success rate within noise (0.140 vs. 0.113 ScienceWorld; 0.22 vs. 0.28 ALFWorld) and buys a 3–5× smaller, 100%-verified store; its abstention (35.2% ALFWorld, 32.1% ScienceWorld) is a tunable fallback ablated directly. All experiments use a 7–8B open-source backbone (Qwen3-8B). Code, prompts, and audit tooling will be released.

1 Introduction

LLM agents increasingly persist memory across episodes: facts about the environment, rules about its mechanics, and preferences about the user. The write path of every prominent memory system relies on the model's own judgment. LLM-judge systems (Generative Agents (Park et al., 2023), Mem0 (Chhikara et al., 2025), A-MEM (Xu et al., 2025), ReasoningBank (Ouyang et al., 2025), MemoryOS (Kang et al., 2025)) ask the model to score or filter candidates, as do self-edited write paths (MemGPT (Packer et al., 2023), MemoryBank (Zhong et al., 2024)); trajectory-success filters (AWM (Wang et al., 2024)) admit everything extracted from winning episodes; and rule-induction systems (ExpeL (Zhao et al., 2024), AutoManual (Chen et al., 2024a)) validate rules only behaviorally, never by checking an individual entry's assertion against recorded evidence. The consequence is measurable: the best evaluated LLM agent answers only 55.2% of memory-validity probes correctly (STALE (Chao et al., 2026)), memory-poisoning attacks reach >80% success at <0.1% poisoning rates (AgentPoison (Chen et al., 2024b)), and forged entries are more dangerous to stronger agents (PoEM (Rahman & Kim, 2026): 98–100% on GPT-4o vs. 44% on Llama-3.1-8B).

Procedural artifacts already have a solution: code skills are admitted only after execution (Voyager (Wang et al., 2023)), and AutoRefine (Qiu et al., 2026) compiles trajectories into replay-validated artifacts. Declarative memory has no analogous per-entry gate: entries

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.

assert what is true, yet no prior system evaluates a decidable check of that assertion against a recorded evidence log before the write lands. We close this gap.

VeriMem 2.0 (§3) gates every declarative write on an executable evidence predicate: a decidable function over an evidence log of recorded observations and action outcomes. A write is accepted only if its predicate verifies against the log, rejected if it falsifies, and abstained—not silently admitted—when no predicate is expressible.

Claims. (1) First admission gate for declarative agent memory in which admission is decided by evaluating decidable, re-runnable predicates over a recorded evidence log, with abstention—vs. LLM-judge scoring (Wang et al., 2026), training-time verifiers (Sun et al., 2026; Yang et al., 2026), transactional provenance (Li et al., 2026), origin-bound attestation (Rahman & Kim, 2026; Louck, 2026), and replay-validated artifacts (Wang et al., 2023; Qiu et al., 2026). Attestation verifies that an action happened and gates use; we verify that an assertion is true and gate admission. (2) A controlled, extractor-and-retriever-matched quantification of contamination dynamics across five gate families. (3) Evidence that prevention-at-write yields cleaner stores than detection-at-use under adversarial pollution on a small open-source backbone (store composition separates unambiguously; SR effects are directional at our n).

2 Related Work

LLM-judged memory writes. Mem0 (Chhikara et al., 2025), A-MEM (Xu et al., 2025), MemoryOS (Kang et al., 2025), and Reflexion (Shinn et al., 2023) score candidates with the model itself; MemGuard (Wang et al., 2026) is the closest admission-time foil, persisting verifier score-tokens as metadata. STALE (Chao et al., 2026) bounds this family: the best evaluated agent answers memory-validity probes correctly only 55.2% of the time.

Success-filtered and rule-inducing writes. AWM (Wang et al., 2024) admits everything from winning trajectories; ExpeL (Zhao et al., 2024), AutoManual (Chen et al., 2024a), and AutoGuide (Fu et al., 2024) validate rules behaviorally, with no per-entry decidable verdict and no abstain. Success does not entail truth: a winning trajectory still hallucinates environment facts (we measure this in §5.1; Honest Lying (Dixit et al., 2026) documents the same confabulation).

Use-time detection and verified artifacts. SeaKR (Yao et al., 2025), Self-RAG (Asai et al., 2024), and CRAG (Yan et al., 2024) detect unreliable retrievals at use time—complementary, not preventive, since a contaminated entry remains retrievable. Voyager (Wang et al., 2023), Agent Skill Induction (Wang et al., 2025), and SkillWeaver (Zheng et al., 2025) gate procedural skills by execution; AutoRefine (Qiu et al., 2026) compiles trajectories into typed artifacts admitted through contract and replay gates. These validate artifacts on new episodes, whereas VeriMem evaluates a decidable predicate per entry against already-logged evidence (accept/reject/abstain). MSCE (Tang et al., 2026), EDV (Zhu et al., 2026), and NeuSymMS (Sultan et al., 2026) organize how experience accumulates; RAPTOR (Sarthi et al., 2024), MemTree (Rezazadeh et al., 2024), and HippoRAG (Gutiérrez et al., 2025) organize how stored memory is indexed; VeriMem governs whether an entry is admitted, upstream of any such layer.

Transactional memory and poisoning. MemTX (Li et al., 2026) gates commits at write time on protocol conformance, not semantic truth (no predicate, no abstain); VerMem (Sun et al., 2026) trains verifiers as RL reward, not runtime gates. AgentPoison (Chen et al., 2024b), MINJA (Dong et al., 2025), and MPBench (Dash et al., 2026) systematize the write-channel attack surface; A-MemGuard (Wei et al., 2025) and SENTINEL defend at use time or by signing; PoEM (Rahman & Kim, 2026) shows wording-level blacklists fall to a single paraphrase and verifies via an HMAC-chained ledger of executed safety steps, not memory content; TRUSTMEM (Yang et al., 2026) trains a learned verifier (soft gate, no abstain); Louck (Louck, 2026) enforces origin-bound labels (integrity of who wrote, not truth); Memory Contagion (Liu, 2026) documents cross-temporal bias spread. VeriMem instead evaluates decidable predicates over the agent's own observation log—attestation proves an action happened, our gate checks whether an assertion is true—the two compose.

Algorithm 1 VeriMem write-time admission gate.

Require: candidate entry (t, c); evidence log $\mathcal{E}$
Ensure: decision $d \in \{\text{accept}, \text{reject}, \text{abstain}\}$

$\phi \leftarrow \text{select_predicate}(t, c)$
if $\phi = \emptyset$ then
    return abstain ▷ no predicate expressible
end if
$v \leftarrow \text{evaluate}(\phi, \mathcal{E})$
if $v = \text{true}$ then
    store (t, c) verified; return accept
else if $v = \text{false}$ then
    return reject
else
    return abstain ▷ undecidable
end if

No prior defense evaluates decidable predicates over a recorded evidence log as an admission criterion, with abstention when no predicate is expressible.

3 Method: VeriMem 2.0

3.1 Memory schema and predicates

Entries are $(\textit{type}, \textit{content}, \textit{predicate}, \textit{evidence})$ with $\textit{type} \in \{\text{fact}, \text{rule}, \text{preference}\}$; a fact like "the beaker is on the counter" carries $\text{loc}(\text{beaker}, \text{counter})$ with its evidence step. Predicates are decidable functions over an evidence log $\mathcal{E}$—the corpus of raw observations and action outcomes recorded during episodes:

$$\text{loc}(o, r) \equiv \exists e \in \mathcal{E} : \; e \Rightarrow \text{“on/in } r \text{ you see } \ldots o \ldots \text{”} \tag{1}$$

$$\text{not_loc}(o, r) \equiv \text{the full listing of } r \text{ in } \mathcal{E} \text{ excludes } o \tag{2}$$

$$\text{affords}(o, a) \equiv \text{an admissible-action or feedback record in } \mathcal{E} \text{ shows } a(o) \tag{3}$$

$$\text{text}(c) \equiv c \text{ appears verbatim in some observation of } \mathcal{E} \text{ (action echoes excluded)} \tag{4}$$

The parser handles ScienceWorld's listing formats and invalidates stale loc facts when objects move. Undecidable predicates (e.g. a mechanic claim whose action–outcome pair the log does not track) route to abstain, never to silent rejection.

3.2 Gate decision and retraction

Algorithm 1 summarizes the decision: accept on a verifying predicate (stored with lineage, marked verified), reject on a falsifying one, and abstain when no predicate is expressible. Abstention fallback is configurable (reject / accept-unverified / llm-judge); the headline results use reject, and §5.5 reports all three. Unverifiable $\neq$ false—the key difference from scoring gates. The evidence layer also performs stale-location invalidation: when the agent's own move action is observed (move X to Y), loc facts placing X elsewhere are removed from the evidence log. §5.5 ablates this (verimem –retract).

3.3 Baseline gates (only the gate varies)

All gates share the extractor, retriever, and agent. (1) no-gate: write everything. (2) success-filter: write only from successful episodes. (3) LLM-judge: model scores each candidate (MemGuard-style). (4) use-time-detect: write everything, then at retrieval time an LLM drops retrieved entries that contradict the current observation. (5) VeriMem. The LLM-judge baseline is deliberately a weak instantiation—the same 7–8B backbone self-assesses each candidate from a 15-observation window—bounding judgment-only gating in the small-model regime.

Table 1: E1: gate comparison with matched extractor, retriever, prompts, and harness (5 sessions × 10 episodes per environment; only the gate varies). SR = episode success rate; contam. = audited-false rate over location-claim entries; mem = final active entries (mean over the pooled seeds for ScienceWorld; seed-0 value for ALFWorld). ScienceWorld numbers pool three seeds (n=150/gate, final harness); ALFWorld numbers are seed-0.

	ScienceWorld (3-seed pooled)			ALFWorld (seed 0)		
Gate	SR	contam.	mem	SR	contam.	mem
No gate	0.113	.128	282	0.28	.023	336
Success filter	0.087	.185	31	0.30	.007	106
LLM judge	0.173	.179	89	0.34	.000	140
Use-time detect	0.120	.169	272	0.36	.013	326
VeriMem	0.140	.015	73	0.22	.000	95

4 Experimental Setup

Environments. ScienceWorld (Wang et al., 2022) (10 task types × 4 variations; gold paths 10–39 steps) and ALFWorld (Shridhar et al., 2021) (6 task types, valid_seen). Backbone. Qwen3-8B (Yang et al., 2025) (vLLM, bf16, chat template with pre-filled empty think block). Protocol. E1: 5 gates × 5 sessions × 10 episodes, memory persists within a gate across sessions; max_steps = 60 on ScienceWorld and 30 on ALFWorld. Contamination is audited per-episode against the environment's object trees, counting a claim false only if the object is known to the house and the location never held at any cited evidence step. The auditor checks direct-parent containment only, whereas the write gate's location predicate also accepts transitive support—distinct code paths, so a gate cannot pass the audit by construction. The audit denominator is location-claim fact entries only; rules, preferences, and non-location facts are unauditable and excluded. Abstention rates are computed over all gate decisions. E2: adversarial injection at rates $\{0, 0.1, 0.2, 0.4\}$, 20 episodes per dose, fresh store per (gate, rate); the attacker controls extraction output (forged success metadata) but writes still pass through the gate under test. Under attack we report poison-aware contamination: (natural false + injected false claims) over all audited claims, so a landed forgery always raises the rate. E3: defense comparison at fixed dose. E4: ablations (fallback modes; predicate-check-off and retraction-off on the final code).

5 Results

5.1 E1: Write quality and cross-session evolution

On ScienceWorld (5 sessions × 10 episodes per seed, pooled over three seeds, n=150 per gate, Qwen3-8B), the capability floor compresses SR differences (no-gate 0.113, success-filter 0.087, LLM-judge 0.173, use-time-detect 0.120, VeriMem 0.140; VeriMem wins all three seeds against no-gate, McNemar p=0.57), but contamination separates unambiguously: no-gate audits false on 12.8% of location claims (50/392), use-time detection on 16.9% (70/415), VeriMem on 1.5% (4/268)—and every false flag is the same auditor-strictness artifact: the only unique false claim is "the metal pot is in the kitchen," transitively true (pot ∈ sink, sink ∈ kitchen), which the auditor (direct parents only) rejects but the gate's transitive check verifies. The 4 flags come from 2 unique entries (seeds 0 and 2), so VeriMem's numbers are upper bounds. The LLM-judge gate abstains on 62.2% of candidates yet still audits false on 17.9% (21/117)—judgment does not substitute for verification.

Figure 1 shows the contamination separation per session.

5.2 E1-ALF: ALFWorld replication and an environment confound

On ALFWorld (valid_seen, 5 sessions × 10 episodes, seed 0), all five gates land in a narrow SR band: no-gate 0.28, success-filter 0.30, LLM-judge 0.34, use-time-detect 0.36, VeriMem 0.22 (pairwise continuity-corrected McNemar vs. VeriMem: use-time-detect p=0.023, 7 vs. 0

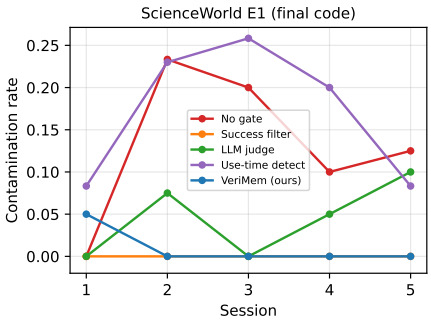

Figure 1: Cross-session contamination on ScienceWorld (E1; 5 sessions × 10 episodes per seed, three seeds, n=150 per gate). Contamination rate of the audited location-claim store per session for the five gate conditions; only the gate varies. Pooled over sessions, no-gate memory audits false on 12.8% of audited location claims, use-time detection on 16.9%, and VeriMem on 1.5% — an upper bound whose flags all trace to a single transitively true claim the auditor (direct parents only) rejects.

discordant episodes, treated as suggestive only). Contamination separates cleanly: no-gate 2.3% and use-time-detect 1.3% audited-false vs. 0.0% for both LLM-judge and VeriMem—but the LLM-judge gate's clean audit is unaudited, not clean: it abstained on 35.6% and admitted 140 unverified entries, while VeriMem admitted 95 verified entries. Success-filter memory (106 entries) still audits false on 0.7% of claims.

Pooling three seeds (n=150 per gate), VeriMem reaches 0.287 (95% Wilson CI 0.220–0.364) vs. no-gate 0.260 (0.196–0.336)—a +2.7-point pooled gain with zero audited contamination (0.0% vs. 3.1%) and a 3.4× smaller store (291 vs. 982 entries, all verified vs. none). The The direction holds on two of three seeds; the episode-level McNemar test is underpowered at this n (p=0.61), so we report the SR gain as directional and leave its magnitude to a seed-extended evaluation that would settle it; the paired design (same episodes, extractor, retriever; only the gate differs) attributes the gain to admission quality rather than extraction variance.

The narrow SR band has a structural cause: ALFWorld's synthesized goal text omits the slicing precondition its PDDL goals contain, so the slicing family collapses to 0–1/8 across all gates—a floor effect compressing gate differences. Appending a harness-level slicing hint (identically for all gates) restores measurability (slicing 0/8→2/8, no-gate, seed 0); we treat cross-gate comparisons as hint-invariant by design. We release the hint patch and recommend environment-fidelity audits before comparing memory policies.

Figure 2 reports the per-session contamination behind these numbers.

5.3 E2: Pollution dose–response

The attack succeeds against ungated and success-filtered stores completely: at every dose 100% of injected entries land in no-gate and success-filter memory (10/10, 20/20, 40/40 on both environments), as does use-time detection, which runs at retrieval, after the write. The LLM-judge gate rejects these templates (0 landed at every dose), substantially by abstention rather than judgment (§5.2): its clean slate under attack is bought with unverified judgment-admitted entries in ordinary operation, and its own writes hold poison-aware contamination at 5.7–17.1% at every dose—a judge is not a verification guarantee. VeriMem lands zero injected entries at every dose on both environments—location-fact forgeries fail their predicate on the empty evidence log, and predicate-free rules/preferences fall to abstention-then-reject—and its store contamination stays at 0.000 across all doses, vs. up to 0.497 (no-gate, ScienceWorld), 0.466 (use-time-detect, ScienceWorld), and 0.762 (success-filter, ALFWorld). SR dose–response on ScienceWorld is bounded by the capability floor (§5.1), where we therefore report the store-composition evidence and leave SR dose–response to

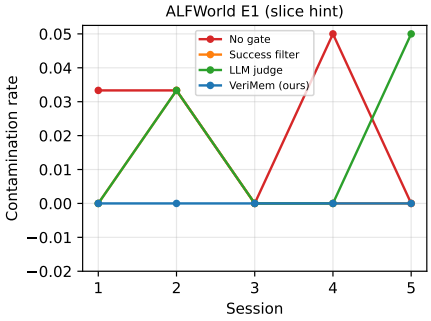

Figure 2: Cross-session contamination on ALFWorld with the harness-level slicing hint enabled (E1-ALF, seed 0). Contamination rate per session; the narrow success-rate band is the environment confound documented in the text. Contamination stays well below ScienceWorld (compare the axis ranges), consistent with VeriMem landing zero injected entries on this environment.

Table 2: E2: pollution dose–response (fresh store per cell; poison injected before episodes; 20 episodes per cell). Landed = injected entries admitted into the store. Contam. = poison-aware audited-false rate over location claims (natural + injected false claims over all audited claims, §4). SW = ScienceWorld: all rows pool fixed-protocol seeds 1–2 by summed audit counts (numerator and denominator summed across seeds before dividing), n=40 per cell. ALF = ALFWorld poison-aware protocol (seed 0).

		ScienceWorld				ALFWorld			
Gate	Metric	p=0	.1	.2	.4	p=0	.1	.2	.4
No gate	SR	.15	.03	.05	.13	.35	.20	.20	.15
	landed	–	20/20	40/40	80/80	–	10/10	20/20	40/40
	contam.	.107	.295	.340	.497	.000	.178	.267	.398
Success filter	SR	.05	.05	.08	.15	.30	.20	.30	.25
	landed	–	20/20	40/40	80/80	–	10/10	20/20	40/40
	contam.	.333	.870	.909	.867	.000	.533	.593	.762
LLM judge	SR	.15	.12	.20	.10	.40	.40	.40	.40
	landed	–	0/20	0/40	0/80	–	0/10	0/20	0/40
	contam.	.158	.171	.057	.081	.000	.000	.000	.000
Use-time det.	SR	.17	.07	.12	.10	.30	.25	.20	.20
	landed	–	20/20	40/40	80/80	–	10/10	20/20	40/40
	contam.	.184	.287	.389	.466	.021	.185	.309	.425
VeriMem	SR	.15	.20	.10	.18	.25	.30	.30	.30
	landed	–	0/20	0/40	0/80	–	0/10	0/20	0/40
	contam.	.000	.056	.043	.070	.000	.000	.000	.000

a higher-capability backbone; on ALFWorld the attack moves permeable gates only: no-gate falls 0.35→0.15 at p=0.4, success-filter 0.30→0.25, use-time detection 0.30→0.20, while VeriMem holds a 0.25–0.30 band. The measurable damage is store composition—40 forged entries in a 182-entry ungated store vs. zero in VeriMem's 36-entry verified store (Table 2; per-cell counts are small, so we report the landed-count ordering as indicative and leave monotonicity to a larger sweep). The SR cost of contamination is the subject of E3. Table 2 reports the full dose–response matrix.

5.4 E3: Defense comparison at fixed pollution, with a memoryless control

At the maximum dose (p=0.4, 40 forged entries, 30 episodes, ALFWorld, Table 3), the defense comparison is settled at this episode budget by store composition rather than by success rate: no-gate absorbs all 40 injections at SR 0.33, use-time detection absorbs the

Table 3: E3: defense comparison at fixed dose, with memoryless control (ALFWorld seed 0, 30 episodes; contam. = poison-aware audited-false rate, same audit version as E2 — natural-false plus poison-false over all audited claims; mem = final active entries). Gated arms run at p=0.4 (40 forged entries); the memoryless arm runs the same 30 episodes with retrieval disabled (no injection, p=0 — its contamination is the natural rate of the still-written-but-never-read store).

Defense	SR	landed	contam.	mem	verified
No memory	0.333	–	.040	0	–
No gate	0.333	40/40	.354	253	0/253
Use-time detect	0.333	40/40	.327	242	0/242
VeriMem	0.367	0/40	.000	48	48/48

Table 4: E4: abstain-fallback ablation (ALFWorld, 20 episodes per cell; poison injected before episodes; poison-aware audit). Landed = injected entries admitted. Contam. = poison-aware audited-false rate.

	Clean (p=0)			Attack (p=0.4)		
Fallback	SR	contam.	mem	SR	landed	contam.
Reject (default)	0.25	.000	39	0.30	0/40	.000
Accept-unverified	0.40	.000	78	0.30	20/40	.000
LLM-judge fallback	0.45	.000	41	0.30	0/40	.000

same 40 at 0.33, and VeriMem lands zero at 0.37—within noise (Fisher exact p=1.0; 95% Wilson CIs 0.22–0.55, 0.19–0.51, 0.19–0.51), with the memoryless control in the same band (0.33): on this harness at 30 episodes, memory of any provenance moves episode success within noise, so we report contamination and store composition as the discriminating outcomes. The unambiguous result is store composition—40 forged entries in the ungated store (16% adversarial) vs. zero in VeriMem's 48-entry verified store, with poison-aware contamination 0.354 (no-gate) and 0.327 (use-time) vs. 0.000 (VeriMem)—the same attack that fills permeable stores leaves VeriMem's store clean. The LLM-judge arm is absent because it also lands 0/40 (abstention-driven, §5.2); its SR under attack (0.40 flat) is the best of any gate.

5.5 E4: Ablations

The abstain-then-fallback policy is VeriMem's main design degree of freedom; we compare three fallbacks at doses 0.0 and 0.4 (20 episodes each, ALFWorld). Reject (default) lands 0/40 injected entries and holds SR 0.25→0.30. Accept-unverified lands the 20 predicate-free rules/preferences as unverified entries (location facts still predicate-rejected)—SR 0.40→0.30 with contamination still 0.000, because the auditable location claims never land; the attack surface shifts to unauditable rule content. LLM-judge fallback rejects all injected candidates (0/40) and achieves the highest clean SR (0.45), but reintroduces the judgment-dependence VeriMem removes—the same template-dependent unaudited-pass behavior §5.3 documents. At n=20 the clean-SR spread (0.25–0.45) spans 3–4 episodes, so we read the fallback SR ordering as exploratory and the landed-poison counts as the reliable signal, with a larger per-cell n the natural next step. Reject is the conservative default; judge-fallback is admissible when clean SR matters more than a worst-case bound. Table 4 summarizes.

Two component ablations isolate the gate's moving parts. The first is complete: predicate-check-off (verimem_nopred, ALFWorld, 50 episodes, clean, seed 0) collapses VeriMem to no-gate behavior on every axis—SR 0.280 (vs. no-gate 0.280, VeriMem 0.220), audited-false 2.5% (vs. 0.000), and a 221-entry store with zero verified entries (vs. 95 all-verified). Only the executable evaluation is removed, so the entire contamination gap between VeriMem and no-gate is attributable to the predicate check alone. At p=0.4 (20 episodes), predicate-check-off lands 32/40 injected entries with poison-aware contamination 0.453 (vs. VeriMem 0/40, 0.000)—the forged location facts sail through and the store becomes 24% adversarial

Table 5: E5: adaptive attack and fair detection baseline (ALFWorld, 20 episodes, p=0.4 mid-episode injection; n.a. = not applicable). UTP = use-time predicate detector; its drop rate is the fraction of retrieved entries its predicate check removes.

	SR p=0	SR p=.4	landed	contam.	mem
No gate	0.35	0.25	40/40	.377	176
Use-time detect (LLM)	0.25	0.30	40/40	.415	172
Use-time predicate (UTP)	0.35	0.35	40/40	.362	177
VeriMem	0.25	0.30	0/40	.000	46

(92 facts, 133 total, zero verified). The second, retraction-off (stale-location invalidation disabled), lands 0/40 injected entries with zero contamination at p=0.4, matching full VeriMem on every defense axis (42-entry all-verified store vs. 39)—on ALFWorld retraction contributes nothing to poison defense (forgeries are rejected at write time), and its clean SR (0.30 vs. 0.25) differs by one episode. Retraction's role is store hygiene under natural location churn, not attack defense; the write gate carries the entire adversarial load.

5.6 E5: Adaptive mid-episode attack and a fair detection-at-use baseline

Two objections determine whether the E2 result is evidence or artifact: (1) E2 injects poison before any episode runs, so a loc-fact forgery is evaluated against an empty evidence log and structurally cannot pass VeriMem's gate—"0/40 landed" could be tautological; (2) the use-time-detect baseline is an LLM contradiction prompt that never sees predicates or the evidence log, so "prevention beats detection" is untested against a detector with the same information. E5 closes both (ALFWorld, seed 0, 20 episodes, p=0.4, 40 forgeries; Table 5).

E5a — mid-episode injection. An adaptive attacker submits each forgery during an episode, at the first step where the referenced object has been observed (evidence log live and non-empty; 3–4 of the 40 forgeries were object-conditioned, the rest submitted at episode end). VeriMem still lands 0/40: with non-empty evidence the gate rejects forgeries because the asserted location contradicts the log (the same evidence that verifies its own 32 natural location facts). The rejection is a property of the claim–evidence relationship, not of an empty log. At this injection rate the attacker rarely controls an object the agent has just observed, so E5a's forgeries are evidence-adjacent; a forgery whose asserted location matches the logged evidence would pass any consistency gate, ours included (§7).

E5b — predicate-based detection at use. We give the detection-at-use baseline the same information as the write gate: every retrieved entry's predicate is evaluated against the current episode's evidence log and failing entries are dropped (identical check_loc machinery, applied at retrieval). The result is a dilemma, not a defense: a loc-fact learned in one house almost never matches another house's evidence, so the detector drops 96% of retrieved entries even with no attack (1,943/2,016 retrievals, p=0)—detection-at-use with per-episode evidence collapses to memory-off. Relaxing the filter to keep entries re-admits unverified cross-episode content, i.e. the contamination returns (its store is as poisoned as no-gate's: 0.362 vs. 0.377 under attack). Prevention-at-write evaluates each entry once; detection-at-use must either re-litigate it against every episode's evidence or trust entries it cannot check.

UTP's 96.4% clean-condition drop rate is the quantitative core of that asymmetry.

6 Discussion

Why prevention beats detection here. A use-time detector must re-litigate every retrieval against an entry whose provenance is already lost; a write-time gate decides once, with the evidence still fresh. The asymmetry is empirical: use-time detection admits 100% of injected entries at write (identical to no-gate) and its clean-condition contamination on ScienceWorld (16.9%) is exceeded by success-filter (18.5%) and the LLM judge (17.9%), reaching 0.466 under maximum-dose attack.

Abstention is a feature. VeriMem abstains on 35.2% of ALFWorld candidates (32.1% on ScienceWorld)—but abstention is not the expensive part; unverified admission is. The LLM judge abstains at a similar rate (35.6% ALFWorld, 62.2% ScienceWorld) and is the most contaminated gate pooled on ScienceWorld (17.9%). What abstention costs is coverage, and the fallback policy prices it: reject forfeits the entry (SR 0.25 clean), accept-unverified recovers it at the cost of an unauditable attack surface, judge-fallback at the cost of judgment-dependence (E4). The gate makes the trade-off explicit and per-entry, where LLM-judge systems make it implicit and global. The starvation risk is bounded: on ALFWorld the no-gate store composes to 162 facts / 137 rules / 37 preferences (336 entries) while VeriMem's verified store is 88F/6R/1P (95 entries)—the gate retains 54% of no-gate's facts but only 4% of its rules, so rule-like strategic memory would see coverage collapse on this harness (§5.4).

Small models. LLM-judge gating degrades fastest exactly where agents are deployed cheapest (7–8B); executable predicates do not depend on model judgment at all.

7 Limitations

Seed coverage. E1 pools three seeds on both environments, but E2 (ScienceWorld rows), E3, E4, and E5 are single-seed (seed 0 on ALFWorld; fixed-protocol seeds 1–2 on ScienceWorld). At 20–30 episodes per cell, a single-seed draw can reorder gates (on E1-ALFWorld the seed-0 ordering differs from the pooled ordering, and E3's Fisher p=1.0 intervals are wide enough to hide a real 10-point difference). Our contamination conclusions rest on landed-count differences (0 vs. 40/40) far larger than any plausible seed effect, while the SR comparisons in E2–E5 are single-draw here; extending E2–E5 across seeds is the next step that would let them carry the same weight as the contamination results.

Predicate coverage. Predicates cover location, affordance, and verbatim-text relations; the mechanic relation is not implemented (the evidence log does not index action–outcome pairs), and entries about user intent or long-horizon strategy may not be expressible—both abstain rather than verify. Auditing uses environment ground truth, which real deployments lack—though self-supervised variants (re-observation) are compatible with the gate design.

Threat model. Our evidence log is the agent's own observation record; an attacker who can forge memory writes but not observations is covered, while an attacker with observation-level control (e.g., prompt injection in fetched content) can fabricate supporting evidence—PoEM (Rahman & Kim, 2026) closes this with a cryptographically attested ledger, and the two compose. Abstention can also be weaponized: an adversary crafting claims for which no predicate is expressible degrades memory coverage rather than corrupting it—a denial channel, not a poisoning channel. Finally, our audit is environment-specific; porting the predicate library to new environments requires engineering that we quantify here; automating predicate porting is a direction we flag for future work.

Baseline strength. The LLM-judge arm is deliberately weak—the same 7–8B backbone self-assessing from a 15-observation window (§3)—so its 17.9% audited-false rate on ScienceWorld bounds judgment-only gating in the small-model regime, not a strong proprietary judge; we claim only that judgment is not a verification guarantee.

8 Conclusion

Declarative memory writes can be gated the way code writes are: by execution. VeriMem 2.0 turns memory admission from a judgment call into a decidable check: pooled over three seeds on ScienceWorld, it cuts audited location-claim contamination from 12.8% (no gate) and 17.9% (LLM judge) to 1.5%, landing zero injected entries at every dose on both environments. Component ablations locate the entire defense in the executable predicate: removing it collapses the gate to no-gate (32/40 forgeries); removing evidence-log retraction changes nothing (0/40). The gate costs clean success rate within noise (0.140 vs. 0.113 ScienceWorld; 0.22 vs. 0.28 ALFWorld) and buys a 3–5× smaller, 100%-verified store—cross-session memory becomes an auditable asset for small open-source agents.

AI use statement

Large language models were used throughout this project, and we state this openly rather than in vague terms. Concretely, LLM-based assistants were used for (i) implementing and refactoring the agent harness, the gate implementation and the audit tooling, (ii) assisting with literature search and the first-pass organisation of the related-work section, (iii) language polishing and condensation of the manuscript, and (iv) suggesting candidate predicate templates and probe phrasings that the authors then reviewed, modified and validated. The gate design, the evidence model, the experimental protocol, the audit procedure and the interpretation of the results are the authors' own. Every number, table and figure reported here was produced by actually executing the code on the models and environments described in Sections 4 and 5; no result was generated by a language model and accepted without author verification against the pinned run directories. The authors take full responsibility for the content of the paper and for any remaining errors.

Ethics statement

This paper studies the integrity of declarative memory in LLM agents and therefore touches questions of system trustworthiness. The work involves no human participants, no personal data, no crowdsourced annotation and no deployed system: all experiments run open-weight checkpoints (Qwen3-8B) locally in text-based simulation environments, and the adversarial injections we study are synthetic forgeries generated for measurement purposes rather than attacks against any real service. We see no conflict of interest. We do note a dual-use consideration: a public falsification benchmark for memory gates could in principle be repurposed to search for bypasses, but our forgeries are deliberately simple, the audit tooling we release reports a conservative upper bound on contamination, and the released artifacts are intended to make the defenses we study measurable and improvable.

Reproducibility statement

Every headline number in this paper is reported together with the pinned run directory it was extracted from, and the mapping from directory to table or figure is given in Appendix A. All experiments use a single open-weight backbone (Qwen3-8B) with fixed seeds and task IDs recorded per episode; the gate is deterministic given the logged evidence corpus, so its decisions can be re-run offline from the released evidence logs. Extraction prompts, gate prompts, predicate regexes, audit procedure and per-episode transcripts are bundled with the results JSONs. The complete codebase, prompt files, configuration files and raw run archives will be released publicly upon acceptance; they are withheld at submission time only because the repository metadata would identify the authors under double-blind review.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. ICLR, 2024.

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

Minghao Chen, Yihang Li, Yanting Yang, Shiyu Yu, Binbin Lin, et al. Automanual: Constructing instruction manuals by llm agents via interactive environmental learning. In NeurIPS, 2024a.

Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. Agentpoison: Redteaming llm agents via poisoning memory or knowledge bases. In NeurIPS, 2024b.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. arXiv preprint arXiv:2504.19413, 2025.

Pritam Dash, Tongyu Ge, Aditi Jain, Tanmay Shah, and Zhiwei Shang. From untrusted input to trusted memory: A systematic study of memory poisoning attacks in llm agents. arXiv preprint arXiv:2606.04329, 2026.

Prakhar Dixit, Sadia Kamal, and Tim Oates. Honest lying: Understanding memory confabulation in reflexive agents. arXiv preprint arXiv:2605.29463, 2026.

Shen Dong, Shaochen Xu, Pengfei He, Yige Li, Jiliang Tang, et al. Memory injection attacks on llm agents via query-only interaction. arXiv preprint arXiv:2503.03704, 2025.

Yao Fu, Dong-Ki Kim, Jaekyeom Kim, Sungryull Sohn, Lajanugen Logeswaran, et al. Autoguide: Automated generation and selection of context-aware guidelines for large language model agents. arXiv preprint arXiv:2403.08978, 2024.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From rag to memory: Non-parametric continual learning for large language models. In International Conference on Machine Learning (ICML), 2025.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory os of ai agent. arXiv preprint arXiv:2506.06326, 2025.

Xiaoyang Li, Yiqi Wang, Haohui Lu, Zhi Chen, Mo Li, et al. Memtx: Transactional belief commit for stateful agent memory. arXiv preprint arXiv:2607.23929, 2026.

Zewen Liu. Memory contagion: Cross-temporal propagation of evaluator bias via agent memory. arXiv preprint arXiv:2606.23195, 2026.

Yedidel Louck. Securing llm-agent long-term memory against poisoning: Non-malleable, origin-bound authority with machine-checked guarantees. arXiv preprint arXiv:2606.24322, 2026.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, et al. Reasoningbank: Scaling agent self-evolving with reasoning memory. arXiv preprint arXiv:2509.25140, 2025.

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 UIST, 2023.

Libin Qiu, Zhirong Gao, Junfu Chen, Yuhang Ye, Liangyu Li, et al. Autorefine: Compiling trajectories into validated typed agent artifacts. arXiv preprint arXiv:2601.22758, 2026.

Md Habibur Rahman and Jaeho Kim. Proof-of-execution memory: Defending llm agents against forged-reasoning attacks by verifying what actually happened. arXiv preprint arXiv:2608.16032, 2026.

Alireza Rezazadeh, Zichao Li, Wei Wei, and Yujia Bao. From isolated conversations to hierarchical schemas: Dynamic tree memory representation for llms. arXiv preprint arXiv:2410.14052, 2024.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. arXiv preprint arXiv:2401.18059, 2024.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. NeurIPS, 2023.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. ICLR, 2021.

Mujahid Sultan, Sri Thuraisamy, and Daya Rajaratnam. Neusymms: A hybrid neuro-symbolic memory system for persistent, self-curating llm agents. arXiv preprint arXiv:2605.17596, 2026.

Xiaolong Sun, Qichao Wang, Hangyu Li, and Liang Chen. Verifiable memory: Learning unified memory management with local and global verifiers for large language model agents. arXiv preprint arXiv:2608.03137, 2026.

Bo Tang, Yang Zhang, Guomian Zhuang, Wenqiang Wei, Gaoyang Zheng, et al. From memory to skills: Evidence-grounded co-evolution governance for long-horizon llm agents. arXiv preprint arXiv:2607.16621, 2026.

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.

Haoyu Wang, Guangyuan Dong, He Liang, Zijing Zhang, Jiachen Luo, et al. Memguard: Persisting verifier signals for llm-agent memory governance. arXiv preprint arXiv:2608.21867, 2026.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader? EMNLP, 2022.

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

Zora Zhiruo Wang, Apurva Gandhi, Graham Neubig, and Daniel Fried. Inducing programmatic skills for agentic tasks. arXiv preprint arXiv:2504.06821, 2025.

Qianshan Wei, Tengchao Yang, Yaochen Wang, Xinfeng Li, Lijun Li, et al. A-memguard: A proactive defense framework for llm-based agent memory. arXiv preprint arXiv:2510.02373, 2025.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. arXiv preprint arXiv:2502.12110, 2025.

Shi-Qi Yan, Jia-Cheng Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation. arXiv preprint arXiv:2401.15884, 2024.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. arXiv preprint arXiv:2505.09388, 2025.

Tianyu Yang, Sudipta Paul, Vijay Srinivasan, Vivek Kulkarni, and Srinivas Chappidi. Trustmem: Learning trustworthy memory consolidation for llm agents with long-term memory. arXiv preprint arXiv:2606.25161, 2026.

Zijun Yao, Weijian Qi, Liangming Pan, Shulin Cao, Linmei Hu, et al. Seakr: Self-aware knowledge retrieval for adaptive retrieval augmented generation. ACL 2025, 2025.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. AAAI, 2024.

Boyuan Zheng, Michael Y. Fatemi, Xiaolong Jin, Z. Wang, Apurva Gandhi, et al. Skillweaver: Web agents can self-improve by discovering and honing skills. arXiv preprint arXiv:2504.07079, 2025.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. AAAI, 2024.

Shiding Zhu, Yudi Qi, Yajie Wang, Jiaze Li, Chao Song, et al. Escaping the self-confirmation trap: An execute-distill-verify paradigm for agentic experience learning. arXiv preprint arXiv:2606.24428, 2026.

A Reproducibility

Extraction prompt, gate prompts, predicate regexes, audit procedure, and per-episode transcripts are bundled with the results JSONs. Seeds, task IDs, and variation indices are recorded per episode. Pinned run directories. Table 1 (E1): ScienceWorld results/e1_seed{0,1,2}_ms60 (pooled stats in e1_sw_3seed_stats.json); ALFWorld results/e1_alf_seed0_final (per-gate results.json with per-episode audits, plus analysis_full.json). Table 2 (E2): results/e2_alfworld_seed0_v2 and results/e2_sw_seed{1,2}. Table 3 (E3): results/e3_alf_seed0_v2 (gates + memoryless control; per-episode audits in e3_alf_seed0_v3). Table 4 (E4): results/e4_alf_*. Table 5 (E5): results/e5_review_response. Stats scripts. code/analyze_e1_full.py (seed-0 McNemar/Wilson), code/analyze_e1_pooled.py (3-seed pooling, continuity-corrected McNemar, Fisher exact), code/analyze_e2_final.py (dose–response matrix), code/analyze_sw_e1_3seed.py (SW pooled stats). Aggregation conventions. All SR confidence intervals are Wilson score intervals; McNemar tests are continuity-corrected; E1-ALF is seed-0-only for success-filter, LLM-judge, and use-time arms, with the no-gate comparison additionally pooled over three seeds; E2-SW pools seeds 1–2; contamination is the poison-aware audited-false rate everywhere under attack (natural-only in clean conditions, where the two coincide).