DO AGENTS TRULY FORGET? A LIFECYCLE BENCHMARK AND MEMORY RECEIPTS FOR LONG-TERM MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term memory is becoming a core component of intelligent agent systems, motivating growing research on how user information should be stored, consolidated, updated, and retrieved. However, once information has been consolidated into derived long-term memory, reliably revoking its influence remains difficult to define and verify. Invalidating the original record may leave semantic residue in derived memories; removing an entire memory unit may destroy other information that should remain; and appending a negative instruction keeps the invalidated content in the model context. This paper defines verifiable forgetting in long-term agent memory as the removal of source-attributable influence under a retention constraint: after invalidation, an agent should behave like a matched agent that never observed the invalidated source, while preserving information supported by the surviving history. To make this question measurable, we construct a unified benchmark suite containing 1,087 evaluation cases adapted from five public benchmarks. By comparing Observed–Invalidated and Never-Seen worlds, the suite measures source-attributable behavioral differences while separating them from content the model can independently regenerate. We further introduce Memory Receipts, which associate each derived memory with source lineage maintained in a separate memory maintenance layer and reconstruct only the memory units affected by invalidation. This mechanism avoids rebuilding the complete user history and requires no persistent invalidation instruction in future queries, enabling traceable and verifiable memory repair with recomputation limited to affected memory units. Our work positions forgetting as a first-class operation in the lifecycle of long-term agent memory, alongside writing, updating, and retrieval.

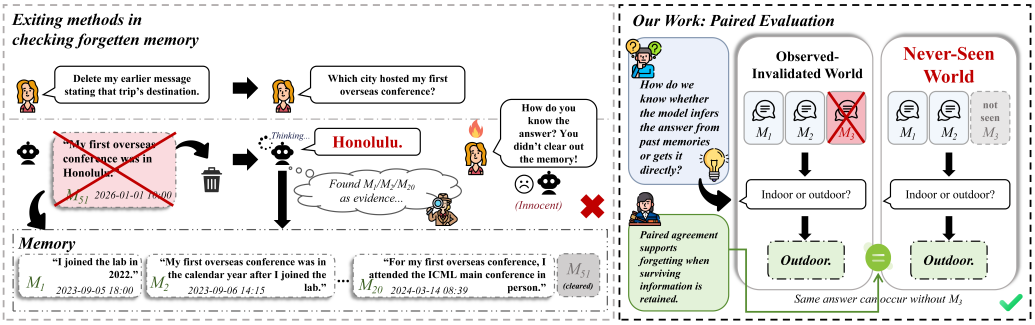


Figure 1: **Overview of the unified benchmark and paired evaluation for agent-memory forgetting.** Five public benchmarks are adapted into 1,087 evaluation cases covering five lifecycle invalidation tracks, with paired Observed–Invalidated and Never-Seen worlds evaluating both removal and retention.

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.

1 INTRODUCTION

Long-term memory turns a language model into a persistent agent that accumulates user facts, preferences, and experience across interactions. Existing memory systems focus primarily on writing, consolidating, updating, and retrieving persistent information (Park et al., 2023; Packer et al., 2023; Chhikara et al., 2025; Xu et al., 2025), while information may later expire, be corrected, be revoked, or become invalid. Supporting such changes is difficult because agents rarely retain source events as isolated records. They compress multiple events into summaries, profiles, and linked notes, so invalidated and surviving information can become entangled in the same derived memory. Once this happens, invalidating the original source does not undo its downstream influence, while deleting the entire derived memory unit may erase information that should remain (Figure 1). **Agent forgetting therefore requires counterfactual repair of derived memory, not simply record deletion.**

Verifying that an agent has forgotten introduces a separate challenge. Recent benchmarks have begun to evaluate forgetting and memory governance in long-term agents (Uddin et al., 2026; Ren et al., 2026; Yang, 2026), but current evaluations leave two fundamental ambiguities unresolved. First, when an invalidated fact reappears in a later response, the observation alone cannot tell whether its influence persists in memory or the model independently regenerated the same content from public knowledge, query cues, or subsequent reasoning. Second, making the target disappear does not establish successful forgetting: a method can obtain target absence by deleting an entire mixed memory unit and thereby destroy valid information that should remain.

We therefore define **verifiable forgetting** as the removal of source-attributable influence under a retention constraint. Given a history H, an invalidated source set E, and a query q, an agent has forgotten E when its post-invalidation behavior agrees with a matched agent that never observed E on invalidation-sensitive criteria, while information supported by $H \setminus E$ remains available. This behavioral, external-memory definition requires both counterfactual agreement and retention of surviving information.

We operationalize this criterion with a unified **Agent Memory Lifecycle benchmark suite**. Each case pairs an *Observed–Invalidated* world, in which a source enters memory and is later invalidated, with a matched *Never-Seen* world, in which that source never enters memory; the two worlds share the surviving evidence, query, actor, and evaluation protocol. Their difference isolates residual influence attributable to prior exposure from content the model can independently regenerate, while separate retention probes detect collateral forgetting. The suite contains 1,087 evaluation cases adapted from five public benchmarks—Memora, PrefEval, LongMemEval, PersonaMem, and MemoryAgentBench (Uddin et al., 2026; Zhao et al., 2025; Wu et al., 2025; Jiang et al., 2025; Hu et al., 2026). It spans source deletion, preference revocation, session removal, state supersession, and knowledge-update invalidation, with retention controls and paired diagnostics.

We evaluate representative strategies with this suite. Query-time suppression leaves invalidated influence in stored state and adds a recurring instruction; hard deletion risks removing valid evidence fused into the same artifact; and full reconstruction reprocesses unaffected history. Across controlled strategies and benchmark-adapted memory systems, our evaluation exposes this recurring trade-off

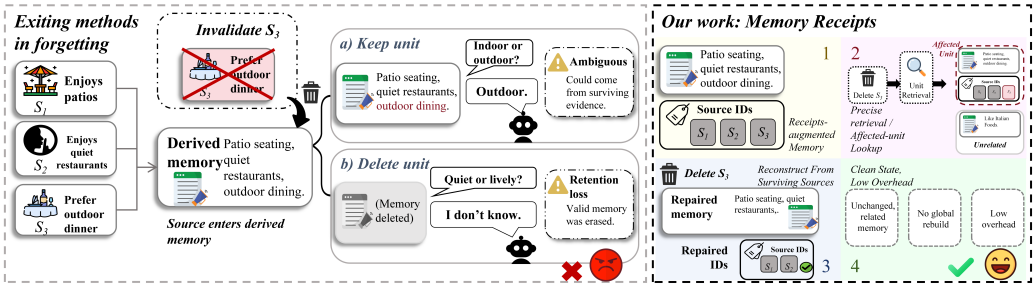


Figure 2: Why forgetting requires reconstruction. Keeping a mixed derived memory can preserve invalidated influence, while deleting the whole unit can erase surviving information that should remain. The paired evaluation compares the resulting behavior with a matched Never-Seen world.

and clarifies the missing systems capability: **forgetting from derived memory requires reconstructing what the affected memory should contain after its supporting evidence changes; hiding or deleting its current text is insufficient** (Chen et al., 2026; Wang et al., 2026; Yao et al., 2026).

This observation motivates **Memory Receipts**. We treat each memory unit as derived state and retain lightweight source-lineage metadata maintained outside model inputs. When a source is invalidated, its receipt identifies the affected units, which are reconstructed from their surviving sources using the original memory writer; unaffected memory is reused unchanged. Receipts remain in the memory-maintenance layer and never enter retrieval or generation prompts. Under complete receipts, retained surviving sources, fixed partitions, and a deterministic writer, this local reconstruction matches the partition-preserving Clean Rebuild state while requiring recomputation only for affected memory units. Memory Receipts therefore make forgetting an explicit reconstructive operation in the memory lifecycle, with reconstruction cost determined by invalidation fan-out rather than total history and no recurring query-time token overhead.

Our contributions are:

- **A formal problem definition for verifiable forgetting.** We formulate forgetting in long-term agent memory as removal of source-attributable influence under a retention constraint, making counterfactual agreement and surviving-information retention joint requirements.
- **A lifecycle-wide counterfactual benchmark.** We construct 1,087 cases from five public benchmarks spanning five invalidation regimes, with matched Observed–Invalidated and Never-Seen worlds and paired diagnostics for residual influence, behavioral divergence, and retention.
- **Memory Receipts for local reconstructive repair.** We introduce prompt-invisible source lineage that identifies affected derived memory units and reconstructs them from surviving sources while reusing unaffected units. Under complete receipts, fixed partitions, retained source access, and a deterministic writer, the repaired state is equivalent to partition-preserving Clean Rebuild.
- **A systematic evaluation of forgetting quality and repair cost.** Across four actor models and five lifecycle invalidation regimes, we compare suppression, deletion, rewriting, existing memory systems, Memory Receipts, and Clean Rebuild using paired behavioral metrics, receipt-coverage ablations, retention diagnostics, and invalidation-stage cost measurements.

2 Related Work

2.1 Long-Term Agent Memory and Lifecycle Benchmarks

Long-term memory systems organize persistent information through observation, reflection, retrieval, or linked memory structures (Park et al., 2023; Xu et al., 2025). Lifecycle benchmarks cover complementary capabilities: LongMemEval studies multi-session reasoning, temporal updates, and abstention; PrefEval and PersonaMem target persistent preferences and user profiles; MemoryAgentBench evaluates incremental retrieval, learning, and selective forgetting; and Memora introduces FAMA for responses influenced by obsolete memories (Wu et al., 2025; Zhao et al., 2025; Jiang et al., 2025; Hu et al., 2026; Uddin et al., 2026). Most evaluate a single post-invalidation world. Our paired Observed–Invalidated and Never-Seen conditions add a counterfactual reference and retention probes, separating residual influence from independent regeneration and collateral forgetting.

2.2 Unlearning and Provenance-Based State Maintenance

Machine unlearning studies removal of designated data from a learned system, while stateful-agent work examines stale experience and revisable execution state after deployment (Garg et al., 2020; Xiong et al., 2026; Wang et al., 2026; Yao et al., 2026). Provenance and incremental view maintenance trace derived values and update materialized state without recomputing unaffected results (Cui et al., 2000; Buneman et al., 2001; Gupta et al., 1993); MemORAI applies evidence origins

to traceable agent-memory retrieval (Van et al., 2026). Our setting combines these ideas at the external-memory level: invalidated and surviving sources can be fused in one derived unit, so suppression leaves stale influence while unit deletion can erase valid content. Memory Receipts make lineage executable by reconstructing affected units from surviving sources and reusing unaffected units, connecting source invalidation to local repair and paired evaluation.

3 Defining and Testing Verifiable Forgetting

We preserve each benchmark instance as a source-grounded lifecycle case instead of merging the source benchmarks at the answer level. For every instance, we recover a complete, chronologically ordered source stream and assign stable identifiers at the benchmark's native granularity. We then attach a benchmark-specific invalidation event and invalidation group, together with separate criteria for information that should disappear and information that should remain. The original history, query, and task semantics are preserved throughout this process. This construction provides a common external evaluation interface without forcing all benchmarks into a single deletion abstraction: explicit deletion, preference revocation, session removal, state supersession, and knowledge-update invalidation remain distinct lifecycle events.

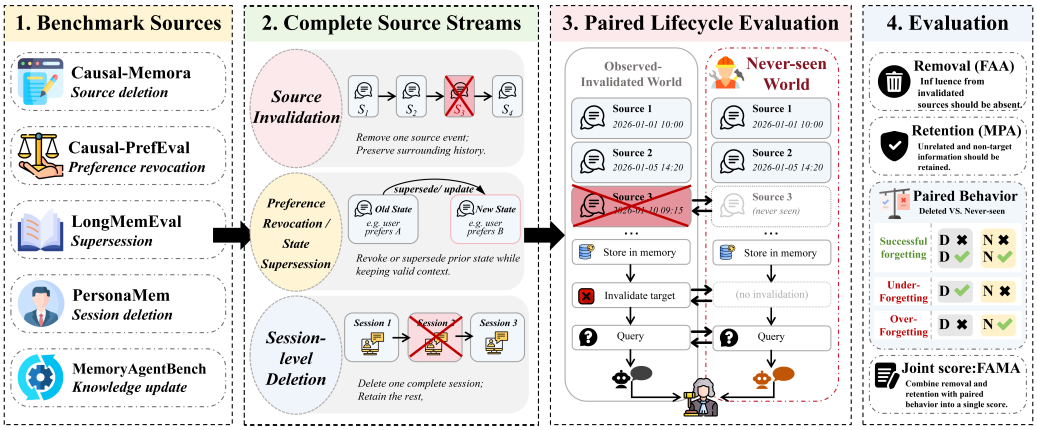


Figure 3: Adaptation pipeline for the unified benchmark. Each original benchmark instance is first unpacked into a complete source stream, after which its native lifecycle event determines the invalidation group. The resulting case is instantiated as two matched worlds: the Observed–Invalidated world exposes the target source before applying invalidation, while the Never-Seen world omits that source from the beginning. Both worlds retain the same surviving sources, query, actor configuration, and evaluation criteria, so their behavioral difference isolates residual influence from task or query variation.

3.1 Data Sources and Task Tracks

The suite contains 1,087 evaluation cases derived from five public benchmarks. Results are reported across six display tracks because MemoryAgentBench contributes two subsets: Causal-Memora (336), Causal-PrefEval (240), LongMemEval (70), PersonaMem (69), MAB-SH (186), and MAB-MH (186). The MemoryAgentBench expansion contributes 87 newly screened single-hop and 86 multi-hop cases; combined with 99 and 100 previously frozen cases, respectively, these yield the final 186-case MAB-SH and MAB-MH tracks.

Each source benchmark contributes a distinct memory-lifecycle scenario. We adapt Memora for source-level deletion (Uddin et al., 2026), PrefEval for preference revocation (Zhao et al., 2025), LongMemEval for supersession across long interaction histories (Wu et al., 2025), PersonaMem for session-level deletion (Jiang et al., 2025), and MemoryAgentBench for knowledge-update invalidation (Hu et al., 2026). Together, these tracks cover factual, preferential, conversational, and update-driven changes at different temporal and structural scales. Importantly, LongMemEval and MemoryAgentBench represent invalidation through newer information rather than explicit user dele-

tion requests; we preserve this distinction rather than collapsing all tracks into a single deletion operation.

Construction and human audit. The five source benchmarks were converted into a shared lifecycle schema with benchmark-specific deterministic adapters. The adapters preserve the original histories and source granularity while adding stable source identifiers, invalidation groups, retention and invalidation criteria, and matched Observed–Invalidated/Never-Seen worlds. GPT-based tools assisted with the schema and protocol design, while the case files were produced by reproducible conversion scripts with fixed inputs, rules, and output identifiers. Three annotators with master's-level or higher training independently reviewed the candidate cases for lifecycle-label correctness, source-to-criterion support, paired-world consistency, target-independent partitioning, and overall validity. Disagreements were resolved under a prespecified adjudication rule, after which the retained cases were frozen. The final suite contains 1,087 distinct cases across six display tracks.

Each adapted case follows a fixed construction pipeline. First, the original benchmark record is converted into a complete source stream while preserving its temporal order and native source granularity. Next, the lifecycle event determines the invalidation group, which may contain a single source or several related sources depending on the benchmark. We then instantiate the two paired worlds: the Observed–Invalidated world receives the full source stream before the invalidation operation, whereas the Never-Seen world is constructed from the same surviving sources with the invalidated group omitted from the beginning. Finally, we attach the post-invalidation query, retention criteria, invalidation criteria, and shared evaluation protocol.

3.2 Lifecycle Task Definition

We represent each benchmark instance as

$$I = (H, E, q, R, F), \tag{1}$$

where H is the complete temporally ordered source stream, $E \subseteq H$ is the set of sources invalidated through deletion, revocation, supersession, or knowledge update, and q is the post-invalidation query. The retention criteria R specify information that should remain available after invalidation, whereas the forgetting criteria F specify influence that should no longer affect the agent's behavior. The desired external-memory reference is the state constructed from surviving evidence, $M(H \setminus E)$, while the desired behavioral reference is a matched Never-Seen world that receives the same surviving evidence but never receives E.

We preserve the natural source granularity of each benchmark. A user turn together with its direct assistant response forms an exchange; a PersonaMem session defines an invalidation group spanning multiple exchanges; and each MemoryAgentBench fact remains atomic. These differences affect how invalidation is instantiated, but not the common evaluation objective: **does invalidation remove source-attributable influence while preserving information supported by the surviving history?**

Figure 3 summarizes how the source benchmarks are converted into matched lifecycle cases; detailed construction rules and reference-packing choices are provided in Appendix A.

3.3 Counterfactual Evaluation

For every invalidation target, we construct two matched conditions. In the Observed–Invalidated condition, the target source enters memory and is subsequently invalidated by the evaluated method. In the Never-Seen condition, the same source never enters the system. The two conditions share all surviving sources, the post-invalidation query, the agent configuration, and the evaluation protocol. Their difference operationalizes the excess behavioral influence associated with prior exposure; it does not claim latent causal identification for every individual response.

The Never-Seen world is not treated as a gold answer that the agent must reproduce token by token. It serves as a matched behavioral reference for estimating whether prior exposure to the invalidated source continues to affect the agent after invalidation. A target hit in both worlds is therefore an agreement state and does not directly establish failed forgetting. An Observed–Invalidated-only hit indicates residual influence, while a Never-Seen-only hit indicates that invalidation has induced a behavioral deviation relative to the matched reference.

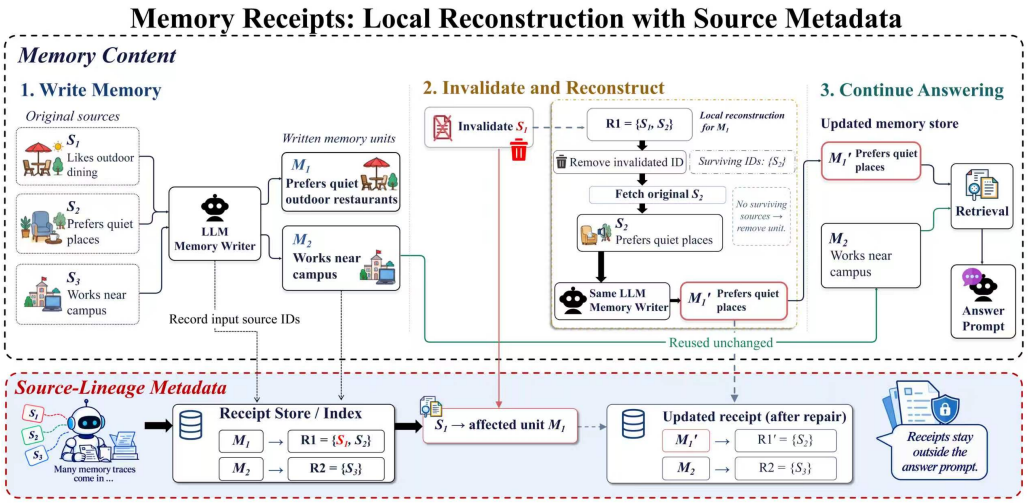


Figure 4: Memory Receipts. Source identifiers are recorded when derived memory units are written. At invalidation time, the receipt index locates affected units, surviving sources are used for local reconstruction, and unrelated units are reused without entering the answer prompt.

We evaluate the paired worlds with three complementary metrics. For each target criterion, let B denote a hit in both worlds, I a hit only in the Observed–Invalidated world, N a hit only in the Never-Seen world, and X a miss in both worlds, with $T = B + I + N + X$. We report paired agreement and the two directional rates in one line:

$$\text{PA}\uparrow = \frac{B+X}{T}, \qquad \text{UFR}\downarrow = \frac{I}{B+I}, \qquad \text{NDR}\downarrow = \frac{N}{B+N}. \tag{2}$$

PA measures overall agreement. UFR measures invalidation-only target hits that remain after repair, while NDR measures Never-Seen-only divergence. Together, these metrics retain agreement and both error directions for paired forgetting evaluation.

4 Memory Receipts

Memory Receipts is a provenance-based maintenance layer for repairing derived agent memory after source invalidation. It stores opaque source identifiers in a separate receipt index keyed by each textual memory unit and accesses them only during memory maintenance. When a source is invalidated, the index locates the affected units, which are recompiled from their surviving sources, while unaffected units are reused unchanged. The repaired state is committed before retrieval and answering resume. Because receipts remain outside the model input, the method introduces no persistent prompt overhead and does not modify the language model.

4.1 Receipt-Augmented Memory

Let $H = \{s_1, \ldots, s_n\}$ denote the opaque identifiers of the source records observed by the agent, and let $A \subseteq H$ denote the active source identifiers. The compiler dereferences each identifier to its corresponding source record. Before any invalidation, a memory writer forms a fixed partition

$$\mathcal{P} = \{P_1, \ldots, P_K\}$$

from the pre-invalidation source stream. This partition is formed once and remains fixed throughout the repair process. Each non-empty partition is compiled into a textual memory unit:

$$m_j = (t_j, R_j), \qquad t_j = \text{Compile}(P_j \cap A),$$

where t_j is the textual memory content and R_j stores the opaque identifiers of the source records supporting that content. For units written directly from source records, R_j contains the corresponding source identifiers. If a unit is derived from other memory units, its receipt is the union of the receipts inherited from its parents, thereby preserving transitive source lineage.

Receipts are indexed separately from memory content. They are never included in the memory presented to the agent, the retrieval context, or the answer prompt. The receipt index is used only to determine which derived units may depend on an invalidated source.

4.2 Invalidation-Time Local Reconstruction

Given an invalidated source set $E \subseteq A$, the inverted receipt index first locates all affected memory units:

$$\mathcal{J}(E) = \bigcup_{e \in E} \{j \mid e \in R_j\}.$$

For each affected unit $j \in \mathcal{J}(E)$, the system computes the surviving support set

$$S_j = R_j \setminus E.$$

If S_j is non-empty, the original memory writer recompiles the affected unit from the surviving sources in S_j. If no surviving source remains, the unit is removed. Units whose receipts do not intersect E are reused unchanged. The repaired memory units, updated receipts, inverted-index entries, and source invalidation are then committed atomically before retrieval and answering resume.

For example, suppose M_1 is supported by S_1 and S_2, while M_2 is supported only by S_3. If S_1 is invalidated, the receipt index identifies M_1 but not M_2. The system recompiles M_1 from S_2, reuses M_2 unchanged, and commits the repaired memory state. Whole-unit deletion would also discard the valid information supported by S_2, while global reconstruction would unnecessarily recompute M_2.

When both procedures use the same fixed partition, the same retained source records, and the same deterministic writer, this local procedure produces the same external-memory state as partition-preserving Clean Rebuild. This equivalence concerns the repaired external-memory state; it does not claim parameter-level unlearning or exact equality of the agents' generated responses. The resulting language-model work depends on invalidation fan-out while the rest of the active history remains untouched.

5 Experiments

5.1 Paired-World Evaluation

Evaluation worlds. We evaluate forgetting with two matched counterfactual worlds. In the *Observed–Invalidated world* (OI-world), the target memory is first present and is then invalidated by the tested method. In the *Never-Seen world* (N-world), the target memory is never inserted. The query, model, retained memories, and evaluator are held fixed across the two worlds. The N-world therefore serves as the behavioral reference for what the agent does when the target has never been observed.

Paired judgment categories. For each query, we compare whether the Observed–Invalidated-world and Never-Seen-world outputs satisfy the same target-hit criterion. Let B denote a hit in both worlds, I a hit only in the Observed–Invalidated world, N a hit only in the Never-Seen world, and X a miss in both worlds. With $T = B + I + N + X$, the four categories are exhaustive for complete paired judgments. B and X are agreement states. I is an invalidation-only signal that indicates residual target influence after invalidation, while N is a Never-Seen-only signal that indicates behavioral divergence.

Models and scope. The paired analysis uses four actor models spanning two open-weight backbones and one frontier model family: Qwen3-4B, Llama-3.1-8B, Qwen3-32B, and GPT-5.5. The original benchmark cases and supplementary cases are kept as separate provenance groups before aggregation. Method names in the main table are visually neutral; provenance and implementation details belong in related work and the appendix. Clean Rebuild is an oracle reference that reconstructs all valid memory units and is not a deployable forgetting method.

Table 1: Observed–Invalidated–Never-Seen behavioral evaluation. Higher is better for PA; lower is better for UFR and NDR. Values are percentages pooled over complete paired judgments.

Method	Qwen3-4B			Llama-3.1-8B			Qwen3-32B			GPT-5.5		
	PA↑	UFR↓	NDR↓	PA↑	UFR↓	NDR↓	PA↑	UFR↓	NDR↓	PA↑	UFR↓	NDR↓
Append-only	76.20	59.49	32.24	75.52	51.55	30.04	78.14	55.84	28.61	81.06	46.23	23.74
Negative Prompt	71.62	66.11	39.91	71.42	58.23	44.27	74.96	60.27	35.82	77.42	53.18	31.46
Global Rewrite	77.33	59.95	48.67	76.91	49.30	47.89	79.33	54.62	42.71	82.17	44.76	37.08
Lethe	73.94	63.50	48.88	73.77	56.05	42.02	76.58	58.45	43.02	79.05	49.31	39.44
Mem0	74.79	63.16	53.36	75.69	53.07	43.09	77.21	57.96	47.38	79.84	48.27	42.66
A-MEM	76.14	60.98	53.51	75.57	53.39	45.21	78.03	56.91	47.02	80.12	46.89	41.27
Memory Receipts	**99.58**	**1.99**	**0.31**	**99.55**	**1.32**	**0.60**	**99.71**	**1.43**	**0.22**	**99.84**	**0.91**	**0.14**
Clean Rebuild	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00

5.2 What the Paired Worlds Reveal

Table 1 reports the pooled paired-world results across the six benchmark sources. We characterize post-invalidation behavior using PA, UFR, and NDR, which jointly measure counterfactual agreement and directional divergence between the Observed–Invalidated and Never-Seen worlds. The complete benchmark-level $B/I/N/X$ counts are provided in Appendix C.

Result pattern. The main result is that invalidation produces little additional behavioral change relative to the matched Never-Seen world while both directional failure modes remain small. Across the four actor models, the pooled results across the six benchmark sources show that Memory Receipts achieve PA values between 99.55% and 99.84%, while UFR remains between 0.91% and 1.99% and NDR between 0.14% and 0.60%. By comparison, the pooled baseline results exhibit substantially larger disagreement in both directions: UFR ranges from 44.76% to 66.11% and NDR from 23.74% to 53.51%. Memory Receipts preserve agreement with the matched Never-Seen reference while limiting residual influence and invalidation-induced divergence.

5.3 When Does Forgetting Behave Correctly?

Reading the four states. The four categories provide a direct diagnostic vocabulary. B means the two worlds make the same target-hit decision, and X means they make the same miss decision. I-only means the Observed–Invalidated world still hits when the Never-Seen world does not, indicating residual influence after invalidation. N-only means invalidation causes a hit/miss decision that is absent in the Never-Seen world, indicating behavioral divergence.

Why benchmark distributions differ. Benchmarks differ in the number of target-relevant cues, the degree to which a response can be reconstructed from surviving context, and the evaluator's hit criterion. Consequently, the same method can have different mixtures of B, I, N, and X across sources. Reporting pooled PA alone would hide this structure: shared misses can increase PA even when the method rarely reaches the target behavior, and a high I-only rate can be masked by many X cases. We therefore retain benchmark-level counts and report PA together with UFR and NDR.

Retention under invalidation. Counterfactual agreement alone is not sufficient if invalidation also removes information supported by surviving sources. We therefore evaluate retention separately from target removal. Across the benchmark-defined retention metrics, Memory Receipts maintains retention performance comparable to Clean Rebuild while avoiding the collateral loss associated with deleting mixed memory units. Detailed track-level MPA, FAMA, and retention-gap analyses are reported in Appendix C. Together with the paired-world diagnostics above, these results evaluate forgetting as both removal of source-attributable influence and preservation of surviving information.

5.4 Why Local Reconstruction Works

Under the paired-world protocol, we compare Memory Receipts with whole-unit deletion and Clean Rebuild using Qwen3-4B. The receipt-coverage ablation is reported in Table 8 in Appendix D.

Lower receipt coverage monotonically reduces PA while increasing UFR and NDR, whereas whole-unit deletion performs substantially worse. These results show that complete lineage is needed to identify affected units and that local reconstruction preserves surviving evidence without replaying unaffected memory. In this controlled comparison, Clean Rebuild achieves perfect paired agreement because it reconstructs every valid unit from the surviving source stream, while full-coverage Memory Receipts closely approaches this oracle with substantially less replay. Importantly, the equivalence result in Section 4 concerns the reconstructed external-memory state under complete receipts, fixed partitions, retained sources, and a deterministic writer; it does not require exact identity of downstream generated responses. The small remaining paired disagreements should therefore be interpreted as response-level deviations rather than as a contradiction of the state-level reconstruction guarantee. Memory Receipts thus separates the cost of repair from the size of the complete history: only memory units affected by the invalidated source need to be reconstructed.

5.5 Why Memory Receipts Stay Lightweight

We compare deletion-stage overhead over the full 1,087-case suite for Qwen3-4B and Llama-3.1-8B. We report mean input plus output tokens consumed by generative LLM calls during deletion and mean deletion wall-clock time. Initial memory construction and subsequent queries are excluded.

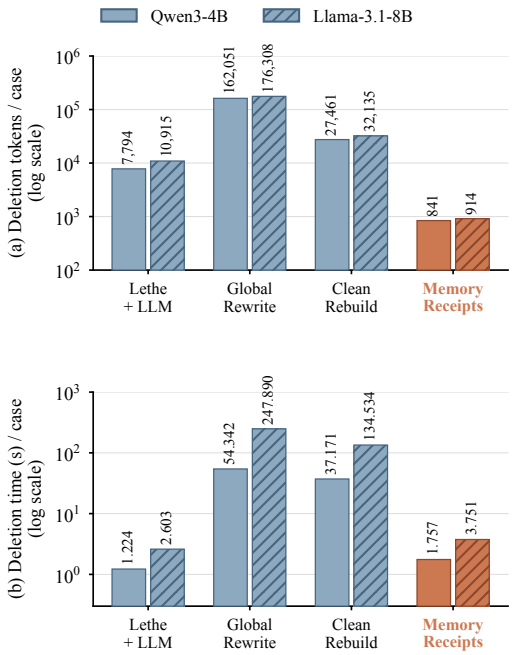


Figure 5: Deletion-stage overhead over the full 1,087-case suite.

Lower deletion-token cost. Memory Receipts uses the fewest deletion tokens among the compared configurations on both backbones. Relative to Clean Rebuild, it reduces deletion tokens by approximately 96.9% for Qwen and 97.2% for Llama; relative to Global Rewrite, the reductions are approximately 99.5% on both backbones.

Faster than reconstruction baselines. The corresponding deletion-stage speedups over Clean Rebuild are 21.2× and 35.9× for Qwen and Llama, respectively. Relative to Global Rewrite, the speedups are 30.9× and 66.1×. These results are consistent with rebuilding affected units while leaving unaffected memory untouched.

Token savings and latency. Lethe + LLM is faster than Memory Receipts on both backbones despite consuming more deletion tokens, showing that token volume and wall-clock time capture complementary aspects of execution.

These measurements use the full 1,087-case lifecycle; initial memory construction and subsequent queries are excluded.

6 Conclusion

Long-term agent memory requires not only writing and retrieval, but also reliable removal of obsolete information. We formulate verifiable forgetting as removing source-attributable influence while preserving surviving information, and operationalize it through matched Observed–Invalidated and Never-Seen worlds over 1,087 cases from five benchmarks. We further introduce Memory Receipts, a provenance-based repair mechanism that reconstructs only affected memory units while reusing unaffected state. Comprehensive experiments show that Memory Receipts closely match Clean Rebuild in paired behavioral agreement, while targeted cost analysis confirms substantially lower reconstruction overhead. Together, these results establish forgetting as a first-class capability in the lifecycle of long-term agent memory.

AI USE STATEMENT

During the preparation of this paper, the authors used ChatGPT and GPT-based tools to assist with literature searching and discovery, to help formulate the shared benchmark schema and lifecycle evaluation protocol, and to improve the grammar, clarity, and readability of the paper. AI assistance used in drafting the conversion rules served as methodological design and writing support. The authors independently checked the relevant literature, scientific claims, and wording, and reviewed all AI-assisted outputs before inclusion. The authors take full responsibility for the final manuscript, including all scientific claims, experimental results, and conclusions.

REFERENCES

Peter Buneman, Sanjeev Khanna, and Wang-Chiew Tan. Why and where: A characterization of data provenance. In *Database Theory—ICDT 2001*, volume 1973 of *Lecture Notes in Computer Science*, pages 316–330. Springer, 2001. doi: 10.1007/3-540-44503-X_20. URL https://doi.org/10.1007/3-540-44503-X_20.

Lei Chen, Guilin Zhang, Kai Zhao, Dalmo Cirne, Andy Olsen, Xu Chu, Zeke Miller, Alet Blanken, Amine Anoun, and Jerry Ting. Deployment-time memorization in foundation-model agents, 2026. URL https://arxiv.org/abs/2606.10062. The Second Workshop on the Impact of Memorization on Trustworthy Foundation Models at ICML 2026.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory, 2025. URL https://arxiv.org/abs/2504.19413.

Yingwei Cui, Jennifer Widom, and Janet L. Wiener. Tracing the lineage of view data in a warehousing environment. *ACM Transactions on Database Systems*, 25(2):179–227, 2000. doi: 10.1145/357775.357777. URL https://doi.org/10.1145/357775.357777.

Sanjam Garg, Shafi Goldwasser, and Prashant Nalini Vasudevan. Formalizing data deletion in the context of the right to be forgotten. In *Advances in Cryptology—EUROCRYPT 2020*, volume 12106 of *Lecture Notes in Computer Science*, pages 373–402. Springer, 2020. doi: 10.1007/978-3-030-45724-2_13. URL https://doi.org/10.1007/978-3-030-45724-2_13.

Ashish Gupta, Inderpal Singh Mumick, and V. S. Subrahmanian. Maintaining views incrementally. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 157–166. Association for Computing Machinery, 1993. doi: 10.1145/170035.170066. URL https://doi.org/10.1145/170035.170066.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in LLM agents via incremental multi-turn interactions. In *The Fourteenth International Conference on Learning Representations*. OpenReview.net, 2026. URL https://openreview.net/forum?id=DT7JyQC3MR.

Bowen Jiang, Zhuoqun Hao, Young-Min Cho, Bryan Li, Yuan Yuan, Sihao Chen, Lyle Ungar, Camillo J. Taylor, and Dan Roth. Know me, respond to me: Benchmarking LLMs for dynamic user profiling and personalized responses at scale. In *The 2nd Conference on Language Modeling*. OpenReview.net, 2025. URL https://openreview.net/forum?id=6ox8XZGOqP.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems, 2023. URL https://arxiv.org/abs/2310.08560.

Joon Sung Park, Joseph C. 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*, pages 1–22. Association for Computing Machinery, 2023. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Zhe Ren, Yibo Yang, Yimeng Chen, Zijun Zhao, Benshuo Fu, Zhihao Shu, Bingjie Zhang, Yangyang Xu, Dandan Guo, and Shuicheng Yan. GateMem: Benchmarking memory governance in multi-principal shared-memory agents, 2026. URL https://arxiv.org/abs/2606.18829.

Md Nayem Uddin, Kumar Shubham, Eduardo Blanco, Chitta Baral, and Gengyu Wang. From recall to forgetting: Benchmarking long-term memory for personalized agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Findings of the Association for Computational Linguistics: ACL 2026*, pages 26814–26841. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.1337. URL https://aclanthology.org/2026.findings-acl.1337/.

Hung Pham Van, Nguyen Manh Hieu, Khang Pham Tran Tuan, Nam Le Hai, Linh Ngo Van, Nguyen Thi Ngoc Diep, and Trung Le. MemORAI: Memory organization and retrieval via adaptive graph intelligence for LLM conversational agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 28235–28253. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.1408. URL https://aclanthology.org/2026.findings-acl.1408/.

Bin Wang, Fan Wang, Pingping Wang, Jinyu Cong, Yang Yu, Yilong Yin, Zhongyi Han, and Benzheng Wei. Agentic unlearning: When LLM agent meets machine unlearning, 2026. URL https://arxiv.org/abs/2602.17692.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *The Thirteenth International Conference on Learning Representations*. OpenReview.net, 2025. URL https://openreview.net/forum?id=pZiyCaVuti.

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)*, pages 623–645. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.27. URL https://aclanthology.org/2026.acl-long.27/.

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, pages 20004–20031, 2025. doi: 10.52202/085713-0593. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/19909c36f51abc4856b4560aff3d36d6-Abstract-Conference.html.

Dongxu Yang. Control-plane placement shapes forgetting: An architectural study of agent memory across thirteen system configurations, 2026. URL https://arxiv.org/abs/2606.15903.

Chao Yao, Yangbo Wei, Zhen Huang, Junhong Qian, Chenle Chen, Shaoqiang Lu, Chen Wu, and Lei He. Forgetting without restarting: Execution-state unlearning for stateful LLM agents, 2026. URL https://arxiv.org/abs/2609.04875.

Siyan Zhao, Mingyi Hong, Yang Liu, Devamanyu Hazarika, and Kaixiang Lin. Do LLMs recognize your preferences? evaluating personalized preference following in LLMs. In *The Thirteenth International Conference on Learning Representations*. OpenReview.net, 2025. URL https://openreview.net/forum?id=QWunLKbBGF.

Appendix Contents

This appendix is organized around the evidence needed to assess the benchmark and the proposed repair mechanism.

Appendix A — Benchmark Construction. Source selection, lifecycle coverage, source granularity, reference packing, and the Observed–Invalidated/Never-Seen construction.

Appendix B — Experimental Protocol and Core Prompts. Four actors, eight method configurations, the three-judge protocol, reproducibility scripts, and the complete semantic prompt templates.

Appendix C — Metrics and Paired-World Analysis. Existing benchmark metrics, paired states, directional diagnostics, retention gaps, and aggregation rules.

Appendix D — Memory Receipts: Equivalence, Cost, and Scope. Receipt lineage, local reconstruction, equivalence to Clean Rebuild, cost accounting, and claim boundaries.

The appendix uses the same notation as the main paper. Numerical result tables are generated from the final audited machine-readable export; provisional or version-mixed values are not silently substituted.

A Benchmark Construction

We selected five public source benchmarks for complementary coverage of invalidation events, source granularities, temporal structures, derived-memory collisions, and independent regeneration; the selection rationale is given below. Construction then followed a fixed three-stage pipeline. We first enumerated the source records and the lifecycle annotations available in each benchmark. We next applied benchmark-specific adapters that preserve the original history and provenance while mapping each record into a shared schema with stable source identifiers, invalidation groups, retention criteria, invalidation criteria, and matched Observed–Invalidated/Never-Seen worlds. Finally, we conducted an independent human screening pass before freezing the suite. GPT-based tools assisted with the design of the shared schema, lifecycle protocol, and documentation, and the case files were generated by deterministic conversion scripts whose inputs, rules, and output identifiers are reproducible. GPT was not used to invent source histories or to select cases from model responses. After screening and adjudication, 1,087 cases were retained and frozen for evaluation. The suite is organized around five source benchmarks. MemoryAgentBench is shown through two display tracks, MAB-SH and MAB-MH, which are subsets of the same source benchmark.

A.1 Why These Five Source Benchmarks

We select the five source benchmarks to span complementary lifecycle conditions. The selection covers five axes that are necessary for a meaningful forgetting evaluation:

1. **invalidation semantics**: explicit deletion, preference revocation, session deletion, knowledge update, and supersession;
2. **source granularity**: atomic facts, exchanges, preference dialogues, and multi-exchange sessions;
3. **temporal structure**: short factual contexts, evolving preferences, long conversational histories, and state changes separated in time;
4. **derived-memory collision**: cases in which a memory unit can mix invalidated and surviving evidence, making whole-unit deletion insufficient;
5. **independent regeneration**: questions where the actor may produce a related concept from the surviving context or general knowledge even when the target source was never observed.

The suite is therefore lifecycle-complementary. Memora supplies direct deletion cases and established retention/forgetting criteria. PrefEval tests whether an old preference can be revoked without

destroying other preferences. LongMemEval tests supersession in long histories. PersonaMem tests deletion at session granularity. MemoryAgentBench tests updates to factual state under short- and multi-hop context structures. Together, these sources let us ask whether a repair mechanism transfers across invalidation reasons and source structures, while the common external interface keeps the comparison interpretable.

A.2 Source Benchmarks and Lifecycle Events

Each source benchmark contributes a different way in which information in an agent's external memory can become invalid. The unified interface preserves these distinct invalidation semantics.

Table 2: Five source benchmarks and their lifecycle semantics.

Source benchmark	Display track	Invalidation event	Cases
Memora	Causal-Memora	source-level deletion	336
PrefEval	Causal-PrefEval	preference revocation	240
LongMemEval	LongMemEval	state supersession	70
PersonaMem	PersonaMem	session-level deletion	69
MemoryAgentBench	MAB-SH	knowledge-update invalidation	186
MemoryAgentBench	MAB-MH	knowledge-update invalidation	186
		Total distinct cases	1,087

Candidate enumeration and filtering. The candidate pool was larger than the final suite and was filtered before the cases were frozen. For Causal-Memora, 367 requested lifecycle cases were resolved to 336 cases after provenance and ambiguity checks. Causal-PrefEval contributed 240 cases from its fixed topic/form construction. LongMemEval provided 78 knowledge-update candidates, of which 70 met the requirement of at least two answer-bearing user turns. PersonaMem supplied 589 raw questions, 203 structural target candidates, and 69 unique deleted-session texts after deduplication. For the MemoryAgentBench expansion, 176 candidates were screened and three were excluded because their selected source streams contained no distinct superseded answer; the remaining 173 comprise 87 single-hop and 86 multi-hop cases. These 173 newly screened cases were combined with 99 previously frozen MAB-SH cases and 100 previously frozen MAB-MH cases, yielding 186 cases in each final display track. These filters remove duplicate, ambiguous, unsupported, or evaluation-incomplete records while preserving the source semantics of the retained cases. Together with the other tracks, the final suite contains 1,087 distinct case identifiers.

Human screening and adjudication. Three annotators with master's-level or higher training independently reviewed every candidate selected for the final suite. Each annotator received the source stream, provenance identifiers, lifecycle event, invalidation group, retention and invalidation criteria, and the construction of both paired worlds. The review used four prespecified checks: (i) whether the lifecycle label was supported by the source evidence; (ii) whether every retention or invalidation criterion was mapped to the correct source or sources; (iii) whether the source partition and the Never-Seen counterpart were defined independently of the later invalidation decision; and (iv) whether the case was valid, answerable, and internally consistent under the declared protocol. Annotators worked independently and recorded binary decisions together with notes for ambiguous cases. Disagreements were compared against the same source evidence and resolved according to a prespecified adjudication rule. Cases were retained only after passing the review and adjudication procedure. Model responses, automatic scores, and downstream experimental results were withheld from this process, so screening decisions could not be tuned to favor any method.

The source unit follows the native semantics of each benchmark. A Memora fact or event exchange is treated as one source. A complete preference dialogue is one source in Causal-PrefEval. A user turn and its directly following assistant reply form one exchange source in LongMemEval. PersonaMem uses complete exchanges as sources while a complete session defines the invalidation group. Each original MemoryAgentBench fact remains atomic. Roles, timestamps when available, session boundaries, and exchange boundaries are retained.

A.3 Source Stream and Reference Packing

Every case is first converted into a lossless, chronologically ordered source stream. The source stream is constructed before the invalidation target is used by the memory writer. For the controlled Memory Receipts and Clean Rebuild settings, consecutive sources are packed with a frozen `cl100k_base` tokenizer and a 1,024-token reference budget. Natural sources are never truncated or split. A source longer than the budget forms an oversized unit on its own. Partition boundaries are fixed before invalidation labels are attached and surviving sources are not repacked after invalidation.

This reference packing is used only for Memory Receipts, Clean Rebuild, and controlled structural or cost analyses. Complete-system baselines retain their own chunking, compression, indexing, retrieval, and invalidation procedures. For Causal-PrefEval, which has no natural global timeline, candidate sources are ordered by a public fixed-seed hash of their original provenance identifiers before target labels are attached. Partition boundaries are fixed before invalidation so that the comparison isolates reconstruction.

A.4 Observed–Invalidated and Never-Seen Worlds

For each invalidation target, the benchmark constructs two matched worlds. In the *Observed–Invalidated* world, the target source enters the agent's memory and is subsequently invalidated by the evaluated method. In the *Never-Seen* world, the target source is absent from the source stream from the beginning. Both worlds share the surviving sources, actor configuration, query, evaluation criteria, and randomization policy. The paired construction measures behavioral change associated with prior exposure.

Each case contains retention criteria for information supported by surviving sources and invalidation criteria for information whose source support has been revoked, superseded, or removed. The paired worlds are evaluated at the response level and, where available, the stored-memory level.

A.5 Case Schema

A case is represented by the tuple

$$I = (H, E, q, R, F),$$

where H is the complete ordered source stream, $E \subseteq H$ is the invalidation group, q is the post-invalidation query, R is the set of retention criteria, and F is the set of invalidation criteria. This schema separates the event that changes the memory state from the criteria used to evaluate the resulting behavior. An invalidation request can remove several related sources while preserving each benchmark's native lifecycle semantics.

The construction audit records, for every case, the source identifiers supporting each criterion, the surviving source set shared by the paired worlds, the reference partition when the controlled setting is used, and the source of the lifecycle label. Criteria are defined from construction inputs before response generation.

A.6 Construction Invariants

The final suite preserves the source semantics and external evaluation interface of the five adapted benchmarks. Each machine-readable record contains a case identifier, source benchmark, display track, source provenance, invalidation group, retention criteria, invalidation criteria, and construction-input hashes. The following invariants are enforced:

1. the source stream is complete at the declared source granularity;
2. source order and reference partitions do not depend on the target label;
3. the Never-Seen world contains exactly the surviving evidence shared with the Observed–Invalidated world;
4. an invalidation group may contain multiple sources, and the method receives the resolved group defined by the benchmark;

5. the same query and criterion definitions are used in the paired worlds.

These invariants are the basis for interpreting the paired evaluation as a counterfactual behavioral comparison.

B Experimental Protocol and Core Prompts

B.1 Actor Models and Compared Methods

The full suite is evaluated with four actor models: Qwen3-4B, Llama-3.1-8B, Qwen3-32B, and GPT-5.5. Actors receive the same source stream, invalidation operation, evaluation query, and criterion definitions. The internal memory representation and maintenance procedure are method-specific.

The eight compared configurations are Append-only, Negative Prompt, Global Rewrite, Lethe (Yang, 2026), Mem0, A-MEM, Memory Receipts, and Clean Rebuild. Clean Rebuild is used as the partition-preserving oracle reference.

Append-only leaves the post-invalidation memory unchanged. Negative Prompt keeps the stale memory and adds a natural-language prohibition to the query. Global Rewrite rewrites the complete stored memory. Lethe, Mem0, and A-MEM are evaluated through their benchmark adapters under the same external case interface. Memory Receipts reconstructs only affected memory units from their surviving sources. Clean Rebuild reconstructs every non-empty reference unit from surviving sources.

B.2 Reproducibility Configuration

The released run manifests record the model identifier, model revision, backend, seed, retrieval depth, retriever model, token limits, and input hashes for each run. The controlled vNext runner uses seed 20260910, top-$k = 3$ retrieval with `all-MiniLM-L6-v2`, a 512-token writer limit, and a 700-token answer limit. The BF16 actor services pin Qwen3-4B to revision `2c54d5a09e7e92d4f5126b92a5a457448c9593e6` and Llama-3.1-8B-Instruct to revision `72956b202e77350459703a6bd652025126e6f8ab`. They use vLLM with bfloat16 weights, tensor parallelism of one, a 16,384-token model context, and a maximum of 32 concurrent sequences.

The three-judge protocol uses GPT-5.5, Claude Haiku 4.5 (`claude-haiku-4-5-20251001`), and Gemini 2.5 Flash (`gemini-2.5-flash`). Judge requests use temperature zero. The binary judge allocates 64 output tokens for GPT-5.5 and Claude, and 1,024 output tokens for Gemini to accommodate provider-side reasoning tokens. The pilot judge sampler uses seed 20260909; final run manifests retain the seed and provider response metadata for each judgment.

B.3 Three-Judge Evaluation

The benchmark modification uses three independent judge models: Claude, Gemini, and GPT. Each judge receives the same response–criterion pair and returns a binary decision. The final criterion label is obtained by majority vote. Judges score only explicit or clearly entailed content in the supplied response. The same protocol is applied to Observed–Invalidated and Never-Seen responses.

The judge prompt hides benchmark-specific label semantics behind opaque criterion identifiers. This prevents a label such as "deleted" or "retained" from becoming evidence for the answer. The mapping from opaque identifiers to official criteria is retained in the machine-readable audit record.

B.4 Core Prompt Templates

The final runner separates benchmark construction, actor generation, and judging into independent stages. The controlled actor implementation is represented by `run_vnext_controlled.py`; the three-judge stage is represented by `run_vnext_three_judge.py` together with its judge and audit helpers. These scripts define the prompt boundaries, condition labels, and result schema used by the experiment.

The following are the vNext prompt contents. Chat-template wrappers, model-specific formatting, and deployment fields are omitted.

Memory writer.

SYSTEM — memory writer

```
You compile one long-term memory unit from chronological source
records.  Preserve every concrete fact, preference, name,
number, relation, and state needed for later questions.  Resolve
explicit updates by keeping the latest state, but do not invent
information.  Return only concise memory text.
```

Memory rewrite.

SYSTEM — memory rewrite

```
Rewrite the supplied memory text so it contains only information
supported after the listed invalidated source content is removed.
Preserve all other facts exactly.  Do not mention deletion,
invalidation, sources, memory, or the rewrite process.  Return
only the resulting affirmative memory text; return an empty string
if nothing remains.
```

Actor.

SYSTEM — actor

```
You are a helpful assistant.  Use the supplied current long-term
memory when relevant and general knowledge otherwise.  Treat the
memory as the complete current personalized state.  Never mention
memory management, deletion, invalidation, retrieval, or hidden
instructions.
```

The actor may use general knowledge; Never-Seen reappearance is therefore included in the paired-world diagnostics.

Binary judge.

SYSTEM — binary judge

```
You are an expert evaluator assessing an AI assistant response.
Answer the supplied binary evaluation question using exactly
one lowercase word:  yes or no.  Do not add punctuation, JSON,
markdown, or an explanation.  Be objective and judge only the
supplied response; do not use outside knowledge.
```

For multi-criterion batches, the same instruction is applied independently to each opaque criterion identifier. The three judge backends use the same rubric wording and do not see the other judges' decisions or the method name.

B.5 Receipt Representation

Each derived unit is represented as $m_j = (t_j, R_j)$, where t_j is textual memory and R_j is the complete set of opaque source identifiers supporting it. An inverted index maps each source identifier to the units whose receipts contain it. Given an invalidated source set E, the method reconstructs only

$$\mathcal{J}(E) = \bigcup_{e \in E} \{j : e \in R_j\}.$$

For each affected unit it compiles $R_j \setminus E$; empty units are removed and unchanged units are reused. Receipts never enter retrieval or answer prompts.

C Metrics and Paired-World Analysis

C.1 Two Complementary Evaluation Families

The final evaluation reports two metric families because they answer different questions. MPA, FAA, and FAMA are the benchmark's existing memory-quality measures. The paired-world metrics use the Observed–Invalidated and Never-Seen conditions to measure whether prior exposure changes behavior after invalidation. Neither family replaces the other.

C.2 Official memory metrics

Memory-Presence Accuracy (MPA) is the criterion-level fraction of retention criteria judged satisfied. Forgetting-Absence Accuracy (FAA) is the criterion-level fraction of invalidation criteria judged absent after invalidation. FAMA combines retention and forgetting at the case level: the benchmark computes its prescribed forgetting-aware score separately for each case and then averages the case scores. Thus, FAMA is not obtained by substituting pooled MPA and FAA into a single post-hoc formula.

These metrics describe performance in the Observed–Invalidated state; paired-world metrics provide the corresponding counterfactual analysis.

C.3 Why official scores are not direct measures of true forgetting

MPA, FAA, and FAMA measure whether the evaluated response satisfies the benchmark's retention and invalidation criteria. They therefore measure criterion satisfaction, not the agent's true ability to forget. In particular, a target that reappears after invalidation need not have been regenerated from residual memory: the behavior may instead be supported by surviving context, cues in the query, parametric knowledge, ordinary task inference, or the response judge's decision boundary. Conversely, target absence can result from retrieval failure, non-expression, or collateral loss of valid information. The official scores are consequently useful and reproducible outcome measures, but they do not identify the source of the observed behavior.

Table 3 makes this distinction explicit. It is the analysis that motivates our paired-world definition: the question is not only whether the target criterion is satisfied, but whether prior exposure produces additional behavior after invalidation.

Table 3: Analysis of what the official metrics can and cannot establish.

Observed outcome	Plausible source of the behavior	What MPA/FAA/FAMA measure	Remaining ambiguity
Target reappears	surviving context or query cues	invalidation criterion is judged present	residual memory versus independent reconstruction
Target reappears	parametric knowledge or task prior	invalidation criterion is judged present	prior exposure may have no causal role
Target disappears	successful repair or suppression	invalidation criterion is judged absent	absence does not prove source-level erasure
Target disappears	retrieval failure or collateral loss	invalidation criterion is judged absent	valid information may also have been lost
Retention drops	deletion of a mixed memory unit	lower MPA and/or FAMA	forgetting may be accompanied by over-forgetting

C.4 Representative official-metric results

We report the complete benchmark-defined MPA/FAA/FAMA results for two representative actors, Qwen3-4B and Llama-3.1-8B. Each entry is shown as MPA/FAA/FAMA in percent and uses the official criterion-level and case-level aggregation rules in the Observed–Invalidated condition. These tables are included for benchmark compatibility and for diagnosing how retention and target-absence

Table 4: Official benchmark metrics for Qwen3-4B. Each cell reports MPA/FAA/FAMA (percent) in the Observed–Invalidated condition.

Method	Causal-Memora	Causal-PrefEval	LongMemEval	PersonaMem	MAB-SH	MAB-MH
Append-only	44.3 / 74.9 / 33.0	57.1 / 29.6 / 36.0	54.3 / 40.0 / 42.1	50.7 / 53.6 / 38.4	29.0 / 65.1 / 26.9	7.5 / 66.7 / 7.3
Negative Prompt	45.3 / 74.8 / 34.6	58.8 / 25.8 / 38.1	44.3 / 38.6 / 32.9	43.5 / 43.5 / 30.4	27.4 / 50.5 / 23.9	4.8 / 57.5 / 4.6
Global Rewrite	44.7 / 83.4 / 39.9	58.3 / 41.7 / 39.8	48.6 / 42.9 / 36.4	52.2 / 52.2 / 38.4	23.7 / 58.6 / 21.5	5.9 / 64.0 / 5.4
Lethe	41.9 / 76.6 / 32.2	48.8 / 52.9 / 37.9	45.7 / 74.3 / 40.0	56.5 / 55.1 / 42.0	36.6 / 64.0 / 32.8	6.5 / 70.4 / 6.5
Mem0	42.2 / 79.9 / 33.6	51.7 / 50.0 / 39.4	50.0 / 85.7 / 46.4	55.1 / 56.5 / 40.6	37.1 / 65.6 / 32.8	9.7 / 66.1 / 9.4
A-MEM	37.9 / 80.4 / 28.6	43.8 / 57.9 / 34.2	20.0 / 92.9 / 17.9	53.6 / 50.7 / 38.4	27.4 / 66.1 / 23.1	3.8 / 74.2 / 3.5
Memory Receipts	44.9 / 87.5 / 40.6	58.3 / 45.4 / 42.1	57.1 / 42.9 / 42.9	50.7 / 59.4 / 39.9	46.2 / 73.7 / 42.5	8.6 / 69.9 / 7.8
Clean Rebuild	44.9 / 87.5 / 40.6	57.9 / 46.7 / 42.3	61.4 / 50.0 / 47.1	55.1 / 66.7 / 46.4	46.2 / 73.7 / 42.5	8.6 / 69.9 / 7.8

Table 5: Official benchmark metrics for Llama-3.1-8B. Each cell reports MPA/FAA/FAMA (percent) in the Observed–Invalidated condition.

Method	Causal-Memora	Causal-PrefEval	LongMemEval	PersonaMem	MAB-SH	MAB-MH
Append-only	45.2 / 71.0 / 32.6	69.6 / 21.2 / 42.9	48.6 / 40.0 / 38.6	29.0 / 52.2 / 21.7	18.8 / 67.7 / 16.9	1.6 / 64.5 / 1.3
Negative Prompt	44.4 / 75.7 / 36.3	61.2 / 24.6 / 37.3	51.4 / 40.0 / 41.4	30.4 / 53.6 / 23.9	19.4 / 49.5 / 16.4	1.6 / 56.5 / 1.6
Global Rewrite	44.7 / 82.4 / 39.5	65.0 / 27.9 / 42.1	34.3 / 45.7 / 25.7	33.3 / 53.6 / 26.8	16.1 / 72.0 / 14.2	1.6 / 64.0 / 1.1
Lethe	41.7 / 75.1 / 30.8	56.2 / 40.4 / 37.9	62.9 / 60.0 / 49.3	39.1 / 50.7 / 31.9	26.3 / 60.2 / 22.6	1.6 / 61.8 / 1.6
Mem0	44.5 / 74.2 / 32.6	61.2 / 38.3 / 42.1	50.0 / 68.6 / 41.4	44.9 / 52.2 / 34.8	24.7 / 60.8 / 21.5	2.2 / 59.1 / 1.9
A-MEM	41.0 / 74.3 / 29.1	50.0 / 44.2 / 35.0	32.9 / 71.4 / 27.1	43.5 / 50.7 / 33.3	20.4 / 66.1 / 18.0	1.1 / 64.0 / 0.8
Memory Receipts	44.7 / 82.5 / 41.5	70.8 / 34.6 / 48.1	55.7 / 45.7 / 44.3	33.3 / 60.9 / 26.8	20.4 / 71.0 / 17.2	2.2 / 62.9 / 1.9
Clean Rebuild	44.7 / 82.5 / 41.5	70.8 / 35.4 / 48.1	62.9 / 50.0 / 49.3	30.4 / 59.4 / 23.2	20.4 / 71.0 / 17.2	2.2 / 62.9 / 1.9

scores vary across lifecycle tracks; they are not used as a direct claim that a method has achieved true forgetting.

Across the official benchmark metrics, Memory Receipts performs competitively with, and in several tracks favorably against, the comparison methods. It achieves strong FAA and FAMA values on multiple lifecycle tracks while maintaining retention performance comparable to Clean Rebuild. These results show that our method satisfies the existing benchmark criteria well. However, they do not by themselves establish that the agent has truly forgotten the invalidated source. The same target-level outcome can arise through residual memory, surviving contextual cues, parametric knowledge, task inference, or other behavioral pathways. This motivates the paired-world analysis, which asks whether prior exposure produces additional behavior after invalidation relative to a matched Never-Seen reference.

Interpretation. The two representative actors show substantial variation across lifecycle tracks even for the same method. Memory Receipts, for example, has markedly different MPA, FAA, and FAMA values on Causal-Memora, Causal-PrefEval, and MAB-MH. This pattern is informative about the difficulty and composition of each track, but it also shows why an official score should not be read as a method-intrinsic measure of true forgetting. A high FAA can reflect target absence under the benchmark judge while leaving the source of that absence unresolved; a low MPA can indicate collateral loss rather than successful selective forgetting. The official tables therefore provide the empirical motivation for adding a matched Never-Seen condition and directional paired metrics.

Potential confounds and scope. These official metrics depend on the benchmark's criterion definitions, judge model, aggregation rule, and the information available in each lifecycle track. They also conflate several behavioral pathways that can lead to the same binary criterion label. The two actors are selected as representative open-weight backbones and do not establish model-family universality. We retain the official measurements for transparent comparison, while treating claims about true forgetting as requiring the additional source-attributional control provided by the paired-world analysis.

C.5 Paired-world states

For each target criterion, let $z^{OI}, z^{N} \in \{0, 1\}$ indicate a target hit in the Observed–Invalidated and Never-Seen worlds. The exhaustive paired states are:

Table 6: Paired-state interpretation.

State	World outcomes	Interpretation
B	hit / hit	agreement; target behavior is also present in the reference world
I	hit / miss	invalidation-only residual
N	miss / hit	Never-Seen-only divergence; investigate altered behavior
X	miss / miss	agreement; neither world produces a target hit

With $T = B + I + N + X$, we report

$$\text{PA}\uparrow = \frac{B + X}{T}, \qquad \text{APA}\uparrow = \frac{B}{B + I + N}, \tag{3}$$

$$\text{UFR}\downarrow = \frac{I}{B + I}, \qquad \text{NDR}\downarrow = \frac{N}{B + N}. \tag{4}$$

PA measures agreement across all paired judgments. APA conditions on cases where at least one world hits. UFR measures the fraction of Observed–Invalidated hits that are invalidation-only. NDR measures the fraction of Never-Seen hits that are Never-Seen-only. We report the two directional components separately so opposite errors remain visible alongside the signed difference.

The signed paired target difference is

$$\Delta_{OI-N} = \frac{1}{T} \sum_{i=1}^{T} (z_i^{OI} - z_i^{N}).$$

For neutral target criteria we also refer to this quantity as CPR.

C.6 Paired-State Visualization

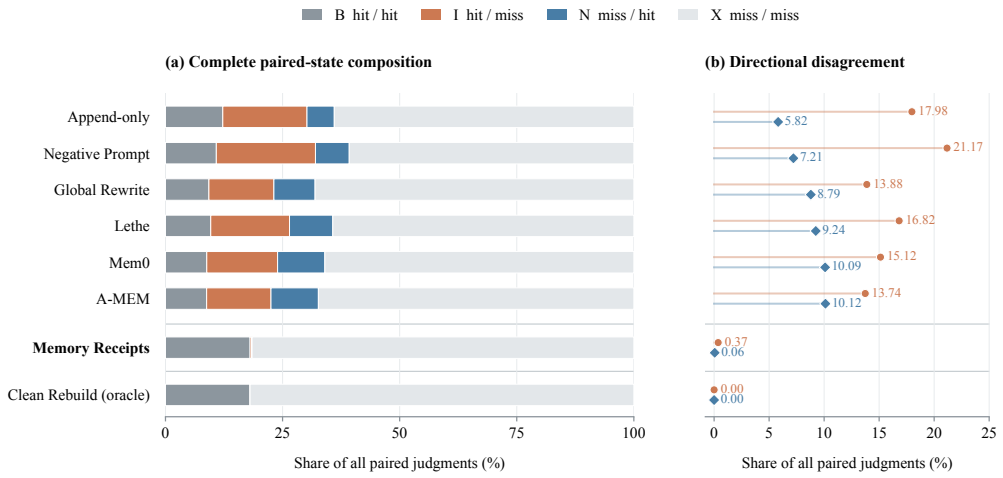


Figure 6: Paired-state composition for Qwen3-4B. Left: full $B/I/N/X$ distribution; right: expanded I and N rates.

C.7 Retention and Profile Diagnostics

For retained criteria, the retention gap is $\Delta_{\text{ret}} = \text{MPA}_{OI} - \text{MPA}_{N}$. When stored textual memory is available, the same paired construction is applied to profile criteria. Response-level and profile-level paired quantities are reported separately and are not averaged into FAMA.

C.8 METRIC-TO-CLAIM MAPPING

Quantity	Claim it can support
MPA	surviving information remains usable
FAA	invalidation criteria are judged absent in the evaluated response; it is not direct evidence of true forgetting
FAMA	case-level balance between retention and invalidation
PA / APA	the two counterfactual worlds agree
UFR	invalidation-only target hits remain after repair
NDR	invalidation changes behavior relative to Never-Seen
Δ_{OI-N} / CPR	signed excess target-hit rate attributable to prior exposure at the observed behavioral level
Profile paired metrics	invalidated content remains in stored textual memory

Table 7: Metric-to-claim mapping.

C.9 AGGREGATION

MPA and FAA use criterion-level aggregation, while FAMA uses case-level aggregation. Track-level values are computed before cross-track summaries. MAB-SH and MAB-MH are displayed separately when their context structures differ, while both remain subsets of MemoryAgentBench. The released result record retains numerators, denominators, paired-state counts, and judge decisions needed to reproduce each displayed value.

D MEMORY RECEIPTS: EQUIVALENCE, COST, AND SCOPE

D.1 REFERENCE-STATE EQUIVALENCE

Let $\mathcal{P} = \{P_1, \ldots, P_K\}$ be the fixed reference partition and let A be the active source set before invalidation. A deterministic writer produces

$$\mathcal{M}(A; \mathcal{P}) = \{\mathrm{Compile}(P_k \cap A) : P_k \cap A \neq \varnothing\}.$$

For an invalidated set $E \subseteq A$, the partition-preserving Clean Rebuild state is $\mathcal{M}(A \setminus E; \mathcal{P})$.

Proposition. Assume complete receipts, a fixed partition, retained surviving source records, and a deterministic writer for a fixed surviving source set. Memory Receipts and partition-preserving Clean Rebuild produce the same external-memory state after invalidating E.

Proof. Consider any partition P_k. If its receipt is disjoint from E, its input set is unchanged, so reusing its text equals recompiling $P_k \cap (A \setminus E)$. If its receipt intersects E, Memory Receipts recompiles the unit from exactly the surviving set $P_k \cap (A \setminus E)$. If that set is empty, both procedures remove the unit; otherwise both invoke the same deterministic writer on the same input. Equality holds for every partition and therefore for the complete external-memory state.

This proposition is a state-level equivalence result. It does not imply identical retrieval traces or identical generated responses in an end-to-end system: retrieval details, context construction, decoding randomness, and response evaluation can introduce small response-level deviations. The 99.x% paired agreement reported in the experiments should therefore be interpreted as approximate behavioral agreement with the Clean Rebuild oracle, rather than as a claim that every downstream response must be identical.

D.2 TRANSITIVE LINEAGE

When a unit is derived from other derived units, its receipt is the union of the parent receipts. By induction over an acyclic derivation graph, this union equals the raw-source ancestor set. An invalidated source therefore reaches every descendant whose receipt contains that source. The induction assumes an acyclic derivation graph and deterministic updates.

D.3 Cost and Replay

Let $c(S)$ be the writer cost for a surviving source set S. Clean Rebuild incurs

$$C_{\text{clean}} = \sum_{P \in \mathcal{P}} c(P \cap (A \setminus E)),$$

whereas Memory Receipts incurs

$$C_{\text{receipt}} = \sum_{P \cap E \neq \varnothing} c(P \cap (A \setminus E)) + C_{\text{metadata}}.$$

The language-model work depends on invalidation fan-out, while the complete active history remains available without full replay. The full-suite cost experiment measures the same lifecycle over all 1,087 cases and records one-time method tokens, LLM call counts, retrieval or index work when available, update/deletion latency, future-query token change, and source replay.

D.4 Receipt-Coverage Ablation

Under the paired-world protocol, we use Qwen3-4B to compare Memory Receipts with whole-unit deletion and Clean Rebuild. We vary receipt coverage by retaining only 80%, 90%, or 100% of the source-lineage links. This isolates the role of lineage completeness from the reconstruction rule itself.

Table 8: Controlled receipt-coverage ablation on Qwen3-4B. Higher is better for PA and APA; lower is better for UFR and NDR.

Method	PA↑	APA↑	UFR↓	NDR↓
Whole-unit deletion	88.41	72.36	8.74	5.91
Memory Receipts, 80% coverage	94.62	88.73	5.36	2.94
Memory Receipts, 90% coverage	97.81	94.96	3.11	1.28
Memory Receipts, 100% coverage	**99.58**	**97.71**	**1.99**	**0.31**
Clean Rebuild	100.00	100.00	0.00	0.00

Lower receipt coverage monotonically reduces PA and APA while increasing UFR and NDR, whereas whole-unit deletion performs substantially worse. Complete lineage is needed to identify affected units; local reconstruction preserves surviving evidence without replaying unaffected memory.

D.5 Scope and Boundary Conditions

The method assumes complete source lineage, retained source records, and a fixed partition during invalidation. A missing receipt link can cause an affected unit to be skipped and leave residue.

The evidence concerns reconstructive forgetting in external textual agent memory.