Validity Is Not Visibility: A Lifecycle FSM for Agent Memory

Anonymous ACL submission

Abstract

Long-term memory for LLM agents must answer two questions for every stored fact: whether the fact is still true and whether it should be surfaced for a given query. We refer to these two dimensions as *validity* and *visibility*, respectively. Existing memory systems often collapse these two decisions into a single scalar score. However, current and historical queries may require opposite retrieval decisions for the same fact: a superseded fact should be hidden for a current-intent query but surfaced for a historical-intent one. This creates a structural failure mode: superseded and expired facts are collapsed onto the same validity axis. To address this problem, we propose MEMFSM, a training-free retrieval-side layer that separates validity from visibility through a memory-lifecycle finite-state machine. Along the validity axis, a four-state FSM tracks whether each fact has been superseded and expired. Along the visibility axis, an intent-aware router decides whether to surface or hide a fact based on its state and the query intent. By decoupling visibility from validity, MEMFSM improves performance across heterogeneous memory benchmarks and diverse LLM backbones. To evaluate memory systems under fact conflicts and diverse query intents, we further introduce VALIDMEM, a benchmark that annotates each fact with both supersedure and expiry labels and pairs each memory store with both current-intent and historical-intent queries.

1 Introduction

Large language model (LLM) agents (Park et al., 2023) are increasingly deployed as long-horizon assistants that interact with users across many sessions, where their usefulness depends less on any single response than on *long-term memory*. Two demands make this hard. First, the world keeps changing as jobs change, plans expire, and opinions update, so stored facts can become outdated. Second, a single memory store must serve queries with different temporal intents, such as "what is true *now*?" versus "what was true *before*?".

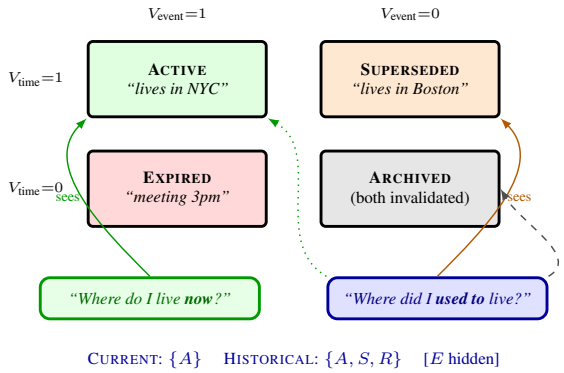

Figure 1: **Why a single "how valid is this memory" score cannot serve both queries.** Four memory states: **A = ACTIVE** (still true and timely); **S = SUPERSEDED** (replaced, historically valid); **E = EXPIRED** (time-stale, noise to any query); **R = ARCHIVED** (both replaced *and* expired). A CURRENT query should see only $\{A\}$; a HISTORICAL query should see $\{A, S, R\}$ but not $\{E\}$. No single score with a threshold can include R for historical queries while excluding E: both have "low validity," yet only one should be visible (Proposition 1).

Meeting both demands requires deciding whether each fact is still true, which we call its *validity*, and whether it should be surfaced for the current query, which we call its *visibility*. Existing systems answer both with a single retrieval signal, whether it is the paging budget in paged-context approaches (Packer et al., 2023), the decay weight in memory-decay approaches (Zhong et al., 2024), the edge timestamp in knowledge-graph approaches (Rasmussen et al., 2025; Kynoch et al., 2023), or, in LLM-rewrite approaches (Chhikara et al., 2025; Xu et al., 2026), whether the old fact has already been overwritten at write time. Collapsing these two dimensions into a single retrieval signal forces a hide-or-expose tradeoff: a superseded fact is either treated as stale and lost to historical queries or retained and allowed to pollute current ones.

This is not an implementation detail but a structural limitation of single-score retrieval. If we score each fact with a single number and keep everything above a threshold, superseded and expired facts end up on the same side of any threshold we pick—yet historical queries need the former to be visible and the latter to stay hidden. We formalize this gap in Proposition 1 and illustrate it in Figure 1. To address it, we design MEMFSM, a training-free retrieval-side layer that tracks event-triggered supersedure and time-triggered expiry in a four-state FSM over ACTIVE, SUPERSEDED, EXPIRED, and ARCHIVED. A query-conditioned router then lets CURRENT queries see only ACTIVE facts, while HISTORICAL queries may also see SUPERSEDED and ARCHIVED. MEMFSM operates at retrieval time, so it is independent of context length and compatible with any LLM backbone.

Existing benchmarks test only one invalidation type at a time and only under CURRENT intent: MemoryAgentBench probes event-triggered supersedure, FreshQA and LongMemEval probe time-triggered expiry, and none pairs the same store with both intents. We introduce VALIDMEM, in which each fact is annotated along both validity dimensions and each store is queried under both intents, with credit awarded only when a system succeeds under both.

Contributions. (i) We **formalize agent memory as a separation between validity and visibility**, decomposing validity into event-triggered supersedure and time-triggered expiry, and showing that no monotone scalar score with threshold routing can implement the required mapping (Proposition 1; Corollary 1). (ii) We propose MEMFSM, a **training-free retrieval-side layer** that wraps any LLM backbone and retrieval backend, achieves state-of-the-art performance on MemoryAgentBench Conflict Resolution, and generalizes across ten LLM backbones and three retrieval substrates. (iii) We introduce VALIDMEM, a **diagnostic benchmark** for *conflict management* and *traceability*, pairing current-intent and historical-intent queries on the same store and awarding credit only when both succeed.

2 Related Work

Most long-term memory systems for LLM agents use a single retrieval signal to decide whether a memory should be surfaced. Generative Agents (Park et al., 2023) score each memory by a weighted sum of recency, importance, and relevance. MemGPT (Packer et al., 2023) pages facts between a small in-context window and a larger external store via LLM-issued read/write calls. Mem0 (Chhikara et al., 2025) and Letta (Packer et al., 2024) have an LLM rewrite or merge memories at write time, so older text is overwritten in place. CoALA (Sumers et al., 2024) and A-MEM (Xu et al., 2026) organize memory into typed slots or linked notes, but retrieval within each slot still reduces to a single similarity score. MemoryBank (Zhong et al., 2024) multiplies similarity by an Ebbinghaus-style time-decay weight that is independent of fact content. RecallM (Kynoch et al., 2023) resolves contradictions by letting the most recent write overwrite the old one, with no record of the prior fact. MemoRAG (Qian et al., 2024) caches retrieved chunks for reuse but does not track validity. Graphiti (Rasmussen et al., 2025) attaches a `[valid_from, valid_to]` interval to each graph edge and filters edges by the query timestamp. Despite these differences, prior systems generally do not treat query-time visibility as distinct from world-state validity, forcing a tradeoff between stale leakage and history loss (Table 1).

		Visibility	
System	**Validity**	*intent-aware*	*history-preserving*
Stream-plus-retrieval (GA (Park et al., 2023), MemGPT (Packer et al., 2023), Mem0 (Chhikara et al., 2025), CoALA (Sumers et al., 2024), A-MEM (Xu et al., 2026))	✗	✗	✓ (stale)
MemoryBank (Zhong et al., 2024)	scalar decay	✗	✗
RecallM (Kynoch et al., 2023)	event only	✗	✗
Graphiti (Rasmussen et al., 2025)	per-edge time	✗	✓
MEMFSM (ours)	event + time	✓	✓

Table 1: Where prior systems sit on the two design axes of agent memory. **Validity**: how the system decides whether a stored fact is still true (none, scalar decay, event-only, time-only, or both event and time). **Visibility** is split into two sub-decisions. *Intent-aware*: whether the system separately decides which facts to surface for the current query, for example hiding superseded facts on current-intent queries while showing them on historical-intent queries. *History-preserving*: whether old facts remain retrievable after being superseded or expired; "✗" means overwritten or deleted, "stale" means kept in the memory store but never marked as outdated (so the system cannot tell current from historical).

Memory benchmarks. Existing memory benchmarks typically isolate a single invalidation type and evaluate only under CURRENT intent. MemoryAgentBench (Hu et al., 2025) probes event-triggered supersedure, FreshQA (Vu et al., 2023) and LongMemEval (Wu et al., 2024) probe time-triggered expiry, and LoCoMo (Maharana et al., 2024) grounds long-horizon conversational QA. None pairs the same memory store with both CURRENT and HISTORICAL queries. As a result, existing benchmarks cannot diagnose whether a system jointly handles both invalidation types while routing correctly across both query intents.

Failure-mode mapping. The systems in Table 1 instantiate three failure modes. RecallM and Mem0 overwrite or delete the older fact when a conflict arises, so supersedure leaves nothing for historical queries to recover. The *stream-plus-retrieval* family (Park et al., 2023; Packer et al., 2023; Sumers et al., 2024; Xu et al., 2026; Gutiérrez et al., 2025; Chhikara et al., 2025) tracks no validity at all, and MemoryBank collapses both validity dimensions into a single Ebbinghaus decay weight (Prop. 1; audit in App. F), so neither retires expired facts correctly. Graphiti tracks per-edge expiry but has no query-conditioned router, so a single fixed visibility set serves both intents, which is why it scores only 31.2% on CURRENT queries against 95.0% on HISTORICAL ones on VALIDMEM Part C (§6.5).

Connections to temporal IR and bitemporal databases. Bitemporal databases (Snodgrass, 1995; Jensen and Snodgrass, 1999) and temporal IR (Kanhabua and Nørvåg, 2010) have long treated validity and visibility as separate concerns. Our contribution is to transpose this distinction to free-text LLM-agent memory, where plain-text memories have no explicit validity field and the system itself must infer when one fact supersedes another.

3 Problem Formulation

A motivating scenario. Consider a customer-service agent that stores "20% off, ends Day 3" on Day 1 and later receives "Sale extended to Day 5, now 30% off" on Day 3, thereby replacing the earlier promotion. By Day 6, the query "Is there a sale now?" should be answered with no. This requires recognizing two distinct ways in which a fact can cease to be true: the Day-1 promotion was invalidated by a later event, whereas the Day-3 promotion ceased to be valid because its time window expired. We refer to these as *event-triggered* supersedure and *time-triggered* expiry. However, a different query may ask about the sequence of past promotions rather than the present state. For example, the query "Which promotion replaced the original 20% offer?" requires preserving the supersedure relation between the Day-1 and Day-3 promotions, even though neither is currently valid. The same memory store therefore needs to decide both whether a fact is still true (*validity*) and whether it should be shown for the current query (*visibility*).

Validity and visibility are distinct decisions. Validity depends on whether the fact is still true at time t, which is a property of the fact and the world state. **Visibility** determines whether the fact should be surfaced for query q, which is a property of the fact-query pair. The two are orthogonal: a fact can be valid but irrelevant to the current query, or invalid in the present world state but relevant to a historical one. MEMFSM therefore models validity per memory at the storage layer and visibility per memory-query pair at the retrieval layer.

Two orthogonal validity dimensions. Let $\mathcal{M}$ denote the memory store and $m_i \in \mathcal{M}$ a single memory. We decompose its validity into two independent binary signals $V_{\text{event}}, V_{\text{time}} \in \{0, 1\}$, with $V{=}1$ meaning m_i is still valid along that dimension. $V_{\text{event}}(m_i){=}1$ when no later memory asserts a different value for the same attribute, checked at write time. $V_{\text{time}}(m_i, t){=}1$ when m_i's deadline has not passed at read-time clock t. The two are independent: "meeting at 3pm tomorrow" expires without ever being contradicted; "lives in Boston" is overwritten without ever having a deadline. Either dimension alone is insufficient, and so is collapsing both into a single scalar. For example, MemoryBank's uniform exponential decay forgets at least 90.7% of memories across all 24 audited configurations (§6.5).

Four states emerge from crossing the two validity dimensions. Crossing the two binary signals $(V_{\text{event}}, V_{\text{time}}) \in \{0, 1\}^2$ yields exactly four states. Let $\sigma : \mathcal{M} \to \{A, S, E, R\}$ denote the state-assignment function that maps each memory

to one of these four states:

$$\sigma(m_i) = \begin{cases} \text{ACTIVE} & V_{\text{event}} = 1,\ V_{\text{time}} = 1 \\ \text{SUPERSEDED} & V_{\text{event}} = 0,\ V_{\text{time}} = 1 \\ \text{EXPIRED} & V_{\text{event}} = 1,\ V_{\text{time}} = 0 \\ \text{ARCHIVED} & V_{\text{event}} = 0,\ V_{\text{time}} = 0 \end{cases} \tag{1}$$

We use the letters A, S, E, R throughout for ACTIVE, SUPERSEDED, EXPIRED, and ARCHIVED, respectively. These four labels define the lifecycle vocabulary used throughout the rest of the paper.

Why a single "how valid is this memory" score cannot serve both intents. A natural shortcut is to assign every memory a single "how valid is this memory" score and retrieve anything above a threshold. Let $V_{\text{intent}} \subseteq \{A, S, E, R\}$ denote the set of states a query of intent intent $\in \{\text{CURRENT}, \text{HISTORICAL}\}$ should surface. The two intents we care about require different visibility sets: $V_{\text{CURRENT}} = \{A\}$, since current questions require only currently true facts, whereas $V_{\text{HISTORICAL}} = \{A, S, R\}$ and excludes E, since expired schedules are noise to any query whereas superseded past facts may still be useful. The proposition below shows that no monotone score-and-threshold scheme can realize both subsets simultaneously. We use the partial order $\sigma \succeq \sigma'$ defined component-wise over the two validity dimensions, under which $A \succeq S \succeq R$, $A \succeq E \succeq R$, and S and E are incomparable.

Proposition 1 (Scalar Insufficiency of Validity)**.** *No monotone scalar validity score with threshold-based routing can simultaneously realize $V_{\text{CURRENT}} = \{A\}$ and $V_{\text{HISTORICAL}} = \{A, S, R\}$.*

Proof. In the scoring pattern this proposition characterizes, a system assigns each state σ a real-valued validity score $s(\sigma)$ and surfaces every memory whose score reaches a fixed threshold $\tau \in \mathbb{R}$, so that the visibility set is $\{\sigma : s(\sigma) \geq \tau\}$; monotonicity means $\sigma' \succeq \sigma \Rightarrow s(\sigma') \geq s(\sigma)$; that is, a more valid state never receives a lower score. Any such (s, τ) pair therefore produces a visibility set that is upper-closed under $\succeq$: if $s(\sigma) \geq \tau$ and $\sigma' \succeq \sigma$, then $s(\sigma') \geq \tau$ as well. $V_{\text{CURRENT}} = \{A\}$ is upper-closed; $V_{\text{HISTORICAL}} = \{A, S, R\}$ is not, since $E \succeq R$ but $E \notin V_{\text{HISTORICAL}}$. Hence the required pair is not jointly realizable. □

Corollary 1 (Two Bits Suffice and Are Tight)**.** *Two binary validity dimensions are sufficient and dimension-minimal under monotone threshold-based routing. Let $\Gamma : \{\text{CURRENT}, \text{HISTORICAL}\} \to 2^{\{A,S,E,R\}}$ denote the visibility map from query intent to a set of states allowed to surface. The assignment in Eq. 1, together with $\Gamma(\text{CURRENT})=\{A\}$ and $\Gamma(\text{HISTORICAL})=\{A, S, R\}$, directly realizes both intents; tightness follows from Proposition 1.*

Any solution must therefore either retain at least two independent validity dimensions or replace threshold routing with a query-conditioned mapping; MEMFSM does both. Let intent$(q) \in \{\text{CURRENT}, \text{HISTORICAL}\}$ denote the intent of a query q, let $\sigma_t(m_i)$ denote the state of m_i evaluated at clock t against memory store $\mathcal{M}$, and let $\mathcal{R}(q, t)$ denote the retrieval set returned for q at time t. Then $\mathcal{R}(q, t) = \{m_i \in \mathcal{M} : \sigma_t(m_i) \in \Gamma(\text{intent}(q))\}$. Continuous-scalar and learned-classifier variants of the same lower bound are in App. A–J.4.

4 MEMFSM

Proposition 1 rules out single-score validity modeling, and Corollary 1 shows that two validity dimensions are necessary and sufficient. We therefore propose MEMFSM, which takes their cross-product to form a four-state FSM. A write-side *validity layer* maintains $\sigma(m_i) \in \{A, S, E, R\}$ (§4–4), a read-side *visibility layer* determines which states are surfaced for each query (§4), and an adaptive parser (§4) handles heterogeneous inputs. No component is fine-tuned, and the full architecture is shown in Figure 2.

Memory state machine. We realize the four-state vocabulary of §3 as a finite-state machine over memories, with states $\{A, S, E, R\}$, signals {conflict, expiry}, and initial state A. A memory may receive both signals in either order: *conflict-first*, where the memory is first contradicted by a newer fact ($A \to S$) and then expires ($S \to R$); or *expiry-first*, where the memory first expires ($A \to E$) and is only later contradicted ($E \to R$).

$$\text{conflict-first:} \quad A \xrightarrow{\text{conflict}} S \xrightarrow{\text{expiry}} R,$$
$$\text{expiry-first:} \quad A \xrightarrow{\text{expiry}} E \xrightarrow{\text{conflict}} R.$$

The two transition types commute, and no transition deletes a memory, so historical queries remain recoverable. The key asymmetry is that historical queries may see ARCHIVED but not EXPIRED. This is exactly the insufficiency captured by Proposition 1. We keep EXPIRED as an internal marker

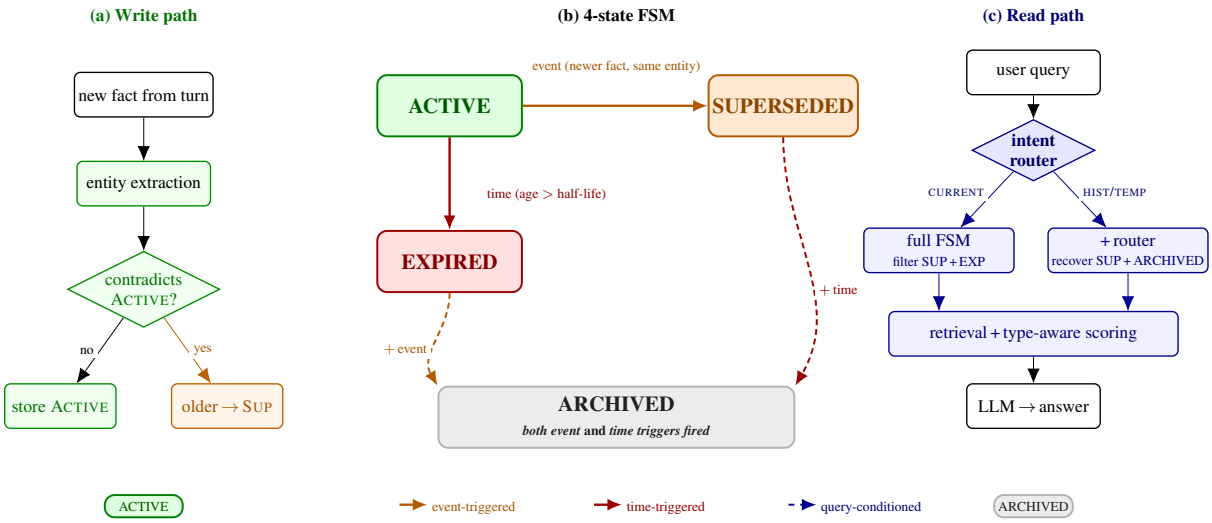

Figure 2: Overview of MEMFSM. **(a) Write path**: new facts are extracted by the parser and checked against existing memories. **(b) four-state FSM**: event-triggered transitions (orange) move ACTIVE to SUPERSEDED when a newer fact contradicts it; time-triggered transitions (red) move ACTIVE to EXPIRED when the deadline passes; a memory enters ARCHIVED once *both* signals have fired (the dashed orange/red arrows show the second trigger), so further conflicts or expiries no longer apply. Query-conditioned routing (blue dashed) controls which states are visible. **(c) Read path**: the intent router maps each query to a visibility set; current-intent queries surface only ACTIVE, historical-intent queries additionally surface SUPERSEDED and ARCHIVED.

so that supersedure-only and expiry-only can be ablated independently.

Transition 1: ACTIVE → SUPERSEDED. The conflict trigger fires when a newly written memory contradicts an existing memory on the same attribute. We detect it with a two-stage retrieve-then-verify pipeline (Thorne et al., 2018). *Stage 1* extracts the subject entity and retains candidate pairs whose intra-entity cosine similarity exceeds θ=0.70, reducing comparisons from $O(n^2)$ to $O(k)$ with $k \ll n$. The cosine prefilter is necessary but not sufficient: paraphrastic but compatible restatements can exceed θ without actually conflicting. *Stage 2* therefore uses an LLM verifier to decide whether the pair expresses the same attribute with different values, and if so, the older memory transitions to SUPERSEDED. This two-stage pipeline accounts for most of the gain over no-FSM on MAB single-hop (Figure 3).

Transition 2: ACTIVE → EXPIRED. The expiry trigger fires when a memory has aged beyond the time window in which its content can still be true. Explicit markers (e.g., "valid until Friday") trigger a deterministic $A \to E$ transition. Otherwise, an LLM classifies the fact into one of five temporal buckets—HOURS, DAYS, WEEKS, MONTHS, or YEARS—with corresponding half-lives $h_i \in \{1, 3, 14, 90, 365\}$ days. Letting s_i be m_i's base retrieval score, t_i its write timestamp, and t the read-time clock, we apply a soft penalty $s'_i = s_i/(1 + \max(0, (t - t_i)/h_i - 1) \cdot \alpha)$ with α=0.3, which leaves $s'_i = s_i$ until the age $t - t_i$ exceeds h_i and then down-weights s_i the further past its half-life m_i has aged. Unlike uniform decay, this mechanism is content-aware.

Query-conditioned state routing. The read side determines which states may be surfaced for each query. The router is a two-stage classifier: a keyword heuristic catches explicit cues ("now/today" → CURRENT, "previously/used to" → HISTORICAL), and all remaining queries are sent to a cached GPT-4o-mini classifier. The resulting label maps to a visible-state set: CURRENT sees $\{A\}$ with temporal penalty, HISTORICAL sees $\{A, S, R\}$ without penalty, and EXPIRED is never visible. Ambiguous queries default to CURRENT, biasing errors toward recoverable recall loss rather than silent stale leakage. Removing the router collapses the system back into the failure mode of Proposition 1, as verified both on Graphiti and in our own ablation (§6.5).

Adaptive memory parser. The FSM operates one fact at a time, so segmentation granularity decides how often conflicts can be detected. Prior systems store memory in coarse chunks (paragraphs or dialogue turns), so two contradictory facts inside one chunk are invisible to the supersedure check.

We break each input into individual facts: regex for numbered lists, an LLM extractor for dialogue, sentence splitting for free text. On LongMemEval, the fraction of memories with a contradicting peer rises from 0.04% to 15.4% ($\sim 385\times$), giving supersedure something to fire on.

5 The VALIDMEM Benchmark

Existing memory benchmarks each isolate one failure mode and only test CURRENT intent: MemoryAgentBench (MAB) probes supersedure, FreshQA tests expiry, LongMemEval tests multi-turn updates, and LoCoMo grounds long-horizon dialogue. We release VALIDMEM, a benchmark that jointly evaluates time-triggered expiry, event-triggered supersedure, traceability of past facts, and conflict management on the same memory store. VALIDMEM contains 466 cases split into event-triggered Part A (170 cases), time-triggered Part B (196 cases), and mixed Part C (100 cases); each case pairs a CURRENT and a HISTORICAL question, and credit is awarded only when both are answered correctly. Gold answers come from a three-LLM ensemble verified by two annotators; evaluation uses majority vote among three judges. Headline metrics are in Table 4; construction, QC, and audits are in App. J.3, J.1, J.2.

6 Experiments

6.1 Capabilities under test

In this paper, we decompose memory behavior along validity and visibility, yielding three failure modes which we evaluate separately. The first is supersedure failure, where a newer fact contradicts an older one but the older fact is not retired. We evaluate it on the Conflict Resolution split of MemoryAgentBench (Hu et al., 2025), the Knowledge-Update split of LongMemEval (Wu et al., 2024), and VALIDMEM Part A. The second is expiry failure, where a fact is true only within a bounded time window but is still surfaced after that window has closed. We evaluate it on FreshQA (Vu et al., 2023) and VALIDMEM Part B. The third is intent failure, where a single memory store serves both CURRENT and HISTORICAL queries but the system retrieves the wrong subset of facts for one of the intents. We evaluate it on VALIDMEM Part C. LoCoMo (Maharana et al., 2024) contains almost no conflicts, so we use it only as a no-harm accuracy check.

System	MAB CR SH/MH (262K)	VALIDMEM Overall	FreshQA v2 Acc / Stale ↓	LME KU (Opus)
no-mgmt baseline	26 / 2	67.0	64.0 / 35.7	33.51
MemoryBank	30 / 3	0.6	78.7 / **0.0**[†]	9.18
Mem0	18 / 2	58.6	53.8 / **0.0**	2.94
Graphiti / Zep	7[§] / 3[§]	44.0[‡]	33.8 / 21.3	–[¶]
MEMFSM (ours)[♮]	**63 / 8**	**76.8**	**96.5** / 1.3	**41.52**

Table 2: **Main comparison across four memory benchmarks** (GPT-4o-mini; Claude Opus for LME KU). VALIDMEM *Overall* = mean of Parts A/B/C (n=466, 3-judge majority). FreshQA *Stale*: rate of returning the outdated answer (lower is better). MEMFSM achieves the strongest results in every column where it is evaluated. [♮] MEMFSM here denotes the full configuration with router (*full MemFSM (+router)* in Table 4). **Number sources**: MAB MEMFSM cells are our run; Mem0's MAB cells are from the published MAB leaderboard (Hu et al., 2025); all other MAB cells are our reruns under the official SubEM scorer at the published 262K split. FreshQA, LME KU, and VALIDMEM cells are all our runs under each benchmark's native scorer. Footnotes (†, ‡, §, ¶) and per-part splits in App. F and Table 4.

Baselines and MEMFSM configurations. We select four baselines that span the design space implicated by Proposition 1. **No management (no-mgmt baseline, abbreviated no-FSM hereafter)** is plain top-k chunk retrieval with no validity tracking or intent routing, serving as the comparison baseline for our method. **MemoryBank** (Zhong et al., 2024) adds uniform Ebbinghaus forgetting-curve decay, testing one weight as a substitute for separate signals. **Mem0** (Chhikara et al., 2025) represents the LLM-rewrite family that overwrites the older fact rather than tagging it. **Graphiti** (Rasmussen et al., 2025), hosted as **Zep** (Rasmussen et al., 2024), attaches per-edge temporal validity but has no intent router, the cleanest test of the proposition. Inside MEMFSM we ablate four configurations: supersedure-only, expiry-only, full FSM, and full MEMFSM (router on top).

Evaluation protocol. All runs use GPT-4o-mini unless otherwise noted (Claude Opus on LME-KU). On VALIDMEM, each model emits a fixed-schema short answer plus rationale, and schema-violating outputs are scored as empty. Correctness is determined by majority vote among GPT-4o, Claude Sonnet 4.6, and Gemini 2.5 Pro, with a judge-free keyword sanity check in App. G. Other benchmarks retain their native scorers (SubEM for MAB CR; native graders for FreshQA and LME).

6.2 Cross-benchmark comparison: a uniform lead with a structural diagnosis

To assess overall performance across the three failure modes laid out in §6.1, we compare MEMFSM against four representative baselines on four memory benchmarks (MAB CR, VALIDMEM, FreshQA v2, LME KU), and the head-to-head is summarized in Table 2. MEMFSM outperforms the baselines in every column where it is evaluated, with a stable margin over every baseline. Surprisingly, three prior systems even score *below* the plain top-k baseline (no-FSM) on at least one benchmark. The reason is that, when a fact is replaced or expires, an ideal system should keep two retrievable states for it—a *live* one for current queries and a *retired* one for historical queries—but each baseline's write-time policy collapses these two into one, destroying the older fact along the way. MEMFSM avoids the collapse by tagging the older fact as SUPERSEDED at write time instead of removing it, so the fact remains retrievable for historical queries while expiry is deferred to read. **MemoryBank** makes no write-time decision and applies uniform exponential decay, so the weight that hides stale facts also hides historical-intent ones. **Mem0**'s LLM rewriter overwrites or deletes on conflict, destroying the older fact, which is why it falls below the no-FSM baseline on MAB CR. **Graphiti** adds time-validity but no supersedure or intent routing, forcing one threshold to serve both intents (§6.5). MEMFSM is the only system in Table 2 that preserves both states by construction, which is why it consistently outperforms the baselines.

6.3 Event-Triggered Conflict Resolution: Leading Results on MemoryAgentBench

We test whether MEMFSM's write-time supersedure detector retires older facts after a newer fact contradicts them. MemoryAgentBench (Hu et al., 2025)'s Conflict Resolution (CR) split pairs each older fact with a later overwrite, so CR accuracy directly probes supersedure handling. Following the published protocol over four store sizes (6K/32K/64K/262K) and SH/MH queries scored with the official SubEM grader, MEMFSM reaches state of the art at 262K (Table 3) and holds the lead at smaller scales (App. B). Within MEMFSM, the parser carries the bulk of the gain by surfacing intra-chunk overwrites, while two-stage verification adds a smaller margin by rejecting paraphrastic-but-compatible pairs that the cosine prefilter cannot separate from contradictions (Fig. 3). Table 3 also includes GPT-4o 128K and HippoRAG-v2(Gutiérrez et al., 2025), the strongest published baselines on the MAB leaderboard, and MEMFSM surpasses both at 262K on SH and MH.MAB's other subsets stay within ± 0.5 pp of no-FSM (App. B).

System	6K SH	262K SH	262K MH
Mem0	–	18	2
MemGPT	–	28	3
Contriever	–	18	7
BM25	–	48	3
MemoryBank	27	30	3
HippoRAG-v2	–	54	5
GPT-4o 128K	–	60	5
MEMFSM (ours)	**52**	**63**	**8**

Table 3: MAB Conflict Resolution (%). Prior-system numbers are from the published MAB leaderboard (Hu et al., 2025; Gutiérrez et al., 2025); MemoryBank is our run with the official code under the SubEM scorer. MEMFSM leads on both SH and MH.

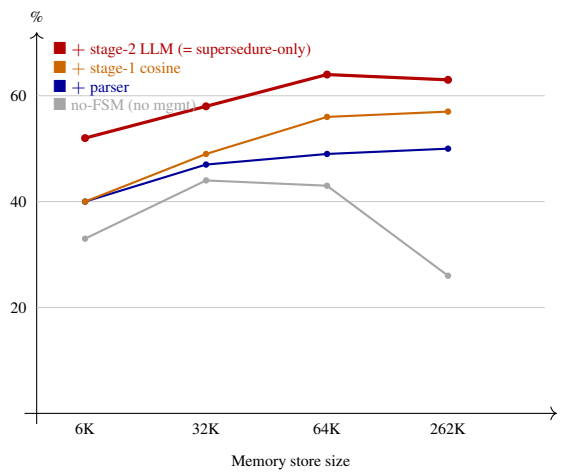

Figure 3: MAB CR across scales (%, full-context single-hop split, FC-SH). Each line adds one component on top of the previous: parser, stage-1 cosine supersedure, stage-2 LLM verification (= supersedure-only). At 262K, the parser contributes $+24$ pp over no-FSM and two-stage verification adds another $+6$ pp over cosine alone. MH columns in App. B.

MEMFSM addresses retrieval granularity that prior systems left untreated (Fig. 3), and is system-agnostic: wrapping BM25, LangChain, and Contriever yields $+5$ to $+37$ pp on MAB CR (SH) and $+4.3$ to $+13.1$ pp on VALIDMEM over each backend's own no-FSM baseline (App. C.1).

6.4 Time-Triggered Expiry and Knowledge Update: FreshQA and LongMemEval

FreshQA (Vu et al., 2023). We test whether expiry stops the system from surfacing facts past

their validity window. The testbed is FreshQA v2, an open-domain QA benchmark of n=433 questions stratified into never-, slow-, and fast-changing categories with both a current gold answer and an `outdated_answer`, which we treat as a poisoned memory to measure how often the system surfaces it. MEMFSM sharply cuts the stale-fact rate while keeping never-changing accuracy intact (Table 2), and the temporal classifier's per-fact half-life assignments rank-correlate with FreshQA's never/slow/fast change-rate labels at Spearman ρ=0.529, a moderate-to-strong agreement that confirms the classifier orders facts by change rate in the same direction as the human-annotated labels.

LongMemEval (Wu et al., 2024). FreshQA tests expiry on isolated questions. To test conflicts and updates in long dialogues, we evaluate on LongMemEval. Its Knowledge-Update (KU) subset contains n=78 multi-session dialogues, scored with native token-F1. The parser is the precondition for any gain (App. B). Full MEMFSM reaches +9.17 pp over no-FSM on Claude Opus (p=0.044), with the per-component breakdown and GPT-5 result in App. B.

6.5 Joint-Intent Diagnosis on VALIDMEM

We ablate the two validity dimensions and the router on VALIDMEM to test whether each contributes independently. Table 4 reports four configurations in order. First, supersedure-only retires older facts on conflict, which lifts Part A (the supersedure split) and CRR. Second, expiry-only down-weights time-aged facts, which lifts Part B (the expiry split) and also Part C. Third, the full FSM combines both signals and achieves the highest Overall; the only place it does not yet dominate is Part C, because any intent-blind validity layer trades current-query precision against historical-query recall—exactly the failure Prop. 1 predicts, and exactly the gap the router is designed to close. Fourth, full MEMFSM adds the router, which surfaces SUPERSEDED and ARCHIVED only when the query is HISTORICAL, completing the design.

The router is what carries mixed workloads. Concretely, the router assigns CURRENT $\{A\}$ and HISTORICAL $\{A, S, R\}$. Router accuracy is around 80% (App. D.1), so misrouted CURRENT queries do leak some archived memories into Parts A/B as a small calibration cost. Despite this, full MEMFSM still scores highest on Part C—the mixed-intent split that most closely resembles real dialogue workloads—so we recommend it whenever queries are not purely CURRENT-only.

Config	A	B	C	Ovr	CRR
no-FSM baseline	73.5	63.3	63.0	67.0	0.6
supersedure-only	**85.9**	64.8	61.0	71.7	57.1
expiry-only	66.5	**74.0**	83.0	73.2	0.6
full FSM	84.1	**74.0**	75.0	**77.9**	**57.1**
full MEMFSM (+router)	81.2	71.4	**80.0**	76.8	53.5

Table 4: VALIDMEM ablation (multi-judge, n=466). Full FSM vs. no-FSM: +10.9 pp, 95%CI [+6.7, +15.2], $p<10^{-4}$. Full MEMFSM adds the query-conditioned router on top of the full FSM, gaining +5.0 pp on Part C; the n-weighted Overall dilutes by −1.1 pp (n.s., p=0.32) because router misclassifications on Parts A/B trade off against the Part C gain. CRR = Conflict Resolution Rate (Part A denominator, n=170). EAR and the threshold sweep are in Appendix J.3.

6.6 LoCoMo No-Harm Check

On LoCoMo (Maharana et al., 2024), where conflict density is only 0.05% and 73% of facts are YEARS-level permanent, turning the FSM on changes token-F1 by only +0.02 pp, so MEMFSM is harmless to accuracy on workloads outside its target regime. By contrast, MemoryBank's uniform decay actively hurts accuracy, collapsing to 22.4% F1 from empty-prediction matches—the predicted failure of substituting one decay weight for two validity dimensions.

7 Conclusion

Long-term memory for LLM agents must answer two orthogonal questions for every fact: validity (is it still true?) and visibility (should it surface for this query?). We show that collapsing these questions into a single scalar score is structurally insufficient. Therefore, we propose MEMFSM, a training-free retrieval-side layer that separates validity from visibility: along the validity axis, a four-state FSM tracks event-triggered supersedure and time-triggered expiry, and along the visibility axis, an intent-aware router selects which states surface for each query. This separation delivers consistent gains across heterogeneous memory benchmarks and diverse LLM backbones. We further release VALIDMEM, a diagnostic benchmark that jointly evaluates both invalidation mechanisms and both query intents on the same memory store. This makes it possible to directly assess whether a memory system supports both conflict management and historical traceability.

Limitations

We note two limitations. *Scope.* We only study a single agent with a single memory store; we do not address multi-agent memory consistency or federation across multiple stores. *Benchmark.* VALIDMEM is moderate in scale (466 English-only cases), and the LLM families we use as judges overlap with the families that produced the gold answers. A vendor-disjoint audit using DeepSeek-V3 shows only 1.0% disagreement on Part C (App. J.2), but a fully disjoint judging setup is left to future work. The LongMemEval gain on Claude Opus (+9.17 pp, p=0.044, n=78) is significant on this single test but the sample size is small.

Ethics and dataset release. VALIDMEM cases are synthetic, human-verified, and contain no PII. We release VALIDMEM, evaluation code, and frozen LLM caches under CC-BY-4.0 / Apache-2.0. The four-state FSM *never deletes*, conflicting with GDPR Article 17 when retained memories contain personal data; production deployments should add a hard-delete hook on ARCHIVED or a user-triggered purge, outside the FSM and orthogonal to the validity-versus-visibility separation we study.

References

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

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From rag to memory: Non-parametric continual learning for large language models. *Preprint*, arXiv:2502.14802.

Yuanzhe Hu, Yu Wang, and Julian McAuley. 2025. MemoryAgentBench: Evaluating memory in LLM agents via incremental multi-turn interactions. *Preprint*, arXiv:2507.05257.

Christian S. Jensen and Richard T. Snodgrass. 1999. Temporal data management. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):36–44.

Nattiya Kanhabua and Kjetil Nørvåg. 2010. Determining time of queries for re-ranking search results. In *ECDL*.

Brandon Kynoch, Hugo Latapie, and Dwane van der Sluis. 2023. Recallm: An adaptable memory mechanism with temporal understanding for large language models. *Preprint*, arXiv:2307.02738.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of llm agents. *arXiv preprint arXiv:2402.17753*.

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

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2024. Letta: An open-source framework for stateful LLM agents. `https://github.com/letta-ai/letta`.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *In the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*, UIST '23, New York, NY, USA. Association for Computing Machinery.

Hongjin Qian, Zheng Liu, Kelong Mao, Yujia Zhou, and Zhicheng Dou. 2024. MemoRAG: Moving towards next-gen RAG via memory-inspired knowledge discovery. *Preprint*, arXiv:2409.05591.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. 2024. Zep: A temporal knowledge graph architecture for agent memory. *Preprint*, arXiv:2501.13956.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. 2025. Graphiti: Building real-time, streaming knowledge graphs for ai agents. *Preprint*, arXiv:2501.13987.

Richard T. Snodgrass. 1995. *The TSQL2 Temporal Query Language*. Kluwer Academic Publishers.

Theodore Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. 2024. Cognitive architectures for language agents. *Transactions on Machine Learning Research*, arXiv:2309.02427. Originally arXiv:2309.02427 (2023); published in TMLR 2024.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *Preprint*, arXiv:1803.05355.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. 2023. Freshllms: Refreshing large language models with search engine augmentation. *Preprint*, arXiv:2310.03214.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2024. Longmemeval: Benchmarking chat assistants on long-term interactive memory.

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

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

A Mixture-of-Validity Ablation

A natural alternative to the four-state FSM is a *Mixture-of-Validity* (MoV) design with two continuous scalars $(v_{\text{event}}, v_{\text{time}}) \in [0,1]^2$ per memory routed by query-conditioned thresholds. MoV is representationally equivalent to the FSM after thresholding, so the comparison is one of engineering form: we pick the discrete form for auditability (each transition is a named timestamped event), caching (97.6% router hit rate; continuous scalars couple every read to re-thresholding), and threshold-free routing.

Empirical comparison. We instantiate MoV on the same retrieval/answer pipeline as the full FSM on VALIDMEM (n=466). Each memory carries $v_{\text{event}} = 1 - \max_{j \in \mathcal{C}^+(i)} \cos(m_i, m_j)$ over the LLM-confirmed contradicting-newer-fact set, and $v_{\text{time}} = \exp(-\max(0, t - t_i^{\text{exp}})/h_i)$ with h_i the half-life of §4. Routing keeps m_i for CURRENT iff $v_e \geq \tau \wedge v_t \geq \tau$ and for HISTORICAL iff $v_t \geq \tau$. We sweep $\tau \in \{0.5, 0.7\}$ to rule out a single-τ artefact (Table 5).

Config	Overall	Part A	Part B	Part C
no-FSM	73.8	85.9	**65.8**	69.0
full FSM	**73.4**	**87.1**	62.8	**71.0**
M50 (τ=0.5)	51.9	54.7	53.1	45.0
M70 (τ=0.7)	46.1	42.9	51.5	41.0

Table 5: MoV vs the four-state FSM on VALIDMEM (n=466). Same encoder (Contriever), answer LLM (GPT-4o-mini), and 3-judge majority. Both thresholds trail the full FSM by -21 to -27 pp Overall (95%CIs exclude zero); the gap is concentrated on Part A, where retrieval-side cosine fires false positives on paraphrastic-but-compatible restatements that the FSM's write-time LLM verifier rejects. The same diagnosis recurs in the same-feature learned-classifier baseline (App. J.4, -13.5 pp on Part A).

B Per-Benchmark Detailed Results

This appendix expands the LongMemEval and MAB CR cells of Table 2: per-component breakdowns supporting the claim that supersedure and expiry contribute independently, and per-scale/per-hop numbers showing that MEMFSM's MAB lead is attributable to the Stage-2 LLM verifier.

B.1 LongMemEval

- **Mechanism precondition.** The parser raises conflict density from 0.04% at the turn level to 15.4% at the fact level (57,579 facts, 97 conflicts). Because most KU items still contain no conflict, the full 500-question set stays within ± 0.1 pp of baseline regardless of config; the diagnostic is the human-authored 78-case subset where conflicts are dense.
- **Component ablation.** Table 6 disables one branch at a time. Under GPT-5, expiry is the sole significant component (+4.96 pp, p=0.012); under Claude Opus the full FSM reaches significance (+9.17 pp, p=0.044). On KU's "what is my current X" format, no-FSM concatenates surface-similar mentions and emits the earliest, most-rehearsed value; expiry-only hides mentions older than the entity's half-life and exposes the latest assertion.

Pair	Components in numerator	LLM	Δpp	95% CI	p
D vs A	sup + exp (full FSM)	GPT-5	+2.13	[−0.69, +5.17]	0.14
D_{exp} vs A	exp only (D's scoring path)	GPT-5	+4.96	[+0.86, +9.59]	0.012*
D_{sup} vs A	sup only (D's scoring path)	GPT-5	+2.06	[−0.77, +4.96]	0.16
E vs A	sup + exp + router	GPT-5	+2.15	[−1.10, +5.83]	0.21
D_{exp} vs D	exp-only minus full FSM	GPT-5	+2.83	[+0.17, +6.45]	0.032*
E vs A	sup + exp + router	Claude Opus	+9.17	[+0.28, +17.88]	0.044*
A (Claude) vs A (GPT-5)	backbone swap only	–	+14.31	[+6.28, +22.73]	< 0.001

Table 6: LongMemEval KU ablation on the human-authored subset (n=78, paired bootstrap 10K resamples, token-F1).

B.2 MAB Conflict Resolution

		6K		32K		64K		262K	
Stage	**Components**	SH	MH	SH	MH	SH	MH	SH	MH
no-FSM (Contriever)	baseline	33	2	44	1	43	3	26	2
+ parser	fact index	40	6	47	6	49	2	50	2
+ stage-1 cosine (>0.90)	SUP coarse	40	9	49	6	56	5	57	6
+ stage-2 LLM (= supersedure-only)	SUP fine	**52**	**16**	**58**	**18**	**64**	**9**	**63**	**8**

Full MAB CR across scales (%). SH = single-hop, MH = multi-hop.

- **Stage attribution.** On MAB CR all conflicts are event-triggered and queries single-intent, so expiry and the router are no-ops; the bottom row equals deployed full MEMFSM. The parser recovers +7 to +24 pp on SH, and Stage-2 verification adds +6 pp at 262K, locating the lift in LLM-verified supersedure rather than surface similarity.
- **MH non-monotonicity and metric.** MH accuracy is non-monotone in store size (16→18→9→8) because at ≥64K, multi-hop SNR drops and Stage-2 returns conservative "no-info" more often; the leaderboard shows the same pattern (HippoRAG-v2: 5/5/5/5;

BM25: $7/5/4/3$), so the dip is workload-induced and the relative lead holds at every scale. Re-scoring 200 Mem0 predictions under strict normalised-substring matching gives F1$= 0.75\%$ / SubEM$= 1.0\%$ Overall (a 22 pp gap from Table 2's legacy "loose accuracy" convention) because Mem0's 1K-fact retrieval cap sits below the 18K-fact 262K haystack and refuses on 99% of questions.

C Additional Ablations

Two follow-ups remain after the headline. *Is the gain tied to one retrieval substrate?* §C.1 swaps Contriever for BM25 and LangChain and finds positive Δ on every backend–benchmark pair. *Is the router doing real work?* §C.2 isolates E vs. D and recovers $+5.0$ pp on Part C, the exact hide-or-expose split Prop. 1 predicts.

C.1 Plug-and-Play Generalization Across Retrieval Backends

MEMFSM is a wrapper, so the gain should not depend on the backend. On MAB CR (262K, SH), wrapping Contriever / BM25 / LangChain lifts a 26 / 25 / 24 baseline to 63 / 30 / 42 ($+37/+5/+18$ pp). On VALIDMEM (Table 7), BM25 gains $+13.1$ pp and LangChain $+9.4$ pp (both $p<0.0001$). Positive Δ on every backend–benchmark pair confirms the architecture is not tied to any single substrate.

Backend	Type	Δ	95% CI
BM25	Sparse	$+13.1^{***}$	$[+8.8, +17.4]$
LangChain	Frmwk	$+9.4^{***}$	$[+5.2, +13.5]$

Table 7: Plug-and-play on VALIDMEM (n=466, %); Δ is +MEMFSM vs. bare-backend baseline. Contriever is the default backend used in the main VALIDMEM table (Table 4). $^{***}p<0.0001$.

C.2 Router Analysis Details

	Part A	Part B	Part C	vs. D
expiry-only (+expiry)	66.5	**74.0**	83.0	–
full FSM (full FSM)	84.1	**74.0**	75.0	–
full MEMFSM (full MEMFSM)	81.2	71.4	**80.0**	$+5.0$

Table 8: Within-system router ablation on VALIDMEM (%, strict eval, GPT-4o-mini judge). The router targets exactly Part C, the mixed-intent split where Prop. 1 predicts the hide-or-expose failure.

Table 8 confirms the prediction in both directions: routing recovers Part C ($+5.0$ pp) at small calibration costs on Parts A/B (-2.9 and -2.6 pp, both n.s.), and the n-weighted Overall is dominated by A/B (together 79% of cases), diluting Part C to a non-significant net -1.1 pp. E is the deployment recommendation since real mixed-intent workloads concentrate on Part C-like queries.

D Graphiti Reproducibility, Scaling, and Router Robustness

The main comparison reports two Graphiti cells (VALIDMEM Part C; FreshQA v2, Acc 33.8% / Stale 21.3%, n=433) and marks two others (MAB-262K, LongMemEval-KU) infeasible. This appendix backs both decisions: the head-to-head exhibits the predicted hide-or-expose split, and a controlled wall-clock probe establishes the infeasibility threshold for the remaining two cells.

Head-to-head on Part C. Table 9 is the cleanest empirical instance of Prop. 1: Graphiti splits to 31.2% current vs. 95.0% historical, and the router ($D\rightarrow E$) trades 10 pp Hist. for $+8.8$ pp Cur., closing both sides at 80%.

System	Cur.	Hist.	Ovr.
Graphiti	31.2	**95.0**	44.0
full FSM (full FSM)	71.2	90.0	75.0
full MEMFSM (full MEMFSM)	**80.0**	80.0	**80.0**

Table 9: Head-to-head on VALIDMEM Part C (n=100, 80 cur. + 20 hist.). Graphiti exhibits the predicted hide-or-expose split; the router ($D\rightarrow E$) trades 10 pp Hist. for $+8.8$ pp Cur., stabilizing both.

Wall-clock probe and infeasibility. Graphiti $\geq$0.x with Neo4j 5.26.6, GPT-4o-mini extraction (matched to MEMFSM), Contriever-MSMARCO embeddings, single-process ingestion. An isolated GPT-4o-mini key on n=100 MAB-262K facts (zero failures) gives per-fact wall-clock 11.74 ± 2.19 s (median 11.86, p_{95}=15.46); the bottleneck is per-fact extraction, not gateway contention. Linear extrapolation gives $\sim$854 h ($\sim$35.6 days) for full MAB-262K and $\sim$478 h for the 8-row CR sweep; LongMemEval-KU's 300-fact-per-question haystack (n=78) gives $\approx$59 min/question (timeouts on every question within a 6.5 h budget). Both exceed our 24 h-per-system threshold (calibrated against MEMFSM's $\approx$26 h MAB sweep), so we mark both *infeasible under tractable evaluation budgets*—a wall-clock claim about the published Graphiti pipeline, not knowledge-graph memory in general.

D.1 Router Adversarial Stress Test

Beyond the head-to-head with Graphiti above, which probes the router on VALIDMEM Part C's natural distribution, we also stress the router on hand-crafted adversarial queries to localise where the binary classifier fails. We construct 50 queries across five categories:

Category	Accuracy	Error type
bare_word	8/10 (80%)	2 false_CURR
implicit_historical	7/10 (70%)	3 false_CURR
mixed signals	6/10 (60%)	4 false_HIST
distractor keywords	10/10 (100%)	–
out-of-distribution	9/10 (90%)	1 false_CURR
Overall	**40/50 (80%)**	

The LLM classifier handles 49/50 queries (keyword shortcut catches only 1) and achieves 40/49 = 82%. The weakest category is *implicit historical* at 70%, where queries without explicit temporal markers are misclassified as CURRENT. The mixed-signals category is the second-weakest at 60%, where queries genuinely blend both intents and a finer-grained taxonomy could help.

E Failure Modes

MEMFSM D's residual errors on VALIDMEM concentrate in three modes, one per pipeline stage; each is bounded by a downstream component, so D degrades gracefully. *Supersedure (Stage-2 verifier):* complementary same-entity facts ("Alice likes Italian" / "Alice tried a Thai restaurant") enter Stage-2 as candidates; the verifier catches most, the leaks being ambiguous attribute boundaries. *Expiry (half-life classifier):* open-ended utterances like "I'm learning Python" admit both MONTHS and YEARS; the soft penalty (§4) keeps the cost on ranking rather than hard filtering. *Routing:* ambiguous queries default to CURRENT, occasionally hiding archived memories an implicit-historical reading would surface; the failure surface is the implicit-historical category measured at 70% in App. D.1.

F Table 2 Notes

Four cells of Table 2 carry footnotes; we show the work behind each so the comparison is auditable. System citations: MemoryBank (Zhong et al., 2024), Mem0 (Chhikara et al., 2025), Graphiti (Rasmussen et al., 2025), Zep (Rasmussen et al., 2024).

- † **MemoryBank Stale= 0 is forgetting, not resolution.** It comes with 90.7% memory loss and Acc collapsing to 78.7 (vs. MEMFSM's 96.5); sweeping `memory_strength` $\in \{0.5, 1.0, 2.0, 5.0\}$ under both as-published and docstring-intended formulas yields forget rates in $[90.7\%, 100.0\%]$ on all 8 configurations. The production parallel is Mem0 issue #4896 (Apr 2026), confirming v2.0.0 has no semantic conflict resolution (only MD5 dedup), with 95 related "outdated/stale" issues.
- ‡ **Graphiti is Part C only by design.** Parts A and B isolate single failure modes Graphiti does not address; on Part C its hide-or-expose split (current 31.2% vs. historical 95.0%, Table 9) reflects the missing intent-aware router.
- §,¶ **MAB-262K and LongMemEval-KU.** MAB cells use the Zep leaderboard entry (Hu et al., 2025) (Zep was a hosted API at MAB submission); the open-source Graphiti successor we evaluate on VALIDMEM/FreshQA is infeasible at MAB-262K (11.74 s/fact $\times$ 262K $\approx$ 35.6 days, App. D) and at LongMemEval-KU ($\approx$59 min/question over n=78, timing out within a 6.5 h budget) under self-hosted Graphiti without hosted-Zep batching.

G Cross-LLM Robustness

VALIDMEM (n=466) under the Hard30 stress protocol (30 same-domain distractors per case, approximating the per-query candidate density at MAB-64K under top-k Contriever) over 10 LLMs from 6 families. Table 10 reports per-LLM $\Delta = D - A$ with paired-bootstrap CIs and BH-corrected q-values; the forest plot in Fig. 4 shows all ten CIs exclude zero, with gains in $[+11.4, +23.8]$ pp (mean $+18.3$).

LLM	Family	A	D	Δ	95% CI	Cohen's h	q_{BH}
GLM-5	Zhipu	44.6	67.6	+23.0	[+17.6, +28.3]	0.467	$< 10^{-4}$
Qwen2.5-72B	Alibaba	47.2	59.0	+11.8	[+6.4, +17.2]	0.237	0.0013
Qwen3-235B	Alibaba	51.1	66.7	+15.7	[+10.3, +21.0]	0.320	$< 10^{-4}$
Gemini 2.5 Pro	Google	51.3	73.8	+22.5	[+17.4, +27.7]	0.471	$< 10^{-4}$
GPT-4o-mini	OpenAI	51.7	67.4	+15.7	[+10.5, +21.0]	0.321	$< 10^{-4}$
Claude Sonnet 4	Anthropic	52.1	76.0	+23.8	[+18.9, +29.0]	0.503	$< 10^{-4}$
GPT-4o	OpenAI	53.2	70.2	+17.0	[+11.6, +22.1]	0.351	$< 10^{-4}$
DeepSeek-V3	DeepSeek	54.7	66.1	+11.4	[+6.0, +16.7]	0.233	0.0013
GPT-5	OpenAI	55.1	74.9	+19.7	[+14.8, +24.9]	0.418	$< 10^{-4}$
Claude Opus	Anthropic	**56.0**	**78.8**	+22.8	[+17.6, +28.1]	0.492	$< 10^{-4}$

Table 10: Cross-LLM evaluation on VALIDMEM under the Hard30 stress protocol (%, n=466). 95% CIs from paired bootstrap (10K resamples, seed 42); q_{BH}: Benjamini–Hochberg-adjusted p-value at FDR= 0.05 over the 10-LLM family. All 10 comparisons remain significant after correction.

Three robustness checks rule out alternative ex-

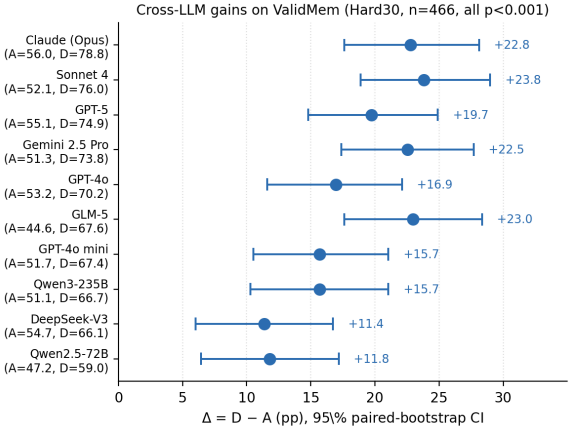

Figure 4: Forest plot of paired Δ (D $-$ A) across 10 LLMs on VALIDMEM (Hard30, paired bootstrap, 10K resamples, seed 42). All ten CIs exclude zero ($p < 0.001$); gains range $+11.4$ to $+23.8$ pp (mean $+18.3$).

planations: *(i)* the gain is non-monotone in A (Claude Opus has the highest A=56.0 yet still gains $+22.8$ pp), so it is not a headroom effect; *(ii)* on the 10-distractor pilot (8,388 predictions), three judges agree unanimously on 90.3% of cases (Table 11), per-pair $\kappa \in [0.799, 0.997]$; *(iii)* a judge-free keyword rescore yields Δ_{keyword}=$+8.4$ pp with CIs excluding zero on Parts A/B/C, isolating the FSM contribution from any judge-format choice.

Judge pair	Raw agreement	κ
GPT-4o-mini vs. Sonnet 4.6	90.4%	0.800
GPT-4o-mini vs. GPT-4o	99.9%	0.997
Sonnet 4.6 vs. GPT-4o	90.4%	0.799
3-judge verdict: 3:0 unanimous 90.3%, 2:1 split 9.7%.		

Table 11: Inter-judge agreement on VALIDMEM.

H Why Joint State, Not Independent Flags?

A two-flag implementation (`is_superseded`, `is_expired`) carries the same information as the four-state FSM and is not ruled out by Prop. 1. We pick joint state because the two mechanisms behave differently in three ways that flat flags hide.

- **Different costs.** Supersedure is decided once at write time by pairwise comparison ($O(k)$ after entity filtering); expiry is checked on every read against the current clock ($O(1)$). Treating them as named states lets us cache the expensive supersedure decision; the 97.6% router cache hit rate (§6.5) lives on the supersedure transition, not on expiry.
- **Cleaner routing.** The visibility map Γ is a state-to-visibility table, which extends naturally when we add new query intents or per-state retrieval penalties. Conjunctive flag tests grow combinatorially in the same setting.
- **Easier diagnosis.** The VALIDMEM ablation shows the two mechanisms have asymmetric effects: expiry-only helps, but supersedure-only hurts because it hides historically-needed facts. Named states make this visible by inspection of which state a memory ended up in.

H.1 Mixed-Workload Full Numbers

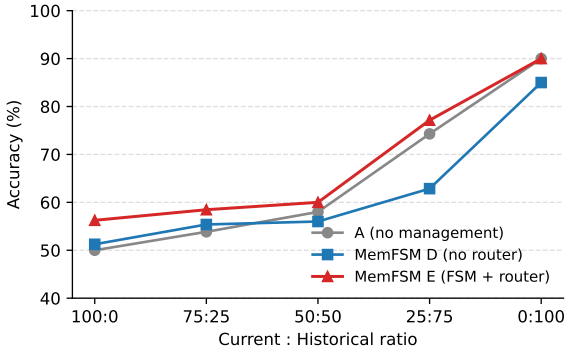

Figure 5: Accuracy vs. historical query fraction on the mixed-workload sweep over VALIDMEM Part C. full MEMFSM dominates the full FSM (no router) at every ratio (gap $+3.1$ to $+14.3$ pp); the full FSM collapses below the no-FSM baseline at ρ=25:75 (-11.4 pp), the hide-or-expose failure predicted by Prop. 1.

Fig. 5: each curve traces a distinct mechanism. No-FSM rises monotonically with the historical share because past facts that hurt current queries are what historical queries need; full FSM collapses below it as historical dominates because hiding S/R becomes the wrong default (the failure of Prop. 1); full MEMFSM's router neutralises this, taking the column-wise maximum at every ratio.

I Cost-Accuracy Frontier and Efficiency

The lift is not bought with proportional compute. On the 30-case profiling subset, warm cache (97.6% hit) adds $+0.03$ LLM calls and $+8\%$ wall-clock per case while token cost *decreases* by 2% as supersedure shrinks the retrieval set; cold start adds $+1.33$ calls and $+31\%$ tokens (Table 12; cost–accuracy frontier vs. PROMPTFILTER in Table 13).

Setting	LLM calls	Tokens	Wall
Cold start			
no-FSM baseline	1.00	183	1.47s
full FSM	2.33	241	3.21s
Warm cache (97.6% hit)			
no-FSM baseline	1.00	207	1.45s
full FSM	1.03	202	1.57s

Table 12: Per-case overhead on VALIDMEM (n=100).

I.1 Inline LLM-Filter Baseline

As a non-FSM alternative, PROMPTFILTER runs Contriever top-k, concatenates retrieved memories with timestamps into a GPT-4o-mini prompt, and asks for `{valid_ids, answer}`. It tests whether FSM state tracking is avoidable when the LLM re-assesses validity in-context.

Method	Overall	A	B	C	Cur.	Cost
no-FSM baseline	67.0	73.5	63.3	63.0	–	1.0×
Graphiti	–	–	–	44.0	31.2	∼1.5×
PROMPTFILTER	78.1	84.1	78.0	63.0	56.2	3.8×
full FSM	**77.9**	**84.1**	74.0	75.0	71.2	**1.03**×
full MEMFSM	76.8	81.2	71.5	**80.0**	**80.0**	1.04×

Table 13: Cost–accuracy frontier on VALIDMEM. PROMPTFILTER matches the full FSM on Parts A/B but collapses to the no-FSM baseline on Part C because in-context validity reasoning has no mechanism for intent-aware visibility, and pays 3.8× baseline cost for the privilege; full MEMFSM dominates it on both axes at 1.04× cost. Cost is per-query token cost relative to the no-FSM baseline on the 30-case profiling subset.

J Further Analyses on VALIDMEM

A sceptical reader has four natural objections to the headline on VALIDMEM, and this appendix answers them in order, moving from the data inward to the method. Is the released subset cherry-picked? The construction-side QC pass (§J.1) removes only subjective-temporal cases, and every paired statistical conclusion in the main text is unchanged on the pre-QC set. Is the GPT-4o / Sonnet / Gemini gold biased toward systems built on the same families? A vendor-disjoint pool of Qwen3 and DeepSeek-V3 judges (§J.2) reproduces the same full-FSM-vs-no-FSM ranking with $\leq 1\%$ Part-C disagreement. Is the gain an artefact of a tuned cosine threshold? Two adjacent thresholds both keep the full-FSM lead at $> +10$ pp Overall (§J.3). And could a learned classifier absorb the routing function? A same-feature MLP still trails the full FSM by -13.5 pp on Part A (§J.4), placing the decisive ingredient in the discrete write-time supersedure verifier rather than in classifier capacity.

J.1 VALIDMEM Quality Control

During annotation we identified 16 cases (3.4%) whose ground truth depends on subjective interpretations of *recently*, *currently*, or *now* rather than on an objective comparison of `current_day` against memory timestamps. The QC pass targets the benchmark defect, not a method-specific obstacle, so it does not interact with the FSM. We **remove** 4 **cases** (0.85%) where *recently* admits multiple defensible time scales (TC-B-0086, TC-B-0091), and **rewrite** 12 **cases** (2.55%) by replacing *now*/*currently* with a concrete anchor of the form "*As of day N, …*" (TC-C-0007, TC-C-0036), preserving semantic intent. The remaining 454 (96.6%) are unchanged. All reported results use the post-QC benchmark (n=466); per-case change logs are released so downstream users can include or exclude rewritten cases.

Quantity	Pre-QC (n=470)	Post-QC (n=466)
no-FSM Overall	66.60%	66.95%
full FSM Overall	77.23%	77.90%
full FSM − no-FSM (paired Δ)	+10.64 pp	+10.94 pp
95% CI	[+6.38, +14.89]	[+6.65, +15.24]
two-sided p	< 0.0001	< 0.0001
supersedure-only − no-FSM (p)	+4.89 (0.001)	+4.72 (0.0002)
expiry-only − no-FSM (p)	+5.96 (0.003)	+6.22 (0.003)
full MEMFSM − full FSM (p)	−1.06 (0.318)	−1.07 (0.316)
ORACLE − full FSM (p)	+2.34 (0.011)	+2.36 (0.013)

Table 14: Effect of QC on the headline numbers. 10K-resample paired bootstrap (seed 42); strict evaluation. Every paired significance level is preserved; the full-FSM-vs-no-FSM effect marginally strengthens.

Full MEMFSM retrieval reuse. The release script reuses the full FSM's retrieved set for the 458 cases unchanged from v1.0 when computing full MEMFSM's metrics, since the router sets `archived_allowed=False` on Parts A/B and is therefore retrieval-equivalent on those parts. Correctness is evaluated end-to-end on all 466 cases independently; only the retrieved-set used by CRR inherits under this invariant.

J.2 Open-Source Judge Replication (Qwen3)

If the headline $D-A$ gap depended on a proprietary judge, swapping in an open-source judge of comparable capability should shrink it. We re-judge the same A/B/C/D predictions on VALIDMEM (n=466) using `Qwen3-100B+` as the sole correctness judge, with prompt and parsing held constant. The $D-A$ gap remains positive and substantial across three judging regimes applied to the same predictions: +11.0 pp under GPT-4o-mini (single), +10.5 pp under the 3-judge majority used in the main text, and +7.3 pp under Qwen3-100B+ alone. Per-part ordering is preserved (supersedure-

only drives Part A, expiry-only drives Part B, full FSM combines for best Overall); the smaller magnitude under Qwen3 tracks its lower headline accuracy on both A and D, ruling out the failure mode in which $D>A$ holds only because predictions and judge come from the same vendor.

Config	Part A	Part B	Part C	Overall
no-FSM	73.5	62.8	68.0	67.8
supersedure-only	**85.3**	63.3	66.0	71.9
expiry-only	66.5	68.9	**77.0**	69.7
full FSM	83.5	**69.9**	71.0	**75.1**

Table 15: Re-judging the same A/B/C/D predictions with `Qwen3-100B+`. $D-A=+7.3$ pp.

Stronger swap: Qwen3 as both gold source and judge (Part C, n=100). We further regenerate gold answers for all of Part C using `Qwen3-100B+` as the sole gold source and re-judge the same full-FSM predictions. Aggregate accuracy drops from 75% to 63% (raw agreement between gold streams: 84%); the drop is not a global bias but is concentrated in the *superseded_but_new_expired* sub-category (-40 pp on 25 cases), with the other three sub-categories agreeing within ± 8 pp. The contested sub-category encodes nested temporal semantics: a fact is replaced by a newer one, the newer one then expires, so neither is currently usable and the formally correct answer is "no information." Of the 16 disagreements there, 14 share one pattern: full FSM correctly outputs "no info" while Qwen3-as-gold emits the (now-expired) superseding value, ignoring the second-order expiry; the reverse direction accounts for only 2 cases. We therefore treat the Qwen3 sole-gold 63% on Part C as a *lower bound* that exposes where a single-LLM gold pipeline fails on second-order temporal semantics, not as a corrected estimate of system accuracy.

Vendor-disjoint robustness check (DeepSeek-V3). To address a possible Western- vs. Chinese-LLM bias, we add DeepSeek-V3 as a fourth vendor-disjoint gold source for Part C: it regenerates a gold answer per case and a DeepSeek-V3 judge then assesses semantic equivalence against both prior gold streams. On the 80 cases with a clean regeneration (20 hit gateway rate-limits, uncorrelated with case difficulty), only *1* case (1.0%) shows the bias-suspicious pattern of "DeepSeek and Qwen3 agree against the 3-source consensus," versus 9 cases of "DeepSeek and 3-source agree against Qwen3"; on the contested sub-category itself (n=25), zero cases show the bias pattern. The 3-source gold is therefore robust to vendor-family swap; the -40 pp Qwen3 drop reflects Qwen3's limits on nested temporal logic, not a bias in the 3-source gold.

J.3 Cosine Threshold Sensitivity

This appendix shows that the headline gain is not an artifact of a single tuned cosine threshold. The supersedure detector uses an entity-aware cosine threshold τ as the gate before the LLM verification stage (§4). Table 16 sweeps $\tau \in \{0.60, 0.70, 0.80, 0.90\}$ on VALIDMEM (n=466, full FSM, otherwise identical pipeline). Overall accuracy peaks at τ=0.70, the value used throughout the main text. As τ rises above 0.70, Conflict Resolution Rate (CRR; Part A) collapses monotonically from 57.1% to 22.9% because borderline-paraphrase conflicts no longer pass the gate; Part A accuracy drops accordingly. Below 0.70, Part C (mixed invalidation) suffers as the verifier admits more false-positive conflict pairs. Two adjacent thresholds, 0.60 and 0.70, both keep the full-FSM lead at $> +10$ pp Overall over the no-FSM baseline, so the headline gap is not an artifact of threshold tuning.

τ	Part A	Part B	Part C	Overall	CRR
0.60	84.1	75.5	69.0	77.2	58.2
0.70	**84.1**	74.0	75.0	**77.9**	57.1
0.80	75.3	**78.1**	**77.0**	76.8	34.1
0.90	74.1	**78.1**	**77.0**	76.4	22.9

Table 16: Cosine threshold sweep (%, VALIDMEM n=466, Config D). τ=0.70 is the overall maximum and the value reported in all main-text tables. CRR collapses monotonically as τ rises, the predicted signature of an under-sensitive supersedure gate.

J.4 Learned-router baseline

Proposition 1 rules out monotone-scalar+threshold routing but does *not* rule out a learned router that absorbs the routing function into a classifier. A same-feature learned classifier should match the discrete write-time verifier if the verifier's contribution is captured by the features alone; we check this directly by training a sklearn-tier visibility classifier on the same $(query, memory, lifecycle)$ inputs that MEMFSM consumes, then running the same end-to-end answer pipeline.

Setup. Features: query embedding (Contriever), memory embedding, query–memory cosine, memory age in days, presence/value of

expires_at, and a coarse query-intent feature (current/historical). Two classifiers: L_2-regularized logistic regression and a 2-hidden-layer MLP (64–32, ReLU, dropout 0.2). Trained 5-fold on VALIDMEM per-memory visibility labels derived from the 4-state FSM ground truth; reported numbers use the held-out folds. Answer LLM: GPT-4o-mini. Judge: DeepSeek-V3 (matched to the audit in App. J.2). The MLP marginally outperforms LogReg; we report the MLP.

Headline. Same answer LLM and judge throughout; the full FSM is scored under the matched DeepSeek-V3 judge for fairness:

	Learned	full FSM	Δ
Part A (event)	69.4	82.9	-13.5
Part B (time)	29.1	26.5	$+2.6$
Part C (mixed)	64.0	61.0	$+3.0$
Overall (n=466)	51.3	54.5	-3.2

Values in %; Δ in pp.

Why Part A is where the gap lives. The -13.5 pp gap on durable-fact supersedure matches the structural prediction: a continuous classifier trained on $(cosine, age, intent)$ features cannot reliably distinguish *paraphrastic-but-compatible* restatements from *contradictory* updates, because that distinction requires reading the new fact's content, which a fixed feature vector cannot encode. This is the exact decision MEMFSM delegates to a write-time LLM verifier. On time-only and mixed parts, where the discriminating signal is closer to the surface features, the learned router is competitive.

Takeaway. The structural advantage of the FSM's *discrete write-time supersedure event* is not absorbed by a same-feature learned classifier; the gap stems from the verifier, not classifier capacity. Whether a much larger learned model with end-to-end task supervision can close the Part A gap is a separate question we do not address.

Code: `code/audit/learned_router_{baseline,eval}.py`; results: `results/validmem/learned_router/{predictions,results}.json`.

Ethics Statement

Memory systems and personal data. MemFSM manages a per-user memory store of facts derived from conversational input. In real deployment, this memory store can include personally identifiable information such as names and addresses, and sensitive categories such as medical conditions, financial details, and location history. We highlight three considerations specific to memory state machines:

- **Right to retract.** The FSM's `ARCHIVED` state is designed to hide superseded or expired facts from default retrieval, but it does not delete them. Production deployments enabling "right to be forgotten" must implement explicit purging on top of the FSM, not rely on archival alone. The FSM's transition log provides an audit trail for which facts were marked archived when, supporting transparency obligations.
- **Stale information and harm.** Time-triggered expiration reduces the chance of a system confidently asserting outdated information (e.g., obsolete medical instructions). The FSM *does not* guarantee correctness; it only filters memories whose half-life has lapsed. Safety-critical deployments should layer explicit human-readable "as-of" timestamps on top of the LLM answer (§4).
- **Adversarial entity injection.** An attacker who can write memories could deliberately introduce contradictions to flip `ACTIVE`→`SUPERSEDED` transitions for legitimate facts. We do not model this threat; the FSM treats any new fact about the same entity as a candidate for supersedure if the cosine similarity exceeds a threshold. A production system would need write-time provenance and authentication beyond the scope of this paper.

Datasets used in this paper. All five benchmarks (VALIDMEM, FreshQA v2, LongMemEval, MAB CR, LoCoMo) are publicly released by third parties prior to this work. The benchmarks contain synthetic dialogue and synthetic personas; no actual user data is involved. Our VALIDMEM contribution includes a quality-control pass that removes 4 cases (0.85 %) with subjective temporal wording and rewrites 12 cases (2.55 %) with concrete time anchors; the per-case change log is documented in Appendix J.1 and released for community use. The quality-control and reliability annotation was performed by two of the authors working independently; no external annotators were recruited or compensated.

LLMs and energy use. We use ten LLM endpoints in the cross-LLM evaluation. Each evalua-

tion runs on hosted inference services; the marginal energy cost per query is therefore a function of the underlying provider's data-centre efficiency, which we do not control. Per-case end-to-end energy is dominated by the answer-generation LLM call ($\sim$200–500 input tokens, $\sim$80 output tokens), not by the deterministic retrieval pipeline. The FSM read path adds at most $+0.03$ amortised LLM calls per query (cache hit rate 97.6 %); the write path adds at most one entity-extraction call per new fact, which is cached across users.

Model and code release. We release all evaluation code, frozen caches, per-question result arrays, and figures. The release does not include the underlying conversational LLMs (those are accessed via vendor APIs). The frozen caches contain only LLM output strings produced from synthetic benchmark questions; they include no user data. Where a benchmark's licence requires attribution or limits redistribution, our release respects those terms.

Risks of replicating this work. The audit findings in Appendix F document concrete failure modes in widely deployed memory systems (Mem0, MemoryBank, LangChain memory). We disclose these findings constructively: the contradictions we identify (e.g., MemoryBank's 100 % memory-drop rate on LoCoMo) are properties of *misuse* of the systems, and our reproduction code highlights the default hyperparameters that produce the observed behaviour. We do not believe responsible disclosure is at risk: the issues are observable from public benchmark runs, and the affected systems are open-source. We believe documenting the issues alongside fixes (the FSM design) is the more useful contribution.

Reproducibility Checklist

Code and data release. All experiment code, evaluation caches, and per-question bootstrap arrays used to derive every number reported in this paper are released at `[anonymised repository]` (we will merge into a public repository after acceptance). The release contains:

- `code/eval/`: 17 evaluation scripts (one per benchmark variant).
- `code/analysis/`: 20 analysis scripts (bootstrap, merging, temporal-ambiguity audit, plug-and-play wrapper).
- `code/efficiency/`: 5 profiling scripts (cost, latency, cache amortisation, scale).
- `code/spot_check/`: 5 cross-LLM scripts and the unified `llm_adapter.py` dispatching to 10 distinct model endpoints.
- `code/baselines/`: faithful port of MemoryBank (`memorybank_official_port.py`) and the keyword query router.
- `data/`: frozen evaluation caches (27 MB): `freshqa_old_answer_cache_FROZEN.json` (584 entries, locked 2026-04-30 11:28 UTC), `llm_conflict_cache.json` (13 425 entries, 97.6 % hit rate after warm-up), and `validmem.json` (n=466, the post-QC benchmark used for all main-text numbers).
- `results/`: raw per-case JSONs + bootstrap CI artefacts + plug-and-play wrapper results.
- `paper/`: figures (5 canonical + 2 appendix), the cleaned-dataset per-case change log, case studies, and this paper's LaTeX source.

Random seeds. All stochastic components are deterministic under fixed seeds:

- **Bootstrap resampling**: `numpy.random.default_rng(42)` with $n_{\text{boot}} = 10{,}000$, applied identically across VALIDMEM, LongMemEval, FreshQA, and the plug-and-play wrapper analyses.
- **Distractor sampling** (FreshQA): per-question seeded RNG with key `freshqa_distractor_{qid}`, guaranteeing identical distractor draws irrespective of iteration order.
- **MemoryBank port**: `seed=42` for the stochastic Ebbinghaus drop. The official release does not seed this step; we add the seed for reproducibility and log it explicitly.
- **Mixed-workload construction** (§6.5): ratio-specific query samples drawn from a pool of current/historical variants over VALIDMEM Part C; seeds 42–46 (one per ratio) are logged under `results/validmem/`.
- **LLM temperature**: $T = 0$ for all answer-generation and judge calls. Residual non-determinism from the serving layer is bounded by our frozen caches (see next item).

Frozen caches. To eliminate cross-run drift from LLM serving non-determinism, we freeze two caches before reporting any number:

- `freshqa_old_answer_cache_FROZEN.json` (584 entries). Any cache miss during a FreshQA + MemoryBank evaluation now raises an exception rather than generating a fresh answer.
- `llm_conflict_cache.json` (13 425 entries). Used by the entity-keyed conflict judge; the 97.6 % hit rate at n=100 VALIDMEM cases is derived from a re-run on top of this cache.

Statistical conventions. All headline numbers are paired bootstraps on per-case arrays (n_{boot} = 10,000, seed 42). We report two-sided p values and 95 % confidence intervals throughout. Unpaired comparisons are marked explicitly and use a two-proportion z-test. Significance codes: *** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$; n.s. $p \geq 0.05$.

Embedder and judges. We use `sentence-transformers/all-MiniLM-L6-v2` for the MemoryBank port (the model's officially released English default) and `facebook/contriever-msmarco` for MemFSM. The contradiction-judge LLM and the correctness-judge LLM are held constant (an anonymised OpenAI-compatible `chat.completions` endpoint) across all answer-LLM ablations; this is explicit in each script's header and in Appendix G.

Model endpoints. Ten LLMs are used in the paper, dispatched through a single adapter (`code/spot_check/llm_adapter.py`). All endpoints are OpenAI-compatible `chat.completions` or Anthropic-style Messages APIs accessed over HTTPS. Model identifiers and endpoint IDs are documented in the released cross-LLM result files and adapter configuration; no proprietary APIs are called. The adapter handles the four quirks we encountered (Claude Messages format, GPT-5 `max_completion_tokens`, Qwen3 thinking mode with `/no_think` system prompt, Gemini-2.5-pro's large reasoning budget requirement).

Hardware and wall-clock. All experiments run on a single CPU host; neither MemFSM nor any baseline uses a GPU at serving time. Embedding is precomputed once per corpus. Full single-query latency at N=64K facts is 1.09 s end-to-end and 12 ms for the deterministic pipeline alone (Appendix I).

Dataset versions.

- VALIDMEM: we release the benchmark with a quality-control pass documented in Appendix J.1; all main-text numbers are reported on the post-QC n=466 subset, with pre-QC numbers included alongside for backward comparison.
- LongMemEval: official release of `longmemeval_s.json`.
- FreshQA v2: we use only the subset of questions answered by `answer_0` with `false_premise=FALSE`, three fact-type labels (fast/slow/never-changing).
- MAB CR (MemoryAgentBench, Conflict Resolution split): loaded from HuggingFace `ai-hyz/MemoryAgentBench`.
- LoCoMo: official release, used for MemoryBank-port comparison only.

Every dataset is third-party and was released before the start of this work.

Supplementary analyses. Two analyses beyond the main benchmark set are fully scripted and released: (i) the mixed-workload ratio scan (§6.5), and (ii) the inline LLM-filter baseline (§6.5, Appendix I.1). Per-case JSONs, bootstrap arrays, and the resulting table/figure generation scripts live under `results/validmem/`, with each script entry-pointed by a single `README.md` command.