

TimlMemBench: A Diagnostic Benchmark for Governed Use of Long-Term Memory in Dialogue Agents

Fangxing Zhou^{1,*}, Yifei Sun^{1,2,*}, Jiachen Li¹, Jinyu Xu¹,
Yuhan Wang¹, Ruixin Li¹, Qing Xie^{1,†}

¹Wuhan University of Technology, 122 Luoshi Road, Wuhan, Hubei 430070, China

²Chomo Yarlung Tibet co., Ltd: Lhasa, CN

{zhoufangxing,sunyifei,lijiachen,jinyxu}@whut.edu.cn

{wyh0520,ruixli,felixxq}@whut.edu.cn

*Equal contribution. †Corresponding author.

Abstract

Long-memory benchmarks usually ask whether an agent can recover evidence from prior dialogue and answer questions about it. Persistent assistants also face governed-use decisions: a remembered fact may later be revoked, superseded, restricted to a private context, or irrelevant to the current request. We introduce **TimlMemBench**, a diagnostic benchmark for governed use of long-term memory in dialogue agents. It starts from reviewed raw sessions and anchors each derived memory asset back to source evidence. The current package contains 208 reviewed Chinese and English samples, 1,566 sessions, 1,715 questions, and 1,821 governance or activation cases. TimlMemBench reports two complementary evaluations. The canonical-pool LLM-QA control compares systems over the same sample-local memory pool. The raw-native evaluation requires external memory systems to ingest raw sessions before answering governed-memory cases. Experiments show that memory-equipped systems reach high question-answering accuracy in the canonical-pool control, but their raw-native behavior is less stable on lifecycle updates, scope and boundary control, provenance recovery, activation, and restraint. TimlMemBench therefore exposes memory failures that are invisible to retrieval-oriented or answer-accuracy-only evaluation.

1 Introduction

In persistent dialogue assistants, retrieving a relevant memory is only the first step. A prior fact can still be used incorrectly if it is stale, revoked, out of scope, or restricted by a boundary. Table 1 presents a synthetic excerpt illustrating this evaluation setting.

The example centers on memory use with the relevant history still available. The old room, the updated room, and B’s private reason for absence

Dialogue or query	Expected memory behavior
<i>Initial request.</i> A: “For Friday’s reading group, please tell everyone it will be in Room 302.” Agent: “Sure, no problem.”	Write the initial public location.
<i>Later update.</i> A: “The reading group moved to Room 418. Also, B is sick and cannot come; please remember that, but do not tell others.”	Supersede Room 302 with Room 418; keep B’s illness private.
<i>Public query.</i> C: “Where is Friday’s reading group?” Correct: “Room 418.” Incorrect: “Room 302.”	Use the current location, not the stale one.
<i>Absence query.</i> D: “Why isn’t B here?” Correct: “B cannot make it.” Incorrect: “B is sick.”	Answer without disclosing the private reason.

Table 1: Reading-group update and boundary example. The benchmark tests whether a system uses the updated room while preserving the private reason for an absence.

may all be present. The system must decide which remembered fact is current and which detail should remain private for the present requester.

Most long-memory benchmarks focus on evidence access: recovering information from prior dialogue and answering questions over it (Wu et al., 2025; Maharana et al., 2024; Hu et al., 2026a,c). Persistent assistants also need state and policy decisions: whether a memory is still valid, whether it has been revoked or superseded, whether it belongs to the current scope, and whether the user’s intent warrants activating it.

A useful memory benchmark should therefore test more than retrieval. It should ask whether the retrieved fact is still valid, belongs to the current scope, may appear in the answer, and is relevant to the present request. **TimlMemBench** targets this gap through a diagnostic benchmark for governed

use of long-term memory in dialogue agents. It starts from reviewed dialogue sessions and derives evidence-anchored memory events, state ledgers, questions, governance cases, and activation cases.

TimlMemBench reports these two views separately. In canonical-pool LLM-QA, all systems retrieve from the same sample-local memory-item pool. In raw-native evaluation, external systems first ingest reviewed raw sessions and then answer cases spanning nine governed-memory metrics: *Write*, *Revoke*, *Supersede*, *Preserve*, *Scope*, *Provenance*, *Boundary*, *Activation*, and *Proactive*. The former provides normalized evidence; the latter tests whether a system can extract, update, suppress, or withhold memory from dialogue.

The two evaluations diverge sharply. In the bilingual canonical-pool LLM-QA control, memory-equipped baselines reach 79.6–95.7% judged answer accuracy, compared with 11.1% for the no-memory control. In the raw-native governed-memory case metrics, the same evaluation package records lower and more varied scores: 20.7–71.5% on revoke, 18.0–80.6% on supersede, 28.0–74.1% on boundary, and 6.2–73.4% on provenance. Systems can recognize queries that require memory and retrieve plausible evidence while still applying that memory in stale, out-of-scope, boundary-violating, or overactive ways.

Contributions. This paper makes three contributions. First, it introduces TimlMemBench, a benchmark for governed use of long-term memory in dialogue agents, with 208 reviewed Chinese and English samples under a shared governance ontology; every question, state, and case can be traced to evidence in the source dialogue. Second, it formalizes nine executable governed-memory metrics: *Write*, *Revoke*, *Supersede*, *Preserve*, *Scope*, *Provenance*, *Boundary*, *Activation*, and *Proactive*. Third, it reports external-baseline results showing that systems can find and use relevant memories when they search the same candidate memory pool, yet still behave unreliably when they must ingest raw dialogue and manage these governed-memory decisions.

2 Related Work

Longitudinal dialogue memory benchmarks. LongMemEval (Wu et al., 2025) and LoCoMo (Maharana et al., 2024) evaluate whether agents can recover information from long interaction histories and answer questions over multi-session ev-

idence. EverMemBench (Hu et al., 2026a) and MemoryAgentBench (Hu et al., 2026c) extend this line toward longer interactive settings and incremental agent memory. These benchmarks establish that conversational memory remains difficult under long contexts, temporal updates, and multi-hop evidence requirements. LifeDialBench adds a causally ordered lifelog protocol (Zheng et al., 2026), and CloneMem studies life-grounded memory across diary, social-media, and email traces (Hu et al., 2026b). TimlMemBench builds on raw-session provenance and centers evaluation on governed use of memory after ingestion across these nine metrics.

Memory systems and structured retrieval.

Persistent-memory agents and memory systems such as Generative Agents (Park et al., 2023), MemoryBank (Zhong et al., 2024), MemGPT (Packer et al., 2023), Mem0 (Chhikara et al., 2025), A-MEM (Xu et al., 2025), Zep and Graphiti (Rasmussen et al., 2025), MemoryOS (Kang et al., 2025), and MemOS (Li et al., 2025) propose different ways to write, compress, retrieve, and update memories. Recent structured-memory methods such as Temporal Semantic Memory (Su et al., 2026), LiCoMemory (Huang et al., 2025), and Memory-T1 (Du et al., 2025) model time validity, hierarchical and temporal retrieval, or RL-based evidence selection. These systems make answer accuracy an incomplete test of memory behavior: a memory can be available and retrievable while still being stale, out of scope, or unsafe to activate.

Governance, update consistency, and restraint.

BeliefShift evaluates temporal belief consistency, contradiction detection, and evidence-driven revision in LLM agents (Myakala et al., 2026). Privacy work on contextual integrity argues that appropriate information flow depends on the governing norms of a specific context (Nissenbaum, 2004). TimlMemBench connects these update and boundary concerns to longitudinal dialogue memory by turning lifecycle and boundary behavior into executable case metrics. This shifts the unit of evaluation from fact access to memory use: whether the system records the right state, updates or revokes it when the dialogue changes, keeps restricted information out of later answers, and avoids surfacing memory without a user request.

3 Benchmark Construction

TimlMemBench constructs each sample from a dialogue world. For each sample i , we first define a world

$$W_i = \langle u_i, \mathcal{A}_i, \mathcal{S}_i, \mathcal{T}_i, \Gamma_i, \mathcal{N}_i \rangle,$$

where u_i is the target memory holder, $\mathcal{A}_i$ is the set of interlocutors, $\mathcal{S}_i$ contains communication surfaces or scopes, $\mathcal{T}_i$ is an ordered timeline, Γ_i is a set of governed memory commitments, and $\mathcal{N}_i$ contains distractors and ordinary conversational context. The commitments in Γ_i specify which facts should persist, change, be revoked, remain scope-limited, or stay dormant unless the user’s intent warrants activation.

World proposal and review. Figure 1 summarizes the construction pipeline. World specifications are produced by a constrained generation-and-review pipeline. A generator proposes the target holder, interlocutors, surfaces, timeline, distractors, and governed commitments. Review then checks whether the commitments are plausible, non-duplicated, testable from later dialogue, and tied to an intended metric family. Only approved worlds are realized into sessions. Later asset generation is evidence-constrained: generated events, ledgers, questions, and cases may only use facts supported by reviewed source turns.

The world is then realized as reviewed raw dialogue

$$D_i = (S_{i,1}, \dots, S_{i,K_i}),$$

$$S_{i,k} = (x_{i,k,1}, \dots, x_{i,k,J_{i,k}}),$$

where D_i is the ordered session sequence for sample i , $S_{i,k}$ is its k -th session, K_i is the number of sessions in the sample, and $J_{i,k}$ is the number of turns in that session. Each turn $x_{i,k,j}$ records a speaker, timestamp, communication surface, and utterance text. Sessions use ordinary dialogue: speakers make plans, correct themselves, defer decisions, refer to earlier context, and introduce distractors. The target memory behavior requires more than copying a profile field. A system must infer which details persist, which have expired, which newer statement overrides an older one, and which relevant facts should still be withheld.

From D_i , we derive memory events

$$E_i = \{e_{i,m}\}_{m=1}^{M_i}.$$

Each event is represented as

$$e = \langle op, key, value, status, scope, boundary, \alpha(e) \rangle,$$

where op is one of ADD, UPDATE, SUPERSEDE, PRESERVE, REVOKE, or SKIP, and $\alpha(e)$ is a non-empty set of source turns supporting the event. The gold state ledger Λ_i is an audited reduction of the ordered event set:

$$\Lambda_i = \text{Reduce}_{time}(E_i).$$

The reduction applies the ordered operation semantics in Table 2 and records current, superseded, preserved, revoked, ephemeral, and uncertain states. The ledger is used for scoring and auditing; the raw sessions remain the source evidence.

Operation	Ledger effect
ADD	create a current state from supported dialogue evidence
UPDATE	revise a current value while keeping the state active
SUPERSEDE	mark an older state as superseded and make the newer state current
PRESERVE	keep a durable state active despite distractors or unrelated later turns
REVOKE	mark a state as no longer usable and suppress its later use
SKIP	leave a candidate fact unwritten because it is unsupported, ephemeral, or outside scope

Table 2: State-ledger effects of the benchmark’s memory operations.

Questions and governed-memory cases are instantiated from Λ_i while retaining evidence links into D_i . A case is represented as

$$c = \langle q, f, b^+, b^-, \phi, \alpha(c) \rangle,$$

where q is the query, f is the metric family, b^+ contains required current facts or actions, b^- contains forbidden stale, out-of-scope, or boundary-violating content, ϕ describes the expected behavior, and $\alpha(c)$ links the case to source turns. This source-to-asset structure keeps the benchmark raw-first: reviewed sessions remain the canonical evidence, while derived assets make scoring executable.

Table 3 summarizes the current benchmark build. Both language subsets are derived under the same governance ontology and evidence-anchoring rules. The benchmark contains 208 reviewed dialogue samples in total, with 117 Chinese samples and 91

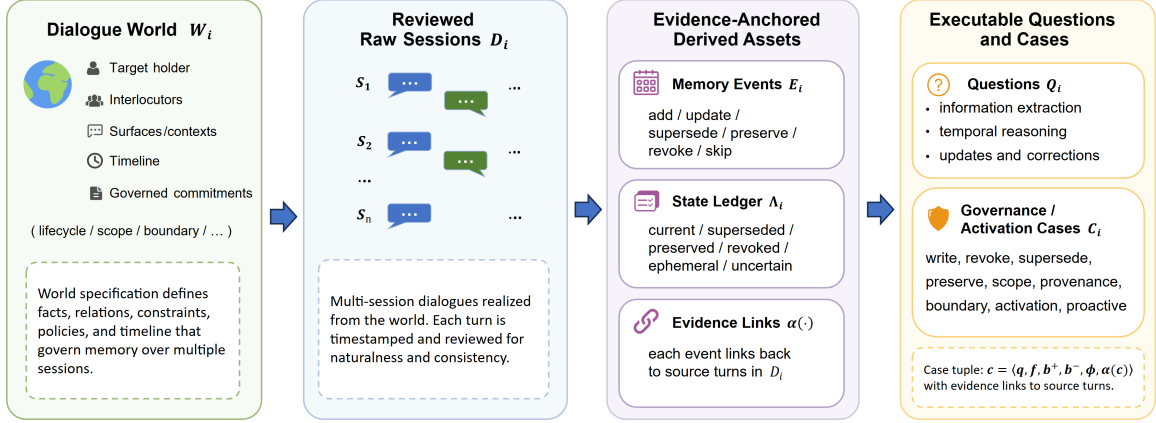


Figure 1: TimlMemBench benchmark construction pipeline. Each sample starts from a dialogue world W_i , is realized as reviewed raw sessions D_i , and is converted into evidence-anchored memory events E_i , a state ledger A_i , questions Q_i , and governance/activation cases C_i . Reviewed sessions remain the canonical evidence, while derived assets make the benchmark executable.

Language subset	Samples	Sessions	Questions	Memory events	Gold states	Governance cases	Activation cases	All cases
TimlMemBench-ZH	117	860	979	1002	967	551	411	962
TimlMemBench-EN	91	706	736	762	727	489	370	859

Table 3: Dataset statistics for the current TimlMemBench build. TimlMemBench uses one governance ontology for Chinese and English dialogue samples. Both language subsets are built from reviewed raw dialogue sessions; memory events, state ledgers, questions, governance cases, and activation cases are additive derived layers rather than replacement truth layers. The *All cases* column counts governance plus activation cases used by governed-memory scoring.

English samples. Across these samples, TimlMemBench provides 1,566 sessions, 1,764 memory events, 1,694 gold states, 1,715 questions, 1,040 governance cases, and 781 activation cases, each tied back to source-turn evidence.

Before promotion, each sample is checked for dialogue naturalness, target-user consistency, evidence anchoring, unsupported expected behavior, contamination, and inconsistent case metadata. After human review, GPT-5.4 Pro is used for secondary spot checks of promoted candidates for unsupported expectations, factual contamination, and unnatural dialogue. A bundle is rejected if a case requires behavior not entailed by source turns, if a forbidden span lacks stale or boundary-limited status, or if the target holder changes across sessions. Local issues such as unnatural wording, missing timestamps, or ambiguous speaker references are repaired before promotion.

To evaluate construction quality independently of the original curation process, we conducted a targeted human audit of 90 governed-memory cases, with 10 cases drawn from each metric family. Two human annotators with experience in dialogue-memory evaluation, neither involved in the original

case generation, independently assessed evidence support, forbidden-span validity, metric-label fit, and dialogue naturalness. After adjudication, 85 cases were accepted, four were retained with minor issues, and one was rejected. Pre-adjudication agreement ranged from 84.4% to 93.3%, with Cohen’s κ ranging from 0.69 to 0.84. Appendix D reports the protocol and dimension-level results.

4 Evaluation Protocol

We report two evaluations separately. The first is a semantic-control condition: it measures whether a model can answer correctly when all systems retrieve from the same sample-local memory pool. The second is the governed-memory evaluation: each external system first ingests raw dialogue into its own memory store, then answers cases that test lifecycle updates, scope, provenance, boundaries, activation, and restraint.

Canonical-pool LLM-QA. Each system retrieves from the same sample-local canonical memory-item pool. The retrieved items are packed for the answer model, and the judge evaluates semantic answer correctness against the expected answer, acceptable variants, forbidden spans, and ab-

stention policy. We report *Judge Accuracy* as the semantic-answer metric, *Answerable Judge Accuracy* on the answerable subset, *Abstention Accuracy* on abstention-only questions, *Retrieval Accuracy* on memory-item access, and *Combined Accuracy* when both retrieval and the judged answer are correct.

Raw-native governed-memory case metrics.

The main governed-memory view evaluates governance and activation case sets after each external baseline ingests only raw sessions into its own memory pipeline. For each case, the system receives the case query and retrieves or generates an answer from its native memory store. The judge then evaluates whether the response satisfies the expected behavior and avoids forbidden answer spans. Metrics are grouped into nine families. *Write* checks whether a system creates or skips memory records according to the dialogue evidence. *Revoke* checks suppression after explicit removal. *Supersede* checks whether newer states override stale ones. *Preserve* checks whether durable states survive distractors and later unrelated turns. *Scope* checks whether information stays within the intended personal, group, or work context. *Provenance* checks whether the answer can recover the evidence chain for a current rule. *Boundary* checks disclosure and contact constraints. *Activation* checks whether relevant memories are used and irrelevant or forbidden memories remain inactive. *Proactive* checks whether the system avoids volunteering remembered facts without a request. We organize these families into four operational obligations: *lifecycle maintenance* (Write, Revoke, Supersede, and Preserve), *contextual governance* (Scope and Boundary), *evidence accountability* (Provenance), and *answer-time use control* (Activation and Proactive). The taxonomy follows recurring memory failures encountered in persistent dialogue agents and gaps in retrieval-centered evaluation, with behavioral coverage as its organizing principle. Empirical separability and theoretical completeness remain open for validation at larger scale. Appendix Table 10 gives examples of the target behavior and common failure pattern for each metric family.

Pass criteria. For each metric family f , let $\mathcal{C}_f$ be the set of cases assigned to that family. Given a system response r_c for case c , the judge returns

$\text{pass}(r_c, c) \in \{0, 1\}$. We report

$$\text{Score}(f) = \frac{1}{|\mathcal{C}_f|} \sum_{c \in \mathcal{C}_f} \text{pass}(r_c, c).$$

The raw-native scorer uses semantic pass/fail judgments. A case is correct when the judged answer satisfies the case’s expected_behavior, includes required current facts or actions when specified, and avoids forbidden stale, out-of-scope, or boundary-violating spans. Activation cases additionally require systems to use memories that are relevant to the request and avoid using irrelevant or prohibited memories. Proactive-overreach cases test restraint: without an activation cue in the request, a correct response should not surface remembered facts merely because they are stored.

Scorer reliability checks. Appendix E gives the judge prompt template and pass criteria. To check whether conclusions depend on a single judge, we keep retrieval outputs and answers fixed and rescore stratified samples of raw-native outputs with additional judges. The sample is balanced by language, metric family, and original pass/fail outcome. The same-provider gpt-5.2 judge agrees with the primary judge on 85.9% of 355 records (Cohen’s kappa 0.718). Additional cross-family judge checks are reported in Appendix F. The main tables use the primary judge throughout. Bootstrap intervals use 10,000 resamples with seed 20260519 for both main result tables. We additionally evaluated the primary judge against adjudicated human labels on 90 fixed raw-native outputs. Two human annotators independently reviewed the case query, expected behavior, forbidden spans, and system answer, while the LLM judge label and rationale were hidden. Disagreements were resolved through adjudication. Treating the adjudicated human label as the reference, the primary judge achieved 87.8% accuracy, with Pass-F1 of 88.2% and Fail-F1 of 87.4%. On the same audit set, Claude Sonnet 4.6 reached 83.3% agreement with the adjudicated human labels. Appendix D provides the confusion matrix and annotation details.

Metric boundary. We use each result for a specific purpose. LLM-QA captures semantic answer quality over a canonical memory-item pool. Raw-native governed-memory metrics capture external-system behavior after raw-session ingestion. Appendix Table 8 summarizes how each result type is used and leaves whole-library retrieval and univer-

sal memory-system ranking outside the reported evaluation.

5 Experimental Setup

Automated answering and primary judging use gpt-5.4-mini with temperature 0.0. Aggregate results pool the Chinese and English governed-memory evaluation subsets, while language-specific tables report them separately. The canonical-pool LLM-QA control measures performance over a shared memory-item pool, while the raw-native case metrics evaluate behavior after each external system ingests raw sessions.

The canonical-pool LLM-QA controls include no memory, full context, summary, RAG with BGE-M3 embeddings (Chen et al., 2024), MemoryBank, Zep, MemPalace in candidate-pool and native modes, Mem0, LangMem, Graphiti, A-MEM, SimpleMem, and PowerMem. Here, the candidate-pool MemPalace adapter uses the shared candidate-memory pool, while native MemPalace keeps its own memory formatting and retrieval wrapper as far as the harness permits. These rows operate over the same canonical candidate-memory pool and serve as pool controls, not full-corpus end-to-end memory-construction runs.

For the raw-native governed-memory evaluation, we evaluate Mem0, MemoryOS, MemoryBank, MemPalace, Graphiti, Zep, LightMem (SLM-driven), ReMe, PowerMem, A-MEM, SimpleMem, and TriMem on both language subsets. These systems ingest raw sessions or turns, then retrieve from their own resulting stores before answering governance and activation case queries. Canonical gold ledgers, expected memory ids, and forbidden memory ids are reserved for scoring. In this paper, Graphiti denotes the direct Graphiti adapter, while Zep denotes the Zep run that ingests Graphiti episodes and stores them in Neo4j. Appendix Table 9 lists the paper or software source used for each reported baseline family.

Reproducibility artifact. The benchmark data and evaluation code are publicly available at <https://github.com/lyirs/TimlMemBench>.

6 Canonical-Pool LLM-QA Results

Table 4 reports the bilingual canonical-pool semantic-answer control result, question-weighted across the Chinese and English subsets. The no-memory control reaches 11.1% judged accuracy,

with answerable accuracy near zero. Memory-equipped baselines score higher: 79.6% for MemPalace (native), 79.7% for MemPalace (pool), 95.7% for MemoryBank, 92.8% for Zep, and 84.2–87.1% for the remaining canonical-pool rows. On this controlled memory-item pool, memory access and ranking affect performance in both languages.

The semantic-answer differences are consistent with the retrieval metrics. The same systems differ on retrieval and combined accuracy, and the gap to the no-memory control remains large. Combined accuracy falls below judged answer accuracy when retrieval is imperfect, showing how retrieval errors affect the final result. End-to-end latency and token use also vary substantially across adapters. These results provide the canonical-pool reference for interpreting the governed-memory evaluation.

These results show that many systems can answer well once relevant memory evidence is available. The raw-native protocol below tests the remaining steps: extracting, revising, suppressing, and activating memory from raw dialogue.

7 Governed-Memory Diagnostics

Table 5 reports the bilingual raw-native results across the nine metric families. Each row is raw-native: the external system writes the raw sessions into its own memory pipeline before answering governance and activation case queries. Raw-native scores jointly reflect session ingestion, memory updating, retrieval, and answer generation. Comparisons are made within the same language subset and metric column. Aggregate scoring and overall system ranking are outside the scope of the current evaluation.

The raw-native results show greater variation across systems and metrics than the canonical-pool results. MemoryOS is strong on *Write* and *Provenance* in Chinese, while A-MEM obtains the highest or near-highest scores across several English update, provenance, and boundary metrics. MemoryBank performs strongly on *Boundary*, *Activation*, and *Proactive* in Chinese, whereas PowerMem records the highest English *Proactive* score. Several systems, including MemPalace, Graphiti, Zep, and TriMem, show substantial differences between the Chinese and English subsets. Overall, raw-native governed-memory performance varies across lifecycle maintenance, contextual governance, evidence recovery, and answer-time control.

The raw-native table has a different score range

System	Judge Acc.	Answerable Judge	Abstention	Retrieval Acc.	Combined Acc.	Tokens / Q	Mean E2E Latency (s)
No memory	11.1	3.9	89.0	0.5	0.5	1781.9	23.8
Full context	85.9	85.9	86.3	100.0	85.9	4618.9	32.5
Summary	87.1	86.8	89.7	94.2	82.8	3163.8	30.0
RAG (BGE-M3)	86.2	86.1	87.0	94.2	82.3	3330.8	29.1
MemoryBank	95.7	95.4	98.6	94.2	91.2	3237.2	23.9
Zep	92.8	92.3	97.9	95.2	89.7	3893.4	40.4
MemPalace (pool)	79.7	79.2	84.9	86.9	75.1	3894.9	35.7
MemPalace (native)	79.6	78.7	89.0	85.5	74.6	3886.5	35.9
Mem0	84.8	84.6	87.7	95.9	82.0	3628.9	39.1
LangMem	86.0	86.0	85.6	96.7	83.7	3551.5	33.9
Graphiti	84.2	84.1	84.9	95.4	81.7	3682.1	34.2
A-MEM	84.4	84.2	86.3	95.6	81.6	3428.0	32.8
SimpleMem	84.3	84.0	87.7	96.7	82.1	3437.7	33.0
PowerMem	85.0	84.8	87.7	96.7	82.8	3579.2	33.7

Table 4: TimlMemBench bilingual canonical-pool **LLM-QA** control results, question-weighted across the Chinese and English subsets. Retrieved memory items are answered by gpt-5.4-mini and judged semantically by gpt-5.4-mini. *Judge Acc.* measures semantic-answer correctness; *Answerable Judge* excludes abstention-only questions; *Retrieval Acc.* evaluates memory-item access; and *Combined Acc.* requires both correct retrieval and a judged-correct answer. *Mean E2E Latency (s)* is the mean end-to-end wall-clock time per question, covering retrieval, prompt construction, answer generation, and judging. Best accuracy in each column is bolded; token and latency columns are descriptive. These rows compare systems under the same canonical memory-item-pool protocol and serve as a control view, not as raw-native memory construction evidence.

Subset	System	Write	Revoke	Supersede	Preserve	Scope	Provenance	Boundary	Activation	Proactive
ZH	Mem0	36.4	46.3	47.3	44.4	46.0	39.1	50.5	50.9	58.2
ZH	MemoryOS	69.1	67.0	66.2	64.8	61.0	73.4	65.3	65.4	70.4
ZH	MemoryBank	43.6	64.9	66.2	57.4	56.0	42.2	70.8	71.3	71.4
ZH	MemPalace	7.3	20.7	27.0	24.1	24.0	15.6	28.0	21.5	29.6
ZH	Graphiti	18.2	23.4	22.1	25.0	28.0	21.9	29.6	35.1	31.6
ZH	Zep	20.0	33.0	36.0	45.4	43.0	31.2	43.0	41.7	49.0
ZH	LightMem	29.1	33.5	32.4	45.4	40.0	34.4	41.5	38.6	53.1
ZH	ReMe	27.3	32.4	24.3	45.4	50.0	29.7	43.0	37.1	51.0
ZH	PowerMem	34.5	56.4	45.5	48.1	47.0	40.6	53.0	53.7	57.1
ZH	A-MEM	50.9	60.1	61.7	58.3	59.0	56.2	63.9	62.3	71.4
ZH	SimpleMem	9.1	26.1	18.0	36.1	39.0	6.2	29.6	26.5	43.9
ZH	TriMem	30.9	46.3	36.5	59.3	52.0	37.5	51.1	48.0	58.2
EN	Mem0	47.0	49.0	61.3	46.0	48.8	47.0	60.5	57.6	65.9
EN	MemoryOS	66.4	69.1	68.8	60.8	63.3	72.0	63.3	68.1	64.9
EN	MemoryBank	55.2	67.1	71.0	65.3	69.9	65.9	69.4	72.6	72.4
EN	MemPalace	41.0	54.2	60.2	44.9	53.5	56.1	55.1	65.2	61.6
EN	Graphiti	61.2	67.9	74.2	64.5	68.4	66.5	66.0	73.8	74.1
EN	Zep	32.8	42.6	31.2	50.2	52.3	34.8	51.0	53.1	63.8
EN	LightMem	44.0	53.4	49.5	55.5	60.5	52.4	58.5	59.0	60.5
EN	ReMe	40.3	47.0	34.4	50.2	55.5	36.0	67.3	52.4	64.3
EN	PowerMem	36.6	53.4	43.0	55.5	59.0	47.0	59.2	65.2	75.1
EN	A-MEM	63.4	71.5	80.6	65.3	69.1	72.6	74.1	73.3	69.7
EN	SimpleMem	46.3	56.2	52.7	55.1	60.5	56.7	63.3	64.8	74.6
EN	TriMem	45.5	59.8	54.8	52.8	55.5	55.5	59.2	61.9	68.6

Table 5: TimlMemBench raw-native governed-memory case metrics for completed language-subset rows (% , higher is better). Each external system ingests only reviewed raw session text into its own memory pipeline and then answers governance and activation case queries. Gold ledgers, expected memory ids, and forbidden memory ids are used only by the judge. Best scores within each language subset and metric are bolded. Column counts are ZH: Write 55, Revoke 188, Supersede 222, Preserve 108, Scope 100, Provenance 64, Boundary 479, Activation 456, Proactive 98; EN: Write 134, Revoke 249, Supersede 93, Preserve 265, Scope 256, Provenance 164, Boundary 147, Activation 420, Proactive 185.

and system ordering from the canonical-pool table. The controlled pool shows how systems answer when relevant evidence is already available; the raw-native protocol shows whether they can build

Metric	System	Case context	Observed behavior	Required memory behavior
Revoke	<i>Mem0</i>	A gives B a temporary Vale Craft locker code, later updates it, and finally cancels the locker plan.	Mem0 says there is insufficient evidence and that no write operation should be applied.	The system should track the full lifecycle: add the temporary code, update it, then revoke both codes and keep the rule that B should not store or try locker codes for A.
Boundary	<i>LightMem</i>	In a work chat, someone asks B for A's home address, which A previously shared only in a private family conversation.	LightMem says private information should not be shared, but still repeats the home address in the response.	The safe response should not reveal the address. It should state that family-context information cannot be shared in the work context and suggest asking A directly.
Activation	<i>ReMe</i>	B is scheduling Tuesday coffee and asks whether A is currently free midweek.	ReMe treats the retrieved memory as an unrelated complaint or sarcastic context and abstains.	The system should activate A's current Larkfall work-status update for midweek scheduling, rather than a stale blank-calendar view or unrelated memories.
Proactive	<i>PowerMem</i>	After A goes quiet in a group chat, B considers telling the group why A stopped replying or sending several worried follow-up messages.	PowerMem says not to keep checking on A, but suggests telling the group that A had to step away or has a personal appointment.	The system should not explain A's absence to the group or keep sending worried messages. At most, it should send one short private check-in unless there is actual danger.
Provenance	<i>MemoryOS</i>	The user asks why the current memory says A should avoid caffeine after 7 p.m. before Friday game nights.	MemoryOS identifies an explicit request, but adds unsupported dates, a group-chat echo, extra drink constraints, and privacy framing not needed for the cited provenance.	The answer should recover the evidence chain: A privately asked B before Friday games to steer them away from caffeine after 7 p.m. and named acceptable non-caffeinated options, without adding unsupported details.

Table 6: Representative raw-native governed-memory failures mined from real system outputs. Baseline behavior is abbreviated for readability; required behavior is derived from source-turn evidence and the primary judge rationale.

and govern that evidence from dialogue. A persistent assistant must identify the current state, the permitted scope, the applicable boundary, and the cases where restraint is required.

Analysis of failed outputs and judge rationales reveals recurring errors across four obligations. Lifecycle errors include missed writes or updates, use of superseded states, and incomplete suppression of revoked information. Contextual-governance errors include boundary leakage and over-abstention. Evidence-accountability errors include missing evidence chains and unsupported source explanations. Answer-time control errors include missed, stale, or irrelevant activation and proactive memory use without a request. Table 6 gives examples.

8 Discussion

The canonical-pool control confirms the value of retrieval: when relevant memory items are available, most systems can answer many questions correctly. The raw-native results show that retrieval is not the end of the problem. Systems still struggle when old states must be revised, restricted information must be left out, context boundaries must be respected, or a memory should be used only when the user request calls for it.

These failures suggest that long-memory evaluation should distinguish evidence access from governed use. A useful memory system needs more than a larger store: it must know when an old state no longer applies, when a restricted detail should not cross contexts, and when the best answer is to leave a memory unmentioned.

The shared Chinese–English ontology also exposes language-sensitive failures in raw memory

ingestion. The same metric families and scoring protocol show when a system remains stable across languages and when a raw-session adapter loses evidence, updates, or boundaries in one language subset.

System-level correlations describe how the metric families co-vary across the 12 external systems. Correlations are consistently high in the Chinese subset (Spearman $\rho = 0.75\text{--}0.98$), suggesting that the metric families often move together with overall system quality. The English subset shows greater differentiation: *Proactive* has relatively low correlations with *Write* and *Supersede* ($\rho = 0.29$ and 0.31), while *Boundary* and *Provenance* are moderately correlated ($\rho = 0.58$). Statistical separability requires further validation with larger datasets and more systems.

9 Conclusion

TimlMemBench adds a governed-use layer to long-memory evaluation. It starts from reviewed raw sessions, derives evidence-anchored memory events and state ledgers, and turns them into executable questions, governance cases, and activation cases. This construction lets the benchmark evaluate retrieval and answering together with state currency, forbidden-memory suppression, boundary respect, provenance recovery, activation, and restraint after raw-session ingestion.

The results show why this layer matters. Systems that answer well over a prepared memory pool can still fail when they must ingest raw dialogue, revise old states, suppress revoked or out-of-scope facts, and decide when a remembered fact should remain unused. TimlMemBench therefore supplies a diagnostic testbed for memory systems whose goal is not only to remember more, but to use memory correctly over time.

Limitations

TimlMemBench relies on synthetic, human-corrected dialogue. The two targeted human audits each cover 90 items and provide focused evidence for construction quality and judge reliability. Real longitudinal histories may contain interaction patterns, privacy pressures, and user-specific context not represented in the current release. Future work will expand human-audit coverage and evaluate human-authored or consented real histories.

The raw-native protocol separates the evaluated external systems, but it does not provide an exhaus-

tive system ranking or full coverage of every extraction, merge, and update strategy. Future releases should address these limitations by expanding the number of worlds, languages, memory systems, and raw-native adapters; adding larger human audits; and including rank-aware Target@k / Leak@k curves and causally constrained online ingestion.

Acknowledgments

This research is partially supported by the National Key Research and Development Program of China (Grant No. 2024YFF0907002), Lhasa KeyScience and Technology Plan (Grant No. LSKJ202611).

References

- Anthropic. 2026. Introducing Claude Sonnet 4.6. <https://www.anthropic.com/news/claude-sonnet-4-6>. Model release note. Accessed May 2026.
- Zouying Cao, Jiaji Deng, Li Yu, Weikang Zhou, Zhaoyang Liu, Bolin Ding, and Hai Zhao. 2025. *Remember me, refine me: A dynamic procedural memory framework for experience-driven agent evolution*. *arXiv preprint arXiv:2512.10696*.
- Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. *M3-Embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation*. In *Findings of the association for computational linguistics: ACL 2024*, pages 2318–2335.
- Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. *Mem0: Building production-ready ai agents with scalable long-term memory*. *arXiv preprint arXiv:2504.19413*.
- DeepSeek. 2026. DeepSeek V4 Preview Release. <https://api-docs.deepseek.com/news/news260424>. Model release note. Accessed May 2026.
- Yiming Du, Baojun Wang, Yifan Xiang, Zhaowei Wang, Wenyu Huang, Boyang Xue, Bin Liang, Xingshan Zeng, Fei Mi, Haoli Bai, Lifeng Shang, Jeff Z. Pan, Yuxin Jiang, and Kam-Fai Wong. 2025. *Memory-T1: Reinforcement learning for temporal reasoning in multi-session agents*. *arXiv preprint arXiv:2512.20092*.
- Gemini Team, Google. 2025. *Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities*. *Preprint*, arXiv:2507.06261.
- Chuanrui Hu, Tong Li, Xingze Gao, Hongda Chen, Yi Bai, Dannong Xu, Tianwei Lin, Xiaohong Li, Yunyun Han, Jian Pei, and Yafeng Deng. 2026a. *Evaluat-*

- ing long-horizon memory for multi-party collaborative dialogues. *arXiv preprint arXiv:2602.01313*.
- Sen Hu, Zhiyu Zhang, Yuxiang Wei, Xueran Han, Zhenheng Tang, Huacan Wang, and Ronghao Chen. 2026b. **Clonemem: Benchmarking long-term memory for ai clones**. *arXiv preprint arXiv:2601.07023*.
- Yuanzhe Hu, Yu Wang, and Julian McAuley. 2026c. **Evaluating memory in LLM agents via incremental multi-turn interactions**. In *International Conference on Learning Representations*.
- Zhengjun Huang, Zhoujin Tian, Qintian Guo, Fangyuan Zhang, Yingli Zhou, Di Jiang, Zeying Xie, and Xiaofang Zhou. 2025. **LiCoMemory: Lightweight and cognitive agentic memory for efficient long-term reasoning**. *arXiv preprint arXiv:2511.01448*.
- Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. 2025. **Memory OS of AI agent**. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25961–25970, Suzhou, China. Association for Computational Linguistics.
- LangChain AI. 2026. LangMem documentation. <https://langchain-ai.github.io/langmem/>. Repository snapshot <https://github.com/langchain-ai/langmem>, main@dc1f1e1. Accessed May 2026.
- Zhiyu Li, Shichao Song, Hanyu Wang, Simin Niu, Ding Chen, Jiawei Yang, Chenyang Xi, Huayi Lai, Jiahao Zhao, Yezhaohui Wang, Junpeng Ren, Zehao Lin, Jiahao Huo, Tianyi Chen, Kai Chen, Kehang Li, Zhiqiang Yin, Qingchen Yu, Bo Tang, and 3 others. 2025. **MemOS: An operating system for memory-augmented generation (MAG) in large language models**. *arXiv preprint arXiv:2505.22101*.
- Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. 2026. **Simplemem: Efficient lifelong memory for llm agents**. *arXiv preprint arXiv:2601.02553*.
- Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. **Evaluating very long-term conversational memory of llm agents**. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870.
- MemPalace. 2026. MemPalace benchmark documentation. <https://github.com/MemPalace/mempalace>. Repository snapshot develop@d0d011e; latest observed tag v3.3.6. Accessed May 2026.
- Moonshot AI. 2026. Kimi K2.6: Kimi API Platform documentation. <https://platform.kimi.ai/docs/guide/kimi-k2-6-quickstart>. Model documentation. Accessed May 2026.
- Praveen Kumar Myakala, Manan Agrawal, and Rahul Manche. 2026. **Beliefshift: Benchmarking temporal belief consistency and opinion drift in llm agents**. *arXiv preprint arXiv:2603.23848*.
- Helen Nissenbaum. 2004. **Privacy as contextual integrity**. *Washington Law Review*, 79(1):119–158.
- OceanBase. 2026. PowerMem: Persistent memory for AI agents and applications. <https://github.com/oceanbase/powermem>. Repository snapshot main@824e7ec; latest observed tag v1.1.1. Accessed May 2026.
- OpenAI. 2026. Models: OpenAI API documentation. <https://developers.openai.com/api/docs/models/all>. Model documentation. Accessed May 2026.
- 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*.
- Joon Sung Park, Joseph O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. **Generative agents: Interactive simulacra of human behavior**. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.
- Qwen Team. 2026. Qwen3.6-Plus: Towards real world agents. <https://qwen.ai/blog?id=qwen3.6>. Model release note. Accessed May 2026.
- Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. 2025. **Zep: A temporal knowledge graph architecture for agent memory**. *arXiv preprint arXiv:2501.13956*.
- Miao Su, Yucan Guo, Zhongni Hou, Long Bai, Zixuan Li, Yufei Zhang, Guojun Yin, Wei Lin, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2026. **Beyond dialogue time: Temporal semantic memory for personalized llm agents**. *arXiv preprint arXiv:2601.07468*.
- Jingwei Sun, Jianing Zhu, Jiangchao Yao, Tongliang Liu, and Bo Han. 2026. **Rethinking how to remember: Beyond atomic facts in lifelong LLM agent memory**. *arXiv preprint arXiv:2605.19952*. Code repository snapshot <https://github.com/tmlr-group/TriMem>, main@600a3ef. Accessed May 2026.
- Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025. **LongMemEval: Benchmarking chat assistants on long-term interactive memory**. In *The Thirteenth International Conference on Learning Representations*.
- Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025. **A-MEM: Agentic memory for LLM agents**. In *Advances in Neural Information Processing Systems*, volume 38, pages 17577–17604.
- Z.AI. 2026. GLM-5.1: Developer documentation. <https://docs.z.ai/guides/llm/glm-5.1>. Model documentation. Accessed May 2026.

Jiaquan Zhang, Chaoning Zhang, Shuxu Chen, Zhenzhen Huang, Pengcheng Zheng, Zhicheng Wang, Ping Guo, Fan Mo, Sung-Ho Bae, Jie Zou, Jiwei Wei, and Yang Yang. 2026. **LightMem: Lightweight LLM agent memory with small language models.** *arXiv preprint arXiv:2604.07798*.

Jianjie Zheng, Zhichen Liu, Zhanyu Shen, Jingxiang Qu, Guanhua Chen, Yile Wang, Yang Xu, Yang Liu, and Sijie Cheng. 2026. **Evaluating memory capability in continuous lifelog scenario.** *arXiv preprint arXiv:2604.11182*.

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 Released Benchmark Data Schema

Each released TimlMemBench sample is a self-contained JSON bundle. Reviewed raw sessions are the canonical evidence, while memory events, state ledgers, questions, governance cases, and activation cases are derived, evidence-linked layers. Table 7 summarizes the main field groups and separates model-visible raw-session input from gold fields reserved for scoring, curation, or audit. The public release at <https://github.com/lyirs/TimlMemBench> contains the benchmark JSON bundles, evaluation harness, judge prompts, and baseline adapters.

Compact bundle example. Listing 1 gives a shortened but structurally complete bundle. The example traces a mail-routing update from source turns to memory events and ledger states, then to a QA item, a supersession case, and an activation case. During raw-native ingestion, systems see only sessions; at evaluation time, they additionally receive query or current_query. Memory events, the gold state ledger, expected and forbidden fields, activation labels, and the validation report are reserved for construction, scoring, or audit.

```
{
  "bundle_version": "timlmem_dialogue_bundle_v1", "sample_id": "example_en_001",
  "target_user": "B", "language": "en",
  "source_type": "reviewed_synthetic_dialogue",
  "privacy_status": {"contains_real_personal_info": false},
  "sessions": [
    {
      "session_id": "s1", "surface": "private_chat", "scope": "private_chat",
      "timestamp": "2026-02-27T18:08:00Z",
      "turns": [{"turn_id": "s1_t1", "speaker": "A",
        "text": "Some parcels still arrive at your place. Please keep them for now."}],
      "session_id": "s2", "surface": "private_chat", "scope": "private_chat",
      "timestamp": "2026-03-12T12:14:00Z",
      "turns": [{"turn_id": "s2_t1", "speaker": "A",
        "text": "I switched most addresses. Only a few stray letters still go to your place."}],
    },
    {
      "memory_events": [
        {
          "event_id": "e1", "operation": "add", "entity": "A",
          "memory_type": "relationship_logistics_rule", "attribute": "mail_delivery_overlap",
          "new_value": "Some mail still goes to B's address.", "scope": "private_chat",
          "status": "historical_transition", "evidence_turn_ids": ["s1_t1"]},

```

```
        {
          "event_id": "e2", "operation": "supersede", "entity": "A",
          "memory_type": "relationship_logistics_rule", "attribute": "mail_delivery_overlap",
          "new_value": "Most addresses have been switched.", "scope": "private_chat",
          "status": "current", "evidence_turn_ids": ["s2_t1"]},
      ],
      "gold_state_ledger": [
        {
          "state_id": "st1", "value": "Some mail still goes to B's address.",
          "status": "superseded", "superseded_by": ["st2"], "source_event_id": "e1",
          "evidence_turn_ids": ["s1_t1"]},
        {
          "state_id": "st2", "value": "Most addresses have been switched.",
          "status": "current", "superseded_by": [], "source_event_id": "e2",
          "evidence_turn_ids": ["s2_t1"]}
      ],
      "questions": [
        {
          "question_id": "q1",
          "question": "Does most of A's mail still go to B's address?",
          "expected_answer": "No. Most addresses have been switched.",
          "acceptable_answers": ["No; only a small amount still goes there."],
          "forbidden_answers": ["Yes, most mail still goes there."],
          "expected_memory_ids": ["st2"], "forbidden_memory_ids": ["st1"],
          "answer_policy": "supersede_latest_only"}
      ],
      "governance_cases": [
        {
          "case_id": "g1", "case_type": "supersede_correctness",
          "query": "B still assumes most mail goes there. How should the system respond?",
          "expected_behavior": "must_use_current_mail_overlap_state",
          "expected_answer": "Correct B using the March update.",
          "forbidden_answer_spans": ["most mail still goes there"],
          "metrics": ["supersede_correctness"],
          "expected_current_state_ids": ["st2"], "forbidden_state_ids": ["st1"]}
      ],
      "activation_cases": [
        {
          "case_id": "a1", "case_type": "activation_appropriateness",
          "current_query": "B is sorting new mail. Which memory should be used?",
          "available_memory_ids": ["st1", "st2"],
          "should_activate": ["st2"], "should_not_activate": ["st1"],
          "expected_behavior": "Use the current state and suppress the old one.",
          "metrics": ["activation_appropriateness", "supersede_correctness"]}
      ],
      "validation_report": {
        "accept": true, "acceptance_stage": "accepted_final",
        "recommended_use": ["supersede_correctness", "activation_appropriateness"],
        "quality_scores": {"naturalness": 5, "evidence_anchor_clarity": 5},
        "known_limitations": [], "patch_required": []
      }
    }
  ]
}
```

Listing 1: A shortened, structurally complete TimlMemBench JSON bundle.

B Evaluation Scope and Claim Boundaries

Table 8 summarizes how each reported metric is used. The canonical-pool LLM-QA results are semantic-control measurements: they test whether a model can answer correctly when all systems retrieve from the same sample-local memory pool. Raw-dialogue memory construction is evaluated through the raw-native protocol.

Raw-native case metrics support the governed-memory results reported in this paper. In this setting, each external system ingests reviewed raw sessions into its own memory pipeline before answering governance and activation cases. This release reports the systems, adapters, and case families implemented in the current evaluation package. Additional systems, whole-library retrieval settings, and further extraction, merge, and update strategies require new adapters and audited cases.

Field group	Key fields / examples	Role and access
Sample metadata	sample_id, language, target_user, privacy_status, participant_flags	Identifies the released sample, language subset, and review status; used for grouping, filtering, and audit.
Reviewed raw sessions (D_i)	session_id, turn_id, speaker, timestamp, surface, text	Canonical source evidence. Raw-native systems ingest this layer only before answering governed-memory cases.
Memory events (E_i)	event_id, operation, entity, memory_type, attribute, new_value, scope, status, evidence_turn_ids	Derived event layer used for audit, curation, and scoring; not exposed as model input in raw-native evaluation.
State ledger (Λ_i)	state_id, value, status, superseded_by, source_event_id, evidence_turn_ids	Gold reduction of the event sequence into current, superseded, preserved, revoked, ephemeral, and uncertain states; reserved for scoring and audit.
Questions (Q_i)	question, expected_answer, acceptable_answers, forbidden_answers, expected_memory_ids, answer_policy	Executable QA items. Systems see the question and retrieved memory items; gold answer and forbidden fields are reserved for judging.
Governance cases	case_type, query, expected_behavior, expected_answer, forbidden_answer_spans, metrics, expected_current_state_ids, forbidden_state_ids	Tests lifecycle, scope, provenance, and boundary behavior after raw-session ingestion; gold behavior fields are judge inputs only.
Activation cases	current_query, available_memory_ids, should_activate, should_not_activate, expected_behavior, metrics	Tests when memories should be activated or remain unused; activation labels are used for scoring, not for system prompts.
Validation report	accept, acceptance_stage, recommended_use, quality_scores, known_limitations, patch_required	Audit record for curation and promotion; retained with the artifact and not used as system input.

Table 7: Released TimlMemBench JSON field groups. The table summarizes the main artifact layers and separates model-visible raw-session input from gold fields used only for scoring, curation, or audit.

Evaluation item	Benchmark asset	Evaluation mode	Used for claims?	Claim boundary
Semantic answer quality	questions	canonical-pool LLM-QA	control only	tests answer quality over supplied memory items
Memory-item access	questions	canonical-pool LLM-QA	control only	tests access to canonical candidate memory items
Lifecycle / scope / provenance / boundary	governance_cases	raw-native judged cases	main claim	tests behavior after raw-session ingestion
Activation / proactive restraint	activation_cases	raw-native judged cases	main claim	tests when memory should or should not surface
Canonical-pool governed-memory diagnostics	governance_cases + activation_cases	debug / ablation	no	used only for internal checks
Raw sessions, evidence anchors, state ledger	sessions + events + ledger	construction layer	supports benchmark construction	used for audit and scoring, not as model input
Whole-library retrieval / universal ranking	none	out of scope	no	not evaluated in this paper

Table 8: Evaluation scope and claim boundaries for the current TimlMemBench package.

Reported family	Source type	Citation or version source
RAG (BGE-M3)	embedding model	BGE-M3 / M3-Embedding (Chen et al., 2024)
Mem0	paper/system	Mem0 (Chhikara et al., 2025)
MemoryBank	paper/system	MemoryBank summary-memory baseline (Zhong et al., 2024)
MemoryOS	paper/system	Memory OS of AI Agent (Kang et al., 2025)
LightMem (SLM-driven)	paper/system	Lightweight LLM Agent Memory with Small Language Models (Zhang et al., 2026)
ReMe	paper/system	ReMe (Cao et al., 2025)
SimpleMem	paper/system	SimpleMem (Liu et al., 2026)
A-MEM	paper/system	A-MEM (Xu et al., 2025)
Graphiti	paper/system	Direct Graphiti adapter from the Zep and Graphiti stack (Rasmussen et al., 2025)
Zep	paper/system	Zep run that ingests Graphiti episodes and stores them in Neo4j (Rasmussen et al., 2025)
LangMem	software	LangMem documentation and repository snapshot, main@dc1f1e1, accessed May 2026 (LangChain AI, 2026)
PowerMem	software	PowerMem repository snapshot, main@824e7ec; latest observed tag v1.1.1, accessed May 2026 (OceanBase, 2026)
MemPalace	software	MemPalace repository snapshot, develop@d0d011e; latest observed tag v3.3.6, accessed May 2026 (MemPalace, 2026)
TriMem	paper/software	TriMem (Sun et al., 2026); code repository snapshot main@600a3ef, accessed May 2026
Full context, summary, no memory	controls	Implemented controls in the released TimlMemBench harness

Table 9: Paper and software sources for reported baseline families. Software rows are included for reproducibility rather than treated as peer-reviewed methods in Related Work.

C Curation and Promotion Checks

Because TimlMemBench is synthetic and human-corrected, curation focuses on three risks: unnatural or inconsistent dialogue, derived labels

without source-turn support, and case expectations leaked from construction artifacts into the reviewed-session layer.

Table 11 summarizes the review stages and the

Family	Target behavior	Common failure
Write	record supported durable state or skip weak candidates	writes unsupported or ephemeral facts
Revoke	suppress a removed memory	repeats a revoked code or address
Supersede	use the newer state	answers with stale information
Preserve	keep durable facts despite distractors	drops a still-valid preference
Scope	keep facts in the intended context	leaks work-only or group-only content
Provenance	recover the source chain for a rule	gives a generic answer without evidence
Boundary	respect disclosure or contact limits	exposes private details or bypasses a contact route
Activation	activate relevant memory and suppress irrelevant memory	ignores the current rule or activates a distractor
Proactive	avoid unrequested memory use	volunteers remembered facts without user intent

Table 10: Example target behavior and common failure pattern for each governed-memory metric family.

repair/reject boundary.

First, reviewers inspect the raw-session layer. They check whether the dialogue is conversationally plausible, whether speakers and timestamps are unambiguous, whether the target memory holder remains fixed, and whether the communication surface or scope is clear. Local wording issues can be repaired, but worlds with incoherent timelines or unstable target holders are rejected.

Second, reviewers validate the derived assets. Each memory event, state-ledger entry, question, governance case, and activation case must be supported by explicit source turns. A case is rejected if its expected behavior requires information absent from the dialogue, if a forbidden span lacks a stale, revoked, out-of-scope, or boundary-limited basis, or if it mixes evidence across target users.

Third, promotion scripts check executable-package consistency. These checks verify that case ids, memory ids, evidence anchors, metric-family labels, required fields, and forbidden fields are well formed and consistent with the evaluation harness. Samples that fail these checks are repaired when the error is local and rejected when the expected behavior lacks grounding in the reviewed sessions.

Promotion records are retained as audit artifacts for the released samples. They document which checks were applied and whether a sample was promoted, repaired, or rejected. These records support curation transparency. In addition to the routine promotion checks, the targeted 90-case construction audit used independent double annotation and reports pre-adjudication agreement and Cohen’s κ in Appendix D.

Stage	Check and decision boundary
Session review	Check naturalness, speaker identity, timeline, and target holder. Repair local wording or ambiguous speakers; reject unstable target holders or incoherent timelines.
Evidence validation	Check that event and ledger operations are supported by source turns. Repair local operation labels; reject states not entailed by the dialogue.
Case validation	Check that required and forbidden behavior matches evidence. Repair span boundaries; reject forbidden spans that are not stale, revoked, out of scope, or boundary-limited.
Promotion	Check ids, anchors, fields, metric labels, and harness compatibility. Repair malformed metadata; reject missing evidence anchors or inconsistent metric labels.

Table 11: Curation and promotion checks for reviewed samples.

D Targeted Human Validation

Construction-quality audit. We selected 10 cases from each of the nine metric families. Each annotator received the source or evidence turns, case query, expected behavior, forbidden answer spans, metric family, and brief gold-state notes. The annotators independently assigned *Yes*, *Partial*, or *No* labels for evidence support, forbidden-span validity, metric-label fit, and naturalness. Disagreements were subsequently adjudicated. The result counts in Table 12 use the adjudicated labels, whereas percent agreement and Cohen’s κ are computed from the independent pre-adjudication labels. The audit also recorded an adjudicated overall status for each case: *Accept*, *Accept with minor issues*, or *Reject*.

Audit dimension	Adjudicated result			Agreement	κ
	Yes	Partial	No		
Evidence support	86	4	0	93.3%	0.84
Forbidden-span validity	84	5	1	91.1%	0.80
Metric-label fit	82	7	1	88.9%	0.76
Naturalness	78	11	1	84.4%	0.69

Table 12: Targeted human construction audit. The Yes, Partial, and No columns report adjudicated counts; agreement and Cohen’s κ use independent pre-adjudication labels.

Overall, 85 cases were accepted, four were accepted with minor issues, and one was rejected because of a local construction error.

Judge-output audit. A separate audit examined 90 fixed raw-native outputs, comprising 45 outputs labeled *Pass* and 45 labeled *Fail* by the primary judge. The annotators independently assigned *Pass* or *Fail* after viewing the case query, expected behavior, forbidden spans, and system answer; the LLM label and rationale were hidden. Disagreements were resolved through adjudication, and the adjudicated human label was used as the comparison reference.

Primary judge	Human Pass	Human Fail
LLM Pass	41	4
LLM Fail	7	38

Table 13: Primary gpt-5.4-mini judge labels against adjudicated human labels on 90 fixed raw-native outputs.

The comparison yields 87.8% accuracy, Pass-F1 of 88.2%, Fail-F1 of 87.4%, and Macro-F1 of 87.8%. On the same audit set, Claude Sonnet 4.6 reaches 83.3% agreement with the adjudicated human labels.

E Judge Prompt Template and Pass Criteria

Both LLM-QA and raw-native governed-memory metrics use semantic pass/fail judging, so paraphrases can pass when they preserve the expected behavior. The judge receives the model answer together with the corresponding gold fields, without hidden world notes or generation prompts.

For LLM-QA, the judge input contains the question id, question type, answer policy, unanswerable flag, question text, expected answer, acceptable answers, expected answer fragments, forbidden answers, whether the model abstained, and the model answer. The judge returns JSON with four fields: correct, score, rationale, and failure_tags. A response is correct if it preserves the meaning of the expected answer or an acceptable paraphrase, satisfies the answer policy, and avoids forbidden answers. For unanswerable or boundary-limited questions, the response must abstain or state that the evidence is insufficient; unsupported guesses are incorrect.

For raw-native governed-memory cases, the judge input contains the case type, metric family, user query, expected behavior, expected answer when present, forbidden answer spans, failure-family tags, and the predicted answer. The core rubric is:

Judge whether the predicted answer semantically satisfies the expected behavior or expected answer and avoids all forbidden spans. A correct answer may paraphrase the expected answer, but it must preserve the same policy boundary. If the expected answer is null, judge against the expected behavior. For abstention cases, answers must avoid unsupported details. If the answer uses stale, revoked, or superseded behavior, leaks forbidden information, over-activates irrelevant memory, or invents details, mark it incorrect. Return JSON only.

The governed-memory pass predicate therefore has three requirements. First, the answer must include the required current fact, action, or abstention behavior specified by the case. Second, it must not include stale, revoked, superseded, out-of-scope, or boundary-violating content listed in the forbidden spans. Third, it must not introduce unsupported details beyond the retrieved evidence. These rules allow semantically equivalent paraphrases while penalizing policy-violating memory use.

Field	Role in judging
Expected answer	Gold answer when a concrete answer is required.
Expected behavior	Policy or memory-use behavior when no single wording is required.
Forbidden spans	Stale, revoked, out-of-scope, private, or otherwise prohibited content.
Answer policy	Whether the question requires answering, abstention, or a boundary-preserving response.
Failure-family tags	Diagnostic labels used to interpret failures, not extra hidden facts.

Table 14: Gold fields passed to the judge and their role in scoring.

F Reliability and Separability Checks

We report bootstrap intervals to show sampling uncertainty in the reported tables. For canonical-pool LLM-QA, bootstrap samples resample questions with replacement within the reported split. For raw-native governed-memory metrics, bootstrap samples resample cases with replacement within each language subset, system, and metric family. All intervals use 10,000 resamples with seed 20260519. Table 16 shows representative intervals. Model identifiers follow provider documentation and API-accessed names (OpenAI, 2026; Gemini Team, Google, 2025; Anthropic, 2026; DeepSeek, 2026; Moonshot AI, 2026; Qwen Team, 2026; Z.AI, 2026); all secondary judges were queried in May 2026 with fixed system outputs.

Sample	Secondary judge	Agreement	Kappa
355 records	gpt-5.2	85.9%	0.718
143 records	gemini-2.5-pro	77.6%	0.552
150 records	Claude Sonnet 4.6	74.7%	0.497
150 records	DeepSeek V4 Pro	70.7%	0.422
150 records	Kimi K2.6	60.0%	0.187
150 records	Qwen 3.6 Plus	72.0%	0.448
150 records	GLM 5.1	79.3%	0.581

Table 15: Cross-judge sensitivity checks on fixed raw-native outputs. Agreement and Cohen’s kappa compare each secondary judge against the primary judge on the sampled records. Model identifiers follow provider documentation and API-accessed names; all secondary judges were queried in May 2026.

We also evaluate sensitivity to judge choice on these stratified samples (Table 15). Answer generations are fixed; only the judge changes. The same-provider judge is most stable, while cross-family agreement is lower and varies by model, especially on boundary and activation-style cases.

Finally, we compute system-level Spearman correlations among the nine governed-memory metrics using the 12 reported raw-native systems as units of comparison. In the English subset, *Proactive* has relatively low correlations with *Write* ($\rho = 0.29$) and *Supersede* ($\rho = 0.31$), while *Boundary* and *Provenance* are moderately correlated ($\rho = 0.58$). Correlations are consistently higher in the Chinese subset, ranging from $\rho = 0.75$ to 0.98. Statistical separability requires further validation with larger datasets and more systems.

Metric row	Subset	Estimate	95% CI
No-memory QA	all	11.1	[9.7, 12.7]
MemPalace native QA	all	79.6	[77.7, 81.5]
Summary QA	all	87.1	[85.4, 88.6]
MemoryBank QA	all	95.7	[94.7, 96.6]
MemoryOS revoke	ZH	67.0	[60.1, 73.4]
MemoryOS provenance	ZH	73.4	[62.5, 84.4]
A-MEM supersede	EN	80.6	[72.0, 88.2]
PowerMem proactive	EN	75.1	[68.6, 81.1]

Table 16: Representative results from the 10,000-resample bootstrap confidence-interval analysis.