EXECUTABLE GOLD SEMANTICS: COMPILING GOLD LABELS FOR AUDITING MEMORY TRANSITIONS IN LANGUAGE AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Memory systems let language agents carry information across long interactions, and they are evaluated by whether the agent eventually answers questions correctly. Whether each individual memory operation was right is rarely measured, because per-operation labels are held to be impractical to obtain. The one operation-level benchmark generates its memory points with a language model and verifies them by hand, and no benchmark records how a displaced belief should be read, who observed a change, or what the agent still owes. We show that once a scenario schema is fixed, these labels can be compiled instead of annotated. A small router of belief-change operators computes, for every event, the correct operation, the successor memory, whether the displaced belief was once true or wrong all along, who observed the change, and the open obligations. We prove that the router is well defined and runs in polynomial time, that every compiled trace satisfies contracts stated over states alone, and that six axioms characterize it, so that dropping any one yields an operator with a recognizable design flaw. The same semantics generates gold with paired counterfactuals, judges systems against it, and defines a training signal and a runtime guard. We release MEMLEDGER, the compiler and a benchmark of 900 sessions in two languages, and use it to ask what each evaluation view can see. On a balanced suite of 1,200 seeded contract violations, questions about final values detect under one percent, the best single question per violation class detects three quarters, and one large language model given the complete record in a single pass detects under ten percent. Changing only how a displaced belief is interpreted inside a real memory framework moves its answers about the past from 2 to 33 correct out of 36. Two author annotators, working without access to the router or to each other's labels, agreed with the compiled labels on 1,199 and 1,177 of 1,200 items, and with each other on 1,178.

1 INTRODUCTION

Memory systems are becoming a first-class component of language agents, and the field evaluates them through final answers: LongMemEval (Wu et al., 2025), LoCoMo (Maharana et al., 2024), MemoryAgentBench (Hu et al., 2026) and AMA-Bench (Zhao et al., 2026) ask whether the agent eventually answers correctly. Training follows the same premise; Memory-R1 (Yan et al., 2025) and Mem-α (Wang et al., 2025) are driven by outcome rewards, and Memory-R1 states that supervised fine-tuning "provides limited help because it is impractical to label every memory operation".

An answer-level evaluation reads the memory once, at the end, through a question. It therefore cannot tell whether each operation along the way was right: a write that should have been a correction, a deletion that left the ledger claiming an update, or a change shown to the wrong party can all leave the final answer intact. HaluMem (Chen et al., 2025) moved one step toward operations by generating persona memory points with a language model, writing dialogues around them and having humans verify the result, and gold-first synthetic generation has a longer history, from bAbI (Weston et al., 2015) and theory-of-mind stories (Le et al., 2019; Wu et al., 2023; Kim et al., 2023) to schema-guided dialogue (Shah et al., 2018; Rastogi et al., 2020). In all of these the label records which facts should be stored. None records how a displaced belief should be interpreted, who observed the change, or what the agent still owes, and none comes with a guarantee that the label is the right one.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

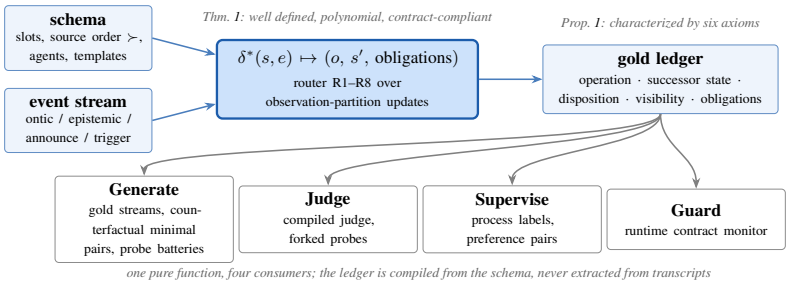

Figure 1: One executable semantics, four consumers. A schema and an event stream are compiled by the router δ^* into a per-step gold ledger recording the operation, the successor state, the disposition of displaced beliefs, who observed the transition and which obligations were discharged; the same function generates data, judges systems, and defines a training signal and a runtime guard.

Those three fields are what an audit of memory needs. The *disposition* of a displaced belief says whether the old value was once true and later changed or was wrong from the start; the two cases give opposite correct answers to any question about the past. The *scope* of a transition says which parties observed it. The *obligations* say what the agent must still do, such as ask when evidence cannot decide. Each of them is fixed by the scenario and the event history: once the schema states which sources outrank which and who hears what, the correct operation at every step is a function of the state and the event. Belief revision (Alchourrón et al., 1985; Katsuno & Mendelzon, 1991) and dynamic epistemic logic (Baltag et al., 1998) supply the operators for single updates of this kind. What has been missing is a router that composes them event by event into the three fields, with a proof that the result is well defined, and an instrument that checks a memory system against it.

This paper supplies both. Given a scenario schema (entities, slots, a reliability order over sources, event templates and visibility), a small router of belief-change operators, run over observation-partition product updates, unfolds a gold ledger for every step (Figure 1). We call the router an *executable gold semantics*, a pure function $\delta^*\colon (\text{state}, \text{event}) \mapsto (\text{operation}, \text{state}', \text{obligations})$. The compiler is well defined and polynomial per event, and every compiled trace satisfies Hoare-style contracts stated over serialized states only, checked by a module independent of the routing rules (Theorem 1). Six axioms from belief revision, evidence ordering and dynamic epistemic logic characterize the router, and dropping any one yields a counterexample operator that names a recognizable design flaw (Proposition 1). Four complexity results say why the fragment stops where it does (Section 3). Because the same function generates and judges, and defines the supervision signal and the guard, we treat it the way software engineering treats an executable specification, by differential testing against the reference implementation (McKeeman, 1998); adapter lemmas turn each idealization behind these uses into a checkable protocol (two-rate channel calibration, a gate-purity bound for rendered text, and decorrelated rendering).

We instantiate the semantics in MEMLEDGER, 600 development and 300 confirmatory sessions in two languages with 40,500 contract-checked transitions and 400 counterfactual pairs, and report four findings under pre-registered decision rules. On 1,200 seeded contract violations, the views that current benchmarks use see few of them: final-value questions under one percent, and a language model reading a lossless full record under ten percent, while questions designed per contract recover three quarters. Inside a real framework, changing the disposition field alone moves answers about the past from 2 to 33 correct of 36. Across three deployed frameworks, history, provenance, visibility and rollback are answered far less often than current values, and language-model extraction of operations falls short of the bar for automatic audit. In a forked-probing study, probing rewrites memory rarely, measurably only with probe-free controls. Two author annotators working without access to the router agreed with the compiled labels on 1,199 and 1,177 of 1,200 items.

Taken together, the results say that the premise behind outcome-only evaluation is a fact about hand labelling of open dialogue: inside a fixed schema, gold compiles and the operation level is measurable, and the fields that answer-level views cannot see are the ones where most violations live. The construction depends on the schema rather than on our domains, so any scenario written inside the fragment of Section 2 admits the same compiler.

2 THE FRAGMENT AND THE EXECUTABLE SEMANTICS

This section fixes the objects, the fragment and the router.

Objects and fragment. A gold system is $G = (S, s_0, E, \mathcal{O}, \delta^*, \ell)$ with a finite state space S generated by a schema, a deterministic transition δ^* and a labelling $\ell\colon S \times E \to \mathcal{O}$. A state $s = (w, \{B_a\}_{a\in A}, \{B_a^b\}_{a,b\in A}, \mathrm{Trig}, \mathrm{Ledger})$ has a slot array w (the world layer), per-agent literal belief bases B_a with support graphs, second-order bases B_a^b (what a takes b to believe; nesting at most two), guarded-trigger states and the record. The system under audit is $(F_\theta\colon M \times E \to M,\ D\colon M \to S)$ with a disclosure head D, the identity on the exported store for explicit-memory systems; probes $\Pi = \{\pi\colon S \to \Delta(\mathrm{Ans})\}$ are realized as text questions. Every theorem in Section 3 is stated for the fragment $\mathfrak{F}$: finite propositional vocabulary; observation-partition events whose cells are either the actual event or carry no information about the written slot (Appendix A); sources ordered by a strict partial order of reliability, with distinct equally ranked sources rejected at schema compile time; triggers as guarded automata; finite slot domains, with numeric comparison confined to guards and probes. We claim nothing outside $\mathfrak{F}$.

Operations and dispositions. $\mathcal{O}$ has thirteen operations, reported in seven families: CREATE; the update family UPDATE, REFINE and SHIFT (with DRIFT a rule-level subtype of UPDATE); REVISE; DELETE; CLARIFY; the trigger family ARM, FIRE, DISMISS; and the no-op family NO-OP, RETAIN, DROP. CLARIFY is first class: when evidence cannot decide, the correct transition leaves the state unchanged, records both candidates and emits a question obligation. Each displaced belief receives a *disposition*, true-until-t (true, then the world changed) or was-wrong@t (wrong all along); ledger entries additionally carry a status (kept, deleted, dissent, ambiguous). The two dispositions are the Katsuno–Mendelzon distinction between update and revision (Katsuno & Mendelzon, 1992) and the valid-time versus transaction-time distinction of bitemporal databases (Snodgrass & Ahn, 1985); in our fragment they are the faithful projection of the event type the schema declares, and their value lies in the questions they license about the past (Section 5.2).

Observation partitions. An observation-partition event is (actual event, alternative set Alt, a partition of Alt per agent, pre, post): each agent observes the cell of its own partition containing the actual event, which is product update with factual postconditions (Baltag et al., 1998; van Benthem et al., 2006; van Ditmarsch & Kooi, 2008) restricted to partitions, the pattern of semi-private announcements. Scope, visibility and audience thereby get a model-theoretic definition, and *cross-audience contamination is updating with too coarse a partition*. Under the cell condition, the product update reduces to per-agent conditioning within literal bases at $O(n\,|A|\,|\mathrm{Alt}|)$ per event, times $|A|$ for the second-order layer (Lemma 1).

Contracts checked independently of the rules. Each operation carries a Hoare-style contract $\{P_o\}\,o\,\{Q_o\}$ whose postconditions come from the postulate systems: Q_{REVISE} encodes success, inclusion, consistency and relevance of base revision (Hansson, 1999) with the was-wrong disposition; Q_{UPDATE} the pointwise KM update semantics (Katsuno & Mendelzon, 1991) with the true-until interval; every Q's visibility component the observation partition; Q_{DELETE} minimality and a trace; Q_{CLARIFY} abstention. The checker is a separate module that reads only serialized pre- and post-states, the event and the emitted entry, and never chooses a route.

The router (rules in order, first match wins).

- **R1** Ontic write $x \leftarrow v$: UPDATE; close the old interval as true-until-t, open a new one.
- **R2** Epistemic input consistent with the target base: CREATE if the slot is empty; on the same value, RETAIN from the same source, REFINE from a strictly stronger source, else NO-OP; add the support edge.
- **R3** Epistemic conflict ($\neg\ell$ in the base) from a source that strictly outranks the current support: REVISE; swap $\neg\ell$ for ℓ with was-wrong@t, then sweep dependents whose support has died.
- **R4** Epistemic conflict from a different, non-dominating source: NO-OP(dissent); the base is unchanged, the ledger records the dissent.
- **R5** Same-source conflict: with an explicit correction marker, as R3; otherwise UPDATE(drift) on a preference or commitment slot and CLARIFY on a fact slot.

R6 ANNOUNCE(audience): apply R2–R5 to the audience only.

R7 Trigger ARM/FIRE/DISMISS: product with the guard automaton; FIRE needs armed $\wedge$ guard(w), otherwise NO-OP(blocked).

R8 Deletion (DELETE if the slot is populated, ledgered; DROP if absent), support SHIFT to a destination slot, and NO-OP(null) for anything else with zero innovation.

The correction marker of R5 is a lexical cue in the schema's templates; on real logs it has to be extracted (Section 5.3).

Table 4 (Appendix G) compiles a six-event scenario in which a dominated third-party claim is recorded as dissent and a private self-correction revises a deadline for two of three agents; at both steps the ledger carries information no final answer does.

3 Theory: Compilability, Characterization and the Cost Map

This section states what the router guarantees: compilability, an axiomatic characterization, and four complexity bounds.

Theorem 1 (Core compilability)**.** *Within $\mathfrak{F}$: (a) the router assigns to every (s, e) a unique (o, s'), so δ^* and ℓ are total; (b) one event costs $O(n\,|A|^2\,|\mathrm{Alt}|)$; (c) every compiled trace satisfies the operation contracts of Section 2; (d) the revision operator of R3 satisfies Hansson's base-revision postulates and its support sweep maintains the invariant that every entry has a live support chain, while the update operator of R1 satisfies the literal-interval form of the KM postulates U1–U4 and U8.*

In words: every event has exactly one gold successor, computing it is cheap, the successor always meets its contract, and the two belief-change operators inside the router are the textbook ones. *Proof idea.* The rule antecedents are decidable predicates that partition the event space, ties being rejected at schema compile time; the cost is one traversal of the support graph times the propagation of Lemma 1; (c) is a mechanical check of each rule against its contract; in a literal base the postulates of (d) hold in a simple form, and the content of the theorem is that the full router, with its sweep and interval bookkeeping, stays inside them (Appendix B).

An axiomatic characterization. Naturally, the objection to any rule system is that it encodes its authors' grading preferences. We answer by stating six axioms on transition operators over $\mathfrak{F}$ and showing that they pin the router down: **A1 Success and well-formedness** (accepted epistemic input enters the base, ontic postconditions hold, the successor is consistent, support-closed and ledger-complete); **A2 Minimal change** (no proper subset of the change set achieves the success, consistency and support-closure clauses of A1); **A3 Source monotonicity** (epistemic input is accepted iff its source strictly outranks the support of every conflicting entry, a marked self-correction ranking above the same source's earlier assertion; otherwise nothing changes and the dissent is recorded; same-value input strengthens support iff its source strictly outranks the current one); **A4 Type fidelity** (ontic events act pointwise on worlds, epistemic events by closest revision; unmarked same-source change on preference and commitment slots is typed as ontic drift); **A5 Partition fidelity** (agent a changes only if the event carries information in a's cell); **A6 Abstention** (when the comparison of A3 is undefined, nothing changes and a clarification obligation is produced).

Proposition 1 (Characterization)**.** *On $\mathfrak{F}$, the state transition of any operator satisfying A1–A6 coincides with that of the router of Section 2, and the axioms are independent: removing any one admits an operator that satisfies the other five and differs from the router.*

Put differently, any operator that accepts what it should, changes as little as it can, respects the source order, keeps changes apart from corrections, respects who heard what, and asks when it cannot decide, is the router. The characterization is relative to the schema's declared parameters (source order, marker convention, drift typing): it makes the router the unique operator consistent with those declarations, and a convention such as recording a restated clock value as an update rather than a retention (Section 5.5) is decided by the schema, not by the axioms. *Proof idea.* Case on event type: a dominated conflict has ℓ forced in and $\neg\ell$ out by A1, the removal set pinned to $\{\neg\ell\}$ by A2, the sweep set by the support invariant and the disposition by A4, which is R3; the other cases follow the same pattern, and Appendix C lists the six counterexample operators. Two schema declarations enter the axioms as parameters (the marker convention in A3 and the drift typing in A4), and the operation

label beyond the state change (CREATE against REFINE, RETAIN against DROP) is a ledger convention outside the axioms. A3 and A6 are instances of non-prioritized revision, where a credibility or trust preprocessor may reject an input (Makinson, 1997; Hansson et al., 2001; Booth & Hunter, 2018); A1–A2 come from AGM (Alchourrón et al., 1985), A4 from the update/revision distinction (Katsuno & Mendelzon, 1992), A5 from dynamic epistemic logic (van Ditmarsch et al., 2007). To reject the router one has to reject one of these, and each rejection is a concrete operator with a recognizable behaviour.

Counterfactual minimal pairs. The contrast structure of the rules yields *counterfactual minimal pairs* for free, (contrast sets (Gardner et al., 2020)): two streams with the same surface prefix, one flipped semantic bit and different gold transitions (R3/R4 flips source dominance, R1/R3 ontic against epistemic, R6 the audience, R2/R8 relevance). The R1/R3 pair is the sharpest: "my mistake, it is June 8" is epistemic, routes to REVISE with was-wrong, and the gold answer to "what was the real deadline on June 5?" is June 8; "the client moved it to June 8" is ontic, routes to UPDATE with true-until, and the gold answer is June 15. The texts differ by a clause and the final state is identical, so answer-level evaluation gives full marks to a system that conflates them while the ledger gives it zero on the pair.

The cost map: four walls. Four results say why the fragment stops where it does (Appendix D); they show that specific extensions are expensive, not that $\mathfrak{F}$ is maximal. **W1 (nesting):** a nesting bound of k costs $O(n\,|A|^k\,|\mathrm{Alt}|)$ per event. **W2 (numbers):** with difference constraints in the base, the canonical minimum-cardinality repair a deterministic gold needs is NP-hard (minimum feedback arc set (Karp, 1972)), so $\mathfrak{F}$ confines numeric comparison to guards and probes. **W3 (non-partition):** general event models lose per-agent factorization (van Ditmarsch et al., 2007). **W4 (infinite domains):** unbounded arithmetic makes consistency undecidable (Matiyasevich, 1993) or doubly exponential (Fischer & Rabin, 1974).

4 One Pure Function, Four Uses

The method is one function, four consumers and three adapters (entry: rendering, extraction, a verification gate; exit: forked probing, channel calibration; evidence: the ledger's deployable form).

Generate. Forward-unfold a schema into gold streams, counterfactual pairs and a five-type probe battery per slot: current value, history, provenance, visibility and rollback (the interpretation of the displaced value). A language model renders event parameters into text and never produces semantics; two adapter lemmas make rendering checkable (Appendix F): Lemma 5 sets the mutual information between surface shortcuts and labels to zero by rendering design (the corpora here predate this control; Section 5.5 measures the residual), and Lemma 6 bounds the impurity of gate-retained renderings by $r_0\hat{m}/(1-f)$.

Judge. The primary instrument is the *compiled judge*: recompile δ^* on the event stream and compare with the observed ledger and disclosed state entry by entry. Where the observed ledger is a corrupted copy of the gold ledger it is a differential test and succeeds by construction (Section 5.1), and where the ledger must be read out of a deployed system its bottleneck is the extraction step (Section 5.3). For systems that expose no ledger, probes read the state; a qualification gate checks by partition refinement that Π distinguishes G (Vafa et al., 2024). It turns out that current-value probes alone fail it, since "believes X from live evidence" and "believes X from withdrawn evidence" answer them identically, the blind spot of static probing (Belinkov, 2022). A probe realized as text is itself a message, so the default protocol *forks*: the probe is asked on a copy of the prefix and the copy is discarded; when a system cannot be forked, the probe becomes an event with a NO-OP contract whose violation is measurable (Section 5.4). Text probes also introduce an answer channel (a system may believe correctly and answer wrongly, or the reverse); Lemma 7 (Appendix F) recovers the belief error rate from the observed error rate given two rates measured in calibration and forced-error states.

Supervise and guard. The ledger is a process label (context, ledger state, event, candidate) $\mapsto$ (operation, contract bits), with compiled gold as positives and the worse side of counterfactual pairs as structural negatives; on real logs the router is a weak teacher whose label error is bounded by

extraction error plus out-of-fragment rate (Appendix F). At run time the guard runs the core each step and shares every line of code with the judge. Both are implemented; their evaluation on deployed systems follows the extraction step that Section 5.3 measures.

5 EXPERIMENTS

Scope. Everything below is measured on sessions compiled from schemas; the labels are correct by construction relative to Section 2, and two blind annotators' agreement with them is reported in Section 5.5. Results carry their corpus: the *development* corpus (600 sessions, 27,000 transitions, compiled with an earlier same-value rule that refined support on any new source) and the *confirmatory* corpus (300 sessions, 13,500 transitions, 400 counterfactual pairs, the semantics of Section 2). Both have seven domains, half English and half Chinese, three tiers, and *one schema skeleton* (four agents, the source order third_party $\prec$ assistant_inference $\prec$ user_direct $\prec$ system_record, two fact slots, a preference and a commitment slot, a clock and a guarded reminder) instantiated with a different vocabulary per domain; sessions average 45 events. A machine qualification suite (replay determinism, route partition, gold separation, corpus isolation) passes in full (Appendix G). Framework runs use Qwen2.5-72B-Instruct (Qwen Team, 2025) on vLLM (Kwon et al., 2023) at temperature 0 with BGE-M3 embeddings (Chen et al., 2024).

5.1 WHICH EVALUATION VIEW SEES WHICH CONTRACT VIOLATION

This experiment asks which of six evaluation views can see a violated contract.

Protocol. Violations are injected by an external mutation specification, a JSON file of twelve classes (Appendix G) that edits serialized ledgers and imports neither the compiler nor the contract predicates; the classes (100 each, on the 300 confirmatory sessions; row labels of Figure 2) cover writes, dispositions, scope, provenance, dissent, triggers, deletion, clarification and temporal order, and every mutated ledger is paired with its unmodified twin. The fault model is record corruption: the injector edits one entry and that step's successor snapshot and does not replay later events, so a corrupted write reaches a final-value question only when no later legal write to the slot follows. We ask what each of six *views* of the same 1,200 pairs can see. The *compiled judge* recompiles δ^* from the events and diffs against the observed ledger; since that ledger is a corrupted copy of the gold ledger, it must see every mutation, so its column is a completeness check, not a detection result. Three deterministic *state projections* answer a question from the observed successor state and flag a case when the answer differs from the twin's: a *final-value* question about the mutated slot at session end (what final-answer benchmarks ask); the *five-type battery* at the mutated step; and, per class, the single *class-specific question* most likely to expose it. These projections know which slot and step to ask about, which favours them, and raise no false alarms by construction; a *belief-snapshot diff* compares per-step belief snapshots without semantics. An *LLM judge* receives the complete observed record (event stream, a lossless encoding of every successor state, the contract list and the rule code; 37k–66k tokens per case, so Llama-3.1-70B-Instruct (Llama Team, AI @ Meta, 2024), temperature 0.2, three seeds, majority vote) and returns one boolean; clean and mutated conditions are judged separately, so its false-alarm rate is measured. Pre-registered rules: compiled recall 100/100 per class; final-value recall at most 50%; two alternative conclusions depending on whether class-specific questions recover the disposition, scope, dissent and provenance families above 70%.

Results. Figure 2 gives the outcome, with source-session-clustered bootstrap intervals in Appendix H. The final-value projection sees 10 of 1,200 violations (0.8%, interval 0.4–1.4): a violation of a memory contract almost never changes the value a final question would read. The five-type battery sees 54.3% (53.4–55.2) and the class-specific question 76.0% (74.7–77.3), so the pre-registered branch that fired is the weaker one: the question types current benchmarks ask are blind, while questions designed for each contract recover much of it. Two classes survive every question: ghost updates (a ledger claiming an update that never happened) and time reorderings (swapped history intervals) corrupt the record of the process, not any value the state exposes. The belief-snapshot diff sees 30.7%. The surprising part is the LLM judge: given the complete record and the rule code, it sees 9.6% (8.6–10.5) with no false alarms and 99.3% vote unanimity; its 115 detections are all wrong-source (84) and wrong-value (31) cases, the two classes whose mutations insert out-of-vocabulary sentinel tokens, and it sees none of the ten structural classes. Complete information did not make the

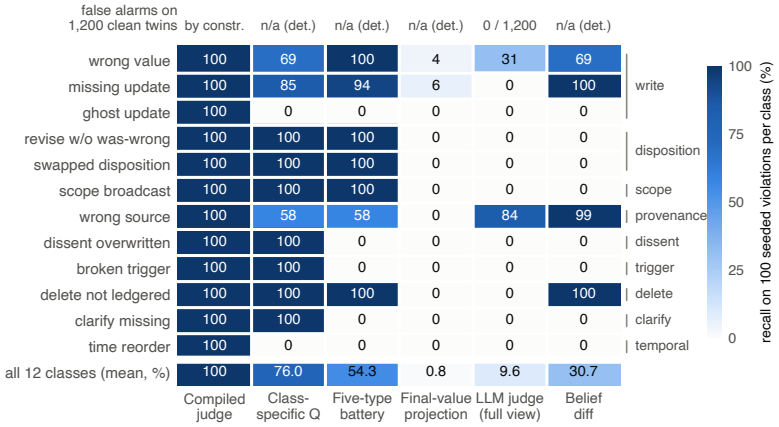

Figure 2: Recall on 1,200 seeded contract violations (twelve classes, 100 each, confirmatory corpus) for six views of the same pairs. The compiled judge's column is by construction; the deterministic projections cannot raise false alarms; the LLM judge (Llama-3.1-70B, complete lossless record, rule code) is the only view whose false-alarm rate is measured, 0 of 1,200.

Table 1: Seeded disposition intervention on native LangMem with Qwen2.5-72B (12 fixtures, 180 native question runs). Only the sentence carrying the disposition of the displaced value differs between arms. Counts are correct answers over questions; the three history questions per fixture are dependent.

Readout	Fault arm	Repaired arm	Fault repeated
History (past value, old record valid, keep or replace)	2 / 36	33 / 36	2 / 36
Controls (current value, unrelated code)	24 / 24	24 / 24	24 / 24

structural contracts legible to this model in this configuration; we do not generalize to other models or prompts (a judge on a compressed view is in Appendix H). Answer-level views are blind where violations live: the structural fields of the ledger are the fields no answer reads.

5.2 Dispositions have downstream consequences

Does a memory framework's behaviour depend on the disposition? We seeded twelve fixtures into a native LangMem (LangChain, 2025) store, each a correct memory whose displaced value was either true-until or was-wrong, and created a *fault* arm by flipping only the sentence carrying that interpretation (current value, old value, other records, identifiers and timestamps unchanged); a *repaired* arm restored it, and a *repeat* arm re-ran the fault. Each arm answered three history questions (the past value, whether the old record had been valid, whether an archive of past dates should keep or replace it) and two controls (the current value, an unrelated reference code), each on a fresh fork with a closed answer vocabulary. History answers went from 2 of 36 correct under the fault to 33 of 36 when repaired (31 improved, none worsened), controls stayed at 24 of 24 in every arm, and the repeat arm reproduced the fault's answers exactly (Table 1); the three remaining errors are keep-or-replace decisions on records that had once been valid. The three history questions share a fixture and are not independent; this is a mechanism check on one contract family, not a prevalence estimate, and it says that the disposition field is load-bearing for a deployed framework's answers about the past.

5.3 Real frameworks: what probes see and why extraction is the bottleneck

Probes. Table 2 runs three frameworks on 50 confirmatory sessions each (2,227 native writes per framework) and asks the five probe types at the terminal state on forked copies, with an explicit *unknown* option never scored as correct. Current values are answered correctly about half the time; visibility and rollback almost never, in every framework. These are development runs on sessions also used for engineering, the denominators keep every failed or truncated session, and two adapter defects

Table 2: Terminal five-type probe accuracy against the compiled reference on 50 confirmatory sessions per framework (one probe of each type per session; denominators are all planned probes, abstentions and missing runs count as wrong; not a ranking). Mem0 ingests through its native `add` path; LangMem runs its memory agent with a BGE-M3 index; Graphiti uses message episodes (8 sessions from a first attempt, 42 from a retry after a GPU fault).

Framework	Completed	Current	History	Provenance	Visibility	Rollback
Mem0 2.0.11	50 / 50	.56	.52	.62	.26	.04
LangMem 0.0.30 (indexed)	47 / 50	.46	.18	.20	.00	.00
Graphiti 0.29.2	8 + 42 / 50	.46	.46	.48	.18	.04

found in a first run are documented in Appendix H: LangMem's answer encoder had overwritten 62 native abstentions with definite answers, 12 of them accidentally correct, and its default store had no retrieval index, so search ignored the query. With both repaired, on the 47 sessions both configurations completed, current-value answers rose from 13 to 23 correct, history fell from 12 to 9, provenance rose from 6 to 10, and visibility and rollback stayed at 0. The numbers say what the probe types expose and how fragile the adapter layer is; they do not rank the frameworks.

Operations. A transition-level audit of a deployed system needs its realized operations, read out of native disclosures. On six development sessions, human annotators observed and adjudicated 249 framework transitions (83 shared steps per framework); on the 50 shared steps they could judge without reservation, the realized operation matched the compiled one for Graphiti 13, LangMem 17 and Mem0 18 times, and annotators could not observe visibility (0 of 234 applicable) or obligations (0 of 87) at all. Unfortunately, automatic extraction falls short of this bar: on the 246 human-known items, of which 80% are NO-OP, Qwen2.5-72B recovered 69.9% and Llama-3.1-70B 61.8%, below the 80.1% of always answering NO-OP, with macro-F1 0.37 and 0.34 over the reference classes; both gave a definite operation on all 54 items the annotators marked unknown, and they reproduced the human ordering of the three frameworks with Kendall τ_b 0.82 and 0.33. So the compiled judge is qualified on ledgers, a real-system contract audit follows the extraction step, and the probe results are read within each framework.

5.4 Probing is an event

If asking a memory system a question can change its memory, every QA-style evaluation measures a system it has just altered. On 100 development sessions we ingested a real prefix into native LangMem (writable agent) and Mem0 (a read-only search-and-answer path), copied the full native state, and ran, for each of three probe phrasings (open question; yes/no question proposing a *wrong* alternative; assertion of the wrong value), an *inline* arm (probe, then continue), a *fork-probe* arm and two probe-free arms (Appendix H, Figure 4). Parent-state isolation held throughout. The inline LangMem probe changed stored content in 0, 3 and 0 of 98 completed sessions, against 0 of 98 in LangMem's own probe-free arm; Mem0 never wrote, while its two probe-free arms differed from each other in 5, 8 and 3 of 100 sessions, so an inline study without a control arm would misattribute nondeterminism of that size. Under the router an asserted wrong value from the same source is a same-source fact conflict (R5), so the correct response records both candidates and asks; of the three LangMem changes, one recorded the conflict (arguably compliant), one paraphrased a memory, and one replaced the stored attribute (plane became train), a last-writer-wins acceptance. We claim no violation rate: grading these needs the extraction step of Section 5.3.

5.5 Human agreement and construct checks

Human agreement. Two annotators from the author team, working without access to the router's code or output and without seeing each other's labels, trained and qualified alike, labelled the 1,200 confirmatory transitions on five fields (operation, current and historical value, disposition, provenance, trigger state). The first agreed with the compiled ledger on all five fields for 1,199 items, the second for 1,177, and the two agreed with each other on 1,178; Cohen's κ is 0.98 on both the operation and the disposition, and visibility is identical on all 1,200 items. Both annotators read a DISMISS event rendered as "the system record *will* cancel the reminder" as a future announcement (NO-OP); we

traced the tense to seven template sentences and corrected them, with no label change. The second annotator labelled a clock date restated by the system record with an unchanged value as RETAIN, where the router treats every ontic write as an UPDATE (17 items), and fired five reminders that had not been re-armed. The figure measures label reliability under the handbook, not the handbook's fit to natural dialogue. The first annotator rated 30 sessions for naturalness at 2.67 of 5, citing field-like phrasing and repetitive event cycles; a two-rater pilot on two revised sessions moved naturalness from 2 to 3. A schema-driven corpus reads like one.

Construct checks. Conditional on the canonical event semantics, a TF-IDF classifier under session-grouped cross-validation predicts the gold operation at AUC 0.49 on the development corpus (worst stratum 0.64–0.68 across three split seeds) and 0.57 on the confirmatory corpus (worst stratum 0.79), against an unconditional full-text control of 0.98; both pass the pre-registered gate of 0.60. The two-rate estimator of Lemma 7 recovers the latent belief error within 0.37 points in the worst of 125 simulated cells, which validates the algebra only. A probe-battery ablation on retrieval and summary baselines (five-type battery 62.7% against current-value probes 18.6%) is in Appendix H.

6 Related Work

Answer-level benchmarks (LongMemEval and V2 (Wu et al., 2025; 2026), LoCoMo (Maharana et al., 2024), MemoryAgentBench (Hu et al., 2026), AMA-Bench (Zhao et al., 2026), Stream-MemBench (Liu et al., 2026), BeliefShift (Myakala et al., 2026), GateMem (Ren et al., 2026)) measure outcomes. HaluMem (Chen et al., 2025) localizes hallucinations to extraction and update operations over LLM-generated, human-verified memory points, without dispositions, scopes, obligations, counterfactual pairs or an axiomatic guarantee. Simulator-compiled gold with rendered text also runs through CLUTRR (Sinha et al., 2019) and TextWorld (Côté et al., 2018); dialogue state tracking (Williams et al., 2016) is the ancestor of the operation taxonomy. Memory-R1 (Yan et al., 2025) and Mem-α (Wang et al., 2025) are driven to outcome rewards by the premise this paper removes; process reward models (Lightman et al., 2024) show what step labels make possible, and Belief-R (Wilie et al., 2024) documents belief revision failures. Graphiti (Rasmussen et al., 2025) keeps bi-temporal validity intervals, an analogue of our dispositions; MemGPT (Packer et al., 2023) and Mem0 (Chhikara et al., 2025) consolidate. Our contracts come from AGM (Alchourrón et al., 1985), KM (Katsuno & Mendelzon, 1991; 1992), base revision (Hansson, 1999), non-prioritized revision (Makinson, 1997; Hansson et al., 2001; Booth & Hunter, 2018) and truth maintenance (Doyle, 1979); iterated revision (Darwiche & Pearl, 1997) is trivial here. Our scope semantics comes from product update with factual change (Baltag et al., 1998; van Benthem et al., 2006; van Ditmarsch & Kooi, 2008), our judge from differential testing (McKeeman, 1998), and our qualification gate from automaton-based world-model evaluation (Vafa et al., 2024).

7 Limitations and Conclusion

Scope. The theorems hold inside $\mathfrak{F}$, which forbids evidence accumulation and equal-rank sources. The corpus is one schema skeleton in seven vocabularies, so all claims are about compiled sessions. Mutation classes are uniformly weighted, so the final-value figure depends on where mutations sit relative to later writes. Real-system results are development runs on three frameworks, read within each framework; on deployed systems the audit covers the state properties probes read (current value, history, provenance), while visibility and obligations need a disclosure interface, and cross-system audit, supervision and guarding all pass through the operation-extraction step that Section 5.3 measures.

Conclusion. Gold labels for memory transitions can be compiled. A small router over observation-partition updates is polynomial, satisfies independently checked contracts, and is characterized by six axioms whose independence proof doubles as a taxonomy of design flaws. On seeded violations, final-value questions see under one percent and a large model reading the full record under ten; one disposition sentence in a real framework moves its answers about the past from 2 to 33 of 36; two blind annotators agreed with the compiled labels on 1,199 and 1,177 of 1,200 items. That per-operation labels are impractical is a fact about hand labelling of open dialogue; inside a schema they compile, and the next step is reading operations out of deployed systems.

AI USE STATEMENT

In this work, we used generative AI tools for rendering the parameters of compiled events into natural-language surface text (the semantics of every event, its gold label and every contract were produced by deterministic code and never by a language model); as the backend of the memory frameworks under evaluation, as the LLM-judge and operation-extraction baselines of Section 5; and for language editing of the manuscript. We have not used generative AI tools for research ideation, for producing or selecting results, or for writing proofs. All AI-assisted text and code was reviewed by the authors; rendered text passed a rule-and-extractor verification gate and the checks of Section 5.5. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

All sessions are synthetic: entities and values come from invented vocabularies with canary identifiers, and no real personal data is processed; the twelve seeded fixtures of Section 5.2 are hand-written fictions. All annotation and rating was carried out by the authors on synthetic text; every answer sheet was completed before any comparison with the compiled labels and was not altered afterwards. The frameworks studied were run locally on pinned versions with local model backends under their published licenses; no commercial product accounts were used. The observer-effect finding is reported so that evaluations can control for it; it provides no means to manipulate deployed systems beyond ordinary questioning.

REPRODUCIBILITY STATEMENT

Appendix G specifies the schema grammar, the compiler loop, the difficulty tiers and the twelve-class mutation specification; Appendices A–F give complete proofs with all assumptions stated; Appendix I reproduces the pre-registered decision rules, the branch that fired and every departure from the plan. The supplement contains the compiler, contract checker, mutation injector, detectors, framework adapters, analysis and statistics code, configuration files with pinned versions (Mem0 2.0.11, LangMem 0.0.30, LangGraph 1.2.7, Graphiti 0.29.2, Qdrant 1.18.0, vLLM 0.6.3.post1, Qwen2.5-72B-Instruct, Llama-3.1-70B-Instruct, BGE-M3), the aggregate result tables as CSV with their source reports as JSON, and all seeds; raw traces and model responses are retained by the authors, and a 20% private split of the benchmark is withheld. Framework runs at temperature 0 are not bit-reproducible: Section 5.4 measures the drift.

REFERENCES

Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2):510–530, 1985.

Alexandru Baltag, Lawrence S. Moss, and Sławomir Solecki. The logic of public announcements, common knowledge, and private suspicions. In *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, pp. 43–56, 1998.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Richard Booth and Aaron Hunter. Trust as a precursor to belief revision. *Journal of Artificial Intelligence Research*, 61:699–722, 2018.

Ding Chen, Simin Niu, Kehang Li, Peng Liu, Xiangping Zheng, Bo Tang, Xinchi Li, Feiyu Xiong, and Zhiyu Li. HaluMem: Evaluating hallucinations in memory systems of agents. *arXiv preprint arXiv:2511.03506*, 2025.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-Embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2318–2335, 2024.

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

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Ruo Yu Tao, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. TextWorld: A learning environment for text-based games. *arXiv preprint arXiv:1806.11532*, 2018.

Adnan Darwiche and Judea Pearl. On the logic of iterated belief revision. *Artificial Intelligence*, 89 (1–2):1–29, 1997.

Jon Doyle. A truth maintenance system. *Artificial Intelligence*, 12(3):231–272, 1979.

Michael J. Fischer and Michael O. Rabin. Super-exponential complexity of Presburger arithmetic. In *Complexity of Computation (Proceedings of the SIAM-AMS Symposium in Applied Mathematics)*, volume 7 of *SIAM-AMS Proceedings*, pp. 27–41. American Mathematical Society, 1974.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1307–1323, 2020.

Kilem Li Gwet. Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology*, 61(1):29–48, 2008.

Sven Ove Hansson. *A Textbook of Belief Dynamics: Theory Change and Database Updating*. Kluwer, 1999.

Sven Ove Hansson, Eduardo Fermé, John Cantwell, and Marcelo Falappa. Credibility limited revision. *Journal of Symbolic Logic*, 66(4):1581–1596, 2001.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in LLM agents via incremental multi-turn interactions. In *International Conference on Learning Representations (ICLR)*, 2026.

Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pp. 85–103. Plenum Press, 1972.

Hirofumi Katsuno and Alberto O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52(3):263–294, 1991.

Hirofumi Katsuno and Alberto O. Mendelzon. On the difference between updating a knowledge base and revising it. In Peter Gärdenfors (ed.), *Belief Revision*, pp. 183–203. Cambridge University Press, 1992.

Hyunwoo Kim, Melanie Sclar, Xuhui Zhou, Ronan Le Bras, Gunhee Kim, Yejin Choi, and Maarten Sap. FANToM: A benchmark for stress-testing machine theory of mind in interactions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 14397–14413, 2023.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th ACM Symposium on Operating Systems Principles (SOSP)*, 2023.

LangChain. LangMem: Long-term memory for LangGraph agents. `https://github.com/langchain-ai/langmem`, version 0.0.30, 2025.

Matthew Le, Y-Lan Boureau, and Maximilian Nickel. Revisiting the evaluation of theory of mind through question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, pp. 5872–5877, 2019.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024.

Guanming Liu, Yuqi Ren, Hansu Gu, Peng Zhang, Weihang Wang, Jiahao Liu, Ning Gu, and Tun Lu. StreamMemBench: Streaming evaluation of agent memory for future-oriented assistance. *arXiv preprint arXiv:2606.14571*, 2026.

Llama Team, AI @ Meta. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

David Makinson. Screened revision. *Theoria*, 63:14–23, 1997.

Yuri V. Matiyasevich. *Hilbert's Tenth Problem*. MIT Press, 1993.

William M. McKeeman. Differential testing for software. *Digital Technical Journal*, 10(1):100–107, 1998.

Praveen Kumar Myakala, Manan Agrawal, and Rahul Manche. BeliefShift: Benchmarking temporal belief consistency and opinion drift in LLM agents. *arXiv preprint arXiv:2603.23848*, 2026.

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

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956*, 2025.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8689–8696, 2020.

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

Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871*, 2018.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, pp. 4506–4515, 2019.

Richard Snodgrass and Ilsoo Ahn. A taxonomy of time in databases. In *Proceedings of the 1985 ACM SIGMOD International Conference on Management of Data*, pp. 236–246, 1985.

Keyon Vafa, Justin Y. Chen, Ashesh Rambachan, Jon Kleinberg, and Sendhil Mullainathan. Evaluating the world model implicit in a generative model. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024.

Johan van Benthem, Jan van Eijck, and Barteld Kooi. Logics of communication and change. *Information and Computation*, 204(11):1620–1662, 2006.

Hans van Ditmarsch and Barteld Kooi. Semantic results for ontic and epistemic change. In *Logic and the Foundations of Game and Decision Theory (LOFT 7)*, volume 3 of *Texts in Logic and Games*, pp. 87–117. Amsterdam University Press, 2008.

Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic*, volume 337 of *Synthese Library*. Springer, 2007.

Yu Wang, Ryuichi Takanobu, Zhiqi Liang, Yuzhen Mao, Yuanzhe Hu, Julian McAuley, and Xiaojian Wu. Mem-α: Learning memory construction via reinforcement learning. *arXiv preprint arXiv:2509.25911*, 2025.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M. Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards AI-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.

Bryan Wilie, Samuel Cahyawijaya, Etsuko Ishii, Junxian He, and Pascale Fung. Belief revision: The adaptability of large language models reasoning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

Jason D. Williams, Antoine Raux, and Matthew Henderson. The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3):4–33, 2016.

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

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

Yufan Wu, Yinghui He, Yilin Jia, Rada Mihalcea, Yulong Chen, and Naihao Deng. Hi-ToM: A benchmark for evaluating higher-order theory of mind reasoning in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10691–10706, 2023.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schütze, Volker Tresp, and Yunpu Ma. Memory-R1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. *arXiv preprint arXiv:2508.19828*, 2025.

Yujie Zhao, Boqin Yuan, Junbo Huang, Haocheng Yuan, Zhongming Yu, Haozhou Xu, Lanxiang Hu, Abhilash Shankarampeta, Zimeng Huang, Wentao Ni, Yuandong Tian, and Jishen Zhao. AMA-Bench: Evaluating long-horizon memory for agentic applications. *arXiv preprint arXiv:2602.22769*, 2026.