

BELIEFMEM: Belief-Revising Memory for Conversational LLM Agents

Felix Meng
XTrace
San Francisco
felixmeng@xtrace.ai

Junda Wu
UC San Diego
La Jolla
juw069@ucsd.edu

Julian McAuley
UC San Diego
La Jolla
jmcauley@ucsd.edu

Abstract

Conversational LLM agents accumulate persistent state across sessions, but existing memory systems treat updates as append-or-replace heuristics, leaving contradictory beliefs to coexist in the store and forcing the answer model to reconcile them at query time. We argue this is the wrong abstraction: conversation is a stream of *belief revisions*, and the right formal lens is the Alchourrón–Gärdenfors–Makinson (AGM) framework, adapted to Hansson belief bases. We present BELIEFMEM, which operationalizes the AGM operator set as the write semantics of a history-preserving *fact* belief base. A uniform LLM-driven resolver classifies every extracted claim into one of four actions (EXP, REV, DUP, CON) and runs at two timings: per round against the recent session, and across sessions over a similarity graph. The store never deletes; superseded beliefs retain predecessor pointers, and a context-selection step decides per query whether to surface the head, the full chain, or neither. On LoCoMo (1,540 questions) BELIEFMEM reaches **85.5%** over Mem0-graph) and **92.3%** with a stronger backbone. On BEAM (400 questions) it reaches **54.5%** (+21.8 over the LIGHT mean).

1 Introduction

Long-horizon conversational agents, personal assistants, coding copilots, customer-support agents, accumulate persistent state about their users across sessions. Most existing memory systems treat this state as a database: extract facts from dialogue, store with vector indices, retrieve at query time. The dominant designs include retrieval over raw transcripts (Lewis et al., 2020), summarization-based memory (Hu et al., 2023), and structured per-fact stores with add–update–delete semantics (Packer et al., 2023; Chhikara et al., 2025; Xu et al., 2026).

This database framing assumes memory updates are largely additive. Conversation is not. Users

correct prior facts (“we moved to SF in March”), *retract* claims with no replacement (“I don’t have a dog anymore”), *narrow* the scope of existing preferences (“for systems work I prefer Rust”), and *refine* temporally-bounded states (“on vacation until Friday”). Treating each as a fresh additive write produces three failure modes that compound over time: contradictions accumulate (both “lives in NYC” and “moved to SF” coexist with equal status); information is destroyed (a scope-narrowing claim gets logged as a supersession of the general preference, erasing it); and retractions do not propagate (a denied claim persists with equal status to all other facts, so downstream readers cannot tell which to trust). At the heart of these failures is a structural limitation: existing systems reduce every fact relationship to a binary write-time decision, supersede or keep both, which works for the obvious cases and breaks for everything else.

We argue conversational memory is a *belief revision* problem, not a database problem. Forty years ago, Alchourrón et al. (1985) formalized the dynamics of rational belief change with three operators, *expansion* ($K + \varphi$), *revision* ($K * \varphi$), and *contraction* ($K - \varphi$), characterized by postulates (Closure, Success, Inclusion, Vacuity, Consistency, Extensionality). Hansson (1999) adapted AGM to *belief bases* (explicit fact lists, no logical closure), which matches what real systems actually store and rejects the controversial Recovery postulate. AGM has been applied to databases, ontology maintenance, and multi-agent reasoning (Fermé and Hansson, 2011), but, to our knowledge, it has not been applied to LLM conversational memory.

To address these limitations, we propose BELIEFMEM, a memory system for conversational AI that operationalizes AGM as its core write semantics. Figure 1 illustrates the framework. Every newly extracted claim is a candidate for one of three actions: silent expansion (no conflict with existing beliefs), revision (replaces an existing be-

lief, marked with a lineage pointer), or contraction (the user has explicitly disavowed a prior belief). The choice is made by an LLM resolver that sees both the new claim and the relevant existing beliefs. The same revision step runs at two timings, *inline* against recent session context after each turn, and *across sessions* at flush over a semantic-similarity graph, and an explicit contraction channel handles AGM disavowals. Following Hansson, the store never deletes: superseded and retracted facts retain lineage pointers traversable to the chain root. At retrieval time, the lineage accompanies each fact, and a dedicated context-selection LLM decides per query whether the answer model needs just the current belief or the full revision history. We summarize our contributions as follows:

- We frame conversational memory as a belief revision problem rather than a database problem. The *binary-trap* diagnosis explains a class of failure modes in current systems (§1, §4).
- We adapt AGM belief revision, in its Hansson belief-base form, to conversational memory: corrections, restatements, and retractions become distinct, first-class belief-change operations over a history-preserving fact base rather than append/replace heuristics.
- We introduce a uniform LLM-driven resolver that assigns each extracted claim to one of four AGM-derived actions (EXP/REV/DUP/CON) over a Hansson belief base, complemented by a history-preserving store with query-time lineage selection that decouples preservation policy from token-cost economics (§4.2–§4.5).
- We validate BELIEFMEM empirically. Under the GPT-4o-mini backbone shared by published baselines, BELIEFMEM reaches 85.5% on Lo-CoMo (+17.1 over Mem0-graph, the strongest prior system); with a stronger backbone we reach 92.3%. On BEAM-100K, BELIEFMEM reaches 54.5% (+21.8 over LIGHT’s four-LLM mean), with the decisive gap on contradiction-resolution (56.2% vs. 1.8–3.7%) (§5).

2 Related Work

LLM memory systems. Memory architectures for conversational LLM agents fall into three families. *Retrieval over raw conversation* stores transcripts in a vector database and retrieves chunks at query time (Lewis et al., 2020); this scales but accumulates redundancy and lacks any notion of

belief consistency. *Summarization-based memory* compresses prior turns into LLM-generated summaries (Hu et al., 2023); this controls context length but is lossy and does not handle corrections explicitly. *Structured fact stores*—the family BELIEFMEM belongs to—extract discrete claims from dialogue and store them with metadata for later retrieval. We focus the remainder of this section on the structured family, where the belief-revision question is most acute.

Belief revision. AGM belief revision (Alchourrón et al., 1985; Gärdenfors, 1988) is the dominant formal theory of rational belief change; Hansson (1991, 1999) extended AGM to *belief bases*—explicit fact lists without logical closure—and rejected the controversial Recovery postulate. Constructive realizations include partial-meet contraction (Alchourrón et al., 1985) and the systems-of-spheres semantics (Grove, 1988). Applications appear in database schema evolution (Flouris et al., 2006), ontology maintenance (Ribeiro and Wassermann, 2009), and multi-agent belief merging. Despite four decades of theoretical development (Fermé and Hansson, 2011), applications to LLM conversational memory have been absent. Recent benchmarks probe belief revision in LLMs directly—e.g., ReviseQA (Helwe et al., 2025) evaluates multi-turn belief revision in logical reasoning—but target model reasoning behavior rather than a persistent conversational-memory store. Concurrent independent work by Park (2026) (Kumiho, March 2026) proves AGM postulate compliance ($K*2$ – $K*6$ plus Hansson’s Relevance and Core-Retainment) over a property-graph memory architecture with a structured MCP write API and an asynchronous “Dream State” consolidation pipeline. BELIEFMEM differs in three respects: (i) it operates directly over LLM-extracted dialogue (no structured-input requirement at the write API), (ii) it performs revision *inline* during extraction rather than asynchronously, and (iii) it targets vector-KB storage rather than property graphs, which removes the need for a dual-store working/long-term split.

3 Preliminaries: AGM Belief Revision

We briefly recall the AGM framework (Alchourrón et al., 1985; Gärdenfors, 1988; Hansson, 1999); readers familiar with it can skip to §4.

Belief sets. A *belief set* K is a set of sentences in some background language $\mathcal{L}$, closed under logical

consequence: $K = \text{Cn}(K)$. We adapt this informally below to a set of natural-language claims emitted by an LLM extractor; the consequences of relaxing deductive closure are discussed in Appendix B.

The three operators. Given a belief set K and a sentence φ , AGM defines: *expansion* $K + \varphi$, which adds φ to K without consistency checks; *revision* $K * \varphi$, which adds φ while preserving consistency, potentially removing existing beliefs that conflict with φ ; and *contraction* $K - \varphi$, which removes φ from K together with enough other beliefs that φ is no longer derivable from the remainder.

The AGM postulates. A revision operator $*$ is rational in the AGM sense iff for all K, φ, ψ : (K1) Closure: $K * \varphi$ is a belief set; (K2) Success: $\varphi \in K * \varphi$; (K3) Inclusion: $K * \varphi \subseteq K + \varphi$; (K4) Vacuity: if $\neg\varphi \notin K$, then $K * \varphi = K + \varphi$; (K5) Consistency: $K * \varphi$ is consistent if φ is consistent; (K6) Extensionality: if $\varphi \equiv \psi$, then $K * \varphi = K * \psi$. For contraction, the analogous postulates include the controversial *Recovery* postulate: $K \subseteq (K - \varphi) + \varphi$.

Belief bases (Hansson). A practical objection to AGM is that real systems do not store deductively-closed belief sets: they store *explicit lists* of beliefs, with closure (if needed) computed on demand. Hansson (1991, 1999) adapted AGM to this setting under the name *belief bases* (B in place of K , with $K = \text{Cn}(B)$). Three consequences are material here. First, the Recovery postulate is rejected on philosophical grounds and is operationally undesirable for conversational memory: a user who retracts “I live in Paris” generally does not want re-asserting it later to silently restore unrelated Paris-derived facts. Second, base contraction admits constructive realizations (kernel contraction, partial-meet over the base) that are computationally tractable on natural-language fact lists. Third, base revision tolerates redundancy, which matches the reality of LLM-extracted memory. BELIEFMEM is structurally a belief-base system: facts are explicit, deduplication is opportunistic rather than mandatory, and Recovery is not implemented.

To make this derivation explicit rather than implicit, Appendix B (Table 6) maps each of BELIEFMEM’s core design decisions to the AGM/Hansson insight it follows from and the failure that insight helps avoid.

4 Belief-Revising Memory

BELIEFMEM treats each dialogue turn as a sequence of belief-change operations over a Hansson *fact base* $B = B^F$, with each extracted fact linked to the session episode in which it was stated. A uniform LLM-driven resolver assigns each newly extracted claim to one of four AGM-derived actions, runs at two timings (inline per round, cluster across sessions), and never deletes: superseded and retracted beliefs retain predecessor pointers traversable to the chain root. We describe the formal setting (§4.1), the operator mapping (§4.2), the two resolvers (§4.3–4.4), and the history-preserving belief base with query-time lineage selection (§4.5). A per-postulate fidelity analysis is detailed in Appendix B.

4.1 Setting and Belief Base

A user and an LLM agent exchange a stream of turns $(u_1, a_1, u_2, a_2, \dots)$. After each batch of turns, an extractor LLM emits a set of *claims* $\Phi_t = \{\varphi_1, \dots, \varphi_n\}$ summarizing what was said. Following the Hansson belief-base discipline (§3), we treat the agent’s persistent memory as an explicit fact list B_t rather than a deductively-closed belief set, and each batch as a sequence of belief-change operations on B_t .

Atomic claims. A single utterance often bundles several distinct assertions—“I live in NYC and work as an engineer” carries two—so the unit of revision is not the utterance but the *atomic claim*: the smallest proposition the AGM operators apply to. BELIEFMEM decomposes each extracted utterance into atomic fact claims and revises them independently, so that a correction touches only the claim it contradicts and leaves unrelated claims active.

Facts and episodes. The revision machinery of §4.2 operates over facts (B^F) at the granularity of an atomic claim: each fact carries a predecessor pointer and a lifecycle status, and contraction $K -$ realizes explicit retirement. Each fact is additionally linked to the *episode* (B^E)—an immutable per-session summary—in which it was stated, which supplies temporal context to the cross-session resolver (§4.4) without itself being revised.

Payoff. An answer LLM that sees the *head* of a fact’s revision chain operates against a consistent active-set view—the current value of each claim—instead of a pile of mutually contradictory statements. The remainder of this section develops the

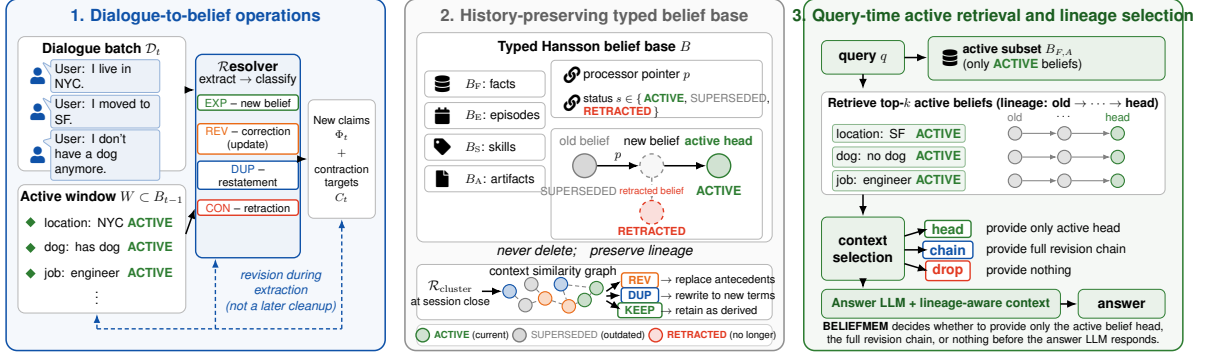


Figure 1: BELIEFMEM treats conversational memory as belief revision over a history-preserving belief base rather than as additive database storage. During ingestion, the inline resolver (R_{inline}) jointly extracts claims from each dialogue batch and classifies them into AGM-derived actions—EXP, REV, DUP, and CON—using the recent active-memory window. The Hansson belief base stores facts with predecessor pointers and lifecycle statuses, together with immutable per-session episodes, preserving superseded and retracted beliefs instead of deleting them. At session close, (R_{cluster}) resolves cross-session conflicts through semantic-similarity clustering. At query time, retrieval operates over the active subset, attaches lineage chains, and a context selector (S) decides whether the answer model should receive only the current belief head, the full revision chain, or no memory context.

operators over B^F .

Running example. Suppose the user says “I live in Boston and work at Company A,” decomposed into two atomic claims φ_1 (lives in Boston) and φ_2 (works at Company A). A later turn “I moved to Seattle” triggers REV on the residence claim *only*, adding φ_3 (lives in Seattle) with predecessor $p(\varphi_3) = \varphi_1$. The active view becomes $\{\varphi_3, \varphi_2\}$ —only the residence claim is superseded—while the lineage $\varphi_1 \rightarrow \varphi_3$ is preserved in storage and the employment claim φ_2 remains an independent active fact. This is the concrete payoff of the atomic-unit discipline: a correction supersedes exactly the conflicting claim and nothing else, so retrieval starts from a uniform active view and additionally expose lineage for history-aware queries.

4.2 Operator Mapping

Each fact $\varphi \in B^F$ carries two metadata fields: a predecessor pointer $p(\varphi) \in B^F \cup \{\perp\}$ (with $\perp$ for chain roots) and a status $st(\varphi) \in \{\text{ACTIVE}, \text{SUPERSEDED}, \text{RETRACTED}\}$. The predecessor pointer is immutable once set; the status is the only mutable state on a fact. We require predecessor pointers to be *irreflexive* ($p(\varphi) \neq \varphi$: no fact is its own predecessor), *order-respecting* (a new fact may point only to a strictly pre-existing belief), and therefore *acyclic*, so the lineage relation forms a forest rooted at $\perp$. We write $B^{F,A} = \{\varphi \in B^F : st(\varphi) = \text{ACTIVE}\}$ for the active subset,

and $B_t^{F,A}$ for its value after turn t . A *resolver*

$$\mathcal{R} : \mathcal{L} \times \mathcal{P}(B_t^{F,A}) \rightarrow \mathcal{A},$$

$\mathcal{A} = \{\text{EXP}, \text{REV}, \text{DUP}, \text{CON}\}$, maps a candidate claim in $\mathcal{L}$ (the natural-language claim space) and a window of active beliefs to one of four actions:

- **EXP(φ): expansion.** Applied as $B^F \cup \{\varphi\}$. Realizes $K + \varphi$.
- **REV(φ, ψ): revision.** φ replaces ψ , with $p(\varphi) := \psi$ and $st(\psi) := \text{SUPERSEDED}$. Realizes $K * \varphi$.
- **DUP(φ, ψ): restatement.** φ is already entailed by ψ ; the base is unchanged. This is the idempotent case of expansion— $K + \varphi = K$ whenever $K \vdash \varphi$ —so DUP (*duplicate*) and EXP (*expansion*) coincide at the belief-set level but differ operationally: EXP inserts a new record whereas DUP skips insertion to avoid a redundant stored claim. Realizes $K + \varphi$ when $\varphi \in B^F$.
- **CON(ψ): contraction.** An existing ψ is withdrawn without replacement via $st(\psi) := \text{RETRACTED}$. Realizes $K - \psi$.

Two implementations of $\mathcal{R}$ run at different timings: $\mathcal{R}_{\text{inline}}$ (§4.3) folds revision into per-round extraction; $\mathcal{R}_{\text{cluster}}$ (§4.4) resolves cross-session conflicts at session boundary. Both write only p and st , never delete a fact.

4.3 Per-Round Resolver

The per-round resolver $\mathcal{R}_{\text{inline}}$ implements $\mathcal{R}$ as a single LLM call that jointly performs extraction

Algorithm 1 Per-round resolver $\mathcal{R}_{\text{inline}}$.

Require: dialogue batch D_t ; active window $W \subseteq B_t^{\text{F},\text{A}}$
Ensure: updated belief base B_{t+1}^{F}

- 1: $(\Phi_t, C_t) \leftarrow \text{LLMEXTRACTANDRESOLVE}(D_t, W) \triangleright$
single LLM call
- 2: $T \leftarrow \emptyset \triangleright$ claimed targets (D3)
- 3: **for** each $\psi \in C_t$ **do** $\triangleright$ contractions first
- 4: **if** $\psi \notin T$ **then**
- 5: $st(\psi) \leftarrow \text{RETRACTED}$; $T \leftarrow T \cup \{\psi\}$
- 6: **end if**
- 7: **end for**
- 8: **for** each $\varphi \in \Phi_t$ with action $a \in \mathcal{A}$ and (optional) target ψ **do**
- 9: **if** $a = \text{DUP}$ **then continue** $\triangleright \varphi$ already entailed by B^{F}
- 10: **else if** $a = \text{EXP}$ **then**
- 11: $B^{\text{F}} \leftarrow B^{\text{F}} \cup \{\varphi\}$
- 12: **else if** $a = \text{REV}$ **and** $\psi \notin T$ **then**
- 13: $p(\varphi) \leftarrow \psi$; $st(\psi) \leftarrow \text{SUPERSEDED}$
- 14: $B^{\text{F}} \leftarrow B^{\text{F}} \cup \{\varphi\}$; $T \leftarrow T \cup \{\psi\}$
- 15: **end if**
- 16: **end for**
- 17: **return** B^{F}

and classification. Given a dialogue batch D_t and an active window $W \subseteq B_t^{\text{F},\text{A}}$, it returns a set of extracted claims Φ_t —each annotated with an action in $\{\text{EXP}, \text{REV}, \text{DUP}\}$ and (when applicable) a predecessor $\psi \in W$ —together with a separate set of contraction targets $C \subseteq W$. Folding revision into extraction makes $\mathcal{R}_{\text{inline}}$ marginally free: the extraction call is unavoidable, and revision adds no additional calls. Algorithm 3 gives the full procedure. For $\mathcal{R}_{\text{inline}}$ to behave well we impose three semantic constraints, stated formally in Appendix H.

4.4 Cross-Session Resolver

$\mathcal{R}_{\text{inline}}$ is bounded by what fits in the window W : a claim made in session 3 may conflict with a claim made in session 17 without ever appearing in the same window. At session boundary we therefore run a second resolver $\mathcal{R}_{\text{cluster}}$ over the newly added facts $F_{\text{new}} \subseteq B^{\text{F},\text{A}}$ and their semantic neighbors. We build an undirected similarity graph $G = (V, E)$ over F_{new} augmented with each new fact’s top- k semantic neighbors, with edges admitted when bidirectional similarity $\max(\sigma(a, b), \sigma(b, a))$ exceeds a threshold τ (the bidirectional max makes the edge order-independent under a reranker σ). Each connected component $c \in \mathcal{C}$ with $|c| \geq 2$ is partitioned by an LLM into *groups* labeled REV, DUP, or KEEP; each member is presented with its enclosing episode, event-date, and recording time, allowing the resolver to distinguish “same state recorded twice” from “two events at different times” (the KEEP label also covers scope-

Algorithm 2 Cross-session resolver $\mathcal{R}_{\text{cluster}}$.

Require: new facts $F_{\text{new}} \subseteq B^{\text{F},\text{A}}$; threshold τ ; similarity σ
Ensure: updated belief base B^{F}

- 1: $V \leftarrow F_{\text{new}} \cup \bigcup_{\varphi \in F_{\text{new}}} \text{NN}_k(\varphi)$
- 2: $E \leftarrow \{(u, v) : u, v \in V, u \neq v, \max(\sigma(u, v), \sigma(v, u)) \geq \tau\}$
- 3: $\mathcal{C} \leftarrow \text{CONNECTEDCOMPONENTS}(V, E)$
- 4: **for** each component $c \in \mathcal{C}$ with $|c| \geq 2$ **do**
- 5: $\mathcal{G} \leftarrow \text{LLMRESOLVE}(c)$; $T_c \leftarrow \emptyset$
- 6: **for** each group $g \in \mathcal{G}$ **do**
- 7: **if** g is REV **then**
- 8: $(\psi_0, \dots, \psi_m) \leftarrow \text{SORTBYTIME}(g)$
- 9: **for** $i = 0, \dots, m - 1$ **do**
- 10: **if** $\psi_{i+1} \notin T_c$ **then**
- 11: $p(\psi_{i+1}) \leftarrow \psi_i$; $st(\psi_i) \leftarrow$
SUPERSEDED
- 12: $T_c \leftarrow T_c \cup \{\psi_{i+1}\}$
- 13: **end if**
- 14: **end for**
- 15: **else if** g is DUP **then**
- 16: $w \leftarrow \text{MOSTRECENT}(g)$
- 17: $st(\psi) \leftarrow \text{SUPERSEDED}$ for all $\psi \in g \setminus \{w\}$
- 18: **end if**
- 19: **end for**
- 20: **end for**
- 21: **return** B^{F}

narrowing). A REV group of N members is time-ordered and emits $N - 1$ revision pairs along the chain $\psi_0 \rightarrow \psi_1 \rightarrow \dots \rightarrow \psi_{N-1}$, preserving the single-target invariant D3; a DUP group elects the most recent member as the canonical survivor and supersedes the rest. The full graph construction, N -peer resolution rules, and cross-group collision check are detailed in Algorithm 4.

4.5 History-Preserving Belief Base

Following the Hansson base discipline, no fact is ever deleted. Revisions and contractions are realized as transitions on the status st and the predecessor pointer p , leaving the full B_t^{F} queryable. The pointer p makes the revision history a traversable forest: from any active fact, the chain $\varphi \rightarrow p(\varphi) \rightarrow p(p(\varphi)) \rightarrow \dots$ exposes the user’s belief evolution back to the chain root (a chain of length d is traversable in $O(d)$ time, bounded by the configured depth cap in Appendix ??).

Query-time lineage selection. Surfacing every chain to every query would bloat the answering context window. We therefore route retrieval through a lightweight *context-selection* step $\mathcal{S}$ that takes the query, the retrieved active facts, and their chains, and emits a per-fact decision in $\{\text{head}, \text{chain}, \text{drop}\}$: keep only the current head, surface the full chain, or drop the fact entirely. This decouples what is *preserved* (all of B_t^{F}) from what *reaches the answer model* (a subset chosen

per query). For factual queries (“*what does the user prefer?*”) the head suffices; for history-aware queries (“*why did the user change their mind?*”) the full chain is load-bearing. The cost is one additional LLM call per retrieval, amortized over the returned set. We measure the impact empirically in §5.5, where the results argue for a sharper $\mathcal{S}$ rather than for always-render or never-render.

5 Experiments

5.1 Benchmarks and Protocol

We evaluate on two long-horizon dialogue benchmarks chosen for direct probing of belief revision.

LoCoMo (Maharana et al., 2024). Multi-session synthetic dialogues averaging tens of sessions per persona pair, with five reasoning categories (single-hop, multi-hop, temporal, open-domain, adversarial). We use the released set of 10 conversations, 1,540 total questions, scored with the LoCoMo LLM-judge protocol.

BEAM (Tavakoli et al., 2025). Long-horizon memory benchmark decomposing each question into atomic “nuggets” with a per-nugget LLM judge, scored across ten ability axes. We evaluate on the released 128K-token scale (20 conversations, 20 probing questions each = 400 questions). The *contradiction-resolution* and *knowledge-update* axes directly probe belief-revision quality.

Models and configuration. All LLM stages ($\mathcal{R}_{\text{inline}}$, $\mathcal{R}_{\text{cluster}}$, context selection $\mathcal{S}$, answer generation) use temperature 0. Embeddings use mxbai-embed-large-v1 (Lee et al., 2024; Li and Li, 2023) truncated to 512 dimensions via Matryoshka representation learning. Reranking uses mxbai-rerank-large-v1. Cluster cosine threshold $\tau = 0.7$ on BEAM ; $\tau = 0.6$ on LoCoMo. Storage backend: pgvector. To preserve comparability and avoid test-set tuning, the retrieval depth top- k is fixed to the value each originating study evaluates its own RAG systems at (rather than swept on the benchmark test sets).

Baselines. We compare against systems spanning the design space of structured fact stores plus a context-window-only upper bound: **No-memory** (answer LLM sees only the question); **Full-context** (the full conversation transcript, truncated only if it exceeds the context window); **Raw RAG** (chunk the transcript, embed, retrieve top- k at query time); **Mem0** (Chhikara et al., 2025) (per-fact LLM call selecting from ADD/UPDATE/DELETE/NOOP); **MemGPT** (Packer et al., 2023) (hierarchical

context with function-call read/write primitives); and **LIGHT** (Tavakoli et al., 2025) (cognitive-architecture method proposed in the BEAM paper itself; episodic + working + scratchpad memory).

5.2 Main Results

BELIEFMEM reaches **92.3%** on LoCoMo (1,421/1,540 questions correct) and **54.5%** on BEAM (400 questions across 20 conversations at the 100K-token scale). Two complementary comparisons isolate the contribution of belief-revising memory.

LoCoMo. The Mem0 paper (Chhikara et al., 2025) reports overall LLM-judge accuracy for seven systems on LoCoMo (GPT-4o-mini backbone): A-MEM (48.4%), OpenAI Memory (52.9%), LangMem (58.1%), Zep (66.0%), Mem0 (66.9%), Mem0-graph (68.4%, strongest published), and full-context (72.9%). BELIEFMEM reaches 92.3%, a **+23.9-point** gain over the strongest published baseline. To isolate the mechanism from the backbone, we additionally run a turn-chunked naive RAG baseline under BELIEFMEM’s own answer model and embedder (top- $k = 10$, 8K-character context); BELIEFMEM gains **+17.8 points** over this backbone-matched control (74.5% $\rightarrow$ 92.3%).

BEAM-100K. The BEAM paper (Tavakoli et al., 2025) reports per-LLM scores at the 100K scale across four backbones. Averaging across the four backbones, the three baselines score: Vanilla full-context 25.0%, RAG 29.5%, and the paper’s proposed LIGHT method 32.7% (range 29.4–35.8%; strongest single backbone is Llama-4-Maverick at 35.8%). BELIEFMEM reaches **54.5%**—**+21.8** over LIGHT’s four-LLM mean and **+18.7** above LIGHT’s strongest single-LLM cell. The decisive gap is on the axis that most directly probes belief revision: **contradiction-resolution**, where BELIEFMEM scores 56.2% against LIGHT’s 1.8–3.7% range across the four BEAM-paper backbones—a **~52-point** margin. These near-zero LIGHT scores are the BEAM authors’ *own* published results across their four backbones under the same released rubric and evaluation codebase—not numbers from our reimplementations—and the same low range holds for the other systems BEAM evaluates at this scale, so it reflects the benchmark’s difficulty rather than a scoring artifact on our side. On knowledge-update BELIEFMEM reaches 38.8%, comfortably within LIGHT’s 30.0–45.0% range; the mechanism is

Table 1: Top-line accuracy on LoCoMo and BEAM. BELIEFMEM rows report the shipped configuration on the public benchmark splits and are annotated with their backbone (gpt-4o-mini vs. gpt-5.4-mini); five-run distributions are given in §5.2. LoCoMo baselines are quoted from Chhikara et al. (2025) Table 2 (GPT-4o-mini backbone, same 1540-question non-adversarial subset); BEAM-100K baselines are quoted from Tavakoli et al. (2025) Table 1, averaged across the four LLM backbones the paper tests (Qwen2.5-32B, Llama-4-Maverick, Gemini-2.0-flash, GPT-4.1-nano). Because the quoted baselines use their own papers’ backbones, the BEAM comparison is cross-backbone and should be read as a reference point, not a causal estimate. Best per-column in **bold**.

System	LoCoMo (%)	BEAM-100K (%)
<i>LoCoMo baselines published in Chhikara et al. (2025) (GPT-4o-mini backbone)</i>		
A-MEM (Xu et al., 2026)	48.4	—
OpenAI Memory	52.9	—
LangMem (Terranova et al.)	58.1	—
Zep (Rasmussen et al., 2025)	66.0	—
Mem0 (Chhikara et al., 2025)	66.9	—
Mem0-graph	68.4	—
Full-context	72.9	—
<i>LoCoMo control (our run, gpt-5.4-mini — our default backbone)</i>		
Raw RAG (turn-chunked, $k = 10$)	74.5	—
<i>BEAM-100K baselines, mean across 4 LLM backbones (Tavakoli et al., 2025)</i>		
Vanilla (full context)	—	25.0
RAG	—	29.5
LIGHT	—	32.7
<i>LIGHT (best single backbone, Llama-4-Maverick)</i>	—	35.8
BELIEFMEM (ours, gpt-4o-mini)	85.5	50.3
Δ vs. Mem0-graph (LoCoMo, backbone-matched)	+17.1	—
Δ vs. LIGHT mean (BEAM-100K)	—	+17.6
BELIEFMEM (ours, gpt-5.4-mini)	92.3	54.5
Δ vs. Mem0-graph (LoCoMo, cross-backbone)	+23.9	—
Δ vs. Raw RAG (LoCoMo, backbone-matched)	+17.8	—
Δ vs. LIGHT mean (BEAM-100K)	—	+21.8
Δ vs. LIGHT best single LLM (Llama-4-Maverick)	—	+18.7

competitive but not dominant on this axis. The published baselines use different LLM backbones than BELIEFMEM; we discuss cross-method comparability in the Limitations section.

Run-to-run variance. The LLM stages run at temperature 0, but LLM-judge scoring and API-side sampling still make results nondeterministic, so we report five independent runs of the shipped configuration. BELIEFMEM scores 91.91 ± 0.21 on LoCoMo (range 91.30–92.47) and 52.94 ± 0.65 on BEAM-100K (range 51.39–54.48), where $\pm$ denotes the standard error of the mean over the five runs. The headline LoCoMo number (92.3%) is the main-paper run; the headline BEAM number (54.5%) is the maximum of the five runs. Even the *lowest* of the five runs preserves the qualitative conclusions: the lowest LoCoMo run still exceeds the backbone-matched Raw-RAG control by +16.8, and the lowest BEAM run still exceeds LIGHT’s strongest published cell by +15.6 (cross-backbone). We report full distributions rather than imply deterministic performance.

5.3 Per-Category Breakdown on LoCoMo

The LoCoMo 4-category decomposition lets us locate the per-category source of the backbone-matched gain over the published Mem0-paper baselines. Table 2 gives BELIEFMEM (gpt-4o-mini) alongside the per-category numbers reported in Chhikara et al. (2025) (all on the same gpt-4o-mini backbone, same 1540-question non-adversarial subset).

BELIEFMEM leads on every category, with the widest gap on *multi-hop* questions (+**33.8** over Mem0-graph)—the category that most directly stresses belief consistency: a multi-hop answer chains across two or more retrieved facts, so a stale or contradicting belief in the chain corrupts the output. The narrower gap on *open-domain* (+12.2) is also informative: open-domain questions can often be answered from a single fact, so the belief-revision machinery contributes less. *Temporal* questions (+14.8) benefit from the lineage chain—the answer model can see the predecessor and disambiguate “*what does the user prefer now?*” from “*what did the user prefer last year?*” The

Table 2: LoCoMo per-category LLM-judge accuracy (%), all systems on gpt-4o-mini. Baseline columns from Chhikara et al. (2025) Table 1; BELIEFMEM from this work. Best per column in **bold**.

System	Single-hop	Multi-hop	Temporal	Open-domain
A-MEM (Xu et al., 2026)	39.8	18.9	49.9	54.1
LangMem (Terranova et al.)	62.2	47.9	23.4	71.1
Zep (Rasmussen et al., 2025)	61.7	41.4	49.3	76.6
OpenAI Memory	63.8	42.9	21.7	62.3
Mem0 (Chhikara et al., 2025)	67.1	51.2	55.5	72.9
Mem0-graph	65.7	47.2	58.1	75.7
BELIEFMEM (ours, gpt-4o-mini)	87.9	81.0	72.9	87.9
Δ vs. <i>Mem0-graph</i>	+22.2	+33.8	+14.8	+12.2

Table 3: BELIEFMEM per-ability scores on BEAM-100K (overall 54.5%, $n = 400$ across 20 conversations). Belief-revision-relevant axes are highlighted.

Ability	Score (%)
Abstention	90.0
Preference-following	73.8
Information extraction	61.7
Contradiction resolution	56.2
Event ordering	54.2
Multi-session reasoning	53.2
Instruction following	41.2
Summarization	40.7
Knowledge update	38.8
Temporal reasoning	35.0

single-hop gain (+22.2) tracks the overall accuracy lift and is most plausibly attributable to higher extraction recall plus the active-set invariant.

5.4 Per-Ability Breakdown on BEAM

The BEAM 10-axis decomposition isolates which abilities benefit from belief-revision machinery versus which depend on other components.

BELIEFMEM’s best abilities (*abstention* 90.0%, *preference-following* 73.8%) come from the active-set invariant: when the answer LLM only sees facts that have not been superseded, it is not distracted by stale preferences and is more willing to abstain when no relevant active fact exists. The two belief-revision-direct axes split. On *contradiction-resolution*, BELIEFMEM reaches 56.2% against LIGHT’s 1.8–3.7% range—a ~ 52 -point margin that is, to our knowledge, the largest gap any system has reported on this axis. The interpretation is straightforward: contradiction resolution requires the active belief set to be free of conflicting claims, which is precisely what $\mathcal{R}_{\text{inline}}$ and $\mathcal{R}_{\text{cluster}}$ enforce. On *knowledge-update*, BELIEFMEM reaches 38.8%, comfortably within LIGHT’s 30.0–45.0% range; the mechanism is competitive but not dominant. The weakest ability is *temporal-reasoning*

(35.0%), which probes a capability orthogonal to the write-time revision mechanism; improving it likely requires explicit episode-ordering metadata and time-anchored retrieval.

5.5 Ablations: Cluster Consolidation and Lineage

We measure the contribution of $\mathcal{R}_{\text{cluster}}$ and of lineage rendering through the context-selection step $\mathcal{S}$ by disabling each in turn while holding $\mathcal{R}_{\text{inline}}$ and the rest of the pipeline fixed. All three configurations use gpt-4o-mini on BEAM-100K (400 questions, 20 conversations) so the comparison is backbone-controlled. *Inline-only* is the floor: only per-round revision fires. *+cluster* adds the cross-session resolver at session boundary. *+cluster+lineage* additionally renders predecessor chains at retrieval time.

Inline carries; cluster and lineage are precise instruments. The dominant driver of BEAM accuracy is $\mathcal{R}_{\text{inline}}$ alone (49.9%), which already beats LIGHT’s four-LLM mean (32.7%) by +17.2 points and produces a 42.5% contradiction-resolution score against LIGHT’s 1.8–3.7% range. $\mathcal{R}_{\text{cluster}}$ and lineage are not sources of additional bulk accuracy: their net contribution on top of inline is +0.4 overall (49.9 $\rightarrow$ 50.3). Each is, however, a precise instrument that targets the axes it was designed for. Table 5 gives the cluster-vs.-inline-only per-axis breakdown: cluster consolidation lifts *information extraction* by +9.6 (cross-session de-duplication cleans the retrieval surface, which is precisely what DUP is designed to do), at a cost of -3.4 on preference-following and -2.5 on knowledge-update where supersession occasionally over-aligns on attribute-update chains. The corresponding lineage-vs.-cluster-only breakdown (Appendix E) shows lineage rendering claims *contradiction-resolution* (+4.4) and *abstention* (+5.0) at a cost of -10.3 on multi-session rea-

Table 4: Ablation on BEAM-100K at gpt-4o-mini. Each row adds one mechanism to the previous configuration. Right column reports per-axis scores for the two belief-revision-direct abilities. Bold cells are the largest entry per column among the ablation rows.

Configuration	Overall (%)	Contradiction-res. (%)	Knowledge-update (%)
Inline only (no cluster, no lineage)	49.9	42.5	45.0
+ cluster $\mathcal{R}_{\text{cluster}}$	51.2	42.5	42.5
+ lineage in context	50.3	46.9	42.5
<i>Reference: full system @ gpt-5.4-mini</i>	54.5	56.2	38.8

Table 5: Effect of cluster consolidation on per-axis BEAM-100K scores (gpt-4o-mini), holding lineage off. The largest gain is on *information extraction* (+9.6): cross-session de-duplication cleans the retrieval surface, which is precisely what $\mathcal{R}_{\text{cluster}}$ ’s DUP action is designed to do. The -2.5 on knowledge-update quantifies the residual rate at which the cluster prompt’s temporal discipline fails (§4.4).

Ability	inline only	+cluster	Δ
<i>Cluster helps (axes that benefit from cleaner retrieval):</i>			
Information extraction	45.5	55.1	+9.6
Temporal reasoning	35.0	38.8	+3.8
Abstention	85.0	87.5	+2.5
Multi-session reasoning	55.7	57.9	+2.2
Event ordering	45.0	46.8	+1.8
Instruction following	43.1	43.8	+0.7
Contradiction resolution	42.5	42.5	0.0
<i>Cluster hurts (axes where supersession over-aligns):</i>			
Preference following	74.4	71.0	-3.4
Knowledge update	45.0	42.5	-2.5
Summarization	28.0	25.9	-2.1

soning and -8.8 on event-ordering where chains add noise to queries that need a focused context.

Implication for $\mathcal{S}$. The context-selection step in our current configuration renders chains by default for all retrieved facts; the data quantifies the cost of that policy and argues for a sharper $\mathcal{S}$ that emits drop or head more aggressively on short-context-preferring queries. We treat this as a methodological finding rather than a tuning issue: *not every query benefits from lineage*, and the revision-history asset only pays off when surfaced selectively. To be precise about what these experiments validate: $\mathcal{S}$ is the selector *interface*, and the configuration evaluated throughout this paper is *chain-forward* (lineage is rendered by default). The ablations here establish the conditional value and cost of lineage; they do *not* validate an optimized dynamic selector policy. A direct always-head / always-chain / dynamic- $\mathcal{S}$ comparison is left to future work. The principal empirically validated contribution of BELIEFMEM is thus *write-time belief revision*, not query-time selector optimization.

Threshold and failure analysis. The cluster phase cosine threshold τ trades recall (lower threshold catches more candidate conflicts) against precision (higher threshold avoids spurious revisions). To locate where BELIEFMEM breaks, we categorize 181 wrong answers from the backbone-matched run by response shape (abstention/extraction miss, list/ordering, confident assertion, mixed) in Appendix D: contradiction-resolution errors concentrate in the *confident assertion* cell (14/24), indicating that BELIEFMEM has the predecessor in the lineage chain but the answer model picks one side without surfacing the conflict as the BEAM rubric requires—an answer-prompt failure, not a memory failure.

6 Conclusion

We have argued that conversational memory for LLM agents is a belief-revision problem, not a database problem, and that the AGM framework—adapted to belief bases per Hansson—is the right formal lens for it. BELIEFMEM operationalizes AGM expansion, revision, and contraction as LLM-driven primitives over a history-preserving belief base, revised at two timings ($\mathcal{R}_{\text{inline}}$ per round, $\mathcal{R}_{\text{cluster}}$ at session flush); the store never deletes, and an LLM-mediated context-selection step decides per query whether to surface a fact’s head, its full lineage, or nothing. On two long-horizon dialogue benchmarks BELIEFMEM reaches 92.3% on LoCoMo and 54.5% on BEAM, with the decisive gap on the axis that most directly probes belief revision. We hope this invites the formal belief-revision community into LLM agent design and gives memory-system designers a more disciplined target than “store more, retrieve better.”

Acknowledgments

This work is partially supported by NSF IIS-2432486. We use AI tools for writing polishing and helping illustration visually.

Limitations

Our current study is primarily limited in scope rather than in mechanism. The evaluation is conducted in English on benchmarks dominated by Anglophone conversational content, leaving cross-lingual validation to future work, although the design itself does not rely on language-specific stop lists. We also assume a single-writer revision setting within each memory scope and do not study concurrent writes from multiple agents, which is outside the intended deployment setting of the present work. Finally, while the system operationalizes the main relationships needed for conversational memory revision—contradiction, temporal update, restatement, and retraction—it does not yet introduce separate primitives for finer-grained cases such as entailment suppression, explicit temporal expiration, or unresolved argumentative tension; these extensions are natural future refinements rather than prerequisites for the proposed framework.

References

- Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. 1985. On the logic of theory change: Partial meet contraction and revision functions. *The Journal of Symbolic Logic*, 50(2):510–530.
- 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*.
- Eduardo Fermé and Sven Ove Hansson. 2011. Agm 25 years: Twenty-five years of research in belief change. *Journal of philosophical logic*, 40(2):295–331.
- Giorgos Flouris, Dimitris Plexousakis, and Grigoris Antoniou. 2006. On generalizing the agm postulates. In *STAIRS*, volume 142, pages 132–143.
- Peter Gärdenfors. 1988. *Knowledge in flux: Modeling the dynamics of epistemic states*. The MIT press.
- Adam Grove. 1988. Two modellings for theory change. *Journal of philosophical logic*, pages 157–170.
- Sven Ove Hansson. 1991. Belief contraction without recovery. *Studia logica*, 50(2):251–260.
- Sven Ove Hansson. 1999. *A textbook of belief dynamics: Theory change and database updating*. Kluwer Academic Publishers.
- Chadi Helwe et al. 2025. ReviseQA: A benchmark for belief revision in multi-turn logical reasoning. In *ICML 2025 Workshop*.
- Chenxu Hu, Jie Fu, Chenzhuang Du, Simian Luo, Junbo Zhao, and Hang Zhao. 2023. Chatdb: Augmenting llms with databases as their symbolic memory. *arXiv preprint arXiv:2306.03901*.
- Sean Lee, Aamir Shakir, Darius Koenig, and Julius Lipp. 2024. [Open source strikes bread - new fluffy embeddings model](#).
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.
- Xianming Li and Jing Li. 2023. Angle-optimized text embeddings. *arXiv preprint arXiv:2309.12871*.
- 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.
- Charles Packer, Vivian Fang, Shishir_G Patil, Kevin Lin, Sarah Wooders, and Joseph_E Gonzalez. 2023. Memgpt: towards llms as operating systems.
- Young Bin Park. 2026. Graph-native cognitive memory for ai agents: Formal belief revision semantics for versioned memory architectures. *arXiv preprint arXiv:2603.17244*.
- 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*.
- Márcio M Ribeiro and Renata Wassermann. 2009. Base revision for ontology debugging. *Journal of logic and computation*, 19(5):721–743.
- Mohammad Tavakoli, Alireza Salemi, Carrie Ye, Mohamed Abdalla, Hamed Zamani, and J Ross Mitchell. 2025. Beyond a million tokens: Benchmarking and enhancing long-term memory in llms. *arXiv preprint arXiv:2510.27246*.
- Alessandra Terranova, Alexandra Birch, and Björn Ross. Evaluating long-term memory for long-context question answering. In *EurIPS 2025 Workshop on Metacognition in Generative AI*.
- Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2026. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604.

A Implementation Details

BELIEFMEM is implemented in Python on top of a pluggable storage backend (PostgreSQL +

pgvector in our experiments; the system also supports an HTTP-based vector backend for production deployments). The core defines three protocols:

- **Memory store:** read/write primitives for facts B^F and episodes B^E , with status and predecessor fields exposed at the column level.
- **LLM client:** an async JSON-extracting interface used by $\mathcal{R}_{\text{inline}}$, $\mathcal{R}_{\text{cluster}}$, the context-selection step, and the answer model.
- **Reranker:** optional cross-encoder used for the bidirectional-max similarity in $\mathcal{R}_{\text{cluster}}$ and at retrieval.

Storage of the belief base. The two subsets of §4.1 are stored in separate indices. B^F stores facts with their lifecycle fields (st, p) as columns alongside the embedding. B^E stores episodes as immutable per-session narrative summaries linked to their constituent facts. The active-set query is an $O(1)$ filter on $st = \text{ACTIVE}$; lineage queries walk p and are bounded by the chain depth d .

Tenancy. Storage is scoped per tenant: every read or write is parameterized by an $(\text{org_id}, \text{user_id}, \text{agent_id})$ triple, enforced at the query layer so retrieval cannot cross tenants. The active-set filter on $st = \text{ACTIVE}$ is an indexed column condition; lineage queries walk the predecessor pointer p to a depth bounded by configuration (default $d = 5$).

Retrieval orchestration. Retrieval orchestrates four stages per query:

1. **Vector search** over the active fact subset $B^{F,A}$ for the top- k facts by cosine similarity.
2. **Reranking** via a cross-encoder over the top- k candidates (optional; we use `mxbai-rerank-large-v1` in all experiments).
3. **Lineage attachment** for each retained fact, walking p up to a configured depth (5 in all experiments).
4. **Context selection** $\mathcal{S}$ (§4.5): an LLM call that decides per retrieved fact whether to render the head, the full chain, or to drop the fact entirely. Episode context for the surviving facts is added next, and the result is character-budgeted to fit the answer model’s context window.

B Postulate Fidelity: Per-Postulate Analysis

We do not claim BELIEFMEM satisfies the AGM postulates exactly. We claim it satisfies them in a principled, defensible way appropriate to the LLM-driven setting. Table 7 summarizes per-phase fidelity, and Table 6 cross-references each core design decision in BELIEFMEM to the AGM/Hansson insight it follows from and the failure that insight helps avoid—making explicit that the AGM framing is not a post-hoc analogy.

Closure (K1). $K * \varphi$ should be closed under consequence. Our claim language is natural language, not a formal logic, so deductive closure does not apply literally. The operative analogue is *inferential consistency*: the active set should not contain claims a downstream reader would judge to entail a contradiction. $\mathcal{R}_{\text{inline}}$ enforces this within a session via the same LLM call that performs extraction; $\mathcal{R}_{\text{cluster}}$ approximates it by similarity-graph clustering, which is a heuristic for entailment, not entailment itself.

Success (K2), Vacuity (K4), Extensionality (K6). Satisfied operationally: a new claim is always added (K2); when no conflicting claim is active, the operator reduces to expansion (K4); semantically equivalent claims receive the same classification because the resolver depends only on claim content (K6, modulo sampling temperature).

Inclusion (K3). AGM Inclusion requires $K * \varphi \subseteq K + \varphi$: revision should add nothing beyond expansion. We *principally violate* (K3) by writing the predecessor pointer $p(\varphi) := \psi$ and the status transition $st(\psi) := \text{SUPERSEDED}$. This is more than expansion produces. We defend the departure: a revision log enables history-aware retrieval (§4.5), which classical AGM is conventionally indifferent to but which a dialogue agent benefits from when answering “why did the user change their mind?”-style queries.

Consistency (K5). The active base $B_t^{F,A}$ should be inferentially consistent. $\mathcal{R}_{\text{inline}}$ enforces this within a session (the same LLM call sees both claims and revises); $\mathcal{R}_{\text{cluster}}$ enforces it across sessions whenever the similarity threshold τ admits the conflicting pair as an edge. False negatives (missed conflicts) and false positives (spurious revisions) are both possible; we measure both empirically (§5).

AGM / Hansson insight	BELIEFMEM design decision	Failure it avoids
Expansion, revision, and contraction are <i>distinct</i> operations (§3, §4.2).	EXP inserts a compatible new claim; REV supersedes an antecedent with a successor and a lineage pointer; CON retracts without replacement; DUP handles an already-entailed restatement.	Collapsing every update into “keep both” vs. “replace/delete”.
Belief <i>bases</i> store explicit premises, not deductive closure (§3, §4.1).	Natural-language claims are first-class records; duplicate detection is opportunistic rather than mandatory.	Requiring deductive closure over noisy LLM-extracted memory.
<i>Recovery</i> is rejected for base contraction (§3; this appendix).	CON marks a claim RETRACTED; a later reassertion does not reconstruct unrelated former beliefs.	Resurrecting stale consequences after a retraction.
The postulates supply an explicit <i>audit</i> vocabulary (§3; this appendix).	A per-postulate fidelity table labels each postulate–resolver pair as satisfied, approximated, or principally violated.	An unqualified claim of exact classical-AGM compliance.
Minimal change via typed atomic units and a single-target invariant (D1/D3) (§4.1–4.2).	A correction supersedes only the conflicting atomic claim; unrelated active claims are left untouched.	Collateral deletion of unrelated beliefs.
History preservation via status transitions and predecessor pointers (§4.5).	Superseded/retracted records stay physically queryable; lineage is traversable; retrieval begins from active heads with head/chain/drop selection.	Presenting stale beliefs as current, or losing auditability.

Table 6: AGM/Hansson insight $\rightarrow$ BELIEFMEM design decision $\rightarrow$ failure avoided. The AGM framing is not a post-hoc analogy: each operator distinction, the explicit-base semantics, the non-Recovery contraction policy, and the audit postulates directly motivate a specific mechanism in BELIEFMEM.

Table 7: Postulate fidelity per resolver. $\checkmark$ = satisfied, $\sim$ = approximated, $\times$ = principally violated. Discussion below.

Postulate	$\mathcal{R}_{\text{inline}}$	$\mathcal{R}_{\text{cluster}}$
(K1) Closure	$\sim$	$\sim$
(K2) Success	$\checkmark$	$\checkmark$
(K3) Inclusion	$\times^\dagger$	$\times^\dagger$
(K4) Vacuity	$\checkmark$	$\checkmark$
(K5) Consistency	$\sim$	$\sim$
(K6) Extensionality	$\checkmark$	$\checkmark$
Recovery (contraction)	$\times$	—

† Inclusion is principally violated by the history-preserving belief base; we defend this as a deliberate departure (see text).

Recovery. The Recovery postulate $K \subseteq (K - \varphi) + \varphi$ requires that re-adding a contracted claim restores the original belief set, which presupposes auxiliary reconstruction state. We do not implement Recovery; we follow Hansson’s view (Hansson, 1991) that it is the wrong target for dialogue memory (a user who retracts “*I live in Paris*” does not expect re-asserting it later to silently restore unrelated Paris-derived beliefs).

C Resolver Algorithms

This appendix gives the full procedural specifications of $\mathcal{R}_{\text{inline}}$ (§4.3) and $\mathcal{R}_{\text{cluster}}$ (§4.4).

Algorithm 3 Per-round resolver $\mathcal{R}_{\text{inline}}$.

Require: dialogue batch D_t ; active window $W \subseteq B_t^{\text{F},\text{A}}$
Ensure: updated belief base B_{t+1}^{F}

```

1:  $(\Phi_t, C_t) \leftarrow \text{LLMEXTRACTANDRESOLVE}(D_t, W)$   $\triangleright$  single LLM call
2:  $T \leftarrow \emptyset$   $\triangleright$  claimed targets (D3)
3: for each  $\psi \in C_t$  do  $\triangleright$  contractions first
4:   if  $\psi \notin T$  then
5:      $st(\psi) \leftarrow \text{RETRACTED}$ ;  $T \leftarrow T \cup \{\psi\}$ 
6:   end if
7: end for
8: for each  $\varphi \in \Phi_t$  with action  $a \in \mathcal{A}$  and (optional) target  $\psi$  do
9:   if  $a = \text{DUP}$  then continue  $\triangleright \varphi$  already entailed by  $B^{\text{F}}$ 
10:  else if  $a = \text{EXP}$  then
11:     $B^{\text{F}} \leftarrow B^{\text{F}} \cup \{\varphi\}$ 
12:  else if  $a = \text{REV}$  and  $\psi \notin T$  then
13:     $p(\varphi) \leftarrow \psi$ ;  $st(\psi) \leftarrow \text{SUPERSEDED}$ 
14:     $B^{\text{F}} \leftarrow B^{\text{F}} \cup \{\varphi\}$ ;  $T \leftarrow T \cup \{\psi\}$ 
15:  end if
16: end for
17: return  $B^{\text{F}}$ 

```

D Failure Analysis

To locate where BELIEFMEM breaks, we categorize the 181 wrong answers (rubric score < 0.5) from the backbone-matched run (gpt-4o-mini, full system) by the surface form of BELIEFMEM’s output: *abstention/extraction miss* (the answer is “no information available” or equivalent), *list/ordering* (the answer is a numbered or bulleted list), *confi-*

Algorithm 4 Cross-session resolver $\mathcal{R}_{\text{cluster}}$.

Require: new facts $F_{\text{new}} \subseteq B^{\text{F},\text{A}}$; threshold τ ; similarity σ
Ensure: updated belief base B^{F}

```
1:  $V \leftarrow F_{\text{new}} \cup \bigcup_{\varphi \in F_{\text{new}}} \text{NN}_k(\varphi)$ 
2:  $E \leftarrow \{(u, v) : u, v \in V, u \neq v, \max(\sigma(u, v), \sigma(v, u)) \geq \tau\}$ 
3:  $\mathcal{C} \leftarrow \text{CONNECTEDCOMPONENTS}(V, E)$ 
4: for each component  $c \in \mathcal{C}$  with  $|c| \geq 2$  do
5:    $\mathcal{G} \leftarrow \text{LLMRESOLVE}(c)$ ;  $T_c \leftarrow \emptyset$ 
6:   for each group  $g \in \mathcal{G}$  do
7:     if  $g$  is REV then
8:        $(\psi_0, \dots, \psi_m) \leftarrow \text{SORTBYTIME}(g)$ 
9:       for  $i = 0, \dots, m - 1$  do
10:        if  $\psi_{i+1} \notin T_c$  then
11:           $p(\psi_{i+1}) \leftarrow \psi_i$ ;  $st(\psi_i) \leftarrow$ 
SUPERSEDED
12:           $T_c \leftarrow T_c \cup \{\psi_{i+1}\}$ 
13:        end if
14:      end for
15:    else if  $g$  is DUP then
16:       $w \leftarrow \text{MOSTRECENT}(g)$ 
17:       $st(\psi) \leftarrow \text{SUPERSEDED}$  for all  $\psi \in g \setminus \{w\}$ 
18:    end if
19:  end for
20: end for
21: return  $B^{\text{F}}$ 
```

dent assertion (the answer affirms one side without surfacing alternatives), and *other/mixed*. This is a programmatic classification over answer text—not manual coding—so it identifies *response shapes* rather than ground causes, but the response shape correlates with the underlying failure mode tightly enough to inform diagnosis.

The failure profile splits the BEAM axes into three regimes.

Belief-revision axes (contradiction-resolution, knowledge-update). Contradiction-resolution failures cluster as *confident assertions* (14 of 24): BELIEFMEM stored both predecessor and successor states (the lineage chain is present in the retrieved context), but the answer model picked one side without surfacing the conflict as the BEAM rubric requires. This is an answer-prompt failure, not a memory failure: the information is in the store and reaches the model. Tighter contradiction-surfacing instructions in the answer prompt should recover most of these failures. Knowledge-update failures fall into *other/mixed* (17 of 23), reflecting heterogeneous ways recency can be wrong (incorrect date, wrong attribute, partial chain).

Non-revision axes (event-ordering, summarization). These axes are dominated by *list/ordering* responses (23/23 and 27/30 respectively), which probe a capability orthogonal to belief revision:

arranging facts into a coherent narrative or chronological sequence. Our per-fact representation has no inherent advantage on these axes; improvement requires explicit episode-ordering metadata and a different render layout, not changes to the revision mechanism.

Extraction-bound axes (instruction-following, information-extraction). *Abstention/extraction miss* dominates in instruction-following (10/18) and is the largest cell in information-extraction (7/16). BELIEFMEM answers “*no information available*” when the gold answer was derivable from a fact that either was not extracted or was not retrieved by the query. The bottleneck here is recall—the recall-first extractor we use is one mitigation, but extraction granularity for instruction-style content is an open direction.

E Per-Axis Ablation Tables

The cluster-vs.-inline-only per-axis breakdown (Table 5) appears in the main body alongside the ablation discussion (§5.5). Table 9 gives the complementary lineage-vs.-cluster-only per-ability decomposition.

Per-category LoCoMo at gpt-5.4-mini. For reference, BELIEFMEM at gpt-5.4-mini scores: single-hop 92.9%, multi-hop 90.7%, temporal 75.0%, open-domain 94.6% (overall 92.3%). The corresponding per-ability scores for the backbone-matched gpt-4o-mini BEAM-100K run (overall 50.3%) are: abstention 92.5, preference-following 71.2, information extraction 53.9, instruction following 48.8, multi-session reasoning 47.6, contradiction-resolution 46.9, knowledge-update 42.5, event-ordering 38.0, temporal-reasoning 32.5, summarization 28.9.

F Prompts

All LLM calls operate at temperature 0. Each stage uses a YAML-defined system prompt with the role-specific contract. We summarize the key contracts; the full prompt files are released with the code.

Extraction ($\mathcal{R}_{\text{inline}}$). System prompt: *recall-first* variant. The prompt presents the dialogue batch D_t plus a numbered list of active session beliefs W , and instructs the model to emit a JSON object containing (i) a facts array of new claims, each optionally carrying a *resolves* field with *session_index* and *action* $\in \{\text{REV}, \text{DUP}\}$, and (ii) a top-level retract array listing W -indices to

Table 8: Failure-mode distribution on BEAM-100K at gpt-4o-mini (181 wrongs / 400 questions). The dominant failure cell per ability is bolded. Contradiction-resolution errors concentrate in the *confident assertion* cell: BELIEFMEM has the predecessor in the lineage chain but the answer model selects one side without surfacing both.

Ability	Abst./extr. miss	List/order	Conf. assert.	Mixed
Contradiction resolution	8	0	14	2
Knowledge update	6	0	0	17
Temporal reasoning	7	5	0	7
Event ordering	0	23	0	0
Summarization	0	27	2	1
Multi-session reasoning	6	6	1	6
Information extraction	7	1	0	8
Instruction following	10	4	0	4
Preference following	2	3	0	1

Table 9: Effect of lineage-in-context on per-axis BEAM-100K scores (gpt-4o-mini). Positive Δ means lineage rendering improves the axis; negative means it adds noise.

Ability	+cluster	+cluster+lineage	Δ
<i>Lineage helps (axes that benefit from seeing the predecessor):</i>			
Abstention	87.5	92.5	+5.0
Instruction following	43.8	48.8	+5.0
Contradiction resolution	42.5	46.9	+4.4
Summarization	25.9	28.9	+3.0
<i>Lineage hurts (axes that need short context):</i>			
Multi-session reasoning	57.9	47.6	-10.3
Event ordering	46.8	38.0	-8.8
Temporal reasoning	38.8	32.5	-6.3
<i>Approximately neutral:</i>			
Information extraction	55.1	53.9	-1.2
Preference following	71.0	71.2	+0.2
Knowledge update	42.5	42.5	0.0

contract. The four-relationship contract, decomposition discipline (D1), event-state distinction (D2), and single-target invariant (D3) of §4.3 are stated in the prompt body with worked examples.

Cluster resolver ($\mathcal{R}_{\text{cluster}}$). System prompt: temporal-reasoning variant. Receives a cluster of facts each rendered with its episode title, event date, recorded-at timestamp, and conversation id. Emits a JSON object with one or more groups, each labeled *rev* / *dup* / *keep_all* (corresponding to REV/DUP/KEEP of §4.4) and listing the cluster members it includes.

Context selection ($\mathcal{S}$). System prompt: per-query lineage selector. Receives the query, the retrieved active facts, and their chains. Emits a JSON object with per-fact decisions in {*head*, *chain*, *drop*}.

Answer. Benchmark-specific variants: a BEAM-tuned answer prompt for the BEAM experiments (stricter abstention, contradiction surfacing, chronological ordering for event sequences) and a LoCoMo-tuned variant for the LoCoMo experi-

ments.

G Compute

Each LoCoMo run (1,540 questions across 10 conversations) consumes approximately 15M tokens at gpt-4o-mini (~\$3 per run) and ~20M tokens at gpt-5.4-mini. Each BEAM-100K run (400 questions across 20 conversations) consumes approximately 25M tokens at gpt-4o-mini (~\$5 per run). Wall-clock per LoCoMo run is ~30 minutes with $\text{qa_concurrency} = 20$, $\text{conv_concurrency} = 1$; BEAM-100K is ~45 minutes per run on the same setting. All runs use a single host with no GPU (the cross-encoder reranker runs on CPU via the local Sentence-Transformers backend).

H Per-Round Resolver Disciplines (D1–D3)

This appendix expands the three semantic constraints summarized in §4.3 for the per-round resolver $\mathcal{R}_{\text{inline}}$.

(D1) Atomic-claim decomposition. Each candidate claim φ is decomposed into its atomic claims,

and the resolver classifies φ by a fixed precedence over the relation between its atomic claims and those of each $\psi \in W$:

- *Enrichment*: some atomic claim in a $\psi \in W$ would be lost under replacement by $\varphi \Rightarrow \text{EXP}$. (Both φ and ψ remain active; no supersession is issued.)
- *Entailment*: every atomic claim of φ is already entailed by some $\psi \in W \Rightarrow \text{DUP}$. (φ is dropped at write time.)
- *Strict subsumption*: φ subsumes $\psi \in W$ and at least one atomic claim contradicts $\psi \Rightarrow \text{REV}(\varphi, \psi)$. (φ becomes the chain head; ψ becomes SUPERSEDED.)
- *Scope-narrowing*: φ adds a scoped exception to ψ while ψ remains true outside that scope $\Rightarrow \text{EXP}$. (Both survive; no supersession.)

The precedence is fixed so that ambiguous cases are resolved consistently regardless of stochastic LLM ordering.

(D2) Event–state distinction. Propositions of the form “*did X at t*”—time-indexed events with past-tense verbs, explicit dates, or milestone language—are not eligible for contraction by a present-tense state change. A user reporting they have stopped running does not undo a marathon they completed in 2022. Only beliefs whose proposition is *current* (ongoing preferences, residences, roles, habits) are candidates for REV or CON. Wrongly contracting a past event is silent data corruption; keeping a stale current-state belief is recoverable at retrieval through lineage rendering (§4.5).

(D3) Single-target invariant. Each $\psi \in W$ is the target of at most one action per round:

$$|\{\varphi : p(\varphi) = \psi\}| + \mathbb{K}[\psi \in C] \leq 1,$$

where C is the set of contraction targets emitted by $\mathcal{R}_{\text{inline}}$. The revision history is therefore a forest of chains, traversable in $O(d)$ for chain depth d . Algorithm 3 enforces D3 by processing contractions before revisions and tracking already-claimed targets.