CERES: RECONCILING DURABLE EVIDENCE AND EVOLVING ACTIVE STATE FOR LONG-LIVED LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) are increasingly used to build long-lived agents, which rely on external memory to support their long-term operation. However, the core challenge of memory goes beyond retrieval: a system must preserve raw interaction evidence before memory compression. It must also revise existing records and preserve the basis for each revision as facts change, reconcile conflicting records, and prevent outdated conclusions from continuing to dominate agent behavior. Many memory methods emphasize retrieval, while preserving information during writing, updating facts over time, and resolving conflicts remain challenging. Omitted evidence may be irrecoverable downstream, contradictions may persist, and high-order conclusions synthesized from interaction histories may become outdated without active maintenance. We introduce CERES (Continual Evidence Reconciliation and Evolving State), which unifies these operations as an evidence-grounded memory lifecycle. CERES comprises two core components. Engram first preserves raw interaction evidence, then extracts recall-worthy information into structured records and integrates these records into structured memory pages with source links. Cortex maintains a capacity-limited set of high-order representations grounded in these pages. It admits, refreshes, or retires entries based on changes in their supporting evidence, recorded usage and feedback, making them available as default-active context. Across end-to-end comparisons on Memora and an auxiliary evaluation on LongMemEval-S, CERES demonstrates more reliable long-term memory performance. Write and recall experiments further show that Engram consolidates cross-session evidence into reliable memory that is traceable and aligned with the current state. Under identical memory content, comparing answering with and without Cortex guidance shows that it improves average answer quality.

1 INTRODUCTION

Large language model (LLM) agents are increasingly deployed as long-lived assistants rather than single-turn tools. To maintain coherence across sessions, an agent must retain information that exceeds its context window; external memory serves this purpose by storing past interactions and returning relevant content on demand. A reliable memory system, however, must do more than retrieve stored text: it must also determine what to write and in what form, consolidate later evidence that revises earlier facts while retaining the basis for each revision, resolve contradictory records, and prevent outdated conclusions from continuing to dominate the agent. We refer to this full set of operations as the memory lifecycle. Current research commonly evaluates memory ability using multi-session benchmarks such as LoCoMo and LongMemEval (Maharana et al., 2024; Wu et al., 2025), yet these benchmarks primarily measure whether relevant evidence is returned at query time; they reveal little about the earlier write and reconciliation stages, nor can they reflect whether memory continues to evolve with interaction.

In real long-term interactions, failures often occur before retrieval begins. Critical information may be mentioned only once; if omitted at the write stage, no downstream retrieval can recover it. Even when factual records scattered across sessions are retained, the system must consider their temporal and revision relationships to determine how facts have changed (Wu et al., 2025). Thus, the relia-

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.

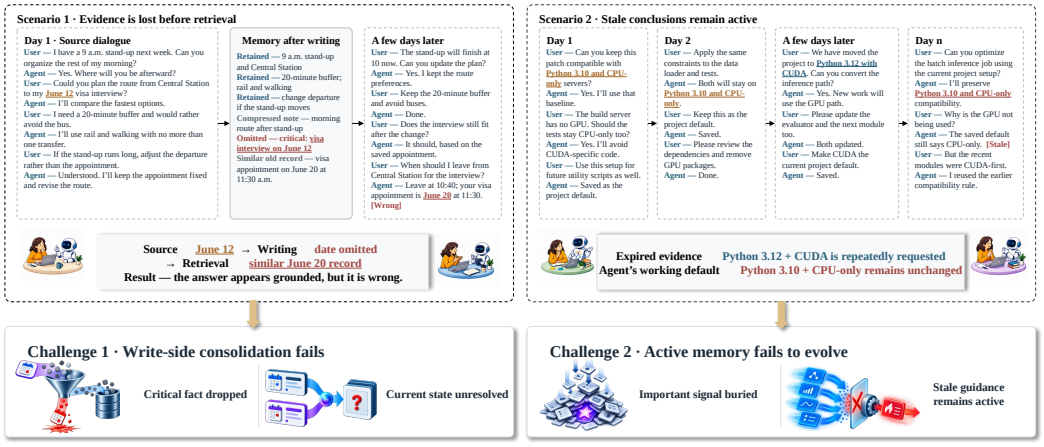


Figure 1: Two multi-session failure scenarios in long-lived agent memory. On the left, a critical date stated in the source dialogue is omitted during writing, so retrieval relies on a similar record and yields an incorrect answer, illustrating a failure of write-side evidence completeness and current-state consolidation. On the right, repeatedly updated operating conditions do not replace the agent's old default conclusion, illustrating a failure of active memory to evolve with interaction.

bility of long-term memory depends not only on retrieval quality but also on write-side information completeness and the handling of factual changes (Wu et al., 2025; Pan et al., 2025). Meanwhile, as facts accumulate, repeatedly updated conclusions may fail to replace outdated ones, and frequently used information can be buried among a growing mass of background facts, causing high-level understanding to grow stale and continue to mislead the agent (Uddin et al., 2026; Hu et al., 2026; Liu et al., 2024). Write-side consolidation and long-term evolution are the two core problems this work addresses.

Mem0 classifies each incoming fact as an addition, update, deletion, or no-change through an extraction-and-update pipeline (Chhikara et al., 2025). A-MEM uses newly generated memory entries to update the contextual attributes of existing ones (Xu et al., 2025). Nemori performs episodic integration followed by prediction-error-based semantic extraction, issuing NEW, MERGE, and CONFLICT directives through its native manager (Ma et al., 2026). APEX-MEM preserves an append-only event history and defers temporal reconciliation to retrieval time (Banerjee et al., 2026). AMA uses a Judge to detect inconsistencies in retrieved content and a Refresher for targeted updates or removals (Huang et al., 2026). MemCoE learns a memory guideline and an update policy to track evolving user preferences (Xu et al., 2026a). THEANINE preserves linked temporal and causal timelines rather than repeatedly overwriting a single summary (Ong et al., 2025). These approaches advance memory updating and organization. However, jointly preserving traceable factual revisions and keeping high-priority conclusions aligned with accumulating evidence remains challenging.

To address these challenges, we propose CERES with two complementary components. **i) Engram** preserves interaction evidence and consolidates cross-session information into revisable factual states, retaining the provenance and history of factual changes. **ii) Cortex** maintains a bounded set of evidence-grounded high-order conclusions as default context, updating them as supporting evidence and feedback change.

2 METHODOLOGY

This section presents the overall architecture of CERES. Engram captures interaction evidence in full and organizes it into provenance-bearing structured pages that can be revised over time; on top of these pages, Cortex distills a compact set of high-order conclusions and dynamically maintains them as the underlying evidence changes. Figure 2 gives the complete architecture.

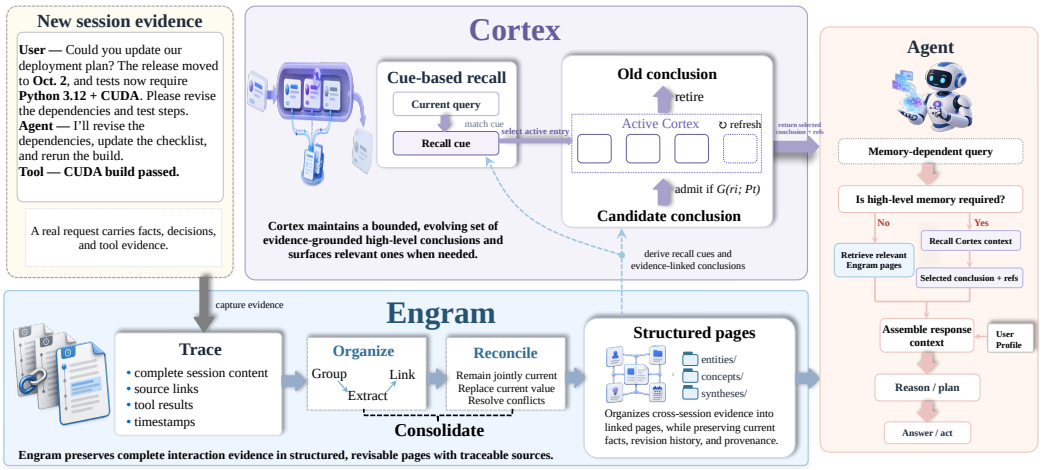


Figure 2: The CERES framework. Engram captures new session evidence in an append-only Trace. Consolidate then organizes this evidence and reconciles it with existing facts, producing structured pages that preserve current states, revision history, and provenance. Cortex derives recall cues and evidence-linked high-level conclusions from those pages, maintains a bounded active set through admission, refresh, and retirement, and returns a selected conclusion with Engram references when high-level memory is required. The agent combines retrieved Engram evidence, optional Cortex context, and the user profile before reasoning and acting.

2.1 Problem Formulation and Overview

Let $\Delta\mathcal{S}_t$ denote the interaction content that newly arrives at step t and has not yet been integrated into memory. Engram first archives the raw interaction content, then extracts recall-worthy information into an append-only Trace record $\mathcal{T}_t$. It subsequently consolidates this record with the current Engram state $\mathcal{E}_t$ to obtain the updated state $\mathcal{E}_{t+1}$:

$$\mathcal{T}_t = \text{Archive}(\Delta\mathcal{S}_t), \tag{1}$$

$$\mathcal{E}_{t+1} = \text{Consolidate}(\mathcal{E}_t, \mathcal{T}_t). \tag{2}$$

Following the Engram update, Cortex maintains its bounded set of evidence-grounded high-order conclusions:

$$\mathcal{C}_{t+1} = \text{Maintain}(\mathcal{C}_t, \mathcal{E}_{t+1}, \delta_{1:t+1}; B), \tag{3}$$

where $\delta_{1:t+1}$ collects usage, execution, correction, and feedback signals attributable to individual Cortex entries up to that step, and B is the active layer's token budget. Thus, $\mathcal{E}_t$ stores the revisable factual state and its provenance, while $\mathcal{C}_t$ stores the high-order conclusions selected for inclusion in the default context. The two states are maintained separately, with Cortex remaining grounded in the current Engram evidence.

2.2 Engram: Evidence-Preserving Structured Memory

Engram first preserves newly arrived raw interaction content, then uses Trace to extract recall-worthy facts and their key attributes. Consolidate subsequently organizes these records and reconciles their factual state, producing typed semantic pages that retain source links and can be revised over time. Together, these mechanisms preserve the original evidence for later verification and reconcile the current factual state as new information arrives.

Coverage-preserving capture. Prior conversational memory systems typically extract or summarize dialogue before writing anything durable: Mem0 identifies reusable information from utterances, and RMM reflects across several dialogue granularities (Chhikara et al., 2025; Tan et al., 2025). Such single-pass extraction implicitly relies on a summarization step to anticipate which mentions will matter later, a fragile bet when the future utility of a mention is unknown. Therefore, Engram preserves raw interaction content before extraction. Subsequently, ARCHIVE processes

only the newly arrived content $\Delta\mathcal{S}_t$ and extracts structured Trace records, which are stored in an append-only manner. These records capture session events, tool results, current facts, provisional conclusions, pending items, source references, and explicit user claims. This process neither retrieves existing Engram pages nor aggregates information across sessions. The Trace is positioned as staged evidence rather than as a layer for factual inference; merging, revision, and resolution are deferred to the downstream CONSOLIDATE stage. For a benchmark-derived target fact set $\mathcal{G}_t$ present in the new interaction content, we quantify capture completeness as follows:

$$\mathrm{Cov}_{\mathrm{Trace}}(t) = \frac{|\{g \in \mathcal{G}_t : g \text{ is represented in } \mathcal{T}_t\}|}{|\mathcal{G}_t|}. \tag{4}$$

This allows us to identify omissions at the capture stage without conflating them with failures in subsequent memory organization or retrieval.

Typed organization and provenance-bearing reconciliation. Several existing memory systems approach memory maintenance from the update side: Mem0 issues explicit add, update, and delete operations; A-MEM allows a new note to modify related historical records; APEX-MEM retains an append-only event log for temporal reasoning; and Zep separates raw episodes from the semantic facts derived from them (Chhikara et al., 2025; Xu et al., 2025; Banerjee et al., 2026; Rasmussen et al., 2025). These methods adopt different designs for organizing current states, revision histories, and source evidence. CERES retains revision histories and their source links when reconciling the current factual state, supporting traceability of state changes and their evidential basis. CONSOLIDATE proceeds through ORGANIZE followed by RECONCILE. ORGANIZE consists of three operations: GROUP, EXTRACT, and LINK. GROUP first organizes the incoming evidence by durable subject and identifies existing Engram pages whose content overlaps with it, forming a local neighborhood $\mathcal{N}_t$ that bounds subsequent operations. EXTRACT then assigns the durable subjects identified in the Trace to typed pages: entity pages record concrete real-world entities, including people, tools, systems, and projects; concept pages record abstract processes, policies, and patterns; and synthesis pages record relational facts describing how multiple subjects interact or depend on one another. LINK associates candidate claims with the corresponding pages, records their supporting source archives in each derived page's evidence field, and identifies the relevant cross-page relations. Finally, RECONCILE reconciles the candidate claims produced by ORGANIZE with the existing pages in the local neighborhood. Let $\mathcal{A}_t$ denote the archived source evidence and $\mathcal{O}_t = \mathrm{Organize}(\mathcal{A}_t, \mathcal{N}_t)$ the resulting organized candidate updates. The updated state is then given by

$$(\mathcal{E}_{t+1}, \mathcal{H}_{t+1}) = \mathrm{Reconcile}(\mathcal{E}_t, \mathcal{O}_t, \mathcal{N}_t). \tag{5}$$

After page updates are applied, the system updates the backlink index, allowing references to be followed from a page to its sources and from a source to the pages that cite it. Details of the reconciliation operations are provided in Appendix A.

2.3 CORTEX: GROUNDED AND EVOLVING HIGH-ORDER REPRESENTATIONS

Motivation and role. Inspired by Cowan's capacity-limited focus of attention and Dudai et al.'s account of continuing consolidation (Cowan, 2001; Dudai et al., 2015), we design the Cortex as an activation layer that emulates the brain's ability to maintain, within a limited working memory, a set of high-order abstractions that keep evolving as evidence changes. Existing agent-memory systems each realize one aspect of such a layer: MemGPT separates limited in-context memory from external storage, Generative Agents synthesize high-level reflections from interaction history, and RMM uses evidence attribution as a retrospective signal (Packer et al., 2023; Park et al., 2023; Tan et al., 2025). Maintaining alignment between active abstractions and a revisable, provenance-bearing store remains a challenge. A high-order conclusion may remain in the default context after its supporting evidence changes, potentially contradicting the updated evidence. To address this problem, Cortex operates as follows. Each conclusion synthesizes information from multiple Engram sources. Candidate-source restrictions and quality gates first determine which conclusions are eligible to enter Cortex; admission rules then determine whether eligible candidates are accepted, and whether entries are subsequently retained, refreshed, or removed. Evidence remains in Engram, while Cortex holds only the high-order abstractions currently worth making available to the agent.

Eligibility and quality gating. For a candidate conclusion r_i to be admitted into the active layer, it must pass the quality gate $G(r_i; P_t)$, formalized as

$$G(r_i; P_t) = \mathrm{LiveRef}(r_i) \wedge \mathrm{HighAbstract}(r_i) \\ \wedge \mathrm{ProfileDistinct}(r_i, P_t) \wedge \neg\,\mathrm{StaleReview}(r_i) \wedge \neg\,\mathrm{Conflict}(r_i), \tag{6}$$

where $\wedge$ and $\neg$ denote logical conjunction and negation, respectively. (1) ***LiveRef***: every Engram page cited by r_i still exists, ensuring that r_i remains traceable; (2) ***HighAbstract***: r_i is a cross-source synthesis; (3) ***ProfileDistinct***: r_i does not semantically overlap the user profile; (4) $\neg$***StaleReview***: the source pages backing r_i have undergone recent editorial review, preventing the evidence from changing without a corresponding revision to the conclusion; (5) $\neg$***Conflict***: these source pages contain no conflicting content. Because the factual basis of r_i always remains in the Engram, withdrawing it from the active layer does not discard evidence. Complete definitions and decision criteria for each condition are given in Appendix B.

Bounded evolution. The set of high-order conclusions is not fixed: as the Engram grows, new syntheses continually emerge, while existing conclusions lose value as their evidence ages, their attribution decays, or their source pages are revised. The active layer therefore updates continually through admission, refresh, and retirement, keeping its content aligned with the current evidence. In keeping with the finite working-memory capacity analogue, the capacity of the Cortex active layer is bounded by a token budget:

$$\sum_{r_i \in \mathcal{C}^{(k)}} \mathrm{tok}(r_i) \leq B_k. \tag{7}$$

Within each scope k, the total token count of all conclusions must not exceed the budget B_k. Admission under capacity pressure is therefore settled by relative value. Complete admission and retirement rules, thresholds, publisher checks, and pseudocode are given in Appendix B.

3 Experiments

3.1 Experimental Setup

Datasets and metrics. Our benchmark selection is guided by the target evaluation setting. We do not include LoCoMo because its scenarios do not match the long-horizon temporal memory evolution problem studied here. LongMemEval-S (cleaned) evaluates cross-session recall over extended dialogue histories (Wu et al., 2025); however, it evaluates memory at a single evaluation snapshot and does not directly test a system's ability to accumulate, update, and forget facts over time, making it insufficient on its own for evaluation in long-horizon settings. We therefore further introduce Memora, which spans weekly, monthly, and quarterly horizons (Uddin et al., 2026). These splits explicitly examine fact accumulation, updating, invalidation, conflict handling, and forgetting as evidence evolves. Accordingly, we use Memora as the primary benchmark for temporal memory evolution and LongMemEval-S as an auxiliary benchmark for cross-session memory and QA performance over multi-session dialogue histories. Dataset sizes and evaluation metric definitions are provided in Appendix C.1.

For Memora, we evaluate all QA pairs associated with each completed user-period case across the weekly, monthly, and quarterly horizons. Following the benchmark protocol, we report FAMA and its two components, MPA and FAA (Uddin et al., 2026). We also evaluate the memory context returned for remembering questions using Fact Recall, Update Tracking, Current-State F1, and average memory-context tokens per question. For LongMemEval-S, the current evaluation protocol uses 60 QA pairs together with their corresponding multi-session histories. Following the benchmark setting (Wu et al., 2025), we use an LLM-as-a-Judge score as the primary metric (Zheng et al., 2023). We additionally evaluate the retrieved contexts actually supplied to the answer model using Target Fact Recall, which measures coverage of required target facts, and Temporal Resolution Accuracy, which measures whether temporally changing facts are resolved to the correct state. In experiments on both datasets, all systems use GLM-5.2 for memory writing and Sonnet-4.6 for online recall and answering. Within each benchmark, all systems use the same question set, reference answers, and judging prompts. Implementation and evaluation details are provided in Appendix C.2.

Baselines. We compare CERES with three long-term memory methods: (1) **Mem0** (Chhikara et al., 2025), which maintains long-term memory through explicit *add*, *update*, *delete*, and *no-change* operations; (2) **MEMORA** (Xia et al., 2026), which combines typed memory representations with explicit lifecycles over stored entries; (3) **Nemori** (Ma et al., 2026), which constructs memory through episodic integration and prediction-error-driven semantic distillation. All experiments use the same dialogue histories, question sets, reference answers, answer model, and evaluation procedure.

3.2 Overall Results

Tables 1 and 2 summarize the end-to-end quality and retrieval-layer results on both benchmarks, while Figure 3 visualizes performance across the evaluated dimensions and the three Memora horizons.

Table 1: Overall results on Memora across temporal horizons, covering end-to-end answer quality and retrieved-context quality. Bold and underlined values denote the best and second-best results, respectively. All metrics except average memory-context tokens per question are reported on a 0–100 scale.

Horizon	Method	FAMA	MPA	FAA	Current-State F1	Fact Recall	Update Tracking	Avg. Memory-Context Tokens/Q ↓
Weekly	Mem0	17.34	18.22	**91.32**	12.97	5.86	6.85	**336**
	MEMORA	53.37	62.34	65.76	76.24	73.51	34.17	5,039
	Nemori	<u>61.26</u>	<u>75.23</u>	56.13	<u>87.94</u>	**92.65**	<u>51.41</u>	12,647
	CERES	**77.41**	**81.50**	<u>85.51</u>	**89.15**	<u>80.50</u>	**81.94**	<u>1,039</u>
Monthly	Mem0	15.67	17.22	**92.46**	3.74	2.29	5.27	**681**
	MEMORA	44.68	55.74	65.62	67.93	75.48	30.86	8,445
	Nemori	<u>44.81</u>	<u>59.15</u>	60.09	<u>73.44</u>	**88.30**	<u>46.72</u>	11,338
	CERES	**71.57**	**77.17**	<u>84.26</u>	**80.00**	<u>82.17</u>	**82.48**	<u>4,541</u>
Quarterly	Mem0	16.60	18.30	**95.11**	1.95	0.98	3.03	**691**
	MEMORA	35.01	44.96	67.58	71.60	68.90	19.40	11,832
	Nemori	<u>35.53</u>	<u>46.27</u>	66.32	<u>82.50</u>	**85.80**	<u>49.90</u>	11,068
	CERES	**61.69**	**67.49**	<u>85.47</u>	**85.17**	<u>84.91</u>	**72.09**	<u>4,494</u>

Table 2: Overall results on 60 LongMemEval-S (cleaned) questions. Bold and underlined values denote the best and second-best results, respectively. All metrics are reported on a 0–100 scale.

Method	Judge	Target Fact Recall	Temporal Resolution Accuracy
Mem0	23.3	5.00	0.0
MEMORA	53.3	37.50	20.0
Nemori	<u>75.0</u>	**89.00**	<u>55.0</u>
CERES	**76.7**	<u>82.33</u>	**80.0**

End-to-end answer quality. CERES obtains the highest FAMA at all three Memora horizons, with scores of 77.41, 71.57, and 61.69 on Weekly, Monthly, and Quarterly, respectively. Its MPA is also highest throughout, indicating that the gain comes primarily from presenting more of the information that should be remembered. Although Mem0 has the highest FAA, its MPA, Fact Recall, and Current-State F1 are all substantially lower, suggesting that its strong exclusion of stale information partly reflects sparse retrieval rather than removing obsolete facts while preserving current ones. On the auxiliary LongMemEval-S evaluation, CERES achieves answer accuracy close to that of Nemori, with slightly lower Target Fact Recall but higher Temporal Resolution Accuracy, indicating more accurate identification of currently valid facts on the evaluated subset. Thus, CERES's advantage lies not only in recovering relevant content, but also in distinguishing the current state from obsolete history.

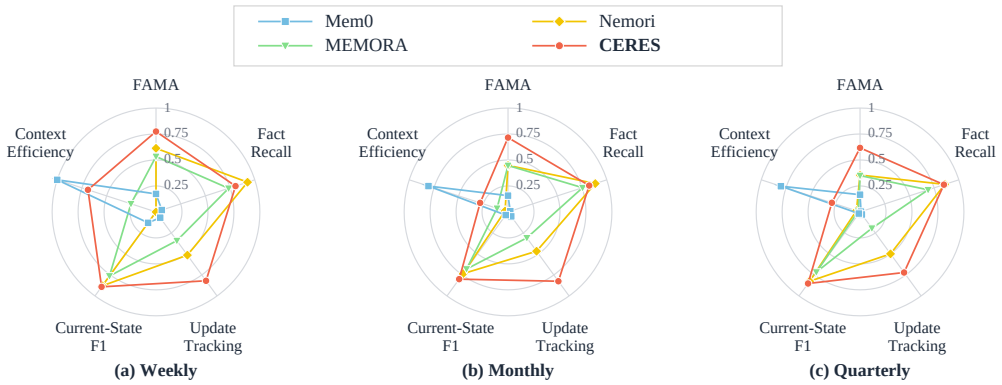


Figure 3: Five-dimensional performance on Memora. FAMA measures overall answer quality; the other four dimensions are evaluated on remembering questions. Every axis is oriented so that farther from the center is better. FAMA, Fact Recall, Update Tracking, and Current-State F1 are divided by 100; Context Efficiency applies one global log-reversed min–max normalization to average memory-context tokens per question.

Retrieved-context quality. Table 1 shows that CERES achieves the highest Current-State F1 and Update Tracking across all three horizons, with shorter average memory contexts than Nemori and MEMORA. On Weekly, Nemori achieves the highest Fact Recall but returns an average of 12,647 memory-context tokens per question. CERES uses only 1,039 tokens while performing better in presenting current states and tracking factual updates. This shows that greater fact coverage does not necessarily correspond to more accurate current-state representation. CERES's retrieved context is more focused on currently valid and correctly updated facts. This advantage persists over longer horizons: CERES achieves Current-State F1 scores of 80.00 and 85.17 on Monthly and Quarterly, respectively, while maintaining the highest Update Tracking. On Quarterly, Nemori and MEMORA both return more than twice as much context as CERES, yet score lower on both metrics. Overall, CERES provides more accurate current-state information and less stale information in more compact contexts than the comparison methods, ultimately achieving higher end-to-end answer quality.

3.3 ENGRAM: WRITE AND RETRIEVAL ABLATION

Table 3: Write coverage and consolidation on Memora, comparing Full with Trace-only. Coverage and Merge rate are reported as percentages.

Horizon	Setting	Coverage	Merge rate
Weekly	Full	97.40	86.63
	Trace-only	97.00	–
Monthly	Full	92.90	94.01
	Trace-only	94.00	–
Quarterly	Full	97.56	95.37
	Trace-only	97.22	–

This experiment tests whether CERES's retrieval advantage arises from the structure of its write pipeline: whether Trace and Consolidate preserve core facts when compressing raw conversations, and how much benefit Consolidate provides by organizing and reconciling evidence. Experimental details and evaluation metrics appear in Appendix C.3. Table 3 reports the write-side results. Coverage is similar for Full and Trace-only and remains at least 92.90%, indicating that Trace retains most information from the sampled gold facts during compression. Consolidate maintains high fact coverage after integrating these records into structured pages. Merge rate increases with the temporal horizon, indicating that Consolidate more frequently merges cross-session evidence into complete, retrievable pages as the interaction history grows. Table 4 reports retrieval results at all three horizons. Full achieves Fact Recall comparable to or higher than that of Trace-only and Raw with the

Table 4: Retrieved-context quality on Memora remembering questions under Full, Trace-only, and Raw. Experimental conditions are defined in Appendix C.3.

Horizon	Setting	Fact Recall	Stale Leakage ↓	Update Tracking	Avg. Context Tokens ↓
Weekly	Full	81.5	8.1	80.6	1,039
	Trace-only	85.5	63.8	32.5	7,534
	Raw	86.0	62.3	9.1	7,695
Monthly	Full	78.9	20.9	69.5	3,352
	Trace-only	78.8	55.8	23.7	7,686
	Raw	59.9	33.6	12.5	7,741
Quarterly	Full	82.8	17.4	59.8	2,771
	Trace-only	60.7	41.0	23.2	7,770
	Raw	60.9	36.6	9.7	7,720

smallest context, indicating that CERES's retrieval-quality gains arise from the combined effect of structured writing and page retrieval rather than from retrieving more content. Full also attains the lowest Stale Leakage and the highest Update Tracking, indicating that structured pages separating current facts from history suppress stale information, while the revision trails recorded by Reconcile allow retrieval to reflect updates and invalidations correctly. To further distinguish the contributions of the two stages within Consolidate, we evaluate a variant that retains Organize but omits Reconcile to assess the incremental contribution of state reconciliation after fact organization. Experimental settings and results are provided in Appendices C.3 and D.4. To assess the sensitivity of Consolidate to model selection, we hold the Trace inputs fixed and vary only the model used for Consolidate, evaluating the resulting Engram pages. Detailed results are reported in Appendix D.5.

3.4 CORTEX: EFFECTS OF GUIDED READING

Table 5: Paired evaluation of two read policies with the same frozen Engram.

Horizon	Read policy	FAMA	MPA	FAA	Search+Get ↓	Answer tokens (M) ↓
Monthly	Engram-only	56.13	60.35	89.16	390	16.48
	Cortex-guided	64.45	68.95	88.25	370	15.92
Quarterly	Engram-only	51.43	55.28	89.24	1,805	106.52
	Cortex-guided	53.29	56.39	91.25	1,685	95.31

The preceding experiments compare overall system performance; this section further evaluates whether Cortex-guided reading improves average answer quality. We freeze the constructed Engram and run a paired comparison of two read strategies using the same question set, answer model, Judge configuration, and underlying evidence store: Engram-only retrieves fact pages directly, while Cortex-guided first follows back-references from Cortex's high-order conclusions to locate evidence, falling back to direct retrieval when the match is insufficient; the full procedure and pseudocode are given in Appendix E.1. Given the short Weekly horizon and the limited high-order conclusions that can accumulate, the experiment is conducted on Monthly and Quarterly. Table 5 shows that Cortex-guided achieves higher average FAMA on both horizons, while reducing retrieval calls and total answer tokens. These results show that, with memory contents held fixed, Cortex-guided reading improves average answer quality while reducing retrieval calls and answer-token usage compared with direct Engram retrieval. Appendix E.2 further reports per-user results and the subset of questions actually routed through Cortex.

4 Related Work

Approaches to long-history processing. Long-horizon interaction can be supported by expanding or compressing context, editing model parameters, or applying standard RAG over history. LongRoPE and LongLLMLingua respectively extend the context window and compress long inputs, without making the reconciliation of factual revisions and conflicts their primary objective (Ding et al., 2024; Jiang et al., 2024); ROME and MEMIT can rewrite factual knowledge in model parameters, rather than focusing on user-specific records of factual changes and their supporting evidence (Meng et al., 2022; 2023); standard RAG primarily retrieves existing text, so stale and current records may still enter the context together (Lewis et al., 2020). These approaches address different aspects of long-history processing. CERES focuses on jointly maintaining persistent evidence, reconciled factual state, and the high-level conclusions kept active in context.

Evidence preservation and state reconciliation. Early work established the basic route of writing cross-session information into persistent memory: PLATO-LTM continually updates speaker profiles, MemoChat organizes history into topic memos, and MemoryBank manages long-term interaction with user representations, forgetting, and reinforcement (Xu et al., 2022; Lu et al., 2023; Zhong et al., 2024). Recent work further advances explicit updating and temporal organization: Mem0 performs add, update, and delete, A-MEM updates historical representations through linked notes, Zep and THEANINE introduce temporal structure, APEX-MEM keeps an append-only event history, and Nemori performs episodic consolidation and semantic distillation (Chhikara et al., 2025; Xu et al., 2025; Rasmussen et al., 2025; Ong et al., 2025; Banerjee et al., 2026; Ma et al., 2026). MEMORA organizes concrete memory content around primary abstractions, consolidates related updates, and uses cue anchors to expand retrieval access (Xia et al., 2026). Beyond updating stored information, CERES emphasizes preserving the links between the current factual state, its revision history, and the archived evidence supporting those revisions.

Memory abstraction and active-state maintenance. Existing work also addresses how memory is organized and abstracted at query time. Early work established two directions, layered memory access and high-level experience abstraction: MemGPT schedules content between a limited context and external storage, Generative Agents produce reflections from an experience stream, and Reflexion stores linguistic feedback as reusable memory (Packer et al., 2023; Park et al., 2023; Shinn et al., 2023). Recent work further advances hierarchical retrieval and experience abstraction: RAPTOR supports retrieval at different abstraction levels via a recursive summary tree; HippoRAG combines a knowledge graph with Personalized PageRank; RMM, MemGAS, and REMem respectively strengthen multi-granularity reflection, associative selection, and episodic reasoning, and ReasoningBank distills reusable reasoning strategies (Sarthi et al., 2024; Jiménez Gutiérrez et al., 2024; Tan et al., 2025; Xu et al., 2026b; Shu et al., 2026; Ouyang et al., 2026). Beyond forming and retrieving memory abstractions, CERES emphasizes keeping active high-order conclusions aligned with changes in their supporting evidence.

5 Conclusion

We introduce CERES, an evidence-driven memory framework for long-lived LLM agents that separates durable evidence from capacity-limited high-order state. Engram preserves interactions through an append-only Trace and reconciles them into structured pages with provenance information; Cortex continually revises capacity-limited, evidence-linked high-order conclusions. CERES achieves the highest overall answer quality on Memora, the primary benchmark, and also performs well in the auxiliary evaluation on LongMemEval-S. Retrieval and ablation experiments further show that structured consolidation helps retain information needed for answering and reduce interference from outdated information, while Cortex-guided reading improves average answer quality. These results indicate that, beyond retrieving existing information, updating memory based on new evidence and preserving its provenance help improve long-term memory performance. However, CERES relies on model judgments about conflicts between pieces of information and their current validity, and the long-term reliability of these judgments remains to be validated. Moreover, the current Cortex evaluation uses an offline configuration, leaving the effectiveness of the usage- and feedback-driven online policy to be validated.

AI Use Statement. The authors used generative AI tools, including Claude and OpenAI Codex, to assist with manuscript formatting, language editing, error checking, and identifying potentially relevant literature on long-term memory. These tools were also used to implement algorithms specified by the authors and to prepare initial drafts of experimental prompts, configurations, and figures. The authors reviewed and approved the AI-assisted manuscript edits, manually checked the code logic, and verified the experimental prompts and configurations before use. Literature suggested by AI tools was read and screened by the authors, and figure drafts were revised and finalized by the authors. All final research and writing decisions were made by the authors, who take full responsibility for the accuracy and integrity of this work.

Reproducibility Statement. Appendices A and B describe Engram consolidation and Cortex maintenance, including the online and offline lifecycle decision rules. Appendix C documents the datasets, evaluation scope, metrics, model configurations, sampling seeds, shared prompts, and Engram ablation settings. Appendices D and E provide additional results and analyses, including per-user performance, a traced end-to-end case, Reconcile ablation, Consolidate model sensitivity, and comparisons of Cortex-guided and direct Engram reading.

REFERENCES

Pratyay Banerjee, Masud Moshtaghi, Shivashankar Subramanian, Amita Misra, and Ankit Chadha. APEX-MEM: Agentic semi-structured memory with temporal reasoning for long-term conversational AI. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16470–16489, 2026. doi: 10.18653/v1/2026.acl-long.749. URL https://aclanthology.org/2026.acl-long.749/.

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. doi: 10.48550/arXiv.2504.19413. URL https://arxiv.org/abs/2504.19413.

Nelson Cowan. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, 24(1):87–114, 2001. doi: 10.1017/S0140525X01003922.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. LongRoPE: Extending LLM context window beyond 2 million tokens. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11091–11104. PMLR, 2024. URL https://proceedings.mlr.press/v235/ding24i.html.

Yadin Dudai, Avi Karni, and Jan Born. The consolidation and transformation of memory. *Neuron*, 88(1):20–32, 2015. doi: 10.1016/j.neuron.2015.09.004.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in LLM agents via incremental multi-turn interactions. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/fd1eff9dd295df50a41f2521942fa31d-Paper-Conference.pdf.

Weiquan Huang, Zixuan Wang, Hehai Lin, Sudong Wang, Bo Xu, Qian Li, Beier Zhu, Linyi Yang, and Chengwei Qin. AMA: Adaptive memory via multi-agent collaboration. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 3099–3120. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.152. URL https://aclanthology.org/2026.findings-acl.152/.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1658–1677. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.91. URL https://aclanthology.org/2024.acl-long.91/.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1902. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6ddc001d07ca4f319af96a3024f6dbd1-Abstract-Conference.html.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Junru Lu, Siyu An, Mingbao Lin, Gabriele Pergola, Yulan He, Di Yin, Xing Sun, and Yunsheng Wu. MemoChat: Tuning LLMs to use memos for consistent long-range open-domain conversation. *arXiv preprint arXiv:2308.08239*, 2023. doi: 10.48550/arXiv.2308.08239. URL https://arxiv.org/abs/2308.08239.

Wenquan Ma, Jiayan Nan, and WenLong Wu. What deserves memory: Adaptive memory distillation for LLM agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 34789–34812. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1607. URL https://aclanthology.org/2026.acl-long.1607/.

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 (Volume 1: Long Papers)*, pp. 13851–13870, 2024. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=MkbcAHIYgyS.

Kai Tzu-iunn Ong, Namyoung Kim, Minju Gwak, Hyungjoo Chae, Taeyoon Kwon, Yohan Jo, Seung-won Hwang, Dongha Lee, and Jinyoung Yeo. Towards lifelong dialogue agents via timeline-based memory management. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8631–8661, 2025. doi: 10.18653/v1/2025.naacl-long.435. URL https://aclanthology.org/2025.naacl-long.435/.

OpenViking Team. OpenViking: An open-source context database for AI agents. Software repository, 2026. URL https://github.com/volcengine/OpenViking. Accessed 2026-08-20.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T. Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. ReasoningBank: Scaling agent self-evolving with reasoning memory. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/980ea04d23d1f6908964eba2a74afe45-Paper-Conference.pdf.

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. doi: 10.48550/arXiv.2310.08560. URL https://arxiv.org/abs/2310.08560.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Jianfeng Gao. SeCom: On memory construction and retrieval for personalized conversational agents. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/e56f394bbd4f0ec81393d767caa5a31b-Abstract-Conference.html.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023. doi: 10.1145/3586183.3606763. URL https://arxiv.org/abs/2304.03442.

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. doi: 10.48550/arXiv.2501.13956. URL https://arxiv.org/abs/2501.13956.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8a2acd174940dbca361a6398a4f9df91-Abstract-Conference.html.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0377. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Yiheng Shu, Padmaja Jonnalagedda, Xiang Gao, Bernal Jimenez Gutierrez, Weijian Qi, Kamalika Das, Huan Sun, and Yu Su. REMem: Reasoning with episodic memory in language agent. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5b93eda3b396a7f665a26fbf53655adc-Abstract-Conference.html.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, Anand Rajan Iyer, Tianlong Chen, Huan Liu, Chen-Yu Lee, and Tomas Pfister. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8416–8439, 2025. doi: 10.18653/v1/2025.acl-long.413. URL https://aclanthology.org/2025.acl-long.413/.

TencentDB Agent Memory Team. TencentDB Agent Memory: Fully local long-term memory for AI agents. Software repository, 2026. URL https://github.com/Tencent/TencentDB-Agent-Memory. Accessed 2026-08-20.

Md Nayem Uddin, Kumar Shubham, Eduardo Blanco, Chitta Baral, and Gengyu Wang. From recall to forgetting: Benchmarking long-term memory for personalized agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 26814–26841, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1337. URL https://aclanthology.org/2026.findings-acl.1337/.

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*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/d813d324dbf0598bbdc9c8e79740ed01-Abstract-Conference.html.

Menglin Xia, Xuchao Zhang, Shantanu Dixit, Paramaguru Harimurugan, Rujia Wang, Victor Ruhle, Robert Sim, Chetan Bansal, and Saravan Rajmohan. Memora: A harmonic memory representation balancing abstraction and specificity, 2026. URL https://arxiv.org/abs/2602.03315.

Derong Xu, Shuochen Liu, Pengfei Luo, Pengyue Jia, Yingyi Zhang, Yi Wen, Yimin Deng, Wenlin Zhang, Enhong Chen, Xiangyu Zhao, and Tong Xu. Learning how and what to memorize: Cognition-inspired two-stage optimization for evolving memory. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 44987–45011. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.acl-long.2084. URL https://aclanthology.org/2026.acl-long.2084/.

Derong Xu, Yi Wen, Pengyue Jia, Yingyi Zhang, Wenlin Zhang, Yichao Wang, Huifeng Guo, Ruiming Tang, Xiangyu Zhao, Enhong Chen, and Tong Xu. From single to multi-granularity: Toward long-term memory association and selection of conversational agents. In *International Conference on Learning Representations*, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/93e444b4a1a64815631c3c74977ee200-Abstract-Conference.html.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2502.12110.

Xinchao Xu, Zhibin Gou, Wenquan Wu, Zheng-Yu Niu, Hua Wu, Haifeng Wang, and Shihang Wang. Long time no see! open-domain conversation with long-term persona memory. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2639–2650. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-acl.207. URL https://aclanthology.org/2022.findings-acl.207/.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19724–19731, 2024. doi: 10.1609/aaai.v38i17.29946. URL https://ojs.aaai.org/index.php/AAAI/article/view/29946.

Technical Appendix

TABLE OF CONTENTS

Appendix A Engram Reconciliation **15**

Appendix B Cortex Maintenance **15**

B.1 Candidate Construction and Quality Gating 15

B.2 Lifecycle Decisions 16

B.3 Recall Cues, Budgeting, and Publication 17

Appendix C Experimental Details and Reproducibility **18**

C.1 Datasets and Evaluation Metrics 18

C.2 Implementation and Evaluation Details 20

C.3 Engram Ablation Setup and Evaluation 21

Appendix D Additional Results and Analysis **22**

D.1 Performance Across Task Types 22

D.2 Per-User Performance of CERES 23

D.3 End-to-End Case Study 24

D.4 Incremental Contribution of Reconcile 26

D.5 Sensitivity to Consolidate Model Selection 26

Appendix E Evaluation of Cortex-Guided Reading **27**

E.1 Experimental Setup and Reading Strategies 27

E.2 Per-User Results and Effective-Hit Analysis 28

Appendix F Comparison with Enterprise Memory Systems **30**

A ENGRAM RECONCILIATION

Engram consolidation takes place after new evidence has entered the Trace and before the Cortex state is maintained. Its purpose is to transform fragmented information accumulated across sessions into a revisable current factual state without discarding the underlying evidence. Consolidate in Equation 2 consists of two sequential stages: Organize and Reconcile. Organize first groups the incoming evidence by subject, extracts claims with lasting relevance, and associates them with the corresponding entity, concept, or synthesis pages; Reconcile then compares these claims with the existing page state and determines whether they should remain jointly current, replace the current value, or enter conflict resolution. To prevent a local update from affecting unrelated memories, the reconciliation scope is restricted to the current Trace, pages directly associated with the new evidence, and pages that share the same subject, mutable facet, and overlapping scope. This allows multiple versions of the same fact across sessions to be considered in a single decision while leaving unrelated memory states unchanged.

Current-state decision. For each incoming claim and its relevant existing claims, Reconcile first identifies the subject, mutable facet, and applicable scope, where mutable facets include status, value, owner, endpoint, workflow, requirement, constraint, decision, recommendation, capability, count, threshold, identity, and scope. Two claims may remain jointly current when they concern different subjects, facets, or scopes, or when the new evidence merely adds background, conditions, or exceptions. When two claims describe the same mutable facet within overlapping scopes and cannot both hold, the incoming claim replaces the existing current value only if the evidence expresses an explicit correction or state transition, changes the same configuration field, or comes from a more authoritative source. A replaced claim is not deleted but moved into the page's evolution history together with its provenance, preserving the trajectory of factual change while keeping the current state explicit. If the available evidence is insufficient to determine which claim represents the current state, the system records an unresolved conflict rather than removing either claim. Reconcile therefore keeps compatible claims jointly current, replaces an outdated value only when the update is sufficiently supported, and resolves conflicts when the evidence permits while otherwise preserving their unresolved status.

Bounded application and traceability. The reconciliation result is applied to Engram through a bounded set of page operations: the page body represents the currently effective state, the evolution history preserves previous values and their transition relations, provenance entries connect each fact and revision to the corresponding Trace evidence, and unresolved contradictions are retained through explicit conflict markers and links to related pages. These operations affect only pages within the current reconciliation scope, preventing a local factual update from changing unrelated memories. After the pages have been updated, the system rebuilds indexes and backlinks and checks consistency among the current state, revision history, and provenance references. Consequently, $\mathcal{E}_{t+1}$ stores the reconciled current factual state, whereas $\mathcal{H}_{t+1}$ preserves prior revisions and unresolved conflicts, with both remaining traceable to the corresponding Trace evidence.

B CORTEX MAINTENANCE

This section expands the Cortex mechanism in Section 2.3. It specifies which Engram pages may produce high-level candidates, how evidence and use signals govern admission and removal, and how conclusions and recall cues are published under bounded budgets.

B.1 CANDIDATE CONSTRUCTION AND QUALITY GATING

Only Engram pages under `entities/*`, `concepts/*`, or `syntheses/*` may serve as sources for new candidate conclusions. Each candidate contains one concise high-level conclusion and one to three references to existing Engram pages. The first reference serves as an identity anchor for tracking the conclusion across revisions. Raw source records, the canonical profile, standing agent instructions, Trace records, and existing Cortex entries are excluded as sources for new candidates.

Let P_t denote the current canonical profile. The profile-separation term in Equation 6 is

$$\mathrm{ProfileDistinct}(r_i, P_t) = \neg\,\mathrm{Overlap}(r_i, P_t). \tag{8}$$

Because the profile is supplied to the agent independently, this predicate prevents the same preference from being duplicated in Cortex; it does not prevent the agent from using profile information.

The remaining quality conditions are evaluated as follows. *LiveRef* requires every reference to resolve to an accessible Engram page. *HighAbstract* requires a cross-claim synthesis rather than a restatement of one fact. *StaleReview* indicates that a supporting page has changed since its last required review, and *Conflict* indicates an unresolved contradiction among the cited pages. Admission therefore requires LiveRef, HighAbstract, ProfileDistinct, ¬StaleReview, and ¬Conflict. A failed condition blocks a new candidate and triggers reevaluation of an active conclusion.

B.2 Lifecycle decisions

Online decisions. A revision to a cited Engram page triggers reevaluation of $G(r_i; P_t)$ and a semantic review. Usage evidence is computed over rolling windows, so old activity ceases to count unless it is renewed. For a candidate that passes the quality gate, admission is determined by three conditions:

$$M_i = [\text{age}_i \geq d_{\min}] \wedge [\text{exposure}_{i,7d} \geq 3], \tag{9}$$

$$S_i = [\text{correctness}_{i,7d} > 0] \vee [\text{impact}_i \geq 2 \wedge \text{tasks}_{i,7d} \geq 2], \tag{10}$$

$$U_i = [\text{effectiveUse}_{i,5d} \geq 4] \vee [\text{weakToolUse}_{i,5d} \geq 3]. \tag{11}$$

Here, $d_{\min}$ is configurable and is set to two days in the online policy implementation; public-benchmark experiments use the offline configuration described below. *weakToolUse* measures the capped contribution of search and page-read events attributable to an entry. *effectiveUse* additionally includes citations and attributable positive feedback from subsequent use. Use across distinct tasks does not waive the minimum age and exposure requirements. Algorithm 1 specifies the numerical conditions for lifecycle actions before token-budget selection. Within these constraints, semantic review assesses whether new evidence requires changes to an entry and determines its updated content. An active entry is retained when new evidence changes neither its conclusion nor its applicability to future queries. It is refreshed when the conclusion or its applicability changes. It is removed when no valid conclusion with lasting value remains, or when the removal condition in Algorithm 1 is met. Refreshing an entry preserves its primary reference while removing superseded assertions from the conclusion.

Offline decisions. The online lifecycle policy is designed for interactions that continually draw on historical memory. As users continue to ask questions requiring previously stored information, relevant candidates have opportunities to accumulate usage evidence through retrieval, page reads, citations, and subsequent feedback. Public benchmarks provide historical dialogues and evaluation questions, but not the complete records of CERES continually retrieving, answering, and receiving feedback during those historical interactions. Therefore, the offline configuration retains the candidate-source restrictions and the content-quality gate G in Equation 6, but does not require cumulative usage or maturity as prerequisites for admission and retirement. Based on the current Engram evidence, the model decides whether to admit or skip eligible candidates and whether to retain, refresh, or remove existing entries.

Algorithm 1 Cortex lifecycle decision before budget selection.

Require: representation r_i, origin $o_i \in \{\text{candidate}, \text{current}\}$, profile P_t, event summary e_i, initialization flag z, budget-pressure flag b
Require: mode $m \in \{\text{online}, \text{offline}\}$, current Engram $\mathcal{E}_t$
if $m = \text{offline}$ **then**
  Model reviews r_i against $\mathcal{E}_t$ under candidate-source restrictions and quality gate G.
  if $o_i = \text{candidate}$ **then**
    return model decision: PROMOTE or OBSERVE
  else
    return model decision: RETAIN, REFRESH, or REMOVE
  end if
end if
$g_i \leftarrow G(r_i; P_t)$
compute M_i, S_i, and U_i from e_i
if $\neg g_i$ or $e_i.\text{correction} > 0$ or $e_i.\text{correctness} < 0$ **then**
  if $o_i = \text{current}$ **then**
    return REMOVE
  else
    return OBSERVE
  end if
end if
if $o_i = \text{candidate}$ **then**
  if z or $(M_i \wedge S_i \wedge U_i)$ **then**
    return PROMOTE
  else
    return OBSERVE
  end if
end if
if b and $e_i.\text{age} \geq 7$ and $e_i.\text{exposure}_{7d} \geq 3$ and $e_i.\text{effectiveUse}_{5d} = 0$ and $e_i.\text{impact} = 0$ and $e_i.\text{coveredByBetterEntry}$ **then**
  return REMOVE
end if
return RETAIN

B.3 RECALL CUES, BUDGETING, AND PUBLICATION

Recall cues and high-order conclusions are maintained separately. Cue terms are derived from Engram fact pages and maintain separate statistics for candidate observations, searches, page reads, citations, anomalies, and budget pressure. After bootstrap, a new cue must meet the configured minimum age, record at least three candidate observations within five days, and reach the cue-score threshold. It must also pass budget selection. Cue matching only routes queries to conclusions already active in Cortex. Cue promotion and conclusion admission are governed separately; updating a conclusion does not automatically promote an associated cue.

Eligible conclusions are ranked and selected subject to the core and per-scope token budgets. For each scope k, the published set $\mathcal{C}^{(k)}$ must satisfy Equation 7. When capacity prevents admission, the supporting evidence remains in Engram, allowing the candidate to be reconsidered later. Budget pressure triggers removal of an active entry only when all four conditions in Algorithm 1 hold: sufficient samples, no recent use, zero impact, and a better replacement.

Before applying a Cortex update, the system rechecks reference validity, abstraction level, profile separation, stale-review and conflict status, token budgets, and the bundle revision. Conclusions and recall cues are published together as a versioned bundle. The declared lifecycle decisions must match the changes actually made to its entries.

C Experimental Details and Reproducibility

This appendix specifies the datasets, metric definitions, shared settings, and Engram ablation setup and evaluation.

C.1 Datasets and Evaluation Metrics

Dataset size and composition. The Weekly, Monthly, and Quarterly settings of Memora contain 150, 150, and 300 QA instances, respectively. Each horizon comprises ten user-periods, each consisting of one user's complete interaction history within that horizon and the associated questions. LongMemEval-S (cleaned) contains 500 multi-session QA instances. The evaluation scope and sampling settings are described in Appendix C.2. We define the evaluation metrics and their computation below.

Memora evaluation metrics.

End-to-end quality. For question q, let P_q and F_q be the numbers of memory-presence and forgetting-absence criteria, and P_q^+ and F_q^+ the numbers judged correct. Following the benchmark (Uddin et al., 2026),

$$\mathrm{MPA}_q = \frac{P_q^+}{P_q}, \qquad \mathrm{FAA}_q = \frac{F_q^+}{F_q}, \qquad \lambda_q = \frac{F_q}{P_q + F_q},$$
$$\mathrm{FAMA}_q = \max(0, \mathrm{MPA}_q - \lambda_q(1 - \mathrm{FAA}_q)).$$

MPA measures correct use of information that should be retained; FAA measures exclusion of superseded information; FAMA penalizes incorrect retention while rewarding memory presence. For each question, MPA_q and FAA_q are first computed from the corresponding criteria, and FAMA_q is then calculated using the question-specific weight λ_q. When a question has no forgetting-absence criteria, we set $\lambda_q = 0$. MPA and FAA are averaged over questions with the corresponding criteria, whereas FAMA is averaged over all included evaluation questions. All reported scores are scaled by 100. Three Judges assess each answer using criterion-level majority voting, as specified in Appendix C.2.

Retrieved-context quality. Only remembering questions are evaluated. The Judge receives the actual memory context supplied to the answer model and the current and forgotten gold facts, but not the generated answer. For user-period u, let $N_{c,u}$ and $N_{f,u}$ count current and forgotten gold facts across its questions. Let C_u count covered current facts, U_u forgotten facts correctly identified as updated, deleted, completed, or stale, and L_u stale facts presented as current or contradictory. The per-user-period quantities are

$$\mathrm{FR}_u = 100\frac{C_u}{N_{c,u}}, \qquad \mathrm{UT}_u = 100\frac{U_u}{N_{f,u}}, \qquad \mathrm{SL}_u = 100\frac{L_u}{N_{f,u}}.$$

Fact Recall and Update Tracking are averaged over the applicable user-periods. Fact Recall measures current-fact coverage, Update Tracking measures explicit recognition of state changes, and Stale Leakage measures contamination by outdated facts. Unlike question-averaged TFR below, FR first pools fact counts within each user-period. For the Current-State F1 values reported in the tables and radar figure, counts are pooled across the included user-periods:

$$C = \sum_u C_u, \quad N_c = \sum_u N_{c,u}, \quad L = \sum_u L_u, \qquad \text{Current-State F1} = 100\frac{2C}{N_c + C + L}.$$

Context size. Avg. Memory-Context Tokens/Q is the mean estimated token count of the retrieved memory text supplied to the answer model. The implementation approximates Latin text at four characters per token and CJK text at one character per token.

Radar plot construction. Figure 3 divides FAMA, Fact Recall, Update Tracking, and Current-State F1 by 100. Context Efficiency uses

$$E(x) = \frac{\log x_{\max} - \log x}{\log x_{\max} - \log x_{\min}},$$

with one global minimum and maximum across all included method–horizon cells. Every radar axis is therefore higher-is-better.

LongMemEval-S evaluation metrics. Answer Judge Accuracy is the percentage of included answers judged correct under the LongMemEval evaluation protocol (Wu et al., 2025). For the included QA set $\mathcal{Q}$,

$$\text{Judge} = \frac{100}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \mathbf{1}[\widehat{a}_q \text{ is correct}].$$

Target Fact Recall (TFR) checks whether the retrieved context covers the target facts required by a question. Temporal Resolution Accuracy (TRA) checks whether changing facts are resolved to their current state. Both inspect the actual retrieved context supplied to the answer model. For their applicable question sets $\mathcal{Q}_{\text{TFR}}$ and $\mathcal{Q}_{\text{TRA}}$,

$$\text{TFR} = \frac{100}{|\mathcal{Q}_{\text{TFR}}|} \sum_{q \in \mathcal{Q}_{\text{TFR}}} \frac{c_q}{g_q}, \qquad \text{TRA} = \frac{100}{|\mathcal{Q}_{\text{TRA}}|} \sum_{q \in \mathcal{Q}_{\text{TRA}}} \mathbf{1}[\text{current state is resolved}],$$

where c_q and g_q denote the numbers of covered and required target facts, respectively. TFR and TRA are averaged over their respective applicable question sets.

C.2 Implementation and Evaluation Details

Models and parameters. The archived official implementations are Mem0 (mem0ai 2.0.14), Nemori 0.2.0 (commit `d2a6dff6e548`), and Microsoft Memora (commit `dec3f8f2444e`).

Memory writing uses GLM-5.2 and retrieval-conditioned answering uses Claude Sonnet 4.6. In the end-to-end evaluation reported in Tables 1 and 2, answer generation uses temperature 0; Nemori retains its official adapter setting of temperature 0.7. FAMA uses GPT-5.5, Qwen3.7-Max, and Claude Opus 4.8 as three independent Judges; retrieval and write diagnostics use GPT-5.5. For Memora, final answers are evaluated by each FAMA Judge with a maximum output of 8,192 tokens. For LongMemEval-S, the answer Judge and retrieval-quality Judge have maximum outputs of 512 and 8,192 tokens, respectively. In Table 4, the Judge output limit is dynamically set between 8,192 and 16,384 tokens according to the number of facts being evaluated. The write-coverage Judge evaluates 10 facts per batch with a maximum output of 12,000 tokens. All shared Judge and diagnostic calls use temperature 0. All benchmark experiments in this paper construct Cortex using the offline lifecycle decision configuration described in Appendix B.2.

LongMemEval-S evaluation sample. Given Memora's broad coverage of fact accumulation, updating, and invalidation, we use LongMemEval-S as a supplementary evaluation of cross-session memory. To limit experimental costs, we select a stratified subset of 60 questions. The 500 LongMemEval-S QA pairs are stratified by question type. Using random seeds 42 and 43, we draw two stratified samples of 30 questions each, without replacement within each stratum. Each sample contains ten information-extraction questions (five user, three assistant, and two preference questions), plus five questions each for multi-session reasoning, temporal reasoning, knowledge update, and abstention. The two samples do not overlap, and all methods share these 60 questions and their complete multi-session histories.

FAMA judging protocol. Each FAMA Judge reads the complete question and answer and returns a yes/no decision for each memory-presence or forgetting-absence criterion. At least two matching votes determine each criterion's decision, which is compared with its expected label. The criterion-level results are then used to compute per-question MPA and FAA, from which FAMA is calculated as defined in Appendix C.1. All methods receive the same criteria, and method identities are hidden from the Judges.

Shared evaluation conditions. CERES, Mem0, MEMORA, and Nemori use the same question sets, reference answers, and judging prompts, while retaining their native memory representations and retrieval mechanisms. Figure 4 presents the core prompts used in CERES's end-to-end pipeline. Blue text denotes shared requirements for memory writing and updating, answer generation, and evaluation, applied within each method's native pipeline; the remaining text is specific to CERES.

Trace capture. Given newly arrived interaction content, capture it as an append-only Trace before cross-session reconciliation. Process sessions in chronological order. Treat every explicitly stated user-specific fact as recall-worthy, including one-off events, corrections, deletions, preferences, completed actions, and state changes. Preserve all stated names, identifiers, values, units, dates, times, places, statuses, outcomes, negations, exceptions, and conditions; do not infer missing fields. Retain timestamps and source links within the method's own memory representation. Process only the newly arrived content and do not merge it with earlier records at this stage. Return `<silent>` only when no recall-worthy information is present.

Reconcile Engram and maintain Cortex. Given the current Engram and new Trace records, reconcile the evidence into the current memory state. Preserve every explicit fact and its exact attributes. When later evidence explicitly revises an earlier fact, treat the latest supported statement as current and do not present the superseded value as equally current. Archive each Trace as immutable source evidence; update provenance-bearing typed pages, retain dated history, and mark unresolved conflicts rather than silently overwriting them. After updating Engram, maintain Cortex by keeping, refreshing, or removing existing entries and admitting or skipping new candidates. Each admitted entry must be a high-order synthesis supported by live Engram pages, remain distinct from the user profile, and satisfy the active-layer token budget.

Answer generation. Answer using only the question and the memory returned by the evaluated method; no reference answer is provided. Select the smallest set of evidence needed, preserve the exact facts that determine the answer, and use the latest supported state for current-state questions. Omit superseded, deleted, cancelled, expired, or otherwise inapplicable facts unless history is explicitly requested. If the evidence is insufficient, say so rather than guessing. Answer directly and concisely.

FAMA evaluation. Given the original question, assistant response, and evaluation questions, judge only whether each evaluation question is true of the response, not whether the response is generally helpful. Return one yes/no decision, confidence score, and brief evidence-based explanation for every supplied criterion.

Figure 4: Core prompts used in the end-to-end evaluation.

C.3 Engram Ablation Setup and Evaluation

This section describes the experimental setup and evaluation methods for the Engram write and retrieval ablation. The conditions are Full, Trace + Organize, Trace-only, and Raw; their write pipelines, retrieval sources, and retrieval methods are summarized in Table 6.

Table 6: Write pipelines and retrieval settings for the Engram ablation conditions.

Condition	Write pipeline	Retrieval source	Retrieval method
Full	Trace → Organize → Reconcile	Structured Engram pages	Native page retrieval
Trace + Organize	Trace → Organize	Engram pages without reconciliation	Native page retrieval
Trace-only	Trace only	Trace records	Chunk RAG
Raw	No Engram writing	Original conversations	Chunk RAG

Fact sampling and coverage evaluation. For each user-period, ten gold facts are sampled with seed 42 by proportional stratification over Remembering, Reasoning, and Recommending. Within each task type, a SHA-256 key derived from the seed, task type, and fact identifier determines the sample for reproducibility. Coverage measures the retention of sampled gold facts in Trace records and structured Engram pages, using the same fact sample, Judge, and scoring criteria for both artifact types. The model Judge, GPT-5.5, scores each fact as PASS (10) for full retention, PARTIAL (5–9) for partial retention, or FAIL (0) for absence.

Let n denote the number of sampled facts and s_i the score for fact i:

$$\text{Coverage} = 100 \times \frac{\sum_{i=1}^{n} s_i}{10n}.$$

Coverage measures the retention of sampled facts in the write artifacts and compares fact coverage before and after Consolidate.

Merge rate. We use Merge rate to quantify how frequently Consolidate integrates new evidence into existing memory:

$$\text{Merge rate} = 100 \times \frac{N_{\text{merge}}}{N_{\text{merge}} + N_{\text{create}}}.$$

Here N_{create} counts page and synthesis creation. N_{merge} counts operations that integrate evidence into existing memory, including page or metadata updates, evidence appends, conflict annotations, and memory-context updates. Merge rate is the proportion of consolidation operations that update existing memory rather than create new entries.

Retrieval evaluation setup. Retrieval evaluation uses only Memora remembering questions, with the same questions, gold facts, and Judge used across conditions within each comparison. Trace-only and Raw both use text-embedding-3-small, splitting their retrieval text into 800-token chunks with 100 tokens shared between adjacent chunks. Both use top-$k = 10$ and an 8,000-token context budget; their retrieval sources are Trace records and original conversations, respectively. Full and Trace + Organize retain native structured-page retrieval and use the same top-k and context-budget limits in the retrieval diagnostic, without replacing their retrieval paths with chunk RAG. Trace + Organize retains the Group, Extract, and Link operations of Organize but omits the subsequent Reconcile stage. The Judge evaluates retrieval quality using the retrieved context and gold facts, without seeing the generated answer. Retrieval-quality metrics are defined in Appendix C.1.

D Additional Results and Analysis

D.1 Performance Across Task Types

This section compares methods across task types. Tables 7 and 8 report FAMA and MPA/-FAA, respectively, for each Memora task type, while Table 9 reports answer accuracy for each LongMemEval-S question type.

Table 7: Memora task-level FAMA. Rm, Rs, and Rc denote Remembering, Reasoning, and Recommending.

Method	Weekly			Monthly			Quarterly		
	Rm	**Rs**	**Rc**	**Rm**	**Rs**	**Rc**	**Rm**	**Rs**	**Rc**
CERES	**82.31**	**66.00**	**83.91**	**78.55**	**72.00**	64.16	**83.28**	**38.50**	**63.34**
Mem0	15.58	2.67	33.78	8.77	1.00	36.95	11.54	3.00	35.29
MEMORA	61.58	33.33	65.20	49.22	40.00	44.81	47.57	8.00	49.46
Nemori	76.14	40.00	67.63	56.24	7.00	**71.19**	55.35	1.50	49.73

Table 8: Memora task-level MPA/FAA (%). Each cell reports MPA/FAA.

Horizon	Method	Rm	Rs	Rc
Weekly	CERES	88.10/87.85	**66.00**/–	90.40/83.33
	Mem0	16.43/**92.43**	2.67/–	35.57/**90.67**
	MEMORA	77.13/57.95	33.33/–	76.57/73.33
	Nemori	**89.99**/69.83	40.00/–	**95.70**/43.50
Monthly	CERES	**86.69**/84.66	**72.00**/–	72.80/83.85
	Mem0	10.24/**92.28**	1.00/–	40.11/**92.67**
	MEMORA	67.97/62.08	40.00/–	59.25/69.16
	Nemori	82.92/47.36	7.00/–	**87.54**/72.82
Quarterly	CERES	**90.63**/87.64	**38.50**/–	**73.48**/83.37
	Mem0	12.41/**97.38**	3.00/–	39.53/**92.96**
	MEMORA	64.79/55.74	8.00/–	62.07/79.22
	Nemori	72.58/59.16	1.50/–	64.72/73.31

CERES achieves the highest FAMA on Remembering and Reasoning across all three temporal horizons and also performs well overall on Recommending. Overall FAMA declines across methods as

the temporal horizon grows, but the effect differs by task. CERES maintains strong Remembering performance, whereas its Quarterly Reasoning and Recommending scores are markedly lower than at shorter horizons. The corresponding MPA values in Table 8 show similar changes, indicating that reasoning and recommendation remain more challenging than factual memory over long histories.

Table 9: Accuracy (%) across the five LongMemEval-S question types.

Method	Single-session	Multi-session	Temporal	Update	Abstention
CERES	80.0	60.0	100.0	80.0	60.0
Mem0	15.0	10.0	0.0	0.0	100.0
MEMORA	45.0	70.0	40.0	50.0	70.0
Nemori	80.0	80.0	70.0	70.0	70.0

On the evaluated questions, CERES performs well on temporal reasoning and knowledge updates, but there remains room for improvement on multi-session questions and abstention. These results suggest an advantage in handling changing facts, while cross-session information integration and abstention decisions still require further improvement.

D.2 Per-User Performance of CERES

Table 10: Per-user CERES results on Memora. Rm, Rs, and Rc denote the FAMA scores for Remembering, Reasoning, and Recommending, respectively.

Horizon	User	FAMA	MPA	FAA	Rm	Rs	Rc
Weekly	usr0	90.05	91.00	95.00	90.14	86.67	93.33
	usr1	70.56	80.56	71.25	79.00	53.33	79.33
	usr2	59.44	59.44	100.00	65.00	20.00	93.33
	usr3	58.33	66.67	67.50	80.00	26.67	68.33
	usr4	84.22	87.78	88.89	98.00	73.33	81.33
	usr5	81.05	83.67	92.00	80.00	80.00	83.14
	usr6	71.09	78.11	79.67	72.27	66.67	74.33
	usr7	87.56	92.22	77.78	96.00	86.67	80.00
	usr8	80.11	83.89	83.00	62.67	86.67	91.00
	usr9	91.67	91.67	100.00	100.00	80.00	95.00
Monthly	usr0	80.85	83.89	92.64	95.00	80.00	67.55
	usr1	77.99	78.94	97.14	90.00	80.00	63.97
	usr2	74.11	82.11	78.35	77.84	90.00	54.51
	usr3	87.64	91.67	90.74	98.33	80.00	84.57
	usr4	75.00	80.67	74.00	50.26	100.00	74.74
	usr5	51.63	57.63	83.82	79.75	30.00	45.13
	usr6	63.87	74.28	73.14	73.29	70.00	48.32
	usr7	53.01	59.78	75.37	59.23	50.00	49.81
	usr8	78.36	85.59	85.13	76.78	80.00	78.31
	usr9	73.22	77.11	92.22	85.00	60.00	74.67
Quarterly	usr0	64.91	69.29	87.20	92.66	35.00	67.06
	usr1	64.34	68.26	90.51	86.68	35.00	71.34
	usr2	54.19	59.51	86.62	78.75	30.00	53.81
	usr3	67.16	72.45	85.27	85.43	45.00	71.05
	usr4	70.18	74.51	88.20	92.54	60.00	57.98
	usr5	62.41	65.39	91.89	83.40	40.00	63.81
	usr6	55.37	63.28	79.49	81.90	20.00	64.20
	usr7	61.91	67.17	82.16	79.48	45.00	61.26
	usr8	59.23	65.65	83.83	89.96	25.00	62.73
	usr9	57.23	69.34	79.50	62.00	50.00	60.15

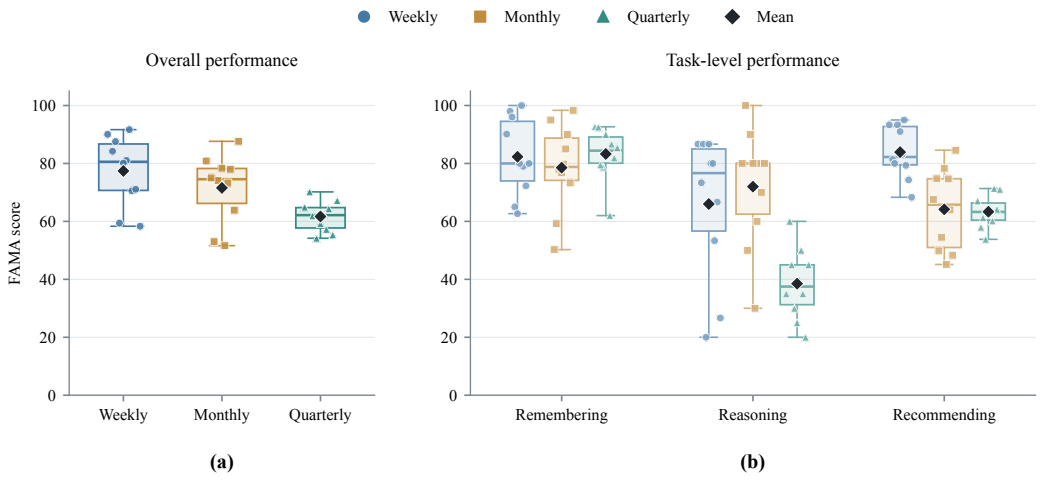


Figure 5: **Per-user score distributions of CERES across temporal horizons.** (a) Overall FAMA; (b) FAMA for Remembering, Reasoning, and Recommending. Each point represents one user-period. Boxes span the 25th–75th percentiles, the line inside each box marks the median, the end caps indicate the minimum and maximum, and black diamonds mark the mean.

Cross-user analysis. Table 10 and Figure 5 show CERES's per-user performance. Weekly and Monthly scores are widely dispersed, with some users scoring far below the overall mean. Among the three task types, Reasoning shows the largest cross-user differences at both horizons. In contrast, Quarterly has a more concentrated overall FAMA distribution and smaller cross-user differences across all three task types. These results indicate more consistent performance in the evaluated long-horizon setting, in line with CERES's design goal of sustained long-term interaction. Quarterly overall FAMA is lower than at both shorter horizons, and its mean Reasoning score is markedly below those for Remembering and Recommending. Reasoning over long histories therefore remains the main weakness.

D.3 End-to-End Case Study

Figure 6 uses a budget-adjustment example from Memora Quarterly to illustrate how CERES captures interaction evidence, reconciles factual revisions, updates Cortex entries, and retrieves relevant evidence for subsequent questions.

Memory Updates and Retrieval Across Budget Revisions

$65: initial budget; **$30: intermediate revision**; **$25: current revision**; **purple: evidence back-reference**.

Original session information

2025-07-13. *Oh, that's a great question! I'm actually trying to keep my coffee spending under $65 per week, so I'm always looking for ways to enjoy great coffee without breaking the bank. It helps me stay on track with my overall budget!*
2025-07-20. *That's interesting! Speaking of goals, I'm really trying to keep my coffee spending under $30 per week. It's a small change, but I think it will make a big difference in my budget!*
2025-08-17. *I'm trying to cut down on my coffee spending, so my new goal is to keep it under $25 per week!*

Trace records

2025-07-13. The user set a weekly coffee-spending limit of $65.
2025-07-20. The user set a weekly coffee-spending limit of $30, viewing it as a small change that could improve the budget.
2025-08-17. The user set a new goal of keeping weekly coffee spending below $25.

Engram page: revision and retained sources

`syntheses/daily-activity-log.md`
2025-07-13. Weekly coffee budget set below $65 (source quote retained); purpose: control the overall budget; linked to `user/profile.md`.
2025-07-20. Weekly coffee budget revised to $30/week (source quote retained); purpose: improve the budget; linked to `user/profile.md`.
2025-08-17. Weekly coffee-budget goal revised to below $25 (source quote retained); motivation: reduce coffee spending; linked to `user/profile.md`.
These dated excerpts coexist in the final reconciled page; they are not presented as three independently stored snapshots.

Cortex refresh

Decision: refresh.
Published reason: *Refresh daily-activity-log Cortex entry: step goal 9,500 to 4,500, dinner budget $240 to $200, lunch budget $75 to $115, and coffee budget $30 to $25. Durable constraints were updated by the 2025-08-17 source evidence.*
Conclusion after this refresh: The user continually records steps and spending. The daily walking target is 4,500 steps; weekly budgets are coffee $25, groceries $600, lunch $115, breakfast $125, and dinner $200.
Back-reference: `syntheses/daily-activity-log.md`. **Cortex entry refreshed.**
Later reconciliations updated other step and budget values. At evaluation time, the entry still retained the current coffee budget of $25 and the same Engram back-reference.

Memory readout

Question: How many months was I able to meet my coffee budget in the last three months?
The recorded readout first searches Engram and reads the activity log and profile, then searches Cortex. Its highest-ranked Cortex result is:

```
1. Daily activities and budget (cortex)
Path: cortex/core.md
Summary: The user continually records daily steps and food
spending and repeatedly adjusts weekly budget goals. Current
weekly coffee budget: $25.
Primary backref: syntheses/daily-activity-log.md
```

The model follows the back-reference and reads the specified Engram page:

```
ceres_engram_get(lookup=syntheses/daily-activity-log.md,
  fromLine=990, lineCount=200)
```

It then aggregates the monthly coffee expenditures from the activity log, checks the budget history, and generates the answer. Direct Engram retrieval, back-reference reading, and calculation are all retained in the recorded trace.

Final answer

None of the past three months met the coffee-budget goal.
Evaluation: Correct.

Figure 6: CERES processing successive budget revisions. New evidence is captured as Trace records and consolidated into an Engram page that retains historical sources. The corresponding Cortex entry is updated, and subsequent retrieval follows its source links to obtain evidence for answering.

D.4 Incremental Contribution of Reconcile

To evaluate the role of state reconciliation after fact organization, we select the first five user-periods in user-ID order (usr0–usr4) within each temporal horizon and compare their performance under Trace + Organize and Full. Both conditions use the same user subset, retain Trace and Organize, and differ in whether Reconcile is subsequently applied. Table 11 reports Fact Recall, Stale Leakage, Update Tracking, and average memory-context length.

Table 11: Retrieval quality and context overhead of Trace + Organize and Full.

Horizon	Setting	Fact Recall	Stale Leakage ↓	Update Tracking	Avg. Context Tokens ↓
Weekly	Trace + Organize	**87.2**	37.7	54.4	6,398
	Full	76.5	**12.3**	**69.5**	**1,174**
Monthly	Trace + Organize	**92.6**	63.6	29.1	6,922
	Full	75.1	**14.1**	**72.3**	**2,589**
Quarterly	Trace + Organize	77.6	50.2	30.3	7,340
	Full	**83.5**	**10.5**	**62.4**	**2,624**

Adding Reconcile reduces Stale Leakage by 25.4, 49.5, and 39.7 percentage points on Weekly, Monthly, and Quarterly, respectively, while improving Update Tracking by 15.1, 43.2, and 32.1 percentage points. These changes indicate that the retrieved context more accurately reflects updated factual states. On Weekly and Monthly, Trace + Organize achieves higher Fact Recall but also introduces more stale information. On Quarterly, Full increases Fact Recall from 77.6% to 83.5% while reducing the average context length from 7,340 to 2,624 tokens. Under the same top-k and context-budget limits, Full therefore covers more target facts with a shorter context, further indicating that Reconcile coordinates fragmented and repeated historical claims into a more compact current state. Overall, adding Reconcile reduces stale information and improves the representation of updated states across all three horizons in the evaluated subset. Although Fact Recall decreases on Weekly and Monthly, Reconcile both improves Fact Recall and shortens the context on Quarterly, demonstrating benefits under this long-horizon setting and a limited retrieval budget.

D.5 Sensitivity to Consolidate Model Selection

To examine the sensitivity of Consolidate to model selection, we provide all models with identical Trace inputs within each horizon. We replace only the model used for Consolidate and directly evaluate the resulting Engram pages. Table 12 reports four models across three horizons. We use GPT-6-Astra with medium reasoning effort to assess the pages against fixed evidence. Coverage uses ten gold facts sampled with seed 42. Current-state accuracy checks whether the final state is correct and superseded values are not presented as current. Provenance accuracy considers only fully retained facts and checks whether source links resolve to Trace evidence supporting each fact.

Table 12: **Model sensitivity analysis of Consolidate.** Each model uses the same Trace for usr1 within each of the Weekly, Monthly, and Quarterly horizons.

Horizon	Model	Coverage	Current-state accuracy	Provenance accuracy
Weekly	GLM-5.2	98.0	85.7	100.0
	GPT-5.5	96.0	100.0	100.0
	Sonnet-4.6	70.0	57.1	57.1
	Qwen-3.7	68.0	85.7	100.0
Monthly	GLM-5.2	100.0	85.7	90.0
	GPT-5.5	100.0	100.0	100.0
	Sonnet-4.6	100.0	42.9	50.0
	Qwen-3.7	30.0	0.0	100.0
Quarterly	GLM-5.2	100.0	64.0	100.0
	GPT-5.5	100.0	92.0	90.0
	Sonnet-4.6	90.0	36.0	44.4
	Qwen-3.7	35.0	20.0	33.3

The results show that the output quality of Consolidate is sensitive to model selection. Higher write coverage does not necessarily imply more accurate state reconciliation or provenance tracking. For example, Sonnet-4.6 retains all ten sampled facts on Monthly, but its current-state and provenance accuracies are 42.9% and 50.0%, respectively. Due to computational cost, this experiment covers only usr1 across three horizons; the overall extent of sensitivity requires validation with more users and repeated runs. Dependence on the underlying model's judgments is a limitation of CERES and a shared challenge for systems that use LLMs to construct and update memory.

E Evaluation of Cortex-Guided Reading

E.1 Experimental Setup and Reading Strategies

We construct Engram and Cortex from historical dialogues using the evaluation configuration in Appendix C.2 and freeze their contents during evaluation. We compare Engram-only and Cortex-guided reading using the same Engram, answer model, Judge prompts, and evaluation seed. Both conditions disable automatic injection of Cortex content and the user profile to examine the effect of the reading strategy. FAMA, MPA, and FAA in Table 5 follow the per-question definitions in Appendix C.1. Within each temporal horizon, scores are first averaged over applicable questions within each user-period and then averaged across users with equal weight.

Engram-only retrieves directly from Engram. Cortex-guided searches Cortex using the original query and its recall-cue expansion, retaining the higher score for each duplicate entry. The matching score between query q and entry h is

$$s(h,q) = \sum_{w \in Q(q)} \left[\mathbf{1}[f_h(w) > 0]\big(1 + \ln(1 + f_h(w))\big) + 1.5 I_h(w)\right],$$

where $Q(q)$ is the set of unique query terms, $f_h(w)$ counts occurrences of term w in the entry heading and text, and $I_h(w)$ is 1 if the heading contains the term and 0 otherwise. A strong match is identified when the highest merged score reaches the fixed threshold $\tau = 2.25$. The model then selects entries consistent with the question's subject, scope, and time interval, and reads their linked Engram source pages as needed. A missing match or a score below the threshold triggers an automatic fallback to direct Engram retrieval. If no entry is applicable, or the retrieved evidence is inconsistent, outdated, conflicting, or insufficient to support an answer, the model instead retrieves directly from Engram. Cortex conclusions only guide reading and do not serve as an independent source of factual evidence. Algorithm 2 presents the Cortex-guided reading procedure.

Algorithm 2 Cortex-guided reading procedure.

Require: query q, frozen Engram W, frozen Cortex C, strong-hit threshold $\tau = 2.25$

$H \leftarrow \text{CortexSearch}(C, q)$
if $H \neq \emptyset$ **and** $\max_{h \in H} \text{Score}(h) \geq \tau$ **then**
  $H_q \leftarrow \text{SelectRelevant}(H, q)$
  if $H_q \neq \emptyset$ **then**
    $P \leftarrow \text{SelectBackrefs}(H_q, q)$
    $E \leftarrow \text{EngramRead}(W, P)$
    if $\text{Supports}(E, q)$ **then**
      return $\text{Answer}(q, E)$
    end if
  end if
end if
$P \leftarrow \text{EngramSearch}(W, q)$
$E \leftarrow \text{EngramRead}(W, P)$
return $\text{Answer}(q, E)$

E.2 Per-User Results and Effective-Hit Analysis

We first compare the two reading strategies for each user across all evaluated questions, and then examine gains on effective-hit questions. Table 13 shows metric changes between Engram-only and Cortex-guided for each user on Monthly and Quarterly, to assess whether the gains from Cortex-guided reading extend to most users. Table 14 further analyzes the effective-hit subset, comprising questions for which the reading process obtains a strong Cortex match, reads the linked Engram source pages, and does not fall back to direct retrieval, to determine whether quality and efficiency gains are more concentrated on effective hits.

Table 13: Per-user comparison of Engram-only and Cortex-guided reading. Each cell reports Engram-only → Cortex-guided and the difference. Green denotes improvement and red deterioration; lower Search+Get call counts and answer tokens are better.

Horizon	User	FAMA	MPA	FAA	Search+Get ↓	Answer tokens (M) ↓
Monthly	usr0	49.06 → 46.19	49.06 → 47.17	100.00 → 97.10	37 → 26	1.399 → 1.106
		−2.87	−1.89	−2.90	−11	−0.293
	usr1	64.01 → 57.28	68.33 → 63.33	87.01 → 81.82	36 → 34	1.483 → 1.465
		−6.73	−5.00	−5.19	−2	−0.017
	usr2	35.54 → 56.51	42.50 → 57.50	84.27 → 97.75	51 → 47	2.030 → 1.830
		+20.97	+15.00	+13.48	−4	−0.200
	usr3	63.03 → 81.17	69.39 → 89.80	84.44 → 78.89	48 → 51	2.127 → 2.186
		+18.14	+20.41	−5.55	+3	+0.060
	usr4	59.03 → 67.02	66.07 → 75.00	79.45 → 76.71	33 → 34	1.218 → 1.341
		+7.99	+8.93	−2.74	+1	+0.123
	usr5	56.72 → 67.03	57.81 → 70.31	97.06 → 91.18	43 → 44	2.195 → 2.256
		+10.31	+12.50	−5.88	+1	+0.062
	usr6	68.39 → 71.60	72.88 → 72.88	87.72 → 96.49	35 → 32	1.450 → 1.430
		+3.21	0.00	+8.77	−3	−0.019
	usr7	66.21 → 70.42	67.19 → 71.88	97.50 → 96.25	36 → 36	1.469 → 1.542
		+4.21	+4.69	−1.25	0	+0.073
	usr8	44.49 → 62.85	52.17 → 69.57	82.02 → 84.27	31 → 28	1.356 → 1.273
		+18.36	+17.40	+2.25	−3	−0.083
	usr9	54.77 → 64.40	58.14 → 72.09	92.13 → 82.02	40 → 38	1.753 → 1.487
		+9.63	+13.95	−10.11	−2	−0.266

Horizon	User	FAMA	MPA	FAA	Search+Get ↓	Answer tokens (M) ↓
Quarterly	usr0	53.17 → 59.39	57.55 → 63.22	87.76 → 89.28	175 → 168	6.926 → 6.882
		+6.22	+5.67	+1.52	−7	−0.044
	usr1	60.50 → 62.29	64.10 → 65.64	89.89 → 90.61	313 → 293	20.713 → 20.016
		+1.79	+1.54	+0.72	−20	−0.697
	usr2	54.39 → 56.64	62.63 → 60.00	77.02 → 90.64	133 → 113	5.949 → 5.036
		+2.25	−2.63	+13.62	−20	−0.913
	usr3	52.05 → 58.36	54.55 → 61.36	91.57 → 89.89	274 → 228	18.326 → 13.095
		+6.31	+6.81	−1.68	−46	−5.231
	usr4	52.93 → 48.30	56.36 → 51.10	89.91 → 91.76	164 → 144	7.960 → 6.802
		−4.63	−5.26	+1.85	−20	−1.158
	usr5	41.85 → 44.80	43.50 → 46.00	95.24 → 96.54	143 → 165	7.862 → 9.110
		+2.95	+2.50	+1.30	+22	+1.248
	usr6	46.45 → 53.86	50.84 → 58.10	88.26 → 88.66	123 → 127	6.066 → 6.084
		+7.41	+7.26	+0.40	+4	+0.019
	usr7	65.03 → 62.78	67.80 → 66.34	92.08 → 89.81	122 → 129	6.277 → 5.952
		−2.25	−1.46	−2.27	+7	−0.325
	usr8	44.25 → 50.27	47.62 → 52.91	90.87 → 92.86	229 → 199	20.790 → 16.749
		+6.02	+5.29	+1.99	−30	−4.041
	usr9	43.67 → 36.16	47.85 → 39.26	89.77 → 92.42	129 → 119	5.650 → 5.590
		−7.51	−8.59	+2.65	−10	−0.060

Table 14: Answer quality and reading overhead on the Cortex effective-hit subset. Arrows indicate Engram-only → Cortex-guided; Search+Get is the total number of search and page-read calls, and answer tokens are the total tokens used during answering for the same subset of questions.

Horizon	#Q	FAMA	Search+Get ↓	Answer tokens (M) ↓
Monthly	33	50.76 → 67.41	85 → 81	3.212 → 3.008
Quarterly	55	47.95 → 56.13	217 → 191	16.121 → 11.952

Results analysis. Table 13 shows that eight of ten users on Monthly and seven of ten on Quarterly achieve higher FAMA under Cortex-guided reading, indicating that gains extend to most users rather than arising only from improvements for a few individuals. Efficiency gains are more pronounced over the longer temporal horizon: total Search+Get calls decrease from 390 to 370 on Monthly and from 1,805 to 1,685 on Quarterly, reductions of approximately 5.1% and 6.6%, respectively. The reduction in answer tokens also increases from approximately 3.4% to 10.5%, indicating that Cortex-guided reading can reduce reading and answering overhead more substantially over longer interaction histories. Table 14 further shows that, on the effective-hit subset where Cortex-guided reading is actually completed, FAMA improves by 16.65 percentage points on Monthly and 8.18 percentage points on Quarterly, both exceeding the average gains on the corresponding complete question sets. On the same effective-hit subsets, answer tokens decrease from 3.212 to 3.008 million on Monthly and from 16.121 to 11.952 million on Quarterly, reductions of approximately 6.4% and 25.9%, respectively. These results indicate that Cortex-guided reading improves answer quality for most users. The larger gains on effective-hit questions suggest that improvements are more pronounced when Cortex successfully guides the model to retrieve supporting Engram evidence, with fewer retrieval calls and lower answer-token usage.

F Comparison with Enterprise Memory Systems

CERES has been deployed in long-term interaction settings within an enterprise. To evaluate its end-to-end performance relative to existing enterprise memory systems, we further conduct a system-level comparison with OpenViking (OpenViking Team, 2026) and TencentDB Agent Memory (TencentDB Agent Memory Team, 2026) on public benchmarks. OpenViking organizes memory, resources, and skills through a virtual filesystem and supports hierarchical context retrieval. TencentDB Agent Memory organizes conversational experience into layered long-term memory for reuse across interactions.

Implementation and evaluation settings. OpenViking and TencentDB Agent Memory use their official implementations and recommended configurations. Each method retains its own memory organization, storage, and retrieval mechanisms. Across all methods, the answer model, evaluation-set construction and sampling procedure, and Judge settings follow the main experiments and are detailed in Appendix C.2; the shared prompting rules for benchmark answering are shown in Figure 4. OpenViking uses server 0.4.12 and the official OpenClaw plugin 2026.6.18, with plugin source tag v0.4.7 (commit `d306eaee46fc`). TencentDB Agent Memory uses the official `memory-tencentdb-v2` package 1.0.0-beta.1, whose plugin version is 2.0.0-beta.1.

Table 15: End-to-end quality of CERES and enterprise memory systems on two public benchmarks. Memora reports FAMA across the Weekly, Monthly, and Quarterly horizons. LongMemEval-S (cleaned) reports Answer Judge Accuracy, Target Fact Recall (TFR), and Temporal Resolution Accuracy (TRA) on 60 QA pairs.

Method	Memora			LongMemEval-S (cleaned)		
	W FAMA	M FAMA	Q FAMA	Judge	TFR	TRA
OpenViking	27.09	23.31	20.93	61.7	64.33	45.0
TencentDB Agent Memory	49.73	42.21	32.58	65.0	63.73	40.0
CERES	**77.41**	**71.57**	**61.69**	**76.7**	**82.33**	**80.0**

On Memora, CERES achieves higher FAMA across all three temporal horizons. On Weekly, CERES reaches 77.41, compared with 27.09 for OpenViking and 49.73 for TencentDB Agent Memory; on Monthly and Quarterly, CERES reaches 71.57 and 61.69, respectively. FAMA declines for all three systems as the temporal horizon grows, suggesting that longer interaction histories continue to challenge memory maintenance and question answering. On LongMemEval-S, CERES also outperforms both comparison systems in answer accuracy, target fact recall, and temporal resolution accuracy. The larger gap in TRA indicates that CERES more accurately identifies the currently valid state of facts in evolving interaction histories, reducing interference from outdated information during answering.