OMEGA: A UTILITY-AWARE LIFECYCLE CO-DESIGN FOR LONG-TERM MEMORY IN LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term memory systems for LLM agents face a fundamental tension: rich extraction and retrieval improve evidence coverage but incur prohibitive lifecycle costs. Controlled analyses on LoCoMo and LongMemEval reveal three design principles: (1) preserve complementary evidence without proliferating memory types; (2) decouple concise matching keys from coherent answer-bearing payloads; (3) expand retrieval coverage before folding evidence into a bounded solver context. We instantiate these principles in Omega[1], which combines session-aligned typed projection, key-payload separated maintenance, and bounded multi-route retrieval with late parent folding. Across three benchmarks, Omega achieves state-of-the-art performance, leading the strongest baseline by an average of 3.59 points while using 25–86% less solver context and cutting write-side calls by 81–88% on LoCoMo. Notably, Omega transfers to the held-out PersonaMem-128K, demonstrating robust cross-benchmark generalization.

1 INTRODUCTION

LLM agents can reason, use tools, navigate interactive environments, and modify software (Yao et al., 2023; Schick et al., 2023; Yang et al., 2024), yet competence within one trajectory does not ensure continuity across sessions. Without persistent memory, assistants lose preferences, decisions, and unfinished work, producing repetition and inconsistent behavior (Wu et al., 2025; Jiang et al., 2025). Consequently, robust long-term memory has emerged as a critical frontier for extending agent capabilities by selectively retaining and reusing prior interactions to sustain coherent behavior across sessions (Park et al., 2023; Gutiérrez et al., 2024; Rasmussen et al., 2025).

A memory system spans three lifecycle stages: construction, which projects interactions into persistent units; maintenance, which organizes those units; and retrieval, which selects solver evidence. To maximize coverage, recent systems heavily enrich these stages with learned boundaries, multiple schemas, and hierarchical stores (Jiang et al., 2026; Hu et al., 2026b;a; Yue et al., 2026). However, this escalating complexity introduces a fundamental tension: it often incurs prohibitive write costs, retrieval competition, and solver distraction that offset recall gains. We therefore ask not whether a component is individually novel, but *under what conditions does added lifecycle complexity provide sufficient marginal utility to be retained?*

To answer this question, we derive three testable design principles for memory systems. **P1: Marginal-utility construction.** Under a fixed retrieval budget, finer boundaries or additional schemas are warranted only when their added evidence offsets write costs and pool competition; richer projection alone does not guarantee higher accuracy. **P2: Decoupling matching keys from answer payloads.** When coherent episodic evidence is longer than its discriminative cues, concise keys should index the memory without replacing its payload, improving access while avoiding unnecessary expansion of the solver context. **P3: Coverage-aware compression.** Diverse recall routes should maximize evidence coverage while compression consolidates overlapping or irrelevant content, preserving complementary evidence within a bounded solver context.

Omega operationalizes these principles through session-aligned episodic–fact projection (P1), key–payload separated maintenance (P2), and bounded multi-route retrieval with late folding (P3). For evaluation, we develop and tune Omega exclusively on LoCoMo and LongMemEval-oracle, then

[1]Our code is publicly available at `https://anonymous.4open.science/r/Omega-5113`.

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.

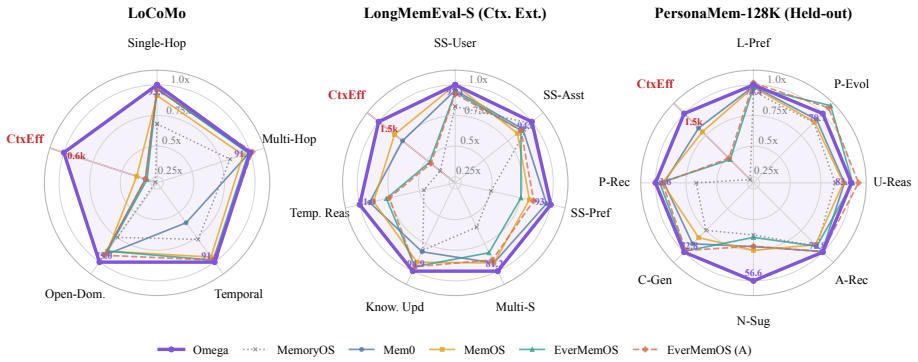


Figure 1: **Category-level accuracy and inference-time context efficiency.** Omega is the 1.0× reference; CtxEff denotes relative context efficiency, computed as normalized inverse context length. EverMemOS uses default retrieval, whereas EverMemOS (A) uses agentic search. *Ctx. Ext.* marks an oracle-to-long-context extension; *Held-out* identifies the benchmark-family holdout. PersonaMem uses Omega's core question-only setting.

freeze all non-budget components for evaluation on length-extended LongMemEval-S and the held-out PersonaMem-128K. Under this evaluation protocol, Figure 1 summarizes the results: Omega matches or surpasses the strongest evaluated baseline across all benchmarks while delivering superior context efficiency. Remarkably, it achieves this with extreme write-side efficiency (81–88% fewer calls and 84–91% fewer input tokens on LoCoMo) and consistently less solver context, demonstrating that robust cross-domain generalization need not incur prohibitive lifecycle costs.

Our contributions are threefold: (1) **Principles:** We formulate three stage-specific design principles that clarify when memory complexity is beneficial and how to empirically test it. (2) **System:** We introduce Omega, a compact instantiation of these principles, supported by controlled diagnostics that isolate the marginal utility of each component. (3) **Generalization:** We provide robust evidence of cross-domain transfer, demonstrating that Omega's favorable accuracy–cost trade-offs persist on context-extended (LongMemEval-S) and strictly held-out (PersonaMem-128K) benchmarks.

2 Related Work

2.1 Long-Term Memory Systems

Long-term memory has evolved from retaining interaction traces to structuring and actively managing persistent state across sessions. Early systems store observations, summaries, or reflections (Park et al., 2023; Packer et al., 2023), while recent systems extend this capability across construction, maintenance, and retrieval. At construction, MIRIX organizes experience into six specialized memory types (Wang & Chen, 2025). Maintenance mechanisms range from time- and importance-sensitive forgetting and reinforcement in MemoryBank (Zhong et al., 2024) to prediction-gap-driven updates in Nemori (Ma et al., 2026). At retrieval, EverMemOS uses multi-round agentic search to assemble task-relevant context (Hu et al., 2026a). However, when these mechanisms are evaluated as coupled pipelines, aggregate gains make it difficult to disentangle the contribution of individual components and assess whether their accuracy benefits justify the associated lifecycle costs. We address this gap through controlled analyses of memory construction, maintenance, and retrieval.

2.2 Memory System Evaluation

Recent studies evaluate memory systems beyond task accuracy. Hu et al. (2026b) compare efficiency–performance trade-offs across memory frameworks by relating F1 to construction time, processing time, per-question latency, and token consumption. Omri et al. (2026) profile 10 systems under long-horizon workloads across construction, retrieval, and generation, emphasizing latency, energy, storage growth, and freshness. Zhou et al. (2026) take a data-management perspective, decomposing agent memory into four modules: representation and storage, extraction, retrieval and

routing, and maintenance. Together, these studies characterize system-level trade-offs through profiling and modular analysis. We instead use controlled, stage-specific interventions to isolate how individual memory choices affect cost and downstream behavior: when construction views warrant their marginal cost, when matching and answering benefit from different representations, and when recall expansion should precede context compression. We trace these effects to write cost, retrieval composition, and solver context, and evaluate the resulting design principles jointly in Omega.

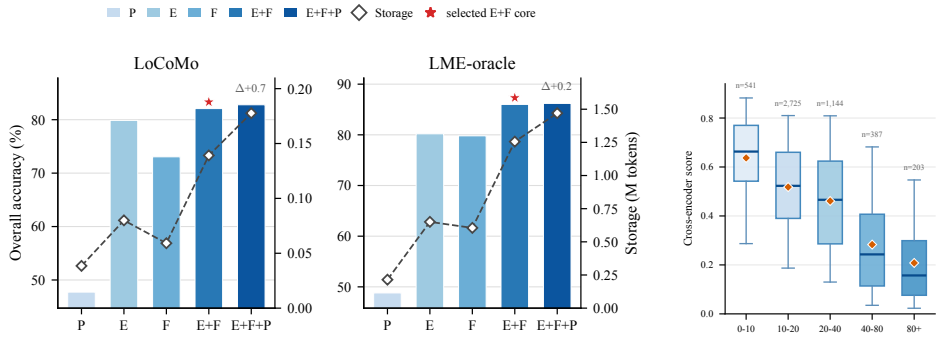


Figure 2: **Preliminary analyses of long-term memory systems.** The diagnostics use the LoCoMo and LongMemEval development settings. **Left:** downstream accuracy and total stored tokens across memory-type combinations; P/E/F denote profile, episodic, and fact, and stars mark the selected E+F configuration. **Right:** cross-encoder reranking scores over 5,000 query and memory item pairs grouped by memory length. Boxes span the first and third quartiles, lines mark medians, whiskers show p5–p95, and diamonds mark means.

3 Methodology

3.1 Preliminaries: Objectives and Pipeline

Given a chronological interaction history $\mathcal{H} = (h_1, \ldots, h_N)$, a long-term memory system aims to preserve useful information and retrieve relevant evidence for downstream tasks. We model this process as three lifecycle stages: construction builds an initial memory pool $\mathcal{P}^0$, maintenance organizes it into a retrieval-ready store $\mathcal{P}^*$, and retrieval selects a bounded context $\mathcal{C}_q$ for the solver A:

$$\mathcal{H} \xrightarrow{W} \mathcal{P}^0 \xrightarrow{M} \mathcal{P}^*, \qquad \mathcal{C}_q = R(q, \mathcal{P}^*; k), \qquad \hat{y} = A(q, \mathcal{C}_q), \tag{1}$$

where W, M, and R denote construction, maintenance, and retrieval operation, k bounds the number of items rendered to the solver, and $\hat{y}$ denotes the output. This pipeline balances two complementary objectives. *Information fidelity* requires preserving sufficient answer-bearing evidence while controlling write and storage costs. *Evidence access* requires making relevant information identifiable and selectively renderable within the solver's context budget. Maintenance exposes retrieval-oriented representations, while retrieval determines which evidence is ultimately presented to the solver.

Memory systems typically improve information fidelity by applying LLM projections to the same conversation, yielding complementary views such as profiles for persistent user state, episodes for coherent temporal context, and facts for atomic propositions. Evidence access, in turn, relies on embedding-based recall followed by cross-encoder reranking. However, these strategies are not naturally aligned: richer construction increases write and storage costs while intensifying retrieval competition, and answer-bearing representations need not provide effective matching surfaces. Both objectives must therefore be optimized jointly under a bounded lifecycle budget.

Figure 2 quantifies this tension under a fixed, lightweight retrieval pipeline. The left panel demonstrates that episodes and facts form a compact, complementary schema: adding profiles yields marginal accuracy fluctuations (at most 0.71 points) while inflating storage by 17–28%. Concurrently, the right panel illustrates that cross-encoder relevance decays with item length, indicating that query-specific signals are diluted in verbose, multi-event memories. These observations crystallize into two core principles: construction should preserve complementary evidence without proliferating types, while maintenance and retrieval must expose concise matching surfaces without

sacrificing coherent answer-bearing content. Together, these principles guide Omega's lifecycle co-design, detailed in Sections 3.2–3.4.

3.2 WRITE SIDE: TYPED MEMORY PROJECTION

Omega begins by establishing session-aligned extraction units. Given the chronological history $\mathcal{H}$, it segments consecutive interactions into natural conversation sessions:

$$\mathcal{U} = (u_1, \ldots, u_n), \qquad u_j = (h_{a_j}, \ldots, h_{b_j}), \tag{2}$$

where $\mathcal{U}$ denotes the ordered sequence of n extraction units, u_j represents the j-th session, and a_j, b_j index its first and last interactions. This partition preserves the local context necessary to resolve coreference, causal links, and temporal relations, while strictly bounding the projection input length. Guided by our preliminary analyses, we restrict the schema to a compact, complementary type set $\mathcal{T}_\Omega = \{\text{EPISODIC}, \text{FACT}\}$. For each type $t \in \mathcal{T}_\Omega$, a frozen LLM projector W employs a type-specific template τ_t to map session u_j into typed memory items:

$$W(u_j; \tau_t) = \{m^t_{j\ell}\}_{\ell=1}^{k_{jt}}, \qquad \mathcal{P}^0 = \bigcup_{j=1}^{n} \bigcup_{t \in \mathcal{T}_\Omega} W(u_j; \tau_t). \tag{3}$$

where $m^t_{j\ell}$ is the ℓ-th item of type t from u_j, k_{jt} is the item count, and $\mathcal{P}^0$ constitutes the initial memory pool. Crucially, separate type-specific calls prevent output budget competition, ensuring both verbose episodes and concise facts are fully expanded without mutual suppression.

Projection follows two constraints. **Coherence–Atomicity:** The episodic projector preserves events, plans, and outcomes in a self-contained narrative, whereas the fact projector extracts each independently verifiable proposition discretely. **Temporal Grounding:** Relative times are resolved against source timestamps; normalized dates are embedded in payloads and, where supported, temporal metadata. For traceability, each item records its type, source pointers, and available salient entities. Representative outputs and prompt templates appear in Appendix Table 10 and Appendix D.

3.3 MAINTENANCE SIDE: KEY–PAYLOAD SEPARATION

Given the initial pool $\mathcal{P}^0$ comprising coherent multi-event episodes and concise atomic facts, the maintenance stage exposes retrieval-oriented representations without discarding answer-bearing context. Episodes pose a particular challenge, as their multiple semantic foci tend to dilute query-specific signals during both first-stage recall and cross-encoder reranking (Figure 2).

To resolve this mismatch, Omega transforms $\mathcal{P}^0$ into a retrieval-ready store $\mathcal{P}^*$ of items $m_i = (p_i, K_i)$, pairing each payload p_i with retrieval keys K_i. While a fact exposes only its atomic payload, an episode retains its coherent payload and, when sufficiently long, invokes a lightweight LLM G_{child} to derive concise child keys, filtered against same-history facts before entering K_i:

$$K_i = \begin{cases} \{p_i\} & t_i = \text{FACT}, \\ \{p_i\} & t_i = \text{EPISODIC}, \ |w(p_i)| < 40, \\ \{p_i\} \cup D_\theta(G_{\text{child}}(p_i), \mathcal{F}_i) & t_i = \text{EPISODIC}, \ |w(p_i)| \geq 40, \end{cases} \tag{4}$$

where t_i denotes the memory type of m_i, $\mathcal{F}_i$ is the fact payloads from the same history, and $w(x)$ the content-word set of x; the gate $|w(p_i)| \geq 40$ skips child generation for already-concise episodes, bounding key-generation cost. The directional filter D_θ prunes a candidate child c if any fact $f \in \mathcal{F}_i$ covers at least a fraction θ of its content words, $\text{cov}(c, f) = |w(c) \cap w(f)|/|w(c)|$, with $\theta = 0.6$. This admission rule leaves payloads unchanged, retaining each surviving child as a retrieval key linked to its coherent parent rather than a standalone payload. Consequently, K_i exposes both original and concise matching surfaces to the retriever, while only p_i is rendered into the solver context $\mathcal{C}_q$.

3.4 RETRIEVAL SIDE: MULTI-ROUTE RETRIEVAL

Given the maintained pool $\mathcal{P}^*$, the retrieval strategy R exploits the retrieval keys exposed by M to allocate the solver's limited context budget to distinct, coherent payloads. For an open-ended query q, Omega employs the original question and a retrieval-oriented rewrite as complementary routes:

$$\mathcal{Q}(q) = \{q, G_{\text{rw}}(q)\}, \qquad \mathcal{E} = \bigcup_{m_i \in \mathcal{P}^*} \{(v, i) \mid v \in K_i\}, \tag{5}$$

where G_{rw} leverages the lightweight Qwen3.5-flash model to generate a self-contained search query, and $\mathcal{E}$ indexes each matching surface v with its parent payload identifier i. For each route $q^{(r)} \in \mathcal{Q}(q)$, lexical and dense recall, followed by key-level reranking, are formalized as:

$$\begin{aligned} L^{(r)} &= \operatorname*{Top}_{c}_{(v,i)\in\mathcal{E}} \mathrm{BM25}(q^{(r)}, v), \quad D^{(r)} = \operatorname*{Top}_{c}_{(v,i)\in\mathcal{E}} \cos\Big(f(q^{(r)}), f(v)\Big), \\ \widetilde{C}^{(r)} &= \mathrm{RRF}_{\mathrm{ch}}\big(L^{(r)}, D^{(r)}\big), \quad C^{(r)} = \mathrm{Rerank}\Big(q^{(r)}, \widetilde{C}^{(r)}\Big). \end{aligned} \tag{6}$$

Here, f denotes the embedding encoder, c is the per-channel candidate depth, and $\mathrm{RRF}_{\mathrm{ch}}$ performs standard channel-level reciprocal-rank fusion. The cross-encoder then reranks the recalled matching surfaces, preserving both lexical and semantic coverage. Omega subsequently applies late parent folding and route-level fusion:

$$\begin{aligned} \Pi^{(r)} &= \mathrm{LateFold}\big(C^{(r)}\big), \quad \Pi_q = \mathrm{RRF}_{\mathrm{rt}}\big(\{\Pi^{(r)}\}_r\big), \\ I_q &= \mathrm{TopK}_k(\Pi_q), \qquad\qquad \mathcal{C}_q = (p_i)_{i\in I_q}. \end{aligned} \tag{7}$$

$\mathrm{LateFold}$ retains each parent payload exactly once, positioned at the highest rank attained by any of its child keys. Performing this folding before $\mathrm{RRF}_{\mathrm{rt}}$ prevents a single memory from casting duplicate votes, while route fusion rewards evidence recovered across multiple query formulations. Finally, Omega constructs the solver context $\mathcal{C}_q$ by rendering the selected payloads with timestamps. The child keys and the rewritten query are discarded as retrieval-specific artifacts, ensuring the solver focuses solely on the original question q and the retrieved evidence. Implementation details, including candidate depths and fusion constants, are provided in Appendix C.

4 Results and Analysis

4.1 Experimental Setup

Benchmarks. We evaluate our approach on three representative long-term memory benchmarks: **LoCoMo** (Maharana et al., 2024), **LongMemEval** (Wu et al., 2025), and **PersonaMem-v1** (Jiang et al., 2025). Departing from prior practices that conflate development and evaluation **phases**, we enforce a clear procedural decoupling to mitigate over-optimization. Specifically, we conduct configuration tuning exclusively on the short-context LoCoMo and LongMemEval-oracle subset, and freeze all non-budget components before final evaluation on LongMemEval-S and PersonaMem-128K. The former extends the oracle setting by introducing a significantly longer, distractor-heavy context, while the latter serves as a strictly held-out benchmark completely isolated from method development. Detailed statistics and the full evaluation protocol are provided in Appendix A.

Baselines. We compare Omega with five prominent memory systems spanning research prototypes and production-oriented platforms: Mem0 (Chhikara et al., 2025), MemOS (Li et al., 2025), MemoryOS (Kang et al., 2025), HyperMem (Yue et al., 2026), and EverMemOS (Hu et al., 2026a). For EverMemOS, we evaluate both its default hybrid retrieval (EverMemOS) and agentic search (EverMemOS (A)) variants. Method-specific descriptions are provided in Appendix B.

Implementation and Evaluation. To ensure a controlled comparison, we standardize the downstream solver and evaluator across all systems. We rerun all baselines using Qwen3.6-Plus under identical settings, while aligning GPT-based results with official reports. Crucially, Omega's core architecture and all non-budget components remain frozen after short-context development. For the primary question-only evaluation on the held-out PersonaMem-128K, we preserve this setup, scaling only the final retrieval budget to top-40 to accommodate the larger context. Full configurations, evaluation metrics and prompts are provided in Appendices C and D.

4.2 Main Results Across Benchmarks

Primary Benchmarks Results. Table 1 and Table 2 report results on LoCoMo and LongMemEval-S. Under the Qwen-based setting, Omega achieves the highest overall LoCoMo score while using only $\sim$0.6K solver-context tokens, just one-third to one-seventh of those used by competing systems. The same trend appears in judge-independent fine-grained metrics, with F1 and BLEU-1 both improving by approximately 2 points. On the more challenging, distractor-heavy LongMemEval-S, Omega achieves a statistically significant 8.00-point lead over the strongest evaluated baseline, with

Method	Avg. Tok.	Single-Hop	Multi-Hop	Temporal	Open-Dom.	Overall (A/F1/B1)
GPT-4.1-mini backbone + GPT-4o-mini judge						
MemoryOS†	5.5k	67.30	59.34	42.26	59.03	60.11
Mem0†	1.0k	68.97	61.70	58.26	50.00	64.20
MemOS†	2.5k	85.37	79.43	75.08	64.58	80.76
HyperMem‡	7.5k	96.08	**93.62**	89.72	70.83	92.73
EverMemOS†	2.3k	**96.67**	91.84	89.72	76.04	93.05
Omega	0.7k	94.41	**93.62**	**93.46**	**79.17**	**93.12**
Qwen3.6-plus backbone + judge						
MemoryOS	2.8k	63.38	75.89	70.09	56.25	66.62/37.43/32.48
Mem0	2.0k	89.89	91.13	54.83	69.79	81.56/44.86/38.85
MemOS	1.6k	84.90	86.17	86.29	66.67	84.29/45.40/39.05
HyperMem	4.3k	**93.94**	89.36	87.85	72.92	90.52/50.08/43.45
EverMemOS	2.1k	90.01	90.43	89.10	66.67	88.44/47.83/41.65
EverMemOS (A)	2.0k	89.66	**93.62**	89.10	69.79	89.03/49.03/42.73
Omega	0.6k	92.98	91.84	**90.97**	**75.00**	**91.23/51.83/45.46**

Table 1: LoCoMo results. Results marked † and ‡ are taken from EverMemOS (Hu et al., 2026a) and HyperMem (Yue et al., 2026), respectively. Best results within each setting are bold.

Method	Avg. Tok.	SS-User	SS-Asst.	SS-Pref.	Multi-S	K-Upd.	T-Reas.	Overall
GPT-4.1-mini backbone + GPT-4o-mini judge								
Mem0†	1.1k	82.86	26.78	90.00	63.15	66.67	72.18	66.40
MemOS†	1.4k	95.71	67.86	**96.67**	70.67	74.26	77.44	77.80
EverMemOS†	2.8k	**97.14**	85.71	93.33	73.68	89.74	77.44	83.00
Omega	0.9k	**97.14**	**94.64**	90.00	**79.70**	**92.31**	**86.47**	**88.20**
Qwen3.6-plus backbone + judge								
MemoryOS	4.2k	80.00	92.86	46.67	48.87	76.92	42.11	60.60
Mem0	2.0k	92.86	85.71	90.00	75.19	78.21	83.46	82.40
MemOS	1.8k	**97.14**	80.36	76.67	75.19	87.18	80.45	82.20
EverMemOS	3.2k	91.43	83.93	70.00	67.67	91.03	70.68	77.40
EverMemOS (A)	3.3k	90.00	83.93	80.00	72.93	91.03	68.42	78.60
Omega	1.5k	**97.14**	**94.64**	**93.33**	**81.20**	**94.87**	**90.98**	**90.40**

Table 2: LongMemEval-S results. Results marked † are taken from EverMemOS (Hu et al., 2026a). HyperMem is omitted from the Qwen-based setting due to prohibitive memory-construction time and cost. Best results within each setting are bold.

the paired 95% CI strictly above zero. Write-side efficiency is equally compelling: relative to the matched multi-stage baselines on LoCoMo, Omega reduces write-side LLM calls by 81–88% and input tokens by 84–91%. These advantages extend to the GPT-based setting, where Omega leads LongMemEval-S by 5.20 points. Complete significance analysis, evaluator sensitivity results, and cost breakdowns are provided in Appendices C and E.4.

Strict Holdout Evaluation on PersonaMem-128K. Table 3 reports results on PersonaMem-128K, which was entirely excluded from Omega's method development. Under the conservative question-only setting, which withholds answer choices from the retriever, Omega achieves 71.43, surpassing the strongest baseline EverMemOS (A) (69.38) by 2.05 points with less than half the solver context. This held-out gain is statistically significant, with the paired 95% CI strictly above zero ($[+0.24, +3.95]$). When option text is available, the optional LLM-free choice-contrast route G_{ccr}, which leverages option-discriminative lexical terms as retrieval signals, further improves the score to 71.76 while reducing context to 1.4K tokens. Together, these results confirm that Omega's core gain does not rely on exposing answer choices during retrieval, while demonstrating that retrieval routing can be enriched beyond the query itself, as structured interface information serves as an effective complementary signal for locating relevant memories.

Method	Avg. Tok.	L-Pref	P-Evol	U-Reas	A-Rec	N-Sug	C-Gen	P-Rec	Overall
MemoryOS	6.3k	70.44	63.34	72.49	66.08	35.33	54.46	55.87	59.70
Mem0	1.8k	73.44	65.69	78.81	70.20	40.73	65.73	77.78	66.04
MemOS	1.9k	72.86	63.93	78.07	64.76	42.47	60.56	78.95	64.87
EverMemOS	3.2k	72.86	**77.13**	84.39	65.90	36.49	71.83	80.70	67.14
EverMemOS (A)	3.1k	**75.52**	75.37	**88.48**	70.49	40.54	71.36	78.95	69.38
Omega ($+G_{\mathrm{rw}}$)	1.5k	74.36	70.67	83.64	70.77	56.56	**72.77**	**83.63**	71.43
Omega ($+G_{\mathrm{ccr}}$)	1.4k	74.25	70.38	85.87	**71.06**	**57.92**	**72.77**	81.87	**71.76**

Table 3: PersonaMem-128K held-out results. $+G_{\mathrm{rw}}$ denotes the primary query-rewriting setting without answer-choice exposure, $+G_{\mathrm{ccr}}$ replaces $+G_{\mathrm{rw}}$ with the LLM-free choice-contrast route derived from all options. HyperMem is omitted for prohibitive construction cost; best in bold.

Cross-benchmark Summary. Across all evaluations spanning Qwen- and GPT-based settings, Omega achieves top-tier performance with high efficiency. Under the primary Qwen-based setting, it matches leading systems on LoCoMo with substantially lower solver context and write-side computation; on LongMemEval-S and PersonaMem-128K, the paired 95% CIs exclude zero, confirming statistically significant gains. These results verify that Omega's architectural design yields a favorable accuracy–efficiency trade-off across diverse long-term memory benchmarks.

4.3 Write-Side Strategy Analysis

Building on the preliminary finding that an episodic–fact pair offers a compact and complementary representation (Section 3.2), we systematically evaluate extraction boundaries and projection regimes on LoCoMo and LongMemEval-oracle. To isolate write-side effects, maintenance and retrieval are held fixed in a simplified configuration. Appendix E reports the full settings and results.

Q1: Which extraction boundary best balances context and cost? We compare natural session boundaries against fixed-size block granularities to isolate boundary effects. As visualized in Figure 3 (left), the finest blocks (Block-S) require 2.2–3.0× the write-side LLM calls of Session and inflate the memory pool by 41–71%, yet improve accuracy by only +0.4 to +0.78 points. Intermediate granularities fare worse still: Block-M is strictly dominated by Session on both benchmarks, delivering lower accuracy at higher call cost. Even at its closest, Session remains within 1.0 point of the Block-S peak while using only one-third to one-half of its calls. This pronounced cost–accuracy divergence shows that finer semantic segmentation yields diminishing returns. We therefore adopt session boundaries as a parameter-free, cost-aware default that avoids excessive fragmentation.

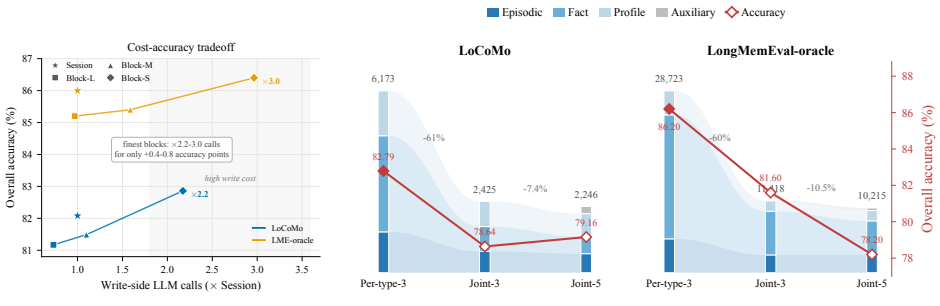


Figure 3: **Write-side strategy diagnostics. Left:** performance versus relative write-side LLM calls across extraction boundaries under Per-type E+F. Block-S/M/L denote fine, medium, and coarse fixed-size blocks connected by solid lines, while stars mark the Session boundary. **Right:** accuracy and pool composition across projection regimes. Bars and bands show composition normalized to Per-type-3; labels give absolute totals, red lines overall accuracy, and diamonds the best regime.

Q2: Can one LLM call effectively extract multiple memory types? To expose competition within joint extraction, we use profile, episodic, and fact as a three-type set and introduce summary and foresight as two additional types. With extraction units fixed, Joint-3 shares one instruction and

output budget across the three-type set, Joint-5 also requests the two additional types, and Per-type-3 assigns each of the original three types a dedicated call. Figure 3 (right) shows that Joint-5 produces fewer cells than Joint-3 despite requesting more types, and only about 40% as many as Per-type-3. Crucially, dedicated calls allow core episodic and fact memories to be fully expanded, recovering fine-grained evidence that is suppressed by type competition in a shared prompt. This leads to accuracy gains of 4.15 and 4.60 points on LoCoMo and LME-oracle. This shows that separate calls recover evidence suppressed by a shared output budget, but at a substantial pool-growth cost. Omega limits that growth by projecting only the complementary episodic–fact pair.

Overall, the write-side results support Omega's session-aligned, per-type projection of a compact episodic–fact set: session boundaries provide a robust cost–accuracy trade-off, dedicated calls prevent type competition, and the compact type set limits pool growth.

4.4 Maintenance-Side Analysis

Motivated by the length disparity observed in Section 3.3, we further evaluate on LoCoMo and LongMemEval-S whether episode-derived children serve more effectively as payloads or retrieval keys, and whether these keys improve the retrieval of their parent episodes.

Q1: Should child statements be stored as payloads or retrieval keys? We evaluate variants positioning children as standalone payloads, parent-attached keys, or both, alongside ablations on parent retention and child deduplication. Overall, Table 4 shows that the deduplicated key-only design outperforms all other variants across benchmarks, yielding an average improvement of 2.84 points over the baseline. Payload-only variants are substantially weaker, especially when the parent is retained, as verbose parent episodes dominate the payload space and render children ineffective during retrieval. This confirms that children function best as *retrieval cues* guiding the model to coherent parent episodes, rather than as standalone evidence. Additionally, deduplication prevents overlapping statements from consuming multiple solver slots, ensuring diverse evidence coverage.

Child placement	LoCoMo	LME-S
Without children	89.55	86.40
Payload + key	90.32	88.80
Payload only: parent kept	85.06	86.40
Payload only: parent removed	88.70	86.80
Key only: all children	90.65	89.60
Key only: deduplicated (ours)	**91.23**	**90.40**

Table 4: Ablation on child-placement strategy.

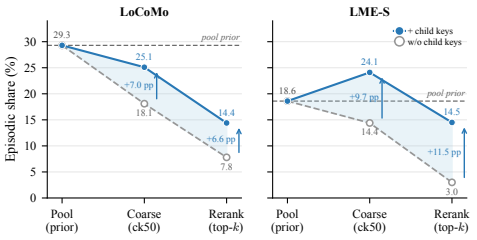


Figure 4: Episodic share by retrieval stage.

Q2: Do child keys improve episode inclusion during retrieval? Figure 4 illustrates the proportion of episodic memories in the full pool, after embedding-based coarse recall, and after reranking, both with and without child keys. Without child keys, episodes are underrepresented after both retrieval stages relative to their proportion in the full pool, revealing a systematic matching disadvantage. Leveraging child keys substantially restores episode inclusion at both stages on both benchmarks; on LongMemEval-S, the reranked share rises from 3.0% to 14.5%, a 4.8× increase. Together with the length diagnostic, this recovery shows that concise child keys improve the matchability of long episodes while preserving their parent payloads as the evidence shown to the solver.

These findings validate Omega's key-payload decoupling strategy: child keys maximize retrievability, while parent-only payloads ensure coherent, non-redundant evidence for the solver. Qualitative traces illustrating this mechanism are provided in Appendix H.2.

4.5 Retrieval-Side Analysis

Building on the established memory representations, the retrieval stack bridges the extensive memory pool and compact solver context. We dissect Omega's retrieval design along two questions: (1) the marginal contribution of each component to downstream accuracy, and (2) the source of residual errors: whether they stem from retrieval failures or representation deficiencies.

Q1: How does each retrieval component affect downstream accuracy? Holding the memory pool and solver fixed, we conduct one-factor ablations on LongMemEval-S across four key dimensions: multi-route recall, key-level reranking, route-level reciprocal-rank fusion (RRF), and payload budget. Specifically, for the fusion ablation, we replace RRF with a naive equal-quota strategy (deduplication followed by round-robin backfilling). As shown in Table 5, each ablation degrades performance by 1.20–5.60 points. Removing reranking entirely causes the largest drop, while reranking parent payloads instead of matched keys incurs a 3.00-point loss, supporting our design of ranking concise keys prior to late fusion. Disabling multi-route recall or replacing RRF with equal quotas also yields consistent drops, confirming the necessity of diverse retrieval paths and robust score fusion. Furthermore, the context budget exhibits a non-monotonic effect: top-20 achieves optimal performance, as excessive payloads introduce irrelevant information that distracts the solver.

Setting	Route	RR	k	Acc.	Δ
Main stack	MR	K	20	**90.40**	–
w/o multi-route	Single	K	20	89.00	−1.40
w/o reranking	MR	–	20	84.80	−5.60
Rerank: K→P	MR	P	20	87.40	−3.00
Route RRF→quota	MR	K	20	88.80	−1.60
Budget: 20→10	MR	K	10	87.80	−2.60
Budget: 20→40	MR	K	40	89.20	−1.20

Table 5: Retrieval ablation on LongMemEval-S. MR denotes multi-route retrieval; K/P denote key/payload reranking; k denotes top-k selection. Δ is relative to the main stack.

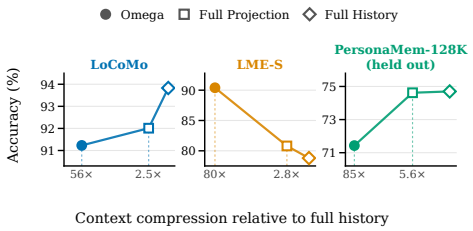


Figure 5: Performance and context length comparison among Omega (Ours), Full Projection, and Full History settings.

Q2: Do the remaining errors arise from retrieval or construction? To distinguish retrieval limitations from construction losses, we compare Omega with two counterfactual controls: Full Projection and Full History (Figure 5). Full Projection supplies all extracted memories directly to the solver, bypassing retrieval, whereas Full History supplies the raw conversation history, bypassing both projection and retrieval. Consequently, Omega achieves massive context efficiency, using 15–29× less context than Full Projection and 56–85× less than Full History. However, this aggressive compression exposes distinct bottlenecks across tasks. On LoCoMo, the performance gap between Full History and Full Projection exceeds that between Full Projection and Omega, suggesting that projection may discard some implicit cross-session information. On PersonaMem, Full Projection nearly matches Full History but exceeds Omega by 3.19 points, consistent with retrieval coverage remaining an important limitation. On LongMemEval-S, Omega substantially outperforms both counterfactual controls, suggesting that selective filtering can outperform exhaustive context in this distractor-heavy setting. Complete results and case studies are in Appendix G and H.1.

Overall, these analyses validate Omega's multi-route, key-reranked, and bounded retrieval design, while revealing that the remaining performance ceiling is task-dependent. This points toward adaptive, query-aware retrieval strategies that dynamically balance evidence coverage and distractor filtering as the next frontier for memory-augmented systems.

5 Conclusion

In this work, we formulate three principles for long-term memory: retain construction complexity only when marginal evidence utility offsets costs, decouple matching keys from answer payloads, and expand recall before compressing solver context. Omega instantiates these via session-aligned projection, key–payload maintenance, and bounded multi-route retrieval. Under a controlled evaluation with a shared evaluation environment and fixed components and method design, Omega matches baselines on LoCoMo with 81–88% fewer write calls, while significantly outperforming them on both the context-extended LongMemEval-S and the held-out PersonaMem-128K. Moreover, it uses 15–29× and 56–85× less context than full-projection and full-history baselines, respectively. Controlled diagnostics across the lifecycle validate the effectiveness of this co-design, confirming the marginal utility of each stage. Therefore, memory systems should not blindly eliminate complexity, but retain lifecycle overhead only when its measurable downstream utility justifies the cost.

AI USE STATEMENT

Generative AI tools assisted with translation, language editing, clarity and presentation feedback. They did not generate experimental data, formulate mathematical claims, design the method or experiments, implement the proposed system, or independently interpret results. The authors reviewed all AI-assisted text, verified the claims against the experimental results and literature, and take full responsibility for the final manuscript.

ETHICS STATEMENT

We use only publicly available benchmarks under their original licenses. Their dialogues are model-simulated, so the experiments involve neither human subjects nor real personal data. Deployment still poses risks: persistent memory can retain sensitive information, infer unwanted or inaccurate traits, and propagate stale or poisoned records. Systems should obtain informed consent, expose records and derived memories with provenance, and support inspection, correction, and deletion. Data minimization, scoped retention and access control, and defenses against leakage and poisoning remain essential but lie beyond this study.

LIMITATIONS

Our experiments remain limited to offline memory question answering over model-simulated English dialogues. LongMemEval-S tests context extension and PersonaMem-128K tests held-out-family transfer, but neither captures tool use, environment-state transitions, online task success, continual-write latency, or error propagation through extended interaction. Solver-context length also measures only one component of end-to-end cost. Future work should evaluate Omega in acting agents under realistic interaction streams, measuring both long-horizon task completion and continual update, retrieval, and inference costs.

REPRODUCIBILITY STATEMENT

The essential methodological and evaluation details required to reproduce the main results of Omega are documented in the paper and supplementary material. Specifically, Appendices A, B, and C document the benchmark and baseline configurations, retriever checkpoints, fusion and rendering budgets, solver–judge pairings, configuration-selection protocols, and evaluation metrics. Appendix D provides the prompt templates used for projection, child-key generation, and query rewriting. The supplementary code provides an executable implementation of Omega's primary pipeline, including memory construction, retrieval, evaluation, scoring, and the associated prompt files, and is mirrored in an anonymous repository (`https://anonymous.4open.science/r/Omega-5113`).

REFERENCES

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. URL https://arxiv.org/abs/2504.19413.

Bernal J Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in neural information processing systems*, 37:59532–59569, 2024. URL https://arxiv.org/abs/2405.14831.

Chuanrui Hu, Xingze Gao, Zuyi Zhou, Dannong Xu, Yi Bai, Xintong Li, Hui Zhang, Tong Li, Chong Zhang, Lidong Bing, and Yafeng Deng. EverMemOS: A self-organizing memory operating system for structured long-horizon reasoning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 45836–45853, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.2125. URL https://aclanthology.org/2026.acl-long.2125/.

Yuyang Hu, Jiongnan Liu, Jiejun Tan, Yutao Zhu, and Zhicheng Dou. Memory matters more: Event-centric memory as a logic map for agent searching and reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 22389–22407, San Diego, California, United States, July 2026b. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1123. URL https://aclanthology.org/2026.findings-acl.1123/.

Bowen Jiang, Zhuoqun Hao, Young-Min Cho, Bryan Li, Yuan Yuan, Sihao Chen, Lyle Ungar, Camillo J. Taylor, and Dan Roth. Know me, respond to me: Benchmarking LLMs for dynamic user profiling and personalized responses at scale. In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=6ox8XZGOqP.

Dongming Jiang, Yi Li, Guanpeng Li, and Bingzhe Li. MAGMA: A multi-graph based agentic memory architecture for AI agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 36848–36865, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1709. URL https://aclanthology.org/2026.acl-long.1709/.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI agent. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25961–25970, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1318. URL https://aclanthology.org/2025.emnlp-main.1318/.

Zhiyu Li, Chenyang Xi, Chunyu Li, Ding Chen, Boyu Chen, Shichao Song, Simin Niu, Hanyu Wang, Jiawei Yang, Chen Tang, Qingchen Yu, Jihao Zhao, Yezhaohui Wang, Peng Liu, Zehao Lin, Pengyuan Wang, Jiahao Huo, Tianyi Chen, Kai Chen, Kehang Li, Zhen Tao, Huayi Lai, Hao Wu, Bo Tang, Zhengren Wang, Zhaoxin Fan, Ningyu Zhang, Linfeng Zhang, Junchi Yan, Mingchuan Yang, Tong Xu, Wei Xu, Huajun Chen, Haofen Wang, Hongkang Yang, Wentao Zhang, Zhi-Qin John Xu, Siheng Chen, and Feiyu Xiong. MemOS: A memory OS for AI system. *arXiv preprint arXiv:2507.03724*, 2025. URL https://arxiv.org/abs/2507.03724.

Wenquan Ma, Jiayan Nan, and WenLong Wu. What deserves memory: Adaptive memory distillation for LLM agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 34789–34812, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. 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 Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

Yasmine Omri, Ziyu Gan, Zachary Broveak, Robin Geens, Zexue He, Alex Pentland, Marian Verhelst, Tsachy Weissman, and Thierry Tambe. Agent memory: Characterization and system implications of stateful long-horizon workloads. *arXiv preprint arXiv:2606.06448*, 2026. URL https://arxiv.org/abs/2606.06448.

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. URL https://arxiv.org/abs/2310.08560.

Joon Sung Park, Joseph C. O'Brien, Carrie Jun 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*, pp. 1–22, 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. URL https://arxiv.org/abs/2501.13956.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2302.04761.

Yu Wang and Xi Chen. MIRIX: Multi-agent memory system for LLM-based agents. *arXiv preprint arXiv:2507.07957*, 2025. URL https://arxiv.org/abs/2507.07957.

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://openreview.net/forum?id=pZiyCaVuti.

John Yang, Carlos Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652, 2024. URL https://arxiv.org/abs/2405.15793.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

Juwei Yue, Chuanrui Hu, Jiawei Sheng, Zuyi Zhou, Wenyuan Zhang, Tingwen Liu, Li Guo, and Yafeng Deng. HyperMem: Hypergraph memory for long-term conversations. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 35237–35254, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1627. URL https://aclanthology.org/2026.acl-long.1627/.

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. URL https://ojs.aaai.org/index.php/AAAI/article/view/29946.

Wei Zhou, Xuanhe Zhou, Shaokun Han, Hongming Xu, Guoliang Li, Zhiyu Li, Feiyu Xiong, and Fan Wu. Are we ready for an agent-native memory system? *arXiv preprint arXiv:2606.24775*, 2026. URL https://arxiv.org/abs/2606.24775.

A BENCHMARK DETAILS

To ensure rigorous evaluation while maintaining computational efficiency, we strategically partition our benchmark usage into pilot experiments and final evaluation. Pilot experiments on LoCoMo and LongMemEval-oracle leverage two distinct, shorter-context datasets to mitigate over-optimizing on a single benchmark. Final evaluation is then conducted on the context-extended LongMemEval-S and the strictly held-out PersonaMem-128K.

Dataset setting	QA type	Total QA	Evaluated QA	#Conv.	Sess./conv.	Turns/conv.
LoCoMo	OE	1,986	1,540	10	27.2	588.2
LongMemEval-oracle	OE	500	500	500	1.9	21.9
LongMemEval-S	OE	500	500	500	47.7	493.5
PersonaMem-128K	MC	2,727	2,727	60	19.5	758.7

Table 6: **Benchmark statistics across varying context scales.** OE and MC denote open-ended and multiple-choice questions, respectively.

LoCoMo (Maharana et al., 2024) contains 10 long dialogues and questions spanning single-hop, multi-hop, temporal, and open-domain reasoning. Following the standard text-only protocol, we report results on its 1,540 non-adversarial questions.

LongMemEval (Wu et al., 2025) contains 500 evaluation instances covering information extraction, multi-session reasoning, knowledge updates, temporal reasoning, and abstention. It provides varying history lengths, among which we specifically utilize two settings. We use the *oracle* setting (retaining only evidence sessions) for short-context pilot experiments to isolate write-side construction choices. We then reserve the *S* setting (moderately long histories) for the final evaluation of the frozen configuration, specifically testing the model's robustness under a more challenging, distractor-heavy context (detailed statistics are recorded in Table 6).

PersonaMem-v1 (Jiang et al., 2025) evaluates evolving user profiles through four-choice in-situ response selection. In this work, we leverage its official 128K setting as a strictly held-out evaluation benchmark. Crucially, it remains entirely isolated from all method development and tuning, serving as an independent testbed to assess true generalization.

B BASELINE SYSTEMS

All baselines strictly utilize their released memory-construction, maintenance, and retrieval pipelines with official configurations. To ensure a fair comparison, we do not alter their architectural components or method-specific hyperparameters. Instead, we standardize only the shared solver and evaluator in the controlled Qwen-based comparisons.

Mem0. (Chhikara et al., 2025) Mem0 extracts salient conversational facts and consolidates them via add, update, delete, or no-op operations prior to retrieval. Thus, its construction phase tightly couples extraction with explicit record-level mutation.

MemOS. (Li et al., 2025) MemOS treats memory as a schedulable resource. Its MemCube abstraction packages content with provenance and version metadata to support storage, retrieval, and evolution across memory types, exposing lifecycle control at the resource level.

MemoryOS. (Kang et al., 2025) MemoryOS organizes dialogues into short-, mid-, and long-term tiers based on retention horizon rather than semantic type. It routes information across these tiers via dialogue-chain FIFO and segmented paging, querying all tiers jointly during generation.

HyperMem. (Yue et al., 2026) HyperMem represents topics, episodes, and facts using a three-level hypergraph, where hyperedges capture higher-order associations. It employs a hybrid lexical–semantic, coarse-to-fine retrieval strategy to select evidence through this graph structure.

EverMemOS. (Hu et al., 2026a) EverMemOS transforms dialogues into episodic, factual, and foresight MemCells, then consolidates them into thematic MemScenes and profiles. We evaluate it under two modes: the default hybrid retrieval (EverMemOS) and MemScene-guided agentic search (EverMemOS (A)). We strictly preserve its released detectors, prompts, and downstream pipelines.

C Implementation and Evaluation Details

Omega Configuration. Omega employs Qwen3-Embedding-4B[2] for dense recall and Qwen3-Reranker-4B[3] for reranking. Each retrieval route returns 50 candidates, and both channel- and route-level reciprocal-rank fusion use $\kappa = 60$. By default, Omega renders the top 20 distinct payloads for LoCoMo and LongMemEval-S. For the primary PersonaMem-128K evaluation (Omega $+G_{\mathrm{rw}}$), the pipeline remains unchanged except for scaling the retrieval budget to top-40. We additionally report an interface-aware variant (Omega $+G_{\mathrm{ccr}}$) that replaces G_{rw} with an LLM-free choice-contrast route. This route constructs queries by extracting option-discriminative lexical terms from candidate options and concatenating them with the original question, which is then used for both BM25 and dense retrieval. Exact implementation details are provided in the supplementary code.

Comparison Protocol. To ensure strict fairness, we standardize the downstream solver and evaluator. In the Qwen-based evaluation blocks, every system (including Omega and all baselines) uses Qwen3.6-Plus for both answering and judging open-ended tasks, with identical solver prompts. For the GPT-based blocks, Omega uses GPT-4.1-mini for answering and GPT-4o-mini for judging, aligning with the cited baseline results. Across all settings, Omega consistently uses the lightweight Qwen3.5-Flash[4] for child-key generation and open-ended query rewriting.

Frozen Evaluation and Scale Adaptation. As detailed in Appendix A, all non-budget components developed on LoCoMo and LongMemEval-oracle remain frozen for the final evaluation (except for the auxiliary G_{ccr} variant). This protocol rigorously evaluates the transferability of the selected configuration to longer, distractor-heavy histories and a held-out benchmark family, while isolating the system's robustness to extended contexts under this frozen setup.

Evaluation Metrics. We strictly adhere to the official evaluation protocols of each benchmark to ensure fair and reproducible comparisons. Specifically, we report LLM-judged accuracy, F1, and BLEU-1 for LoCoMo; utilize the official task-specific evaluator for LongMemEval; and employ Exact Match (EM) for PersonaMem-128K. Crucially, all primary metrics and judging models are pre-defined prior to experimentation to ensure objective evaluation and mitigate potential bias. To further validate the robustness of these evaluations, we conduct comprehensive sensitivity analyses across different reference qualities and independent judge models, as detailed below.

Evaluator Sensitivity and Robustness. To ensure our conclusions are not artifacts of a specific evaluator, we further evaluate the sensitivity of system rankings to reference quality and judge choice. As shown in Tables 7 and 8, correcting references via `LoCoMo-Audit`[5] shifts absolute scores by 0.78–1.23 points but strictly preserves the system ranking. Furthermore, replacing the primary Qwen3.6-plus judge with either DeepSeek[6] or GPT-4o-mini[7] changes absolute scores, while Omega consistently maintains the top position across both benchmarks under every judge configuration. These checks strongly support evaluator robustness. Together with the paired bootstrap analysis below, they comprehensively bound both evaluator bias and sampling uncertainty.

	Reference sensitivity		Judge sensitivity		
System	**Orig.**	**Audit (Δ)**	**Qwen3.6-Plus**	**DeepSeek (Δ)**	**GPT-4o-mini (Δ)**
MemoryOS	66.62	67.86 (+1.23)	66.62	68.31 (+1.69)	73.96 (+7.34)
Mem0	81.56	82.40 (+0.84)	81.56	83.12 (+1.56)	85.58 (+4.02)
MemOS	84.29	85.06 (+0.78)	84.29	85.84 (+1.55)	89.09 (+4.80)
HyperMem	90.52	91.69 (+1.17)	90.52	90.84 (+0.32)	92.86 (+2.34)
EverMemOS	88.44	89.48 (+1.04)	88.44	88.64 (+0.20)	91.36 (+2.92)
EverMemOS (A)	89.03	90.00 (+0.97)	89.03	89.94 (+0.91)	92.47 (+3.44)
Omega	**91.23**	**92.21** (+0.97)	**91.23**	**92.14** (+0.91)	**93.70** (+2.47)

Table 7: **LoCoMo reference and judge sensitivity.** Reference sensitivity compares Orig. and LoCoMo-Audit under Qwen3.6-plus; judge sensitivity compares Qwen3.6-plus with independent DeepSeek-v4-flash-0731 and GPT-4o-mini judgments on the same frozen answers. All Δ use unrounded scores relative to primary Qwen3.6-Plus reference.

[2] `https://huggingface.co/Qwen/Qwen3-Embedding-4B`
[3] `https://huggingface.co/Qwen/Qwen3-Reranker-4B`
[4] `https://help.aliyun.com/en/model-studio/qwen3-5-flash`
[5] `https://github.com/dial481/locomo-audit`
[6] `https://huggingface.co/deepseek-ai/DeepSeek-V4-Flash-0731`
[7] `https://developers.openai.com/api/docs/models/gpt-4o-mini`

System	Qwen3.6-Plus	Judge sensitivity DeepSeek (Δ)	GPT-4o-mini (Δ)
MemoryOS	60.60	64.40 (+3.80)	62.00 (+1.40)
Mem0	82.40	86.20 (+3.80)	81.40 (−1.00)
MemOS	82.20	86.60 (+4.40)	83.40 (+1.20)
EverMemOS	77.40	84.60 (+7.20)	77.40 (+0.00)
EverMemOS (A)	78.60	85.60 (+7.00)	78.60 (+0.00)
Omega	**90.40**	**93.60** (+3.20)	**90.80** (+0.40)

Table 8: **LongMemEval-S judge sensitivity.** The primary Qwen3.6-plus judgments serve as the reference, while DeepSeek-v4-flash and GPT-4o-mini evaluate the same frozen answers using the official prompt. All Δ use unrounded scores relative to primary Qwen3.6-Plus reference.

Statistical Significance. While preceding analyses bound evaluator bias, we employ paired cluster-bootstrap resampling to quantify sampling uncertainty. Table 9 details the primary comparisons, resampling natural independent units (e.g., conversations) to preserve intra-cluster correlations. On LoCoMo, the 95% confidence interval (CI) crosses zero, indicating Omega and HyperMem are statistically tied; practically, however, Omega requires substantially less solver context and write-side cost. On LongMemEval-S and PersonaMem-128K, Omega's gains over Mem0 and EverMemOS (A) remain strictly positive throughout their 95% CIs. Crucially, the PersonaMem significance holds even under a conservative question-only control (excluding G_{ccr} and answer-choice exposure), confirming robust, statistically significant improvements.

Benchmark	Comparator	Omega	Baseline	Δ	Paired 95% CI	Clusters
LoCoMo	HyperMem	91.23	90.52	+0.71	$[-0.76, +2.06]$	10 conversations
LongMemEval-S	Mem0	90.40	82.40	+8.00	$[+4.60, +11.60]$	500 instances
PersonaMem-128K	EverMemOS (A)	71.43	69.38	+2.05	$[+0.24, +3.95]$	60 personas

Table 9: **Paired cluster-bootstrap significance analysis.** Confidence intervals are computed by resampling the benchmark's natural independent units (Clusters) to account for intra-cluster correlations. The PersonaMem comparison strictly uses the conservative question-only setting.

D Prompt Templates for Memory and Query Operations

This section presents paper-form operational prompt specifications for Omega's core workflows, spanning memory extraction, child-key generation, and query rewriting. The boxes summarize the operative objectives, constraints, input/output schemas, and runtime placeholders, with wording and line breaks adapted for readability. Braced expressions denote runtime placeholders, while the exact executable strings are provided in the supplementary code. To clarify the modular design, we first present a Shared Projection Contract that governs all extraction tasks, followed by type-specific instructions. By default, Omega employs the episodic and fact projectors, whereas the profile extractor is included to support memory-type ablations. Exact executable strings including the Summary and Foresight templates are provided in the supplementary code.

Shared Projection Contract

Act as a typed long-term-memory extractor. Use the detailed instruction for the active memory type and only the supplied chronological source records.

Global rules:

1. Return JSON only: an array of objects. Each object must contain `memory_type`, `content`, `source_record_ids`, and `metadata`; do not output confidence scores.
2. Select `source_record_ids` only from the supplied `record_id` values. Preserve answer-bearing dates, durations, quantities, named events, and proper nouns in `content`, not only in metadata.
3. Cover distinct supported propositions without duplication. Resolve pronouns and vague references to a stated name or an unambiguous role such as "the user," but never invent a name.
4. Return an empty array when no useful memory exists. Recommended metadata include salient entities and, when supported, state, relation, event-time, decision, trigger, or risk fields.

Input: `{source_records}`, a JSON list whose records contain `record_id`, `session_id`, `turn_id`, `speaker`, `timestamp`, and `text`.

Profile Extraction

Extract user or person profile information using an explicit-information versus implicit-traits distinction.
Extraction criteria:

1. **Explicit information:** directly stated objective facts or current status, including basic information, health status, skills, stable preferences, constraints, recurring needs, background, goals, work context, or clear habits.
2. **Implicit traits:** a stable psychological profile, personality tag, decision style, social preference, value, or behavioral tendency supported by repeated or sufficiently strong evidence.

Extraction rules:

1. Output in the same language as the source records.
2. Extract only information about the person supported by the records. Do not treat assistant suggestions or another participant's claims as the user's profile unless attribution is explicit.
3. Avoid one-off, trivial, short-term, outdated, or contradicted states unless metadata marks them as temporary.
4. For implicit traits, prefer short reusable tags such as `[Risk-Averse]`, `[Data-Oriented]`, or `[Socially-Driven]`, and include behavioral evidence in metadata when available.
5. Deduplicate semantically similar items; prefer one enriched, state-like memory over multiple duplicates.
6. Include category or type hints, entities, an evidence summary, and a stable `state_key` in metadata when supported.

Input: `{source_records}` **Output:** the shared JSON array with `memory_type = profile`.

Atomic Fact Extraction

Extract atomic facts optimized for factual retrieval. Extract a fact when the source supports a standalone claim that can be verified independently. Each fact should express exactly one coherent unit of meaning: an action, emotion, reason, plan, decision, preference, opinion, or factual statement.
Extraction rules:

1. Output in the same language as the source records, and base every fact strictly on those records; do not infer hidden information.
2. Use explicit attribution: state who said, did, wanted, decided, felt, or planned something. Resolve pronouns to specific names where possible.
3. Split multiple ideas into separate facts; do not merge logically distinct claims.
4. Filter greetings, phatic communication, and low-value chatter unless they convey specific factual or emotional content.
5. Also extract substantive factual content produced by the assistant: recommendations, named suggestions, definitions, explanations, computed values, or single-answer replies. Attribute it explicitly (e.g., "The assistant recommended ..."). Skip generic acknowledgments, hedging, transitions, and social filler. Keep every fact atomic and self-contained; route ordered multi-step responses to episodic memory.
6. Resolve relative times against the source record's `timestamp`. Put the absolute date in both `content` and `metadata.time`, retain the original phrase in `metadata.time_relative`, and leave an unresolved expression only when no timestamp is available.
7. Include entities, relation hints, time, attribution, or `fact_kind` in metadata when supported.

Input: `{source_records}` **Output:** the shared JSON array with `memory_type = fact`.

Episodic Memory Extraction

Convert source records into complete, independently retrievable event memories. Extract a concise third-person narrative when the source describes a concrete event, interaction, decision, plan, emotional change, state change, or experience.
Extraction rules:

1. Output in the same language as the source records.
2. Preserve important names, time, location, emotions, decisions, plans, outcomes, quantities, proper nouns, special items, and causal relationships.
3. Keep chronological order and use specific names instead of ambiguous pronouns where possible.
4. Resolve relative time expressions (e.g., "yesterday," "last week," or "two weeks ago") against the source record's `timestamp`. Put the absolute date in `content`, retain the original phrase in `metadata.time_relative`, and leave a bare relative expression only when no timestamp is available.
5. Remove greetings, filler, and redundant wording, but do not drop retrieval-critical details.
6. Do not turn stable preferences into episodes unless a specific event reveals them.
7. If the assistant reply is a structured or ordered whole whose sequence carries meaning—a multi-step plan, recommendation list, walkthrough, or narrative answer—capture it as one coherent episode. Preserve both the order and the triggering request, attribute it to the assistant, and retain answer-bearing specifics. Route single-proposition assistant answers to factual memory.
8. Include a title, event time, entities, decision, plan, emotion, or outcome in metadata when supported.

Input: `{source_records}` **Output:** the shared JSON array with `memory_type = episodic`.

Child-Key Generation

Split the memory below into self-contained atomic events only if it describes multiple distinct events. Output only a JSON object of the form `{"atomic": [str, ...]}`.

Generation rules:

1. If the memory describes a single event or topic, return it unchanged as a one-element list.
2. Make every item independently readable: repeat the date (or its best approximation) and the subject's full name, resolve all pronouns, and replace references such as "the event" or "there" with the explicit event or place.
3. Copy entities, names, dates, and facts from the memory without adding unsupported information or paraphrasing away retrieval-critical specifics.
4. When splitting, return two to four items; do not fragment a coherent narrative.

Input: Memory: `{content}` **Output:** only `{"atomic": [str, ...]}`.

Query Rewriting

Given a user's latest message or question from a long-running conversation, produce a JSON object with exactly two fields.

1. `derived_query`: one retrieval query that reformulates the message as a search for the user's **past** experiences, activities, and previously stated preferences about its core topic. Keep concrete topic entities (names, activities, and objects) from the message; drop the current attitude framing.
2. `topic_keywords`: 3–8 core topic keywords or entities copied verbatim from the message (nouns or noun phrases; no attitude words).

Example: For "I've decided to avoid food festivals until they showcase authentic culture," output the derived query "user's past experiences attending food festivals and cultural food events."
Input: Message: `{query}`.
Output: Only the JSON object `{"derived_query": str, "topic_keywords": [str]}`; no other text. The original query remains a separate retrieval route, and the rewrite is used only to retrieve evidence; the solver answers the original query.

E Detailed Write-Side Ablations

This section provides a quantitative complement to the write-side analyses presented in Section 4.3. We characterize memory-type representations and conduct comprehensive ablations across memory-type combinations, projection regimes, and extraction boundaries. All experiments are conducted on LoCoMo and LongMemEval-oracle, using Qwen3.6-plus for both projection and question answering. Consistent with the simplified configuration adopted in the main analysis, memory maintenance and retrieval are fixed to hybrid RRF over Qwen3-Embedding-4B, with a candidate depth of 50 and a top-20 selection. We vary only the construction choice under study.

Type	Form	Representative projection
Profile	persistent state	*(a)* Caroline attends an LGBTQ support group, which helps her feel accepted and gives her courage to embrace herself. *(b)* The user wants to focus on learning new skills and expanding knowledge in different areas, and finds workshops and lectures helpful for this goal.
Episodic	temporal narrative	*(a)* On May 7, 2023, Caroline attended an LGBTQ support group. She found the transgender stories shared there inspiring and felt happy and thankful for the support, which made her feel accepted and gave her courage to embrace herself. *(b)* On May 27, 2023, the user attended a workshop on "Effective Time Management" at the local community center. The user found the workshop helpful in their effort to focus on learning new skills and expanding their knowledge.
Fact	atomic proposition	*(a)* Caroline attended an LGBTQ support group on 7 May, 2023. *(b)* The user attended a workshop on "Effective Time Management" at the local community center on 2023/05/27.

Table 10: Representative typed projections from (a) LoCoMo and (b) LongMemEval. Within each pair, all three types share a source unit and overlapping evidence; outputs are reproduced verbatim.

E.1 Memory-Type Characterization and Examples

Representative typed memory items. Table 10 presents aligned projection triplets, illustrating the distinctions among the three memory types. Within each triplet, the three memory items are constructed from the same extraction unit and shared source evidence, yet differ in their representational focus: the profile abstracts persistent user state, the episode preserves temporally and causally structured context, and the fact distills a self-contained proposition. These qualitative distinctions highlight that the three memory types encode complementary aspects of the underlying information.

Memory-length statistics and retrieval implications. To quantify these distinctions, we examine length distributions and their impact on retrievability. Figure 6 shows factual cells concentrate in short regimes, while episodic cells skew toward longer contexts. Echoing the signal dilution in Figure 2 (right), our reranker diagnostics reveal a pronounced negative correlation between cell length and relevance. This confirms that long episodes serve as poor matching surfaces, validating our design of using concise keys for coherent episodic payloads.

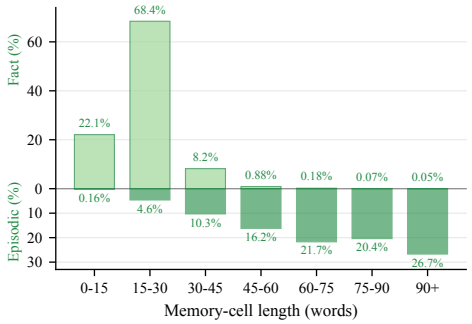


Figure 6: **Memory-cell length distributions.** Length distributions of factual and episodic memory cells, with percentages normalized separately within each type.

E.2 Memory-Type Composition and Ablations

Memory-type contribution and cost trade-off. Table 11 quantifies the performance and storage trade-offs across all combinations of profile, episodic, and fact memories under a fixed downstream pipeline. Across LoCoMo and LongMemEval-oracle, the *Episodic + Fact* configuration consistently emerges as the most effective dual-type setup. Incorporating the Profile type yields marginal accuracy variations while inflating stored tokens by 17–28%. This cost–performance trade-off provides empirical support for Omega's core design choice of adopting the compact *Episodic + Fact* representation, which retains downstream effectiveness while avoiding the additional redundancy and context overhead associated with full profile injection.

Memory types	Overall	Projected cells	Lift	Stored tokens
LoCoMo				
Profile	47.73	1,527	–	38,504
Episodic	79.87	1,385	–	79,986
Fact	73.05	3,261	–	59,162
Profile + Episodic	81.12	2,912	+1.25	118,490
Profile + Fact	74.81	4,788	+1.76	97,666
Episodic + Fact	82.08	4,646	+2.21	139,148
Profile + Episodic + Fact	82.79	6,173	+0.71	177,652
LME-oracle				
Profile	48.80	3,812	–	214,591
Episodic	80.20	5,344	–	650,527
Fact	79.80	19,567	–	605,093
Profile + Episodic	82.20	9,156	+2.00	865,118
Profile + Fact	83.60	23,379	+3.80	819,684
Episodic + Fact	86.00	24,911	+5.80	1,255,620
Profile + Episodic + Fact	86.20	28,723	+0.20	1,470,211

Table 11: Memory-type ablations on the two development benchmarks.

Projection regime ablation and cell composition. Table 12 provides the detailed per-type breakdown underlying Figure 3 (right). Notably, Summary and Foresight in Joint-5 occupy merely 60 and 339 cells in LME-oracle, confirming that auxiliary types receive minimal allocation under shared prompts. In contrast, Per-type-3 yields a $\sim$2.5$\times$ pool expansion, dominated by Fact cells.

Projection regime	Overall	Proj. cells	Episodic	Fact	Profile	Summary	Foresight
LoCoMo							
Joint-3	78.64	2,425	728	838	859	–	–
Joint-5	79.16	2,246	645	578	780	73	170
Per-type-3	82.79	6,173	1,385	3,261	1,527	–	–
LME-oracle							
Joint-3	81.60	11,418	2,801	6,880	1,676	–	–
Joint-5	78.20	10,215	2,801	5,379	1,636	60	339
Per-type-3	86.20	28,723	5,344	19,567	3,812	–	–

Table 12: Projection-regime ablation and per-type cell counts on the two development benchmarks. The LME-oracle Joint-3 total includes 61 uncategorized cells; dashes denote unrequested types.

Extraction-boundary sensitivity across write-side strategies. Table 13 provides the quantitative results underlying Figure 3 (left) under Per-type E+F, and additionally includes Joint-5 as a shared-budget diagnostic. Echoing the main-text tradeoff, finer boundaries yield only modest accuracy gains under both strategies while incurring substantially higher write-side and storage costs, with Session strictly dominating Block-M on both benchmarks. Beyond this shared trend, Per-type E+F exhibits markedly greater boundary robustness than Joint-5, whose accuracy varies far more widely across boundaries, a pattern consistent with greater competition for a shared output budget, further validating Omega's per-type write-side design.

	Per-type E+F (Omega)			Joint-5		
Boundary	Overall	LLM calls	Proj. cells	Overall	LLM calls	Proj. cells
LoCoMo						
Session	82.08	544	4,646	79.16	272	2,246
Block-L	81.17	398	4,234	77.01	199	1,956
Block-M	81.49	598	4,916	78.18	299	2,360
Block-S	82.86	1,182	6,545	81.36	591	3,099
LME-oracle						
Session	86.00	1,896	24,911	78.20	948	10,215
Block-L	85.20	1,836	24,263	77.00	918	9,652
Block-M	85.40	3,008	31,493	77.60	1,504	12,961
Block-S	86.40	5,622	42,577	81.20	2,811	17,635

Table 13: Extraction-boundary ablation under two write-side strategies. Per-type E+F is Omega's write-side strategy and corresponds to the left panel of Figure 3; Joint-5 serves as a shared-output-budget diagnostic. The LLM calls columns count projection calls only. Block sizes (L/M/S, in records) are 30/20/10 for LoCoMo and 16/8/4 for LME-oracle.

E.3 Boundary-Detection Granularity Analysis

This section provides additional experiments to examine whether LLM-based semantic boundary detection provides sufficient utility to justify its cost when session boundaries are already available. We compare EverMemOS with a session-as-cell reference that treats each session as a single extraction unit, isolating the benefit of finer-grained semantic segmentation.

Macro-level statistics. As detailed in Table 14, out of 470 eligible sessions ($\geq$ 3 turns) in a 496-session diagnostic sample from LongMemEval-S, the detector internally splits only 46 sessions (9.8%). Consequently, it yields 626 memory cells—a mere $1.26\times$ increase over the 496-cell session-as-cell reference—indicating that the detector predominantly preserves the original session structure.

Fine-grained distribution. As demonstrated in Figure 7, the same pattern holds on a separate 148-session sample from LongMemEval-S. The detector produces 176 MemCells, only $1.19\times$ the reference, with 90.5% of sessions remaining unsplit and 95.3% yielding at most two MemCells.

Metric	Value
Total source sessions	496
Sessions with ≥ 3 turns	470
Sessions split internally	46 (9.8%)
Avg. boundaries per split session	2.83
Max. boundaries in one session	10
Additional MemCells	130
Session-as-cell total	496
Detector-output total	626
MemCells/session ratio	1.26×

Table 14: **EverMemOS boundary detector.** Aggregate granularity statistics for a 496-session LongMemEval-S diagnostic

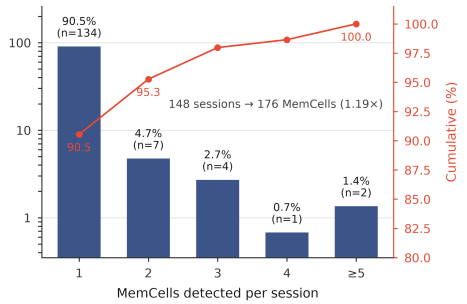


Figure 7: **EverMemOS session granularity.** MemCells per session and cumulative share across 148 LongMemEval-S sessions.

The detector usually preserves sessions and adds only selective splits. Its extra LLM stage yields 1.26× more cells, favoring sessions as the lower-cost default without ruling out gains elsewhere.

E.4 Write-Side Cost Analysis

Table 15 quantifies offline write-side memory construction costs on LoCoMo, tallying all LLM calls and input/output tokens across boundary detection, abstraction, extraction, and key generation. Stage totals may slightly differ from row sums due to independent rounding. Compared to the multi-stage pipelines of HyperMem and EverMemOS, Omega reduces LLM calls by 81–88% and input tokens by 84–91%. This drastic reduction demonstrates that Omega's compact, session-aligned projection lowers the lifecycle overhead without relying on expensive, cascaded extraction stages.

Stage	Calls	In Tok. (M)	Out Tok. (M)
MemoryOS			
Profile analysis	1,095	2.60	1.02
Multi-topic summary	5,882	0.94	0.09
Page keyword gen.	7,725	0.92	0.05
Knowledge ext.	1,095	0.81	0.02
Summary keyword gen.	3,954	0.32	0.02
Total	**19,751**	**5.59**	**1.20**
Mem0			
Memory ext. (total)	**5,882**	**54.40**	**0.16**
HyperMem			
S1: Boundary det.	5,082	6.22	0.41
S2: Episode abs.	663	0.88	0.27
S3: Topic ext.	3,418	13.50	3.05
S4: Fact ext.	470	2.33	1.02
Total	**9,633**	**22.93**	**4.74**
EverMemOS			
S1: Boundary det.	5,519	11.50	0.39
S2: Episode abs.	354	0.84	0.07
S3: Fact ext.	354	0.74	0.19
Total	**6,227**	**13.09**	**0.65**
Omega			
Episode ext.	272	0.98	0.24
Fact ext.	272	0.99	0.30
Episodic key gen.	618	0.16	0.05
Total	**1,162**	**2.13**	**0.59**

Table 15: Offline write-side cost on LoCoMo; token counts are in millions.

F Additional Retrieval Ablations

Cross-dataset sensitivity on LoCoMo. Complementing the LongMemEval-S analysis, Table 16 reports retrieval-side sensitivity on LoCoMo non-adversarial questions. The absolute drops are smaller than on LongMemEval-S, consistent with LoCoMo being less retrieval-limited, but every perturbation remains below the main stack. This agreement across the two open-ended benchmarks supports the coordinated retrieval design without treating the LoCoMo analysis as held-out evidence.

Setting	Route	RR	k	Acc.	Δ
Main stack	MR	K	20	**91.23**	–
w/o multi-route	Single	K	20	90.84	-0.39
w/o reranking	MR	–	20	88.44	-2.79
Rerank: K$\rightarrow$P	MR	P	20	89.94	-1.29
Route RRF$\rightarrow$quota	MR	K	20	90.65	-0.58
Budget: 20$\rightarrow$10	MR	K	10	89.03	-2.20
Budget: 20$\rightarrow$30	MR	K	30	90.39	-0.84

Table 16: Retrieval ablation on LoCoMo. MR denotes multi-route retrieval; K/P denote key/payload reranking; k denotes top-k selection. Δ is relative to the main stack.

G Full-Context Comparisons

Protocol and context accounting. All conditions share the Qwen3.6-plus solver and benchmark-specific evaluation and differ only in solver context. *Omega* uses its standard projected-and-retrieved context; *Full Projection* renders every visible projected item, bypassing retrieval; and *Full History* serializes source records chronologically as speaker–timestamp–utterance sequences, bypassing both projection and retrieval. Table 17 reports the resulting context lengths. No length-based truncation is applied, and no request exceeds the model context window.

PersonaMem-128K

Method	Avg. Tok.	L-Pref	P-Evol	U-Reas	A-Rec	N-Sug	C-Gen	P-Rec	Overall
Omega ($+G_{\mathrm{rw}}$)	1,460	74.4	70.7	83.6	70.8	56.6	72.8	83.6	71.43
Full Projection	21,919	74.8	84.5	93.7	77.1	48.8	80.3	90.1	74.62
Full History	123,578	77.8	84.2	92.9	71.6	48.3	81.7	88.9	74.70

LongMemEval-S

Method	Avg. Tok.	SS-User	SS-Asst.	SS-Pref.	Multi-S	K-Upd.	T-Reas.	Overall
Omega	1,478	97.14	94.64	93.33	81.20	94.87	90.98	90.40
Full Projection	42,091	97.14	94.64	80.00	66.17	82.05	80.45	80.80
Full History	117,645	100.00	100.00	43.33	62.41	82.05	81.20	78.80

LoCoMo

Method	Avg. Tok.	Single-Hop	Multi-Hop	Temporal	Open-Dom.	Overall
Omega	570	92.98	91.84	90.97	75.00	91.23
Full Projection	12,854	92.39	94.68	94.08	73.96	92.01
Full History	31,997	95.36	95.04	94.70	73.96	93.83

Table 17: Complete quantitative comparison with Full Projection and Full History baselines.

Table 17 provides the category-level results summarized in Figure 5. The controls were constructed after the primary benchmark evaluations and were not used to revise Omega. The results show consistent context-efficiency gains for Omega, while the gap to exhaustive controls varies across benchmarks, revealing different remaining sources of error. Full Projection and Full History are diagnostic counterfactuals rather than strict upper bounds, as they also alter context volume and organization. Query-level examples are provided in Section H.1.

H Diagnostic Case Studies

This section bridges the gap between aggregate quantitative metrics and item-level model behavior through detailed case studies. Section H.1 localizes projection and retrieval bottlenecks using full-context counterfactuals. Section H.2 analyzes child-key placement during maintenance, and Section H.3 further isolates the effects of non-budget retrieval components.

H.1 Full-Context Counterfactual Diagnostics

Figure 8 presents diagnostic case studies from each benchmark. For representative queries, we trace the gold evidence, retrieved contexts, and outputs of Omega, Full Projection, and Full History, linking each outcome to its underlying mechanism.

LoCoMo: Projection Loses an Implicit Cross-Session Link

Question: Where did Jolene and her partner find a cool diving spot? **Gold:** Phuket
Evidence: one session places the couple in Phuket for several weeks; a later session reports that they tried scuba diving and found a cool dive spot.

Omega ×	Full Projection ×	Full History (correct)
Context: 568 tokens; top-20 retrieval. Both sessions appear.	**Context:** 16,487 tokens; 558 projected items.	**Context:** 27,399 tokens; 681 dialogue records.
Answer: "during a scuba diving lesson."	**Answer:** "during a scuba diving lesson."	**Answer: Phuket**.
Analysis: Both sessions are retrieved, so the error reflects their implicit relation rather than a recall failure.	**Analysis:** The same error with all 558 projections rules out top-k truncation and identifies a relation lost during projection.	**Analysis:** Raw dialogue preserves the trip context needed to ground the later diving report in Phuket.

LongMemEval-S: Selective Retrieval Disambiguates an Update

Question: What was the amount I was pre-approved for when I got my mortgage from Wells Fargo? **Gold:** $400,000
Evidence: the user reports $350,000 on August 11 and updates the amount to $400,000 on November 30; both statements name Wells Fargo.

Omega (correct)	Full Projection ×	Full History ×
Context: 1,206 tokens. The dated values occupy ranks 1–2.	**Context:** 40,114 tokens; 755 projected items.	**Context:** 126,485 tokens; 396 dialogue blocks.
Answer: $400,000.	**Answer: $350,000**.	**Answer: $350,000**.
Analysis: Retrieval colocates the dated values, allowing the solver to apply recency and select the update.	**Analysis:** Both values are present, but redundant derivatives favor the stale amount.	**Analysis:** The reports remain 34 sessions apart, and the solver rejects the later value. Availability alone is insufficient.

PersonaMem-128K: Retrieval Misses a Decisive State Change

Query: The user is preparing an international themed dinner; which option best describes the evolution of their cooking-show preference?
Gold: (c)
Evidence: the user initially liked cooking shows, later canceled them because the content became repetitive, and now pursues cooking through an international dinner party.

Omega ×	Full Projection (correct)	Full History (correct)
Context: 1,883 tokens; top-40 retrieval.	**Context:** 30,047 tokens; 702 projected items.	**Context:** 162,005 tokens; 738 dialogue records.
Answer: (b), "enjoys them more than ever."	**Answer: (c)**, "no longer enjoys them as much."	**Answer: (c)**, "no longer enjoys them as much."
Analysis: The top-40 window omits the later cancellation and dinner plan, producing a retrieval-coverage failure.	**Analysis:** The complete pool answers correctly, showing that projection preserved the state change but retrieval missed it.	**Analysis:** Raw history gives the same answer with $5.4\times$ more context, indicating little additional construction loss.

Figure 8: Diagnostic case studies across three benchmarks. For each query, we show the gold evidence and compare Omega, Full Projection, and Full History in terms of context size, answer, and analysis. Red highlights indicate model outputs.

H.2 Maintenance-Side Ablation Diagnostics

Figure 9 provides qualitative evidence for the child-placement ablations in Table 4. Case 1 illustrates how alternative placements either fail to match long episodes, thereby validating the retrieval-witness role of child keys, or consume solver slots with redundant information, thus justifying the necessity of deduplication. Case 2 further contrasts Omega's parent-attached keys with standalone child payloads, highlighting the context fragmentation that occurs when children replace parents. In both cases, cards list the diagnostic items by their final retrieval rank and type.

Maintenance Case 1: Counting Events Across Six Sessions

Question: How many charity events did I participate in before the "Run for the Cure" event? **Gold:** 4
Gold set: Dance for a Cause; Walk for Wildlife; charity golf tournament; and Food for Thought gala.

Omega: deduplicated child keys (correct)
Retrieved:
#1 [Episode] On Oct. 15, participated in "Run for the Cure," ran 5 km, and raised $250.
#6 [Fact] Participated in "Dance for a Cause" on May 1.
#13 [Fact] Attended a charity golf tournament on July 17.
#14 [Fact] Volunteered at the Food for Thought gala on Sept. 25.
#20 [Fact] Volunteered at Walk for Wildlife in June.

Answer: 4: Dance, Walk, Golf, and Food for Thought.

Without children ×
Retrieved:
#1 [Episode] On Oct. 15, participated in "Run for the Cure," ran 5 km, and raised $250.
#6 [Fact] Participated in "Dance for a Cause" on May 1.
#11 [Episode] Volunteered at the Food for Thought gala on Sept. 25.
#14 [Fact] Volunteered at Walk for Wildlife in June.
MISS: July 17 charity golf tournament

Answer: 3: Dance, Walk, and Food for Thought.

Child as payload + key ×
Retrieved:
#2 [Episode] On Oct. 15, participated in "Run for the Cure," ran 5 km, and raised $250.
#6 [Fact] Participated in "Dance for a Cause" on May 1.
#7 [Fact] Has event-planning experience from Walk for Wildlife.
#10 [Episode] . . . planned a golf tournament similar to one attended on July 17 . . .
#11 [Child] Planned a similar golf tournament after attending one on July 17.
#19 [Fact] Volunteered at the Food for Thought gala on Sept. 25.
MISS: dated June Walk for Wildlife evidence

Answer: 3: Dance, Golf, and Food for Thought.

Child replaces parent ×
Retrieved:
#4 [Child] On Oct. 15, participated in "Run for the Cure" and ran 5 km for breast-cancer research.
#6 [Fact] Participated in "Dance for a Cause" on May 1.
#10 [Child] Planned a similar golf tournament after attending one on July 17.
#12 [Child] Has event-planning experience from Walk for Wildlife.
MISS: dated June Walk and Sept. 25 Food for Thought evidence

Answer: 2: Dance and Golf.

Maintenance Case 2: Retrieving a Childhood Music Memory

Question: Which artists did Calvin listen to as a child? **Gold:** Tupac and Dr. Dre.

Omega: deduplicated child keys (correct)
Retrieved:
#1 [Episode] Recalled a childhood road trip and "California Love" by Tupac and Dr. Dre.
#4 [Fact] Calvin is going on tour with Frank Ocean.
#8 [Fact] Sang "California Love" by Tupac and Dr. Dre with his father.

Answer: Tupac and Dr. Dre.

Without children ×
Retrieved:
#5 [Fact] Calvin is going on tour with Frank Ocean.
#14 [Fact] Calvin recorded a song with Frank Ocean.
#16 [Fact] Calvin started touring with Frank Ocean.
MISS: childhood Tupac and Dr. Dre evidence

Answer: Frank Ocean.

Child as payload + key (correct, redundant)
Retrieved:
#1 [Episode] . . . childhood memory of "California Love" by Tupac and Dr. Dre . . .
#2 [Child] Childhood memory: "California Love" by Tupac and Dr. Dre.
#4 [Child] Sang the song with his father on a road trip.
#10 [Fact] Sang "California Love" by Tupac and Dr. Dre with his father.
#13 [Child] "California Love" always makes Calvin smile.

Answer: Tupac and Dr. Dre.

Child replaces parent ×
Retrieved:
#3 [Fact] Calvin is going on tour with Frank Ocean.
MISS: the childhood episode and its answer-bearing children and fact from the final top-20

Answer: No childhood artists are provided.

Figure 9: LongMemEval-S and LoCoMo traces for the child-placement ablations in Table 4 (query IDs `a3838d2b` and `conv-50:qa:40`). Both cases compare Omega's deduplicated key-only placement with no children, payload-plus-key children, and parent replacement. Only diagnostic items are shown; **[Child]** denotes a standalone episode-derived payload. Missing evidence and model outputs are highlighted in blue and red, respectively.

H.3 Retrieval-Side Ablation Diagnostics

Figure 10 traces two representative retrieval scenarios to illustrate the impact of candidate ranking and route diversification. Each case compares Omega's full retrieval stack against ablated variants to isolate the contribution of individual components.

Retrieval Case 1: Completed Flights Versus Travel Plans

Question: What is the order of airlines I flew with from earliest to latest before today? **Gold:** JetBlue → Delta → United → American.
Notation: Ranks refer to final parent-payload order after late folding. MISS marks absent answer-bearing evidence; . . . marks omitted content within a displayed payload.

Main stack (correct)
Retrieved:
#4 [Episode] . . . reported a terrible American Airlines entertainment experience while flying NYC–LAX on Feb. 10 . . .
#6 [Fact] Experienced a one-hour United delay from Boston to Chicago on Jan. 28 due to air-traffic control.
#8 [Episode] . . . earned 10,000 Delta SkyMiles after completing a Boston–Atlanta round trip on Jan. 15 . . .
#10 [Episode] . . . decided to book a future JetBlue flight from Boston to Fort Lauderdale . . .
#20 [Fact] Recently took a JetBlue red-eye from San Francisco to Boston and slept for almost the entire flight.
Answer: JetBlue → Delta → United → American.

Route RRF → quota ×
Retrieved:
#4 [Episode] . . . earned 10,000 Delta SkyMiles after completing a Boston–Atlanta round trip on Jan. 15 . . .
#5 [Episode] . . . reported a terrible American Airlines entertainment experience while flying NYC–LAX on Feb. 10 . . .
#8 [Episode] . . . considered American or Delta for Miami and recalled a Jan. 28 United delay . . .
#9 [Episode] . . . booked a future JetBlue flight and requested a Delta upgrade for the return leg . . .
MISS: completed JetBlue
Answer: Delta → United → American → JetBlue.

Without reranking ×
Retrieved:
#10 [Episode] . . . considered American or Delta for Miami and recalled a Jan. 28 United delay . . .
#11 [Fact] Had a terrible American Airlines entertainment experience on a NYC–LAX flight on Feb. 10.
#14 [Fact] Planned a March spring-break trip to Fort Lauderdale on Spirit Airlines.
MISS: completed JetBlue and Delta
Answer: United → American.

Rerank payloads before folding ×
Retrieved:
#1 [Fact] Had a terrible American Airlines entertainment experience on a NYC–LAX flight on Feb. 10.
#11 [Episode] . . . considered American or Delta for Miami and recalled a Jan. 28 United delay . . .
#18 [Fact] Planned a March spring-break trip to Fort Lauderdale on Spirit Airlines.
MISS: completed JetBlue and Delta
Answer: United → American → American.

Retrieval Case 2: Recovering an Option-Discriminative Preference

Query (abridged): Restaurant reviewing evolved from a weekend hobby into a responsibility after the user's opinions began affecting local eateries. **Gold:** (b)
Option contrast: the earlier interest was (a) reviewing cocktails, (b) mixing cocktails, (c) writing about cocktails, or (d) cooking.

With multi-route retrieval: $q + G_{\mathrm{ccr}}$ (correct)
Retrieved:
#7 [Fact] Creating a signature cocktail reignited the user's interest in mixing cocktails.
#9 [Fact] Started mixing cocktails at home and hosting cocktail nights with friends.
#11 [Episode] A cocktail-making workshop rekindled a passion for cocktail mixing . . .
#13 [Fact] Got back into cocktail mixing after attending the workshop.
Answer: (b) mixing cocktails.

Without multi-route retrieval: q only ×
Retrieved:
#1 [Fact] Returned to reviewing restaurants recently.
#3 [Episode] Returned to restaurant reviewing after a friend's success in food blogging . . .
#8 [Episode] A cooking series inspired the user to cook more and try new techniques . . .
#16 [Fact] Started a food blog about cocktail recipes and cooking experiences.
MISS: explicit cocktail-mixing memories (ranks #7, #9, #11, and #13 in the left-hand multi-route condition)
Answer: (d) cooking.

Figure 10: **Qualitative retrieval traces.** Case 1 examines reranking and fusion on LongMemEval-S (Table 5), whereas Case 2 examines auxiliary-route query diversification on PersonaMem-128K (Table 3). Missing evidence and model outputs are highlighted in blue and red, respectively.

Diagnostic conclusion. These qualitative traces demonstrate that Omega's performance stems from an orchestrated lifecycle co-design rather than isolated component improvements. Crucially, the maintenance stage validates that child keys function as precise retrieval witnesses, resolving the matching disadvantage of long episodes without introducing redundant or fragmented payloads. Finally, the retrieval stage confirms that key-level reranking and multi-route fusion are indispensable for recovering diverse, option-discriminative evidence. Together, these mechanisms ensure that the solver receives coherent, non-redundant, and highly relevant context within a strict budget.