MEMORANDA: MEMORY-ANCHOR-DRIVEN ARCHITECTURE FOR LONG-TERM CONVERSATIONAL MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Answering questions about long conversations can require evidence spread across multiple events, temporal relations, and evolving states. Existing memory systems face a trade-off between reducing query-time computation and preserving historical details. Organizing evidence when each question arrives accommodates its specific information needs, but requires repeated search and selection across questions. Compressing history in advance can reduce query-time work, but must decide what to retain before future questions are known, potentially omitting details needed by later answers. We present **Memor**y-**An**chor-**D**riven **A**rchitecture (MemorAnDA; hereafter MEMO), which makes history organization reusable without limiting future answers to compressed information. Memory anchors connect organized memories to their original records, allowing the system to use prior organization to find relevant evidence and recover the details needed for each question. This design reduces the need to repeatedly organize history at query time while preserving access to information that compression may omit. Across three matched build/reader model groups, the primary 8K configuration improves answer accuracy over the corresponding LightMem baseline by 1.40–9.80 percentage points on LongMemEval-S and 7.60–10.20 points on LoCoMo.

1 INTRODUCTION

Long-term conversational memory gives language models access to information that accumulates across days, sessions, and changing user goals. The difficulty is not only the size of this history. A question may require evidence from several events, temporal relations, or intermediate states. Answering whether a plan changed, for example, can require finding when it was proposed, which constraints were introduced later, and what decision followed. Multi-turn memory tasks therefore require systems to connect references across turns and preserve the small set of records that jointly supports a later answer (Katsis et al., 2025; Maharana et al., 2024). This evidence is distributed across the conversation rather than contained in a single sentence, especially when user profiles, preferences, and plans evolve over time.

Existing memory systems differ in both how they represent history and when they perform retrieval work. LightMem organizes and consolidates information during construction (Fang et al., 2026), while T-Mem precomputes triggers that make stored memories reachable through indirect cues (Guo et al., 2026). Other designs use question-dependent graph traversal or selectively activate deeper reasoning (Lin et al., 2026; Zhao et al., 2026; Tang et al., 2026). These approaches reflect a tension between reusable history organization and question-specific evidence access. Searching and organizing history for each question allows evidence selection to follow the current information need, but repeats work across questions, increasing query-time computation and, when language models perform this work, token consumption. Compressing history in advance can reduce these recurring costs, but requires deciding what to retain before future questions are known. Details omitted during compression may later be essential for distinguishing states or connecting events, leaving subsequent answers incomplete or incorrect if the original evidence is no longer accessible. The challenge is therefore to make prior organization reusable without restricting future answers to the information retained in a compressed memory.

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.

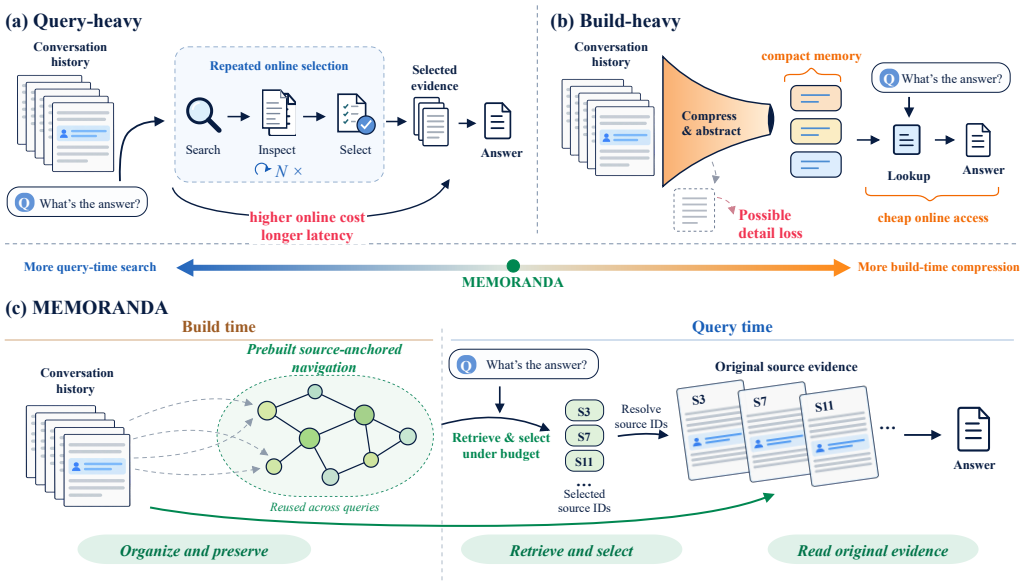

Figure 1: MEMO organizes long-term conversation histories before questions arrive, preserves the original records, and selects a bounded evidence set for each question.

To address this trade-off, we introduce ***Memory-Anchor-Driven Architecture*** (**MemorAnDA**; hereafter MEMO), which separates reusable history organization from question-specific evidence selection. The system organizes semantic, temporal, and state-change connections in advance while preserving the original conversation in full. For each question, direct retrieval finds matching records, while the organized views help uncover related evidence that may be difficult to find from its original wording alone. Memory anchors connect these views to their supporting records, allowing both forms of retrieval to contribute original evidence to a budgeted reader context. This design reuses prior organization without requiring compressed representations to retain every detail a future question may need. Retrieval and selection require no additional language-model calls.

Our main contributions are:

- We introduce MEMO, a conversational memory architecture that combines reusable history organization with question-specific access to original evidence, allowing future answers to draw on details beyond those retained in compressed representations.
- We develop an anchor-based mechanism that integrates direct and view-based retrieval through shared source records. It selects complementary evidence under a fixed budget by accounting for overlapping sources and their incremental support and token cost, without additional language-model calls for retrieval or selection.
- We evaluate MEMO across two benchmarks and three matched build/reader model groups. At the primary 8K budget, it improves accuracy over LightMem by 1.40–9.80 percentage points on LongMemEval-S and 7.60–10.20 points on LoCoMo. Controlled experiments on LoCoMo further support the contribution of multi-view evidence access.

2 RELATED WORK

Long-term conversational memory and benchmarks. LoCoMo and LongMemEval test cross-session recall, temporal relations, and changing user states (Maharana et al., 2024; Wu et al., 2025a). MTRAG and CORAL evaluate conversational retrieval and generation (Katsis et al., 2025; Cheng et al., 2025b); other benchmarks examine memory utilization, forgetting, and personalization (Jia et al., 2025; Uddin et al., 2026; Shen et al., 2026a; Xiao et al., 2026).

Memory construction and management. LightMem filters, organizes, and consolidates history (Fang et al., 2026). A-MEM and Mem0 organize and update persistent memories (Xu et al., 2025; Chhikara et al., 2025); SeCom and StructMem use conversational structure to define memory units (Pan et al., 2025; Xu et al., 2026a). Adaptive distillation selects what to retain (Ma et al., 2026b), while learned construction and feedback-based policies adapt memory to downstream use (Shen et al., 2026b; Ma et al., 2026a; Yu et al., 2026). Hierarchy and scheduling govern storage and maintenance (Hu et al., 2025; Kang et al., 2025; Li et al., 2025c). Small-model controllers combine retrieval with offline consolidation (Zhang et al., 2026); GAM separates event encoding from associative consolidation (Wu et al., 2026). MEMO links retrieval representations to separately preserved original evidence.

Navigation and structured retrieval. Reusable access paths arise from pre-storage reasoning and graph associations (Kim et al., 2025; Zhang et al., 2025c; Gutiérrez et al., 2024), relational and temporal indices (Jiang et al., 2026; Huang et al., 2026), and contextual or tree-based organization (Yang et al., 2026; Liu et al., 2026; Wang et al., 2025b). T-Mem makes scenes or items accessible through write-time triggers (Guo et al., 2026), sharing our goal of preparing access paths before questions arrive. MemWeaver retrieves knowledge and evidence across memory types (Ye et al., 2026); HERO preserves raw dialogue and uses query anchors for iterative graph traversal (Lin et al., 2026). Mnemis combines similarity retrieval with hierarchical global selection (Tang et al., 2026); HyMem selectively invokes LLM-based retrieval (Zhao et al., 2026). Retrieval agents and inference paths support event reasoning (Banerjee et al., 2026; Xu et al., 2026b), as do temporal graphs, timeline summarization, and reflection (Rasmussen et al., 2025; Bao et al., 2025; Bazaga et al., 2025). MEMO combines direct and compiled retrieval through shared sources, selecting groups of original records under an evidence budget without query-time generative retrieval.

Evidence selection and answer faithfulness. Decomposition and diversified retrieval expose relevant details (Li et al., 2025a; In et al., 2025); query decomposition and evidence-tree search connect multi-hop support (Cheng et al., 2025a; Zhu et al., 2025; Sun et al., 2025). Graphs also connect passages for summarization and QA (Zhang et al., 2025a; Yuan et al., 2026; He et al., 2026), with representation granularity affecting GraphRAG performance (Nishida et al., 2026). Context condensation and question-aware chunk selection control reader input (Li et al., 2025b; Sheng et al., 2025). Set selection accounts for coverage and redundancy (Lee et al., 2025); MEMO applies this principle to overlapping source groups. Oracle retrieval does not eliminate reasoning failures (Geng et al., 2025). Faithfulness methods address evidence–answer conflicts (Zhang et al., 2025b; Bi et al., 2025; Wang et al., 2025a), while source-referencing and RAG evaluation distinguish evidence use from answer quality (Wu et al., 2025b; Hagström et al., 2025; Cruz Blandón et al., 2025; Tang et al., 2025). Our evidence metrics and failure analysis retain this distinction.

3 MEMORANDA

MEMO separates reusable history organization from question-specific evidence selection through shared source references. We first define the source representation and construction process, then describe how direct and view-based retrieval support budgeted selection of original evidence. Figure 2 summarizes the system.

3.1 PROBLEM SETUP AND DESIGN OBJECTIVE

Let H be a conversation history and $\mathcal{R}(H)$ its raw records, each containing original text, speaker, timestamp, session identifier, and source ID. The build stage produces $M = \mathrm{Build}(H; \theta_b)$ with a fixed model and configuration θ_b, without access to future questions, answers, or evaluation annotations. For a question q, MEMO uses M and $\mathcal{R}(H)$ to select source IDs S_q. The design objective is to cover complementary retrieval signals while satisfying $L(S_q) \leq B$, where L counts tokens in the rendered original records. The evidence budget excludes the question, instructions, and reader output. Generated items supply retrieval signals; only their resolved raw records form the factual evidence given to the reader.

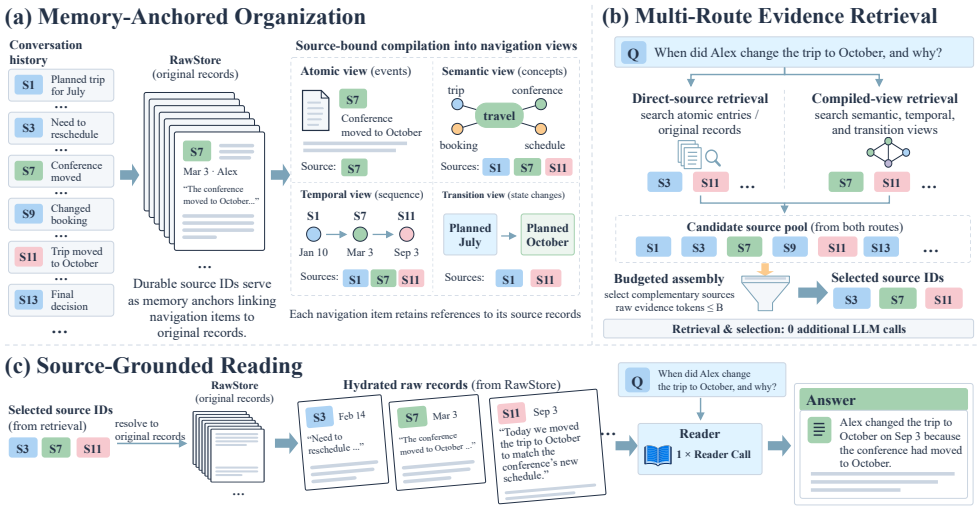

Figure 2: MEMO overview. Query-agnostic compilation creates multiple navigation surfaces linked to durable source IDs. Query-time retrieval fuses the routes, selects sources under a fixed budget, hydrates the original records, and invokes one reader.

3.2 MEMORY ANCHORS AND SOURCE IDENTITY

For each history, MEMO preserves the original records as $\mathcal{R}(H) = \{(s_i, x_i, m_i)\}_{i=1}^{n}$, where s_i is a stable source ID, x_i is the unmodified text, and m_i contains speaker, time, and session metadata. These source IDs implement memory anchors, connecting each navigation item to one or more supporting records. This separation allows navigation representations to summarize or reorganize history while keeping omitted details accessible in the original evidence. Shared source identity also lets the selector combine support from different views while accounting for each original record only once. Shared IDs expose overlap between representations and remain stable when navigation items are regenerated, keeping source identity independent of how history is organized. Every navigation item must have non-empty, resolvable source references, which are retained through retrieval, selection, and evidence recovery. Storage and identifier details are provided in Appendix B.

3.3 REUSABLE MULTI-VIEW HISTORY ORGANIZATION

Future questions may depend on individual facts, connections between events, or changes in an entity's state. A single representation need not expose all of these equally well. MEMO therefore retains atomic entries for direct matches and constructs semantic, temporal, and transition views for complementary retrieval cues. Semantic views express conceptual connections, temporal views capture event order, and transition views highlight changes in state. These views are designed to make supporting records reachable even when their wording differs from the question. Each retains references to its supporting sources through memory anchors.

Construction proceeds one session at a time. The build model identifies *evidence groups* that associate facts and temporal relations with supporting records, and *state snapshots* that describe an entity's attributes at a given time. Deterministic projection rules turn evidence groups into semantic views and, when supported, temporal views. Snapshots of the same normalized entity and attribute are compared across sessions; a change in value produces a transition view linked to the supporting records. Snapshots themselves are not searchable items. Redundancy filters remove semantic views that closely duplicate their original sources and near-duplicate semantic or temporal views from the same group. The resulting organization is reused across questions, while the selection of evidence remains question dependent.

For retrieval, each item is represented as $e_j = (t_j, z_j, I_j, S_j, \tau_j)$: retrieval text, normalized embedding, contributing entries, supporting source IDs, and representation type with metadata, re-

spectively. Atomic and derived items share the same embedding model. Appendix B specifies the construction rules and filters; Appendix F provides the build prompt.

3.4 MULTI-ROUTE RETRIEVAL AND BUDGETED SELECTION

For a question q, MEMO retrieves candidate evidence through two complementary routes. The direct route searches atomic entries, each corresponding to an original conversation record, for content semantically similar to the question. This route provides access to explicitly stated facts and fine-grained details. The view-based route searches derived semantic, temporal, and state-transition representations, which make conceptual connections, event order, and changing states explicit. These representations provide additional retrieval cues for supporting records whose original wording may not closely match the question. Both routes retain the source references of their retrieved items, allowing their candidates to be combined and selected as original evidence.

Both routes use the same embedding model. The normalized question vector z_q and item vectors z_j are scored by $r_j = \cos(z_q, z_j)$. The direct and view-based candidate sets, C_{raw} and C_{comp}, retain the top 120 atomic items and top 360 derived items, respectively. The merged candidate set $\mathcal{C}(q) = C_{\text{raw}} \cup C_{\text{comp}}$ is ordered by similarity. Each candidate retains its score, embedding, and supporting source-ID set S_j. These source references place candidates from both routes in a common source space, where different retrieval cues may point to overlapping original records.

Selection accounts for redundancy in both retrieval signals and source records. Similarity clustering groups repeated signals, while source grouping identifies candidates that would supply the same evidence. Candidates are processed in relevance order and assigned to the first matching cluster leader, or become a new leader if none matches. Let $c(j)$ denote item j's cluster and j_k the leader of cluster k, with weight $w_k = [\max(0, r_{j_k})]^2$. Items with identical source-ID sets form one selection action a with source set S_a. The selected actions A supply the source union $S(A)$.

The value of an action depends on which sources have already been selected. A cluster is covered when at least one of its candidates has all supporting sources selected: $\mathcal{K}(S) = \{c(j) : e_j \in \mathcal{C}(q),\ S_j \subseteq S\}$. Adding action a therefore covers $\mathcal{K}_{\text{new}}(a \mid A) = \mathcal{K}(S(A) \cup S_a) \setminus \mathcal{K}(S(A))$. This coverage is dynamic: adding a source can complete the support for several candidates, including candidates in other clusters. The selector therefore evaluates the additional support provided by an action relative to the current evidence set. Cluster coverage is a proxy for complementary retrieval support, not a measure of answer completeness.

For actions with positive gain and added cost, the priority is

$$\rho_q(a \mid A) = \frac{\sum_{k \in \mathcal{K}_{\text{new}}(a|A)} w_k}{\sum_{s \in S_a \setminus S(A)} \widehat{c}_s}, \tag{1}$$

where $\widehat{c}_s$ estimates the token cost of one formatted source block, including metadata and separators. The numerator measures newly covered retrieval support, while the denominator counts only sources not already selected. Thus, a record referenced by multiple candidates incurs its evidence cost only once. After each admission, missing sources, coverage gains, costs, and priorities are recomputed.

Starting from an empty set, the selector considers actions in descending priority and resolves their new sources. Admission requires the complete tentative packet's exact rendered length to satisfy $L(S(A) \cup S_a) \leq B$. An over-budget action is skipped as a whole for the current state, but can be reconsidered after another admission. Selection stops when no action with positive gain and added cost fits the budget. Records are never truncated. The result is $S_q = S(A)$. Retrieval and selection use embeddings and deterministic operations, requiring no query-time language-model calls. Appendix B details clustering, tie-breaking, cost estimation, and rendering.

The selected source IDs resolve to original evidence $E_q = \text{Hydrate}_{\mathcal{R}(H)}(S_q)$, which the reader uses to answer in one call, $\hat{y} = R(q, E_q)$. This evidence contains the original text with speaker and timestamp information. Derived representations guide evidence access and selection, while the reader receives the supporting original records as its factual context. Appendices B and F provide the rendering details and reader instruction.

Table 1: **Category-level accuracy (%) on LoCoMo.** B and O denote evidence and output budgets; 8K is primary. Per backbone: bold = best, underline = second best; best Overall is italicized. Dashes = unavailable.

Method	Multi-hop	Temporal	Open-domain	Single-hop	Overall
GPT-4o-mini					
FullContext	68.79	53.27	59.38	84.90	73.77
NaiveRAG	56.38	58.57	59.38	72.53	65.84
LightMem	62.41	**73.52**	50.00	77.76	72.34
A-MEM	68.09	61.37	**65.62**	77.41	71.62
Mem0	41.49	36.76	51.04	62.54	52.60
MemOS	59.22	61.68	51.04	71.11	65.71
MEMO (B=4K, O=512)	75.18	61.68	63.54	81.09	74.87
MEMO (B=8K, O=512)	**80.85**	66.67	61.46	86.80	79.94
MEMO (B=16K, O=1024)	79.79	66.67	**65.62**	**87.28**	***80.26***
GPT-4.1-mini					
FullContext	86.88	80.06	63.54	**92.63**	87.14
NaiveRAG	54.26	66.98	60.42	70.63	66.23
LightMem	74.11	77.57	54.17	82.40	78.12
A-MEM	73.05	66.67	64.58	78.12	73.96
Mem0	42.91	49.84	52.08	59.93	54.22
MemOS	60.99	58.26	61.46	72.18	66.56
MEMO (B=4K, O=512)	83.69	80.69	66.67	85.97	83.25
MEMO (B=8K, O=512)	**88.65**	83.49	**70.83**	90.96	87.73
MEMO (B=16K, O=1024)	87.59	**85.36**	**70.83**	91.08	***87.99***
Gemini-2.5-Flash					
FullContext	70.92	50.47	48.96	85.73	73.38
NaiveRAG	52.84	61.68	55.21	65.28	61.62
LightMem	67.73	76.01	44.79	79.55	74.48
A-MEM	64.18	70.40	54.17	72.53	69.42
MemOS	63.48	68.85	58.33	67.30	66.36
MEMO (B=4K, O=512)	81.56	67.91	47.92	83.47	77.66
MEMO (B=8K, O=512)	**88.30**	**78.50**	**59.38**	88.70	***84.68***
MEMO (B=16K, O=1024)	**88.30**	71.96	54.17	**90.37**	83.90

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Experimental details. Each model group uses the same model for build and reader: GPT-4o-mini (`gpt-4o-mini-2024-07-18`), GPT-4.1-mini (`gpt-4.1-mini-2025-04-14`), or Gemini-2.5-Flash. The build stage sees only the dialogue and its metadata; future questions, answers, and evaluation annotations are unavailable. The primary configuration uses an 8,192-token answer-facing evidence budget and a 512-token output limit. We additionally evaluate 4,096/512 and 16,384/1,024, reusing the same built memory and changing only the evidence budget and reader output allowance.

Datasets and baselines. We evaluate LongMemEval-S (Wu et al., 2025a) and LoCoMo (Maharana et al., 2024). For LoCoMo, we evaluate the non-adversarial QA categories (Cat1–4) and exclude adversarial questions (Cat5). We report LLM-judged accuracy using the scoring protocol specified in Appendix F. We compare MEMO with FullContext, NaiveRAG, LightMem (Fang et al., 2026), A-MEM (Xu et al., 2025), Mem0 (Chhikara et al., 2025), and MemOS (Li et al., 2025c). On LongMemEval, Mem0 is reported only for GPT-4o-mini. Each baseline follows its documented construction and retrieval interface. LongMemEval accuracy uses all 500 questions, counting missing answers from incomplete runs as incorrect; Appendix A reports completion counts.

Metrics and accounting. We report answer accuracy (ACC), Evidence Recall, and All-Evidence Recall. Evidence Recall is the average fraction of gold sources retrieved per question, whereas All-Evidence Recall is the fraction of questions for which every gold source is retrieved. On LoCoMo, ACC uses all 1,540 questions, while the two evidence metrics use the 1,527 questions with gold evidence annotations. GPT-4o-mini judges ACC with a common rubric and judge settings across methods for each benchmark and question type. Build and query tokens, calls, and runtime are

Table 2: **Category-level accuracy (%) on LongMemEval-S.** B and O denote evidence and output budgets; 8K is primary. Per backbone: bold = best, underline = second best; best Overall is italicized. Dashes = unavailable.

Method	Single-user	Single-asst.	Multi-session	Temporal	Update	Preference	Overall
GPT-4o-mini							
FullContext	87.14	94.64	54.89	51.13	69.23	6.67	62.20
NaiveRAG	85.71	**98.21**	51.88	47.37	71.79	60.00	64.20
LightMem	94.29	26.79	**75.94**	**70.68**	**91.03**	**73.33**	73.80
A-MEM	91.43	94.64	49.62	44.36	66.67	46.67	61.60
Mem0	91.43	35.71	51.13	47.37	65.38	40.00	55.60
MemOS	74.29	66.07	30.83	25.56	57.69	23.33	43.20
MEMO (B=4K, O=512)	94.29	92.86	69.17	62.41	83.33	56.67	75.00
MEMO (B=8K, O=512)	**95.71**	92.86	72.18	62.41	82.05	46.67	***75.20***
MEMO (B=16K, O=1024)	**95.71**	94.64	70.68	62.41	79.49	40.00	74.20
GPT-4.1-mini							
FullContext	97.14	**100.00**	63.91	72.93	**87.18**	53.33	78.00
NaiveRAG	91.43	98.21	54.14	55.64	78.21	53.33	68.40
LightMem	95.71	41.07	75.94	73.68	79.49	80.00	75.00
A-MEM	97.14	96.43	53.38	46.62	76.92	56.67	66.40
MemOS	81.43	67.86	39.10	27.07	71.79	26.67	49.40
MEMO (B=4K, O=512)	97.14	96.43	72.18	75.19	85.90	76.67	81.60
MEMO (B=8K, O=512)	97.14	98.21	**78.95**	**78.95**	84.62	**83.33**	***84.80***
MEMO (B=16K, O=1024)	**98.57**	98.21	77.44	75.94	82.05	80.00	83.20
Gemini-2.5-Flash							
FullContext	88.57	91.07	69.92	78.95	79.49	76.67	79.20
NaiveRAG	90.00	**98.21**	54.14	56.39	87.18	76.67	71.20
LightMem	97.14	25.00	75.19	80.45	**89.74**	**83.33**	76.80
A-MEM	88.57	96.43	52.63	53.38	80.77	63.33	67.80
MemOS	87.14	64.29	36.84	30.08	71.79	16.67	49.40
MEMO (B=4K, O=512)	95.71	91.07	75.19	82.71	84.62	56.67	82.20
MEMO (B=8K, O=512)	**98.57**	96.43	**78.20**	79.70	83.33	60.00	83.20
MEMO (B=16K, O=1024)	**98.57**	**98.21**	76.69	**84.21**	82.05	53.33	***83.60***

recorded separately from judge costs. Section 4.3 reports the LoCoMo cost profile; Appendix E gives the full ledgers and accounting definitions.

4.2 MAIN RESULTS

Tables 1 and 2 report category-level and overall accuracy under the three matched build/reader backbones, with 8K as the primary configuration. On LoCoMo, MEMO improves over the matched LightMem (Fang et al., 2026) baseline by 7.60, 9.61, and 10.20 percentage points for GPT-4o-mini, GPT-4.1-mini, and Gemini-2.5-Flash, respectively. Multi-hop accuracy rises by 14.54–20.57 points; Single-hop and Open-domain accuracy also improve for all three readers. Temporal accuracy is more model dependent: it improves with GPT-4.1-mini and Gemini-2.5-Flash but falls below LightMem with GPT-4o-mini. At 8K, MEMO outperforms all evaluated baselines in overall accuracy across the three backbones, including FullContext, with a bounded evidence set.

On LongMemEval, MEMO improves over LightMem by 1.40, 9.80, and 6.40 points for the three readers. The category profiles are less uniform than on LoCoMo. GPT-4.1-mini gains on multi-session, temporal, update, and preference questions, and Gemini-2.5-Flash gains most clearly on assistant-side extraction and multi-session questions. With GPT-4o-mini, the large recovery in assistant-side extraction is partly offset by lower scores on several reasoning categories. MEMO provides the best overall score among the bounded-memory systems in each block. With Gemini-2.5-Flash, MEMO at 8K reaches 83.20%, compared with 79.20% for FullContext. These aggregate gains coexist with category-specific weaknesses, especially on preference questions for GPT-4o-mini and Gemini-2.5-Flash.

Across evidence budgets, the largest LoCoMo improvement occurs from 4K to 8K, with gains of 4.48–7.02 points. Increasing to 16K adds at most 0.32 points for the GPT readers and lowers Gemini accuracy by 0.78 points. LongMemEval also shows limited additional benefit beyond 8K: 16K lowers accuracy for both GPT readers and adds 0.40 points for Gemini. Thus a larger packet does not consistently produce a better answer, supporting 8K as a common operating point across the two benchmarks. All three settings reuse the same build. Because 16K also doubles the output al-

Table 3: **Accuracy and cost on LoCoMo.** Build tokens are in thousands; Tokens/Q are unscaled. Per backbone: bold/underline = best/second best, excluding Query Calls; best ACC is italicized. Dashes = no build or unavailable.

Method	ACC (%)	Build			Query	
		Tokens (k)	Calls	Tokens/Q	Tokens/Q	Calls/Q
GPT-4o-mini						
FullContext	73.77	–	–	–	25,418.08	1.0000
NaiveRAG	65.84	–	–	–	712.00	1.0000
LightMem	72.34	879.10	344	570.85	2,687.98	1.0013
A-MEM	71.62	10,327.06	11,754	6,705.88	1,456.66	1.0000
Mem0	52.60	16,771.81	11,495	10,890.79	2,196.27	1.0000
MemOS	65.71	10,188.86	5,882	6,616.14	**556.93**	1.0006
MEMO (B=4K, O=512)	74.87	**836.17**	**272**	**542.97**	4,266.24	1.0000
MEMO (B=8K, O=512)	79.94	**836.17**	**272**	**542.97**	8,285.24	1.0000
MEMO (B=16K, O=1024)	***80.26***	**836.17**	**272**	**542.97**	9,593.77	1.0000
GPT-4.1-mini						
FullContext	87.14	–	–	–	25,417.85	1.0000
NaiveRAG	66.23	–	–	–	703.74	1.0000
LightMem	78.12	880.70	334	571.88	3,052.09	1.0019
A-MEM	73.96	11,758.19	11,754	7,635.19	1,874.28	1.0013
Mem0	54.22	13,834.56	9,933	8,983.48	2,810.21	1.0000
MemOS	66.56	10,395.67	5,882	6,750.44	**545.56**	1.0000
MEMO (B=4K, O=512)	83.25	**847.16**	**272**	**550.10**	4,311.28	1.0000
MEMO (B=8K, O=512)	87.73	**847.16**	**272**	**550.10**	8,411.72	1.0000
MEMO (B=16K, O=1024)	***87.99***	**847.16**	**272**	**550.10**	11,750.79	1.0000
Gemini-2.5-Flash						
FullContext	73.38	–	–	–	32,713.40	1.0000
NaiveRAG	61.62	–	–	–	1,089.64	1.0000
LightMem	74.48	1,221.66	347	793.28	3,495.18	2.0000
A-MEM	69.42	9,194.48	11,754	5,970.44	1,784.24	1.0000
MemOS	66.36	10,893.33	5,882	7,073.59	**920.09**	1.0000
MEMO (B=4K, O=512)	77.66	**982.04**	**272**	**637.69**	4,933.66	1.0000
MEMO (B=8K, O=512)	***84.68***	**982.04**	**272**	**637.69**	10,788.38	1.0000
MEMO (B=16K, O=1024)	83.90	**982.04**	**272**	**637.69**	11,840.27	1.0000

lowance, that comparison evaluates a complete resource configuration rather than isolating evidence length alone (Appendix C).

4.3 LoCoMo Cost Profile

Table 3 separates build costs, amortized over 1,540 questions, from physical reader prompt and completion tokens per question. The same build is reused across all three evidence budgets, so the accuracy changes above incur no additional construction calls. At 8K, MEMO uses 0.80–0.96× LightMem's build tokens and 272 construction calls versus 334–347, gaining 7.60–10.20 accuracy points at 2.76–3.09× the query tokens. Compared with FullContext, MEMO uses approximately one-third of the reader tokens while attaining higher accuracy in all three model groups. These comparisons identify the LoCoMo trade-off: MEMO allocates more reader context than LightMem, but retains a substantially smaller context than the full history. Retrieval and source selection add no language-model calls, and answering requires one reader call per question. Build totals for A-MEM, Mem0, and MemOS include locally monitored failed attempts and do not denote provider-billed usage. Full cost ledgers for both benchmarks and the accounting definitions appear in Appendix E.

4.4 Navigation Mechanism

Figure 3 compares retrieval-route configurations with the reader and judge fixed. Atomic-only uses the top-120 atomic entries without compiled views; Derived-only removes the direct atomic/raw route. All variants deliver original records to the same reader. Full MEMO improves over Atomic-only by 3.90 and 3.83 accuracy points for GPT-4o-mini and GPT-4.1-mini, respectively. All-Evidence Recall rises from 76.49% to 84.41% and 86.71%, so answer gains coincide with more complete annotated support. Derived-only reaches only 55.97% and 71.17% accuracy, indicating that compiled views complement direct access to original records. The variants share the evidence-budget cap, but their delivered evidence lengths differ. Paired uncertainty estimates appear in Appendix D. In a question about John's children, Atomic-only retrieves Kyle but misses Sara; a se-

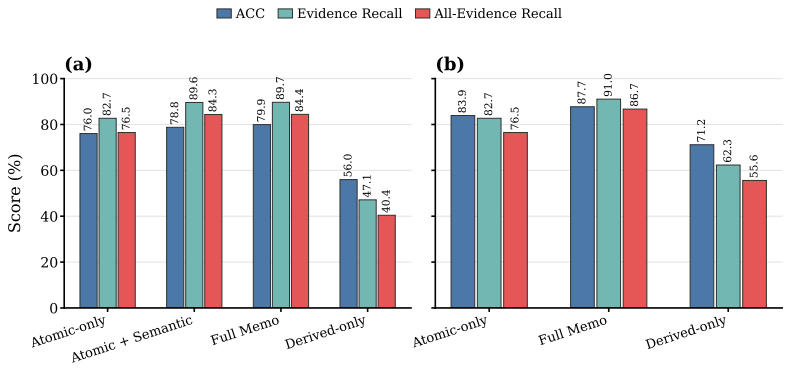

Figure 3: LoCoMo route ablations with (a) GPT-4o-mini and (b) GPT-4.1-mini. ACC uses 1,540 questions; ER/AER use 1,527 annotated questions. Full MEMO uses both routes. Atomic-only retrieves top-120 atomic items without compiled views; Derived-only removes direct atomic/raw retrieval. Atomic + Semantic reruns retrieval without temporal/transition views and replenishes candidates (Appendix D).

mantic family-related cue containing neither name recovers Sara's original record through its source reference (Appendix D.5).

To examine whether the improvement is explained solely by a larger evidence packet, we expand atomic retrieval to 480 candidates while keeping the selector, reader, judge, and evidence budget fixed. With GPT-4o-mini on LoCoMo, expanded atomic retrieval reaches 75.58% accuracy with 8,183.62 evidence tokens per question, whereas MEMO reaches 79.94% with 8,101.47 tokens. The mixed configuration retrieves up to 120 atomic and 360 derived items, matching the total retrieval-entry quota of the atomic control. Under the same selection policy and nearly equal delivered evidence lengths, multi-view retrieval improves accuracy by 4.35 percentage points (Appendix D.2). After source deduplication, the atomic and mixed pools contain 468.14 and 207.06 sources per question, respectively. The accuracy gain thus occurs with fewer distinct candidate sources under the same selector.

Evidence recovery alone does not ensure a correct answer: 63.1–76.2% of wrong LoCoMo answers with gold annotations already have all annotated sources in the evidence packet. The LongMemEval route comparison likewise improves evidence coverage without consistently improving answer accuracy. These results distinguish access to supporting records from the reader's ability to interpret and combine them (Appendix D).

5 CONCLUSION

We introduced MEMO, which combines reusable history organization with question-specific selection of original evidence under a fixed budget. Evaluations on LongMemEval-S and LoCoMo show improved answer accuracy over matched LightMem baselines, while LoCoMo component analyses support the complementarity of direct and view-based retrieval. These findings support using organized representations to improve evidence access while preserving original records as the factual basis for answering.

6 LIMITATIONS

Our evaluation focuses on question answering over text-based conversation histories that are available before memory construction. It therefore does not establish how the same organization performs when conversations continue to grow and previously recorded facts are revised. We also use fixed evidence budgets across questions, although the amount of supporting evidence needed can vary with question complexity. Extending the framework to incremental memory updates, question-adaptive evidence allocation, and multimodal conversations remains future work.

AI USE STATEMENT

We used generative AI tools, including OpenAI Codex, to assist with literature discovery and reference checking, manuscript drafting and revision, LaTeX formatting, experimental code implementation, and the generation of illustrative figures. AI assistance in manuscript preparation included suggestions on structure, wording, and interpretation of results. Experimental model assignments are listed in Section 4.1 and Appendix A, while reader and evaluation prompts are provided in Appendix F. The authors manually reviewed and tested the AI-assisted code, checked the reported results against experimental logs, and verified the technical content of the generated figures. The authors take responsibility for the final text, claims, code, figures, and reported results.

REFERENCES

Pratyay Banerjee, Masud Moshtaghi, Shivashankar Subramanian, Amita Misra, and Ankit Chadha. APEX-MEM: Agentic semi-structured memory with temporal reasoning for long-term conversational AI. In 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. 16470–16489, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.749. URL https://aclanthology.org/2026.acl-long.749/.

Chenlong Bao, Shijie Li, Minghao Hu, Ming Qiao, Bin Zhang, Jin-Tao Tang, Shasha Li, and Ting Wang. R2A-TLS: Reflective retrieval-augmented timeline summarization with causal-semantic integration. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 752–766, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.40. URL https://aclanthology.org/2025.findings-emnlp.40/.

Adrián Bazaga, Rexhina Blloshmi, Bill Byrne, and Adrià de Gispert. Learning to reason over time: Timeline self-reflection for improved temporal reasoning in language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28014–28033. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1358. URL https://aclanthology.org/2025.acl-long.1358/.

Baolong Bi, Shaohan Huang, Yiwei Wang, Tianchi Yang, Zihan Zhang, Haizhen Huang, Lingrui Mei, Junfeng Fang, Zehao Li, Furu Wei, Weiwei Deng, Feng Sun, Qi Zhang, and Shenghua Liu. Context-DPO: Aligning language models for context-faithfulness. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10280–10300. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.536. URL https://aclanthology.org/2025.findings-acl.536/.

Rong Cheng, Jinyi Liu, Yan Zheng, Fei Ni, Jiazhen Du, Hangyu Mao, Fuzheng Zhang, Bo Wang, and Jianye Hao. DualRAG: A dual-process approach to integrate reasoning and retrieval for multi-hop question answering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31877–31899. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.acl-long.1539. URL https://aclanthology.org/2025.acl-long.1539/.

Yiruo Cheng, Kelong Mao, Ziliang Zhao, Guanting Dong, Hongjin Qian, Yongkang Wu, Tetsuya Sakai, Ji-Rong Wen, and Zhicheng Dou. CORAL: Benchmarking multi-turn conversational retrieval-augmented generation. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1308–1330, Albuquerque, New Mexico, April 2025b. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.72. URL https://aclanthology.org/2025.findings-naacl.72/.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. In *ECAI 2025 - 28th European Conference on Artificial Intelligence*, volume 413 of *Frontiers in Artificial Intelligence*

and Applications, pp. 2993–3000. IOS Press, 2025. doi: 10.3233/FAIA251160. URL https://doi.org/10.3233/FAIA251160.

María Andrea Cruz Blandón, Jayasimha Talur, Bruno Charron, Dong Liu, Saab Mansour, and Marcello Federico. MEMERAG: A multilingual end-to-end meta-evaluation benchmark for retrieval augmented generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22577–22595, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1101. URL https://aclanthology.org/2025.acl-long.1101/.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. LightMem: Lightweight and efficient memory-augmented generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=dyJ0GWpjJB.

Shangyi Geng, Wenting Zhao, and Alexander M Rush. Great memory, shallow reasoning: Limits of kNN-LMs. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 471–482, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-190-2. doi: 10.18653/v1/2025.naacl-short.40. URL https://aclanthology.org/2025.naacl-short.40/.

Weidong Guo, Dakai Wang, Zixuan Wang, Hui Liu, and Yu Xu. T-mem: Memory that anticipates, not archives, 2026. URL https://arxiv.org/abs/2606.15405.

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

Lovisa Hagström, Sara Vera Marjanovic, Haeun Yu, Arnav Arora, Christina Lioma, Maria Maistro, Pepa Atanasova, and Isabelle Augenstein. A reality check on context utilisation for retrieval-augmented generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19691–19730, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.968. URL https://aclanthology.org/2025.acl-long.968/.

Liqi He, Zuchao Li, Hao Huang, and Ping Wang. TRACE: Traversal retrieval-augmented chain of evidence for document understanding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9806–9825. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.445. URL https://aclanthology.org/2026.acl-long.445/.

Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. HiAgent: Hierarchical working memory management for solving long-horizon agent tasks with large language model. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32779–32798, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1575. URL https://aclanthology.org/2025.acl-long.1575/.

Zhengjun Huang, Zhoujin Tian, Qintian Guo, Fangyuan Zhang, Yingli Zhou, Di Jiang, Zeying Xie, and Xiaofang Zhou. LiCoMemory: Lightweight and cognitive agentic memory for efficient long-term reasoning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 36842–36858, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1835. URL https://aclanthology.org/2026.findings-acl.1835/.

Yeonjun In, Sungchul Kim, Ryan A. Rossi, Mehrab Tanjim, Tong Yu, Ritwik Sinha, and Chanyoung Park. Diversify-verify-adapt: Efficient and robust retrieval-augmented ambiguous question answering. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1212–1233, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.56. URL https://aclanthology.org/2025.naacl-long.56/.

Zixi Jia, Qinghua Liu, Hexiao Li, Yuyan Chen, and Jiqiang Liu. Evaluating the long-term memory of large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19759–19777, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1014. URL https://aclanthology.org/2025.findings-acl.1014/.

Dongming Jiang, Yi Li, Guanpeng Li, and Bingzhe Li. MAGMA: A multi-graph based agentic memory architecture for AI 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. 36848–36865, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. 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 *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25961–25970. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1318. URL https://aclanthology.org/2025.emnlp-main.1318/.

Yannis Katsis, Sara Rosenthal, Kshitij Fadnis, Chulaka Gunasekara, Young-Suk Lee, Lucian Popa, Vraj Shah, Huaiyu Zhu, Danish Contractor, and Marina Danilevsky. mt RAG: A multi-turn conversational benchmark for evaluating retrieval-augmented generation systems. *Transactions of the Association for Computational Linguistics*, 13:784–808, 2025. doi: 10.1162/tacl.a.19. URL https://aclanthology.org/2025.tacl-1.36/.

Sangyeop Kim, Yohan Lee, Sanghwa Kim, Hyunjong Kim, and Sungzoon Cho. Pre-storage reasoning for episodic memory: Shifting inference burden to memory for personalized dialogue. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 22096–22113, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1204. URL https://aclanthology.org/2025.findings-emnlp.1204/.

Dahyun Lee, Yongrae Jo, Haeju Park, and Moontae Lee. Shifting from ranking to set selection for retrieval augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17606–17619. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.861. URL https://aclanthology.org/2025.acl-long.861/.

Yanhong Li, David Yunis, David McAllester, and Jiawei Zhou. Context-efficient retrieval with factual decomposition. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 178–194, Albuquerque, New Mexico, April 2025a. Association for Computational Linguistics. ISBN 979-8-89176-190-2. doi: 10.18653/v1/2025.naacl-short.16. URL https://aclanthology.org/2025.naacl-short.16/.

Zhenyu Li, Yike Zhang, Tengyu Pan, Yutao Sun, Zhichao Duan, Junjie Fang, Rong Han, Zixuan Wang, and Jianyong Wang. FocusLLM: Precise understanding of long context by dynamic condensing. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31087–31101, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1500. URL https://aclanthology.org/2025.acl-long.1500/.

Zhiyu Li, Shichao Song, Hanyu Wang, Simin Niu, Ding Chen, Jiawei Yang, Chenyang Xi, Huayi Lai, Jihao Zhao, Yezhaohui Wang, Junpeng Ren, Zehao Lin, Jiahao Huo, Tianyi Chen, Kai Chen, Kehang Li, Zhiqiang Yin, Qingchen Yu, Bo Tang, Hongkang Yang, Zhi-Qin John Xu, and Feiyu Xiong. Memos: An operating system for memory-augmented generation (mag) in large language models, 2025c. URL https://arxiv.org/abs/2505.22101.

Yuanhua Lin, Yile Li, Zhiyuan Zhao, Jing Shang, and Jian Sun. Hero: Human-profile enhanced retrieval optimization framework for long-term agent memory, 2026. URL https://arxiv.org/abs/2608.22310.

Yifan Simon Liu, Ruifan Wu, Liam Gallagher, Jiazhou Liang, Armin Toroghi, and Scott Sanner. Semantic XPath: Structured agentic memory access for conversational AI. In Greg Durrett and Ping Jian (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 286–296, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-392-0. doi: 10.18653/v1/2026.acl-demo.28. URL https://aclanthology.org/2026.acl-demo.28/.

Weitao Ma, Xiaocheng Feng, Lei Huang, Xiachong Feng, Zhanyu Ma, Jun Xu, Jiuchong Gao, Jinghua Hao, Renqing He, and Bing Qin. Fine-mem: Fine-grained feedback alignment for long-horizon memory management. 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. 19666–19684, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.900. URL https://aclanthology.org/2026.acl-long.900/.

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

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

Noriki Nishida, Rumana Ferdous Munne, Shanshan Liu, Narumi Tokunaga, Yuki Yamagata, Fei Cheng, Kouji Kozaki, and Yuji Matsumoto. Dissecting GraphRAG: A modular analysis of knowledge structuring for factoid question answering. *Transactions of the Association for Computational Linguistics*, 14:627–655, 2026. doi: 10.1162/tacl.a.615. URL https://aclanthology.org/2026.tacl-1.29/.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Jianfeng Gao. SeCom: On memory construction and retrieval for personalized conversational agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xKDZAW0He3.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory, 2025. URL https://arxiv.org/abs/2501.13956.

Yiting Shen, Kun Li, Wei Zhou, and Songlin Hu. Mem2ActBench: A benchmark for evaluating long-term memory utilization in task-oriented autonomous 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. 8173–8190, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.370. URL https://aclanthology.org/2026.acl-long.370/.

Zhiyu Shen, Ziming Wu, Fuming Lai, Shaobing Lian, and Yanghui Rao. MemBuilder: Reinforcing LLMs for long-term memory construction via attributed dense rewards. 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. 27868–27887, San Diego, California, United States, July 2026b. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1284. URL https://aclanthology.org/2026.acl-long.1284/.

Boheng Sheng, Jiacheng Yao, Meicong Zhang, and Guoxiu He. Dynamic chunking and selection for reading comprehension of ultra-long context in large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31857–31876, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1538. URL https://aclanthology.org/2025.acl-long.1538/.

Hao Sun, Hengyi Cai, Yuchen Li, Xuanbo Fan, Xiaochi Wei, Shuaiqiang Wang, Yan Zhang, and Dawei Yin. Enhancing retrieval-augmented generation via evidence tree search. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24116–24127. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1175. URL https://aclanthology.org/2025.acl-long.1175/.

Zecheng Tang, Keyan Zhou, Juntao Li, Baibei Ji, Jianye Hou, and Min Zhang. *L-CiteEval*: A suite for evaluating fidelity of long-context models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5254–5277, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.263. URL https://aclanthology.org/2025.acl-long.263/.

Zihao Tang, Xin Yu, Ziyu Xiao, Zengxuan Wen, Zelin Li, Jiaxi Zhou, Hualei Wang, Haohua Wang, Haizhen Huang, Weiwei Deng, Feng Sun, and Qi Zhang. Mnemis: Dual-route retrieval on hierarchical graphs for long-term LLM memory. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23914–23928. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1096. URL https://aclanthology.org/2026.acl-long.1096/.

Md Nayem Uddin, Kumar Shubham, Eduardo Blanco, Chitta Baral, and Gengyu Wang. From recall to forgetting: Benchmarking long-term memory for personalized agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 26814–26841, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1337. URL https://aclanthology.org/2026.findings-acl.1337/.

Fei Wang, Xingchen Wan, Ruoxi Sun, Jiefeng Chen, and Sercan O Arik. Astute RAG: Overcoming imperfect retrieval augmentation and knowledge conflicts for large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30553–30571, Vienna, Austria, July 2025a. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1476. URL https://aclanthology.org/2025.acl-long.1476/.

Zhaowen Wang, Xiang Wei, Kangshao Du, Yiting Zhang, Libo Qin, Yingjie Xia, and Li Kuang. CSTree-SRI: Introspection-driven cognitive semantic tree for multi-turn question answering over extra-long contexts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 26502–26525. Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.acl-long.1286. URL https://aclanthology.org/2025.acl-long.1286/.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/d813d324dbf0598bbdc9c8e79740ed01-Abstract-Conference.html.

Junjie Wu, Gefei Gu, Yanan Zheng, Dit-Yan Yeung, and Arman Cohan. Ref-long: Benchmarking the long-context referencing capability of long-context language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23861–23880. Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.acl-long.1162. URL https://aclanthology.org/2025.acl-long.1162/.

Zhaofen Wu, Hanrong Zhang, Fulin Lin, Wujiang Xu, Xinran Xu, Yankai Chen, Henry Peng Zou, Shaowen Chen, Weizhi Zhang, Xue Liu, Philip S. Yu, and Hongwei Wang. GAM: Hierarchical graph-based agentic memory 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. 34647–34664, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1600. URL https://aclanthology.org/2026.acl-long.1600/.

Jianfei Xiao, Xiang Yu, Chengbing Wang, Wuqiang Zheng, Xinyu Lin, Kaining Liu, Hongxun Ding, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. AlpsBench: An LLM personalization benchmark for real-dialogue memorization and preference alignment. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3489–3497. ACM, July 2026. doi: 10.1145/3805712.3808634. URL https://doi.org/10.1145/3805712.3808634.

Buqiang Xu, Yijun Chen, Jizhan Fang, Ruobin Zhong, Yunzhi Yao, Yuqi Zhu, Lun Du, and Shumin Deng. StructMem: Structured memory for long-horizon behavior in LLMs. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 122–146. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.acl-short.12. URL https://aclanthology.org/2026.acl-short.12/.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, volume 38, pp. 20004–20031, 2025. doi: 10.52202/085713-0593. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/19909c36f51abc4856b4560aff3d36d6-Abstract-Conference.html.

Xiucheng Xu, Bingbing Xu, Xueyun Tian, Zihe Huang, Rongxin Chen, Yunfan Li, and Huawei Shen. Chain-of-memory: Lightweight memory construction with dynamic evolution 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. 11618–11631, San Diego, California, United States, July 2026b. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.534. URL https://aclanthology.org/2026.acl-long.534/.

Ruozhen Yang, Yucheng Jiang, Yueqi Jiang, Priyanka Kargupta, Yunyi Zhang, and Jiawei Han. Grounding agent memory in contextual intent. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 12008–12042, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.584. URL https://aclanthology.org/2026.findings-acl.584/.

Juexiang Ye, Xue Li, Xinyu Yang, Chengkai Huang, Lanshun Nie, Lina Yao, and Dechen Zhan. MemWeaver: Weaving hybrid memories for traceable long-horizon agentic reasoning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 12928–12956, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.630. URL https://aclanthology.org/2026.findings-acl.630/.

Yi Yu, Liuyi Yao, Yuexiang Xie, Qingquan Tan, Jiaqi Feng, Yaliang Li, and Libing Wu. Agentic memory: Learning unified long-term and short-term memory management for large language model 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. 21457–21483, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.981. URL https://aclanthology.org/2026.acl-long.981/.

Long Yuan, Kaiwen Tian, Zi Chen, Bolong Zheng, and Chuan Ma. HiGoE: Hierarchical graph of evidence to enhance retrieval-augmented generation for long-context summarization. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19703–19724. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.902. URL https://aclanthology.org/2026.acl-long.902/.

Haozhen Zhang, Tao Feng, and Jiaxuan You. Graph of records: Boosting retrieval augmented generation for long-context summarization with graphs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23780–23799. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.acl-long.1159. URL https://aclanthology.org/2025.acl-long.1159/.

Jiaquan Zhang, Chaoning Zhang, Shuxu Chen, Zhenzhen Huang, Pengcheng Zheng, Zhicheng Wang, Ping Guo, Fan Mo, Sung-Ho Bae, Jie Zou, Jiwei Wei, and Yang Yang. Lightweight LLM agent memory with small language models. 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. 12914–12929, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.588. URL https://aclanthology.org/2026.acl-long.588/.

Qinggang Zhang, Zhishang Xiang, Yilin Xiao, Le Wang, Junhui Li, Xinrun Wang, and Jinsong Su. FaithfulRAG: Fact-level conflict modeling for context-faithful retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21863–21882. Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.acl-long.1062. URL https://aclanthology.org/2025.acl-long.1062/.

Yujie Zhang, Weikang Yuan, and Zhuoren Jiang. Bridging intuitive associations and deliberate recall: Empowering LLM personal assistant with graph-structured long-term memory. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 17533–17547, Vienna, Austria, July 2025c. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.901. URL https://aclanthology.org/2025.findings-acl.901/.

Xiaochen Zhao, Kaikai Wang, Xiaowen Zhang, Chen Yao, and Aili Wang. Hymem: Hybrid memory architecture with dynamic retrieval scheduling, 2026. URL https://arxiv.org/abs/2602.13933.

Rongzhi Zhu, Xiangyu Liu, Zequn Sun, Yiwei Wang, and Wei Hu. Mitigating lost-in-retrieval problems in retrieval augmented multi-hop question answering. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22362–22375, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1089. URL https://aclanthology.org/2025.acl-long.1089/.

Appendix guide. Appendix A specifies the experimental protocol; Appendix B gives implementation details and validation; Appendix C reports additional results; Appendix D examines mechanisms and errors; Appendix E provides cost accounting; and Appendix F reproduces the prompts.

A Experimental Protocol and Datasets

This appendix provides the protocol details, additional result tables, and accounting definitions used for the experiments in Section 4. The main paper reports the primary comparison at the 8K evidence budget, while the appendix records the exact benchmark populations, numerator and denominator counts, route-level ablations, and cost ledgers used to produce those results.

A.1 Benchmark Populations

Table 4 summarizes the evaluated datasets. LongMemEval-S is evaluated over 500 independently built histories. For LoCoMo, we use the 1,540 non-adversarial questions defined in Section 4.1 and score all methods with the same LLM judge. Adversarial questions are outside this evaluation scope. Table 5 gives the exact question-type distribution and the ability tested by each category.

Table 4: Dataset populations used in the main experiments.

Dataset	Histories	Processed sessions	Questions	Notes
LongMemEval-S	500	23,867	500	Each question is associated with an independently built history.
LoCoMo	10	272	1,540	Cat1–4 are used as the primary semantic QA population.

Table 5: Question types and counts in the evaluated benchmark files. For LongMemEval-S, counts follow the released `question_type` field used by the category-level results. LoCoMo Cat5 is reported for dataset completeness but excluded from the primary 1,540-question population.

Dataset	Question type	Count	What the question tests
LongMemEval-S	Single-session-user	70	Recall of information stated by the user within one session.
LongMemEval-S	Single-session-assistant	56	Recall of information stated by the assistant within one session.
LongMemEval-S	Single-session-preference	30	Use of remembered user information to produce a personalized response.
LongMemEval-S	Multi-session	133	Aggregation or comparison of user information across two or more sessions.
LongMemEval-S	Temporal-reasoning	133	Reasoning over explicit time references and session timestamp metadata.
LongMemEval-S	Knowledge-update	78	Recognition of changes in a user's state and use of the updated information.
LoCoMo	Cat1: Multi-hop	282	Synthesis of information from multiple dialogue sessions.
LoCoMo	Cat2: Temporal	321	Temporal reasoning using time-related cues in the conversation.
LoCoMo	Cat3: Open-domain	96	Combination of conversational evidence with commonsense or world knowledge.
LoCoMo	Cat4: Single-hop	841	Answering from evidence contained in one dialogue session.
LoCoMo	Cat5: Adversarial	446	Detection of misleading or unanswerable premises; excluded from primary results.

The six LongMemEval-S rows sum to 500 and are the exact denominators used in Table 2. The benchmark also defines abstention as an ability: 30 of the 500 questions are false-premise variants whose IDs end in `_abs`. In the released file used here, these items retain their source `question_type`: six are included under Single-session-user, twelve under Multi-session, six under Temporal-reasoning, and six under Knowledge-update. Thus an ability-oriented decomposition is 470 answerable questions plus 30 abstention questions, but adding a separate 30-row category to the six released-label counts would double count those items. For LoCoMo, Cat1–4 sum to the primary 1,540 questions; including Cat5 gives 1,986 total questions.

A.2 Model Configuration

All main experiments use matched build and reader models. The judge is held fixed across model groups so that changes in answer accuracy reflect the memory and reader setting rather than a change in evaluator. Table 6 lists the model assignment for each stage.

Table 6: Model assignment for the matched-backbone experiments.

Group	Build model	Reader model	Judge model
GPT-4o-mini	`gpt-4o-mini-2024-07-18`	`gpt-4o-mini-2024-07-18`	`gpt-4o-mini-2024-07-18`
GPT-4.1-mini	`gpt-4.1-mini-2025-04-14`	`gpt-4.1-mini-2025-04-14`	`gpt-4o-mini-2024-07-18`
Gemini-2.5-Flash	Gemini-2.5-Flash	Gemini-2.5-Flash	`gpt-4o-mini-2024-07-18`

A.3 Baselines and Evaluation Population

FullContext sends the serialized conversation history directly to the reader. NaiveRAG retrieves raw or chunk-level text with a simple retrieval interface and does not run a generative memory-construction stage. LightMem, A-MEM, Mem0, and MemOS follow their corresponding memory-construction and retrieval interfaces. For all compared rows, answer accuracy uses the same judge model, scoring prompt, and judge settings as MEMO within each benchmark and question type.

A.4 Run Completion and Failure Accounting

A Calls/Q value below one reflects incomplete runs; it does not measure a reduction in algorithmic query cost. A-MEM recorded 496, 486, and 497 physical query calls for GPT-4o-mini, GPT-4.1-mini, and Gemini-2.5-Flash, respectively. The updated GPT-4o-mini Mem0 run records 490 physical query calls and 278 correct answers out of 500 questions. GPT-4o-mini MemOS produced judged answers for 495 histories. Aggregate accuracy uses the full 500-question population, with unavailable answers counted as incorrect and no reader or judge call fabricated for them.

MEMO records reader failures, judge failures, retry attempts, and provenance checks as separate ledger entries. A failed reader or judge call is not replaced by a fabricated answer or verdict. When an approved content-filter recovery is used, the original failed attempt and the recovery attempt remain in the ledger. This policy affects accounting but does not change the build artifacts.

B Implementation Details

MEMO's construction and retrieval modules run alongside the baseline implementations in the evaluation framework. They preserve the original conversation records, construct reusable navigation items from the history, and resolve every selected item back to raw records before the reader is called. The official LightMem baseline path is kept unchanged.

B.1 Source Records and Identity

The original-record store is labeled RawStore in Figure 2. Each record contains a durable `source_id`, dataset identifiers, speaker, timestamp when available, and the original dialogue text. For LoCoMo, the source key is derived from the sample identifier and the dataset-native dialogue-turn identifier, while the original identifiers are also stored as separate fields. This prevents collisions across conversations and allows retrieval results, packed evidence, and gold evidence analysis to use the same source identity layer. RawStore files are persisted as JSONL and are read-only during retrieval and answer generation.

B.2 Construction and Indexing

The build model receives one dialogue session at a time. A fixed structured output contains at most three evidence groups and four state snapshots per session, with no more than six source turns in an item. An evidence group yields a concise semantic surface and, when supported by multiple cited turns, a temporal surface. State snapshots with the same normalized entity and attribute are

compared across sessions to form transition surfaces when their values change. MEMO admits at most three semantic and three temporal surfaces per session and 32 transition surfaces per history. A semantic surface whose cosine similarity to one of its single raw sources is at least 0.92 is removed as redundant; the duplicate threshold between semantic and temporal surfaces from the same group is 0.95. The complete construction instruction appears in Appendix F.

Atomic records and admitted navigation surfaces are embedded with `all-MiniLM-L6-v2`. Vectors are L_2 normalized, and retrieval uses cosine similarity. Every searchable row stores its surface type and complete set of durable source IDs. The three main model groups use the same construction schema, deterministic projection rules, embedding model, and thresholds; only the build model changes as specified in Table 6.

Table 7: Shared construction and retrieval settings across the three model groups.

Setting	Value / scope
Construction input	One session per request
Evidence groups / state snapshots	At most 3 / 4 per session
Source turns per generated item	At most 6
Semantic / temporal surfaces	At most 3 / 3 per session
Transition surfaces	At most 32 per history
Semantic–raw redundancy threshold	Cosine similarity 0.92
Same-group surface duplicate threshold	Cosine similarity 0.95
Embedding model	`all-MiniLM-L6-v2`
Vector normalization / retrieval score	L_2 normalization / cosine similarity
Atomic / compiled retrieval depth	120 / 360
Cluster-leader matching threshold	Cosine similarity 0.90
Evidence tokenizer	`o200k_base`
Evidence budget / output limit	4,096 / 512; 8,192 / 512; 16,384 / 1,024

B.3 RETRIEVAL AND DYNAMIC SOURCE SELECTION

Candidate retrieval and fusion. For each question, MEMO embeds the question with the same embedding model and ranks all rows by cosine similarity, with retrieval ID as the deterministic tie breaker. The direct route retains the highest-ranked 120 atomic rows and the compiled route retains the highest-ranked 360 semantic, temporal, or transition rows. The retained rows are merged in their original global similarity order. No query-stage language model is used.

Clustering and source actions. The selector in Equation 1 processes retained candidates in global query-relevance order. With normalized embeddings, each candidate is compared only with existing cluster leaders and assigned to the first leader whose cosine similarity is at least 0.90. If none matches, it becomes a new leader. Each candidate belongs to exactly one cluster; assignments depend on processing order. A cluster's weight is the squared positive query similarity of its leader, its highest-ranked member. Candidates with identical source-ID sets form one action regardless of source-ID order. The first candidate represents that action and supplies the order of its sources. Actions retain their first-appearance order, which also breaks priority ties.

Dynamic source completion. For candidate j, let $M_j(A) = S_j \setminus S(A)$ denote its missing sources. Action a completes this candidate exactly when $M_j(A) \subseteq S_a$, equivalently $S_j \subseteq S(A) \cup S_a$. Its gain includes every previously uncovered cluster containing a candidate completed by this addition, not only clusters containing the action's own representative or identical-source candidates. A cluster is covered if at least one member has all of its sources selected. For example, after selecting $\{x, y\}$, adding $\{z\}$ can complete candidates supported by both $\{x, z\}$ and $\{z\}$, crediting both clusters if they were previously uncovered. Coverage is recomputed from source completeness, so previously covered clusters remain covered.

Costs, admission, and stopping. For each source block b_s, the runner precomputes $\widehat{c}_s = \max(1, \mathrm{Tokens}(b_s + \backslash\mathtt{n}\backslash\mathtt{n}))$ using `o200k_base`. Blocks include timestamp, speaker, original text, and formatting. Priority uses the sum of these estimates for previously unselected sources; the denominator is not an exact count of the complete packet. Actions with nonpositive gain or added

cost are ineligible. Before admission, all new sources of an action are appended to the selected IDs, and the complete tentative packet is rendered and counted exactly. This count, including benchmark-specific role grouping, determines whether the packet satisfies B. An over-budget action is disabled only for the current selected-source state, without adding partial sources or updating coverage. After any successful admission, all actions are reconsidered with updated missing sources, coverage, incremental costs, and priorities. Selection stops when no positive-gain, positive-cost action can pass the exact budget check.

Algorithm 1 summarizes the selection loop. The coverage function $\mathcal{K}$ and priority ρ_q are defined in Section 3.4. Estimated source costs determine priority; the exact rendered packet length determines admission.

Algorithm 1 Dynamic source selection under an exact evidence budget

Input: Ranked candidates $\mathcal{C}(q)$, original records, source-cost estimates $\widehat{c}_s$, and evidence budget B.
Output: Ordered source IDs S_q and original evidence E_q.

1. Cluster candidates in relevance order and assign weights w_k. Group identical source sets into actions, retaining first-appearance order.
2. Initialize selected actions $A \leftarrow \emptyset$ and ordered source list $L \leftarrow [\,]$.
3. **Repeat:**
   (a) Recompute each action's new sources, newly covered clusters, gain, and estimated incremental cost relative to $S(A)$.
   (b) Retain actions with positive gain and cost. Sort by descending $\rho_q(a \mid A)$, breaking ties by first appearance. Set $admitted \leftarrow$ false.
   (c) **For each** action a in this order:
      i. Form L' by appending only new source IDs to L, in the representative candidate's source order.
      ii. Render the complete original-evidence packet E' from L', including benchmark-specific role grouping.
      iii. **If** $\mathrm{Tokens}(E') \leq B$: set $A \leftarrow A \cup \{a\}$, $L \leftarrow L'$, and $admitted \leftarrow \mathrm{true}$; **break** the action loop. Otherwise continue to the next action.
   (d) **If** no action was admitted, stop. Otherwise restart the iteration, reconsidering all actions under the updated source state.
4. Return $S_q = L$ and $E_q = \mathrm{Render}(L)$.

B.4 Evidence Rendering and Validation

Evidence rendering. New source IDs are appended in action-admission order, without duplicates. Within an action, they follow the source-ID order of its representative candidate. LoCoMo concatenates source blocks in this order, using `[timestamp] speaker: original text`. LongMemEval formats each block with a timestamp line followed by speaker and original text, then applies the existing role-grouped rendering to separate user and assistant records. Thus source admission order and final display order need not coincide. The same complete-packet rendering is used for admission checks and the reader context; no original record is truncated.

Evidence budgets. The budget B covers only hydrated raw evidence supplied to the reader. It excludes the reader instruction, question, and output. The primary setting is $B = 8192$ with a 512-token output limit. The 4K/512 and 16K/1024 settings reuse the same built memory and retrieval rules. Because the 16K setting also changes the output limit, it is a complete resource configuration rather than a single-factor budget ablation.

Source integrity and budget validation. Table 8 reports source-resolution and budget checks for the six primary runs. These checks establish reference integrity and packet compliance, not the semantic correctness of every derived item.

Table 8: Provenance and packet validation over the six primary 8K MEMO runs: three model groups on two datasets.

Check	Requirement	Status
Query-agnostic build	Build uses dialogue text and metadata, without future questions, answers, or gold evidence.	Pass
RawStore resolution	All 948,209 selected source references resolve in the corresponding RawStore.	100%
Raw evidence rendering	Reader-visible evidence is hydrated from original conversation records.	Pass
Budget enforcement	All 6,120 rendered packets satisfy the configured 8K evidence budget.	100%
Retrieval LLM calls	Main retrieval and source selection use embeddings, index operations, and deterministic packing.	0

The denominator contains 500 LongMemEval and 1,540 LoCoMo packets for each of the three model groups. Source references are counted once per packet occurrence; the same raw record may be selected for different questions. Every selected reference is resolved before rendering, and each rendered packet is counted with the configured tokenizer before its reader call. The component-ablation manifests apply the same checks and also record provenance resolution of 1.0.

C ADDITIONAL MAIN RESULTS

Table 9 gives the exact correct-answer counts for all budget settings, and Figure 4 maps those counts to the physical tokens processed by the reader. All three points reuse the same built memory. The 16K configuration also raises the output limit from 512 to 1,024 tokens and therefore represents a complete operating point rather than a single-factor input-budget ablation.

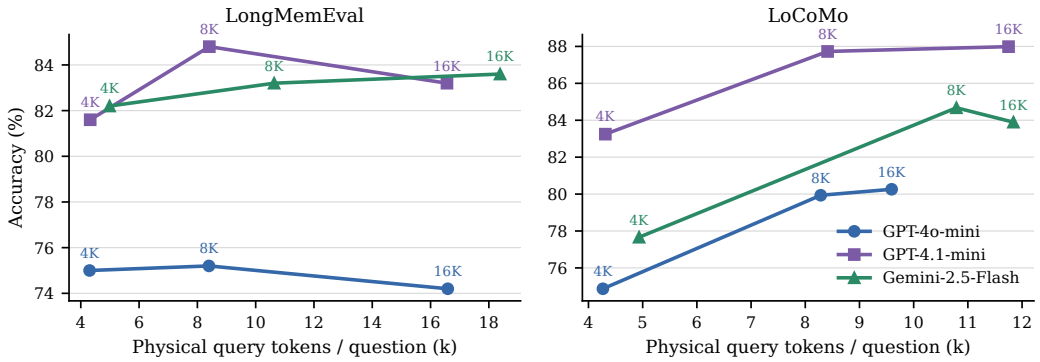

Figure 4: Accuracy versus physical query tokens per question. Point labels denote the configured answer-facing evidence budget.

Table 9: Exact correct-answer counts underlying the budget curves. Parentheses report accuracy in percent.

Dataset / model	4K / 512	8K / 512	16K / 1024
LongMemEval / GPT-4o-mini	375/500 (75.00)	376/500 (75.20)	371/500 (74.20)
LongMemEval / GPT-4.1-mini	408/500 (81.60)	424/500 (84.80)	416/500 (83.20)
LongMemEval / Gemini-2.5-Flash	411/500 (82.20)	416/500 (83.20)	418/500 (83.60)
LoCoMo / GPT-4o-mini	1,153/1,540 (74.87)	1,231/1,540 (79.94)	1,236/1,540 (80.26)
LoCoMo / GPT-4.1-mini	1,282/1,540 (83.25)	1,351/1,540 (87.73)	1,355/1,540 (87.99)
LoCoMo / Gemini-2.5-Flash	1,196/1,540 (77.66)	1,304/1,540 (84.68)	1,292/1,540 (83.90)

Figure 5 breaks the 8K results down by benchmark category. On LoCoMo, Single-hop is the strongest category for all three readers, and Multi-hop also benefits strongly from the reusable navigation structure. Open-domain remains the most difficult category, reaching 59.38% with Gemini-2.5-Flash, while Temporal accuracy varies across readers. On LongMemEval, all three readers perform well on single-session extraction, while multi-session and temporal questions remain harder.

Preference accuracy varies sharply across readers; this category also has the smallest population ($n = 30$), so its percentage is correspondingly sensitive to individual items.

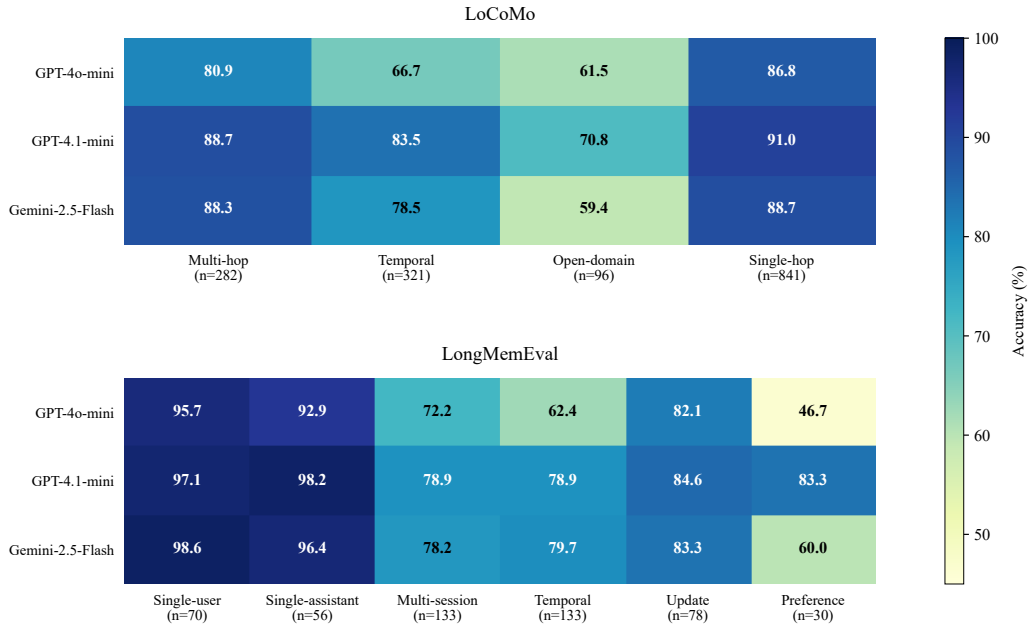

Figure 5: Question-category accuracy at the primary 8K configuration. The number of evaluated questions is shown below each category.

D Mechanism and Error Analysis

D.1 Ablation Definitions and Evidence Metrics

The route study tests whether derived views add useful access paths beyond direct atomic retrieval. All variants use the same reader and evidence-budget cap. Table 10 distinguishes route removal with renewed retrieval from masking within a fixed candidate pool.

Table 10: Candidate-pool construction for the route ablations.

Variant	Retrieval and candidate-pool rule
Full MEMO	Retrieve up to 120 atomic and 360 compiled items (semantic, temporal, and transition).
Atomic-only	Remove the compiled route; retrieve up to 120 atomic items. The expanded control uses 480.
Derived-only	Remove the direct route; retrieve compiled items only.
Atomic + Semantic	Remove temporal/transition types and recompute the eligible top-ranked pool, replenishing candidates.
Temp./Trans. fixed mask	Mask temporal/transition rows in the Full pool; preserve the remaining ranks and do not refill the vacated slots.

Evidence metrics. For a question with gold source set G and retrieved source set R, Evidence Recall is $|G \cap R|/|G|$. All-Evidence Recall is one when $G \subseteq R$ and zero otherwise. These metrics are computed after retrieval outputs are mapped to durable dataset source IDs. They are reported only for questions with gold evidence annotations.

ACC uses all 1,540 LoCoMo questions; ER and AER use the 1,527 questions with gold evidence annotations.

D.2 Route Ablation Results

Full MEMO improves accuracy and evidence recall over Atomic-only for both readers, while Derived-only performs substantially worse. With GPT-4o-mini, Atomic + Semantic reaches 78.77% in Figure 3, whereas the fixed-pool mask reaches 79.09% in Table 11. These are different interventions, as specified in Table 10.

Table 11: Route ablations on LoCoMo. ER denotes Evidence Recall and AER denotes All-Evidence Recall. The fixed-pool mask removes temporal and transition rows from the Full MEMO pool without retrieving replacements.

Model	Variant	Correct / N	ACC	ER	AER	Evd. tok./Q	Qry. tok./Q
GPT-4o-mini	Atomic-only	1,171 / 1,540	76.04	82.70	76.49	4,958.79	5,142.25
GPT-4o-mini	Full MEMO	1,231 / 1,540	79.94	89.67	84.41	8,101.47	8,285.24
GPT-4o-mini	Temp./Trans. fixed mask	1,218 / 1,540	79.09	89.59	84.35	8,093.67	8,278.00
GPT-4o-mini	Derived-only	862 / 1,540	55.97	47.13	40.41	5,588.95	5,772.35
GPT-4.1-mini	Atomic-only	1,292 / 1,540	83.90	82.70	76.49	4,958.79	5,204.16
GPT-4.1-mini	Full MEMO	1,351 / 1,540	87.73	91.05	86.71	8,171.68	8,411.72
GPT-4.1-mini	Derived-only	1,096 / 1,540	71.17	62.32	55.60	7,637.00	7,861.59

Expanded atomic retrieval control. Table 12 examines candidate composition with the dynamic selector held fixed. All three configurations use GPT-4o-mini for build and reading, the same `main_judge_v1` evaluator, an 8,192-token original-evidence cap, and a 512-token output limit. Accuracy uses all 1,540 LoCoMo questions; all reader and judge calls completed successfully. Atomic-480 expands only the atomic retrieval depth. Full MEMO combines up to 120 atomic and 360 compiled entries. These quotas count retrieval entries, not distinct original sources: multiple entries can share a source, and one compiled entry can reference several sources. After source resolution and deduplication, Atomic-480 and Full MEMO yield 468.14 and 207.06 candidate sources per question on average. Both deliver nearly the full evidence budget. The 4.35-point accuracy difference therefore supports mixed retrieval under the shared dynamic selection policy, rather than establishing a comparison between selection policies.

Table 12: Expanded atomic retrieval on LoCoMo with GPT-4o-mini and a fixed dynamic selector. Evidence tokens are the actual mean original-evidence length delivered to the reader.

Configuration	Correct / N	ACC (%)	Evd. tok./Q
Atomic-only (top-120)	1,171 / 1,540	<u>76.04</u>	4,958.79
Atomic-only (top-480)	1,164 / 1,540	75.58	8,183.62
Full MEMO (120 atomic + 360 derived)	1,231 / 1,540	***79.94***	8,101.47

D.3 Paired Comparisons and Uncertainty

The paired results in Table 13 show that the Full route improves over Atomic-only for both readers. Removing the raw route produces a much larger loss, which confirms that derived navigation is complementary to direct access rather than a replacement for it. The fixed-pool mask has a smaller interval that crosses zero, so this contrast does not establish a stable aggregate contribution from temporal and transition items by themselves.

Table 13: Paired LoCoMo accuracy differences. Intervals use 10,000 paired conversation-cluster bootstrap samples over the ten conversations (seed 42).

Reader	Comparison	Δ ACC (pp)	Wins / losses	95% CI (pp)
GPT-4o-mini	Full − Atomic-only	+3.90	119 / 59	[2.50, 5.49]
GPT-4o-mini	Full − fixed-pool mask	+0.84	54 / 41	[−0.14, 1.95]
GPT-4o-mini	Full − Derived-only	+23.96	412 / 43	[20.85, 26.35]
GPT-4.1-mini	Full − Atomic-only	+3.83	103 / 44	[2.24, 5.29]
GPT-4.1-mini	Full − Derived-only	+16.56	288 / 33	[13.26, 19.79]

For each comparison, correctness differences are computed question by question before resampling. One bootstrap replicate samples ten conversations with replacement and includes every question belonging to each sampled conversation. Its accuracy difference is the sum of paired correctness differences divided by the total number of sampled questions, so conversation clusters with different numbers of questions retain their observed sizes. The reported interval is the 2.5th to 97.5th percentile over 10,000 replicates; no multiple-comparison correction is applied.

D.4 Evidence Recovery and Answer Errors

Figure 6 separates wrong LoCoMo answers by whether every annotated source was already present in the evidence packet. For all three readers, most errors occur after complete annotated evidence has been retrieved: 69.7% for GPT-4o-mini, 63.1% for GPT-4.1-mini, and 76.2% for Gemini-2.5-Flash. The remaining errors still expose retrieval headroom, but this decomposition locates many failures after annotated-source recovery. The cases in Appendix D.5 distinguish a reader's temporal reasoning error from a mismatch with an ambiguous benchmark label.

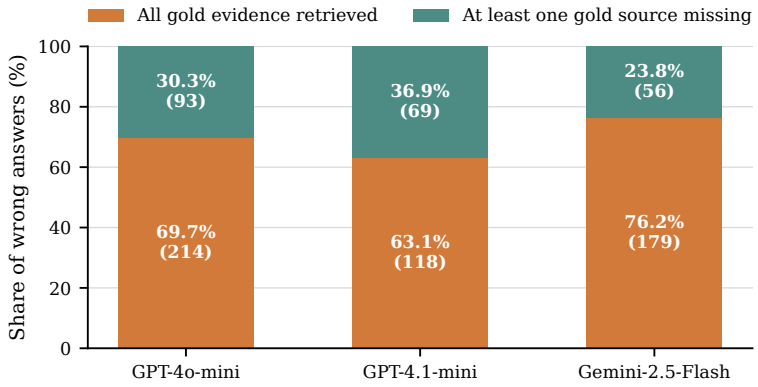

Figure 6: Decomposition of wrong LoCoMo answers at 8K for the 1,527 questions with gold evidence annotations. Counts are shown in parentheses.

LongMemEval route analysis. The same GPT-4o-mini component protocol produces a mixed QA pattern on LongMemEval: Atomic-only reaches 75.80% ACC with 95.62% ER and 92.34% AER, while Full MEMO reaches 75.20% ACC with 97.61% ER and 95.74% AER. The additional recovered support does not improve the aggregate answer score in this setting. Retrieval coverage and answer accuracy must therefore be evaluated separately when assessing the benefit of a navigation route.

D.5 Qualitative Examples

The following cases are taken from the completed GPT-4o-mini LoCoMo runs at the 8K evidence budget. They illustrate where reusable organization helps and where the remaining error occurs after the evidence has already been recovered.

D.5.1 A NAVIGATION SUCCESS CASE

Table 14: A paired case in which a semantic navigation item recovers the second source needed for the complete answer.

Field	Content
Question	What are the names of John's children?
Gold answer	Kyle, Sara.
Raw source 1	`locomo:conv-41::D8%3A4`: *Our one-year-old is so cute, his name is Kyle!*
Raw source 2	`locomo:conv-41::D22%3A7`: *That picture was from a trip we took last year for my daughter Sara's birthday.*
Navigation item	*John's family motivates him to strive for change.* → `locomo:conv-41::D22%3A7`
Atomic-only	*The only child's name mentioned in the conversation is Kyle.* (wrong)
Full MEMO	*John has two children, named Kyle and Sara.* (correct)

Direct retrieval ranks the Kyle record at 20 but the Sara record at 206, outside the Atomic-only depth of 120. The semantic item ranks fourth and carries the durable source ID of the Sara record. Full MEMO therefore hydrates both original turns and returns the complete answer, while Atomic-only omits Sara and is judged wrong. The navigation text does not contain Sara's name; its contribution is the additional path to the original record.

D.5.2 FAILURE CASES

Temporal normalization after successful retrieval. For the question, When did Joanna plan to go over to Nate's and share recipes?, the gold answer is 5 November 2022. MEMO retrieves the complete gold evidence, timestamped 4 November 2022: Thanks so much, Nate! Sure! I'll come over tomorrow if that's fine. Evidence Recall and All-Evidence Recall are both 100 percent, but the reader answers 5 June 2022. The judge marks the answer wrong. The retrieval stage succeeded; the error is the reader's conversion of a relative date into an absolute date.

Answer mismatch after successful retrieval. For the question, What is Caroline's relationship status?, the selected evidence includes Caroline's statement, It'll be tough as a single parent, but I'm up for the challenge! Evidence Recall and All-Evidence Recall are both 100 percent. The reader nevertheless says that the conversation does not provide explicit relationship information and therefore does not produce the benchmark's expected answer, Single. The phrase single parent supports that annotation in the benchmark, but it need not determine a person's current partner status in every context. We therefore treat this as an answer–benchmark mismatch after successful retrieval rather than as an unambiguous logical error.

These cases clarify why retrieval metrics and answer accuracy are reported separately. Reusable navigation can recover a missing bridge, as in Table 14, but complete annotated evidence does not guarantee that the reader will perform the benchmark's expected temporal normalization or interpretation.

E COMPUTATIONAL COST

If one built history serves N questions, its cumulative cost is $C(N) = C_{\text{build}} + N\,C_{\text{query}}$, where C_{query} is the average per-question cost. Build accounting covers model calls, history preprocessing, embedding, and associated runtime. Query accounting covers question encoding, retrieval, candidate fusion, selection, original-record recovery, and reader calls. Evidence tokens are recorded separately from physical reader tokens. Changing the evidence budget reuses the same build artifacts; judge calls, tokens, and runtime are accounted for separately as evaluation costs.

The experiments separate reusable build cost, per-question query cost, and evaluation cost. Build tokens include prompt and completion tokens from construction calls. Query tokens include the reader-facing physical prompt and completion tokens recorded by the provider. Judge tokens are

used only for evaluation and are not included in build or query cost. Total build cost depends on the amount of history processed, while Build Tokens/Q also depends on how many evaluated questions share that memory. The number of questions does not, by itself, explain the construction-token ratio between MEMO and another method; that ratio also reflects their construction prompts, memory units, and update schedules.

Table 15: Construction cost profile for MEMO and LightMem at the matched 8K comparison point.

Dataset	Model	MEMO tok. (k)	LightMem tok. (k)	Ratio	MEMO calls
LongMemEval	GPT-4o-mini	97,122.736	19,654.218	4.94	23,875
LongMemEval	GPT-4.1-mini	98,077.872	22,390.584	4.38	23,891
LongMemEval	Gemini-2.5-Flash	113,132.100	27,972.439	4.04	23,923
LoCoMo	GPT-4o-mini	836.171	879.104	0.95	272
LoCoMo	GPT-4.1-mini	847.158	880.702	0.96	272
LoCoMo	Gemini-2.5-Flash	982.042	1,221.658	0.80	272

Table 16: Reader-token comparison between MEMO and LightMem at the 8K operating point.

Dataset	Model	MEMO query tokens/Q	LightMem query tokens/Q	Ratio
LongMemEval	GPT-4o-mini	8,397.482	838.414	10.02
LongMemEval	GPT-4.1-mini	8,410.384	853.068	9.86
LongMemEval	Gemini-2.5-Flash	10,632.806	1,391.880	7.64
LoCoMo	GPT-4o-mini	8,285.235	2,687.980	3.08
LoCoMo	GPT-4.1-mini	8,411.721	3,052.087	2.76
LoCoMo	Gemini-2.5-Flash	10,788.375	3,495.177	3.09

The Time/Q column in Table 17 adds the recorded query embedding, history loading and normalization, ranking, selection, and physical reader-request durations, divided by the full question population. It is a processing-time total from concurrent batch runs, not single-request wall-clock latency. Recorded retries and failed attempts are included when present, and model groups and budget settings were executed under different API load. The non-monotonic GPT-4o-mini values, including 33.605 seconds at 8K, should therefore be used for accounting rather than as a controlled latency comparison.

For the updated GPT-4o-mini Mem0 run, 490 of 500 histories have complete build archives. Their observed usage is 3,837,531,716 prompt tokens, 630,029,429 completion tokens, and 4,029,141 build calls, with usage recorded for every archived request. The build estimates in Table 17 scale these totals by $500/490$; the ten unclosed histories are not included in the observed totals. The archives comprise 409 histories using `mem0_joint_checkpoint_v1` and 81 using `mem0_joint_checkpoint_skip_terminal_v1`. For LoCoMo MemOS with GPT-4o-mini, the ten completed histories total 9,306,485 prompt tokens and 882,372 completion tokens across 5,882 build calls; Table 3 reports this observed usage.

For LongMemEval MemOS, build totals are extrapolated from completed-history means to 500 histories, separately for each model. GPT-4o-mini, GPT-4.1-mini, and Gemini-2.5-Flash have 1, 70, and 66 completed histories, with 1,072,758, 72,094,513, and 70,293,478 observed build tokens, respectively. Recorded failed or unclosed attempts add 0, 2,055,280, and 111,960,039 tokens; these are excluded from the completed-history means. Thus the corresponding observed totals, including these attempts, are 1,072,758, 74,149,793, and 182,253,517 tokens across 550, 36,179, and 84,701 calls. Requests without usage records are not counted. The GPT-4o-mini projection is based on only one completed history and does not establish a representative full-run cost. Build estimates exclude reader and judge usage; per-question build tokens divide the projected total by 500.

Table 17: **Accuracy and cost on LongMemEval-S.** Build tokens are in thousands; Tokens/Q are unscaled. Per backbone: bold/underline = best/second best, excluding Query Calls; best ACC is italicized. Dashes = no build or unavailable. Mem0 GPT-4o-mini and MemOS build totals are extrapolated to 500 histories from 490 and 1/70/66 completed histories, respectively; query costs are observed.

Method	ACC (%)	Build			Query		
		Tokens (k)	Calls	Tokens/Q	Tokens/Q	Calls/Q	Time/Q (s)
GPT-4o-mini							
FullContext	62.20	–	–	–	113,431.79	1.0000	36.89
NaiveRAG	64.20	–	–	–	2,380.21	1.0040	16.57
LightMem	73.80	**19,654.22**	**11,923**	**39,308.44**	**838.41**	1.0000	16.41
A-MEM	61.60	770,050.45	493,561	1,540,100.90	3,307.75	0.9920	5.85
Mem0	55.60	4,558,735.86	4,111,368	9,117,471.72	2,745.94	0.9800	7.91
MemOS	43.20	536,379.00	275,000	1,072,758.00	873.67	1.0000	**4.61**
MEMO (B=4K, O=512)	75.00	97,122.74	23,875	194,245.47	4,300.86	1.0000	4.62
MEMO (B=8K, O=512)	***75.20***	97,122.74	23,875	194,245.47	8,397.48	1.0000	33.61
MEMO (B=16K, O=1024)	74.20	97,122.74	23,875	194,245.47	16,590.42	1.0000	7.99
GPT-4.1-mini							
FullContext	78.00	–	–	–	113,468.94	1.0040	37.24
NaiveRAG	68.40	–	–	–	2,307.92	1.0000	8.20
LightMem	75.00	**22,390.58**	**12,116**	**44,781.17**	853.07	0.9680	17.17
A-MEM	66.40	843,595.83	492,909	1,687,191.67	3,526.81	0.9720	5.08
MemOS	49.40	514,960.81	251,286	1,029,921.61	**701.54**	1.0000	**3.13**
MEMO (B=4K, O=512)	81.60	98,077.87	23,891	196,155.74	4,322.77	1.0080	4.45
MEMO (B=8K, O=512)	***84.80***	98,077.87	23,891	196,155.74	8,410.38	1.0020	6.47
MEMO (B=16K, O=1024)	83.20	98,077.87	23,891	196,155.74	16,562.87	1.0020	7.30
Gemini-2.5-Flash							
FullContext	79.20	–	–	–	130,393.91	1.0000	6.09
NaiveRAG	71.20	–	–	–	3,025.43	1.0000	2.27
LightMem	76.80	**27,972.44**	**11,997**	**55,944.88**	1,391.88	1.0000	2.32
A-MEM	67.80	689,058.50	493,026	1,378,117.00	3,343.48	0.9940	3.93
MemOS	49.40	532,526.35	247,538	1,065,052.70	**840.36**	1.0000	**1.47**
MEMO (B=4K, O=512)	82.20	113,132.10	23,923	226,264.20	4,988.43	1.0000	3.32
MEMO (B=8K, O=512)	83.20	113,132.10	23,923	226,264.20	10,632.81	1.0000	3.62
MEMO (B=16K, O=1024)	***83.60***	113,132.10	23,923	226,264.20	18,384.53	1.0000	5.44

Table 18: LightMem LoCoMo construction decomposition in the matched runs.

Model	Extraction calls	Update calls	Total calls	Extraction tokens	Update tokens
GPT-4o-mini	263	81	344	825,797	53,307
GPT-4.1-mini	263	71	334	832,471	48,231
Gemini-2.5-Flash	263	84	347	1,141,387	80,271

The difference between LongMemEval and LoCoMo follows from both history scale and amortization. LongMemEval builds many independent histories, so its total construction cost is large and is shared by only 500 benchmark questions. LoCoMo contains fewer processed sessions and many questions per conversation, so the same type of reusable organization is amortized over more questions for each built conversation. The comparison with LightMem also depends on each method's construction prompts and update schedule; LightMem makes extraction and update calls, while MEMO schedules one base compilation request per processed session and records retries or failed attempts separately. At the 8K point, this regime gives MEMO 4.04–4.94× the LightMem build tokens and 7.64–10.02× the LightMem query tokens on LongMemEval. These figures are reported separately from the LoCoMo profile because each LongMemEval history serves one benchmark question rather than a large shared question set.

F PROMPTS

F.1 BUILD COMPILER PROMPT

The same query-agnostic instruction and structured output schema are used for all three build models. Each request contains turns from exactly one session; every turn is assigned a local slot that resolves to a durable RawStore source ID after generation. The instruction is:

MEMO Build Compiler

You are the build-time compiler for MEMO. Use only the supplied turns from exactly one conversation session. The raw turn text is authoritative. Do not use or produce questions, answers, QA annotations, gold evidence, categories, evaluation labels, reader instructions, confidence, reasoning, speculation, or unsupported facts.

Return exactly one JSON object with the fixed keys `evidence_group_0`, `evidence_group_1`, `evidence_group_2`, `state_snapshot_0`, `state_snapshot_1`, `state_snapshot_2`, and `state_snapshot_3`. Each key is either null or one grounded object. A session may contain both groups and snapshots; use null only when no grounded item fits a slot.

Each EvidenceGroup contains one concise, self-contained `canonical_fact` with no unresolved pronouns. Cite one through six exact supplied `source_slots`. Keep `canonical_fact` concise, at most about 40 build tokens, factual, and useful as a retrieval key. Include short entity strings. `temporal_relation` is null or an object with `left_event`, `right_event`, and exactly one relation: `BEFORE`, `AFTER`, or `SAME_TIME`. Only provide a temporal relation when the cited turns support it directly.

Each StateSnapshot is one grounded observation with `entity`, `attribute`, `value`, `source_slots`, and `timestamp`. Copy the supplied `session_timestamp` exactly into `timestamp`. Cite only supplied source slots. StateSnapshot is an intermediate object and is not itself a searchable key.

Do not add prose outside the JSON object.

The request schema additionally requires every non-null group and snapshot to contain exactly the fields named above. Evidence groups allow at most six distinct source slots and eight entity strings. The user message serializes the conversation identifier, session identifier, session timestamp, and the session turns as `turn_000`, `turn_001`, and so on, including speaker, timestamp, and unmodified text. Outputs that cite an unavailable slot or fail source resolution are rejected rather than inserted into the index.

F.2 READER PROMPT

All MEMO configurations use the same reader instruction. The placeholders are filled with the hydrated original records, the optional reference date supplied by LongMemEval, and the current question.

MEMO Reader

Answer the question using the provided conversation evidence. When combining, comparing, or calculating information is necessary, you may include brief relevant evidence and calculations. Check that the facts you use satisfy the conditions in the question and that your conclusion agrees with those facts. If the evidence is insufficient, say what is missing rather than filling the gap with unsupported assumptions. Give a clear answer, retaining the items and qualifications needed to answer the question.

Conversation evidence:

{evidence}

Question reference date: {question_date}

Question: {question}

Answer:

The reference-date line is omitted for examples without a benchmark-provided question date. The main experiments use one reader call with temperature 0, `top_p`=1, and seed 42. The output limit is 512 tokens for the 4K and 8K configurations and 1,024 tokens for the 16K configuration.

F.3 LOCOMO LLM-AS-JUDGE PROMPT

The following prompt defines the LLM-as-judge protocol used for LoCoMo in our experiments. The judge returns a JSON object whose `label` is either `CORRECT` or `WRONG`.

LoCoMo LLM-as-Judge

Your task is to label an answer to a question as CORRECT or WRONG. You will be given the following data:

(1) a question posed by one user to another user,

(2) a gold ground-truth answer,

(3) a generated answer,

which you will score as CORRECT or WRONG.

The point of the question is to ask about something one user should know about the other user based on their prior conversations.

The gold answer will usually be a concise and short answer that includes the referenced topic, for example:

Question: Do you remember what I got the last time I went to Hawaii?

Gold answer: A shell necklace

The generated answer might be much longer, but you should be generous with your grading: as long as it touches on the same topic as the gold answer, it should be counted as CORRECT.

For time related questions, the gold answer will be a specific date, month, year, etc. The generated answer might be much longer or use relative time references such as last Tuesday or next month, but you should be generous with your grading: as long as it refers to the same date or time period as the gold answer, it should be counted as CORRECT. Even if the format differs, for example May 7th versus 7 May, consider it CORRECT if it is the same date.

Question: {question}

Gold answer: {gold_answer}

Generated answer: {generated_answer}

First, provide a short one-sentence explanation of your reasoning, then finish with CORRECT or WRONG.

Do NOT include both CORRECT and WRONG in your response, or it will break the evaluation script.

Just return the label CORRECT or WRONG in a JSON format with the key as label.

The LoCoMo judge uses `gpt-4o-mini-2024-07-18`, temperature 0, and JSON response mode. No explicit output-token cap, `top_p`, or seed is set for this request.

F.4 LongMemEval LLM-as-Judge Prompts

LongMemEval selects a rubric according to the benchmark question type. The judge returns `yes` or `no`. The five templates used in our evaluation are reproduced below.

Standard Tasks (Single-session User/Assistant and Multi-session)

I will give you a question, a correct answer, and a response from a model. Please answer yes if the response contains the correct answer. Otherwise, answer no. If the response is equivalent to the correct answer or contains all the intermediate steps to get the correct answer, you should also answer yes. If the response only contains a subset of the information required by the answer, answer no.
Question: {question}
Correct Answer: {answer}
Model Response: {response}
Is the model response correct? Answer yes or no only.

Temporal Reasoning Tasks

I will give you a question, a correct answer, and a response from a model. Please answer yes if the response contains the correct answer. Otherwise, answer no. If the response is equivalent to the correct answer or contains all the intermediate steps to get the correct answer, you should also answer yes. If the response only contains a subset of the information required by the answer, answer no. In addition, do not penalize off-by-one errors for the number of days. If the question asks for the number of days, weeks, or months, and the model makes an off-by-one error, the model response is still correct.
Question: {question}
Correct Answer: {answer}
Model Response: {response}
Is the model response correct? Answer yes or no only.

Knowledge Update Tasks

I will give you a question, a correct answer, and a response from a model. Please answer yes if the response contains the correct answer. Otherwise, answer no. If the response contains some previous information along with an updated answer, the response should be considered as correct as long as the updated answer is the required answer.
Question: {question}
Correct Answer: {answer}
Model Response: {response}
Is the model response correct? Answer yes or no only.

Single-session Preference Tasks

I will give you a question, a rubric for desired personalized response, and a response from a model. Please answer yes if the response satisfies the desired response. Otherwise, answer no. The model does not need to reflect all the points in the rubric. The response is correct as long as it recalls and utilizes the user's personal information correctly.
Question: {question}
Rubric: {answer}
Model Response: {response}
Is the model response correct? Answer yes or no only.

Abstention Tasks

I will give you an unanswerable question, an explanation, and a response from a model. Please answer yes if the model correctly identifies the question as unanswerable. The model could say that the information is incomplete, or some other information is given but the asked information is not.
Question: {question}
Explanation: {answer}
Model Response: {response}
Does the model correctly identify the question as unanswerable? Answer yes or no only.

The LongMemEval judge uses `gpt-4o-mini-2024-07-18`, temperature 0, `top_p`=0.8, and a 2,000-token output cap. All judge calls and tokens are recorded as evaluation cost rather than build or query cost.