CONTEXTFACT: REMEMBERING FACTS IN CONTEXT FOR LONG-TERM CONVERSATIONAL AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Agents built on large language models must maintain continuity across repeated interactions with users. However, long-term conversations scatter facts across turns and sessions, making it difficult to retrieve the right evidence with enough context. Memory for such conversations must therefore preserve facts with their context while keeping evidence selectively accessible. We first study how memory construction and retrieval-unit boundaries affect answerability and evidence access. Reading the entire memory bank reveals gaps that cannot be resolved by retrieval selection alone, showing that some context must be preserved during memory construction. Adjacent-turn grouping improves evidence access at matched unique-turn budgets, while wider windows do not consistently help. This motivates keeping fact-specific context local while organizing cross-session evidence separately. We propose ContextFact, which links context-preserving fact memories with source-grounded entity representations. Within sessions, it gathers the context needed to interpret each fact. Across sessions, it aligns recurring referents and revisits the source dialogue to reconstruct coherent entity representations linked to the supporting facts. At query time, ContextFact retrieves fact and entity views, reranks them independently, and uses entity-linked expansion to surface related facts; complete entity representations then provide organized cross-session evidence. With Qwen3.5-27B as the backbone, ContextFact achieves state-of-the-art performance while using only 21% of the evidence tokens consumed by the strongest evaluated baseline, obtaining 86.49% overall judge accuracy on LoCoMo and 91.00% on LongMemEval-S. Controlled ablations support context-preserving fact boundaries and source-grounded entity reconstruction, while entity readout improves multi-hop accuracy beyond adding more facts.

1 INTRODUCTION

Large language models are increasingly deployed as personal agents that interact with the same users across many conversations. Such agents must preserve user context over time without requiring users to repeatedly restate their preferences, experiences, decisions, and plans (Maharana et al., 2024; Wu et al., 2025; Zhong et al., 2024; Packer et al., 2023). The evidence needed for a response often spans multiple turns and sessions: a plan may be introduced in one conversation, qualified or revised in another, and referred to again later. Simply retaining the entire history is not sufficient, because models can struggle to locate and use relevant information in long contexts (Liu et al., 2024). Long-term memory must therefore preserve the meaning of past interactions while exposing a compact, relevant set of evidence when needed.

Existing systems construct memories through fact extraction (Chhikara et al., 2025), topical segmentation (Pan et al., 2025), or contextualized events (Zhou & Han, 2025), and organize them through linked representations (Xia et al., 2026; Sun et al., 2026). Yet a central question remains: what information should stay together in a memory, and how should related evidence remain selectively accessible? We investigate this question on LoCoMo (Maharana et al., 2024) with two controlled diagnostics. First, even when the model reads each system's complete memory bank without retrieval, several banks underperform the raw dialogue, so retrieval alone cannot explain the gap. Yet reading the entire dialogue is not ideal either: an oracle that reads only the sessions containing annotated evidence improves multi-hop and temporal accuracy. Second, under a fixed turn budget, grouping adjacent turns improves single-hop evidence access, but wider windows bring no consistent further

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.

gain. Together, these findings motivate a design principle: memory units should preserve the context needed to interpret each fact, while keeping distinct facts independently retrievable and evidence across sessions selectively accessible.

Guided by this principle, we propose *ContextFact*, which preserves the context of each fact and connects related facts across sessions through the entities they describe. Within each session, ContextFact builds one memory around each fact, gathering the context needed to interpret it, such as its reasons or outcome, even when that context appears in nonadjacent turns. Further, to bring together information about the same entity (e.g., a person, project, or object) scattered across sessions, ContextFact constructs an *entity view*. During fact extraction, it also creates session-level entity entries, each linked to the fact memories that describe that entity. It aligns these entries across sessions and revisits the source turns of the linked facts to verify identities, resolve grouping errors, and recover details missing from the extracted facts. The resulting *source-grounded entity representations* organize information across sessions while retaining links to the associated fact memories. At query time, the full configuration, *ContextFact-E*, retrieves both fact memories and entities and follows these links once: retrieved facts activate their entities, and the retrieved and activated entities surface related facts that retrieval alone would miss. The entity view thus serves two roles: its links recover facts beyond direct retrieval, and its representations give the answering model organized evidence about each entity across sessions.

We evaluate ContextFact on LoCoMo and LongMemEval-S. With Qwen3.5-27B, ContextFact-E achieves 86.49% overall judge accuracy on LoCoMo (Fig. 1) and 91.00% on LongMemEval-S. It exceeds Memora (P; Xia et al., 2026), the strongest evaluated baseline, by 2.20 and 1.80 percentage points, respectively, while using approximately 79% fewer evidence tokens on both benchmarks. The fact-only variant, ContextFact-F, also performs strongly with substantially shorter evidence inputs. Controlled comparisons on LoCoMo show that complete fact memories outperform sentence splits and atomic facts derived from the same bank in overall accuracy under the same evidence budgets. Construction ablations support rule-guided temporal annotation and source-grounded entity reconstruction, while readout ablations show that complete entity representations improve multi-hop accuracy. Our contributions are fourfold:

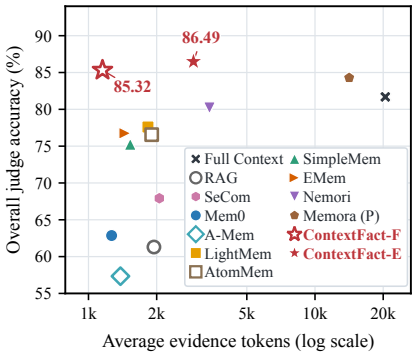
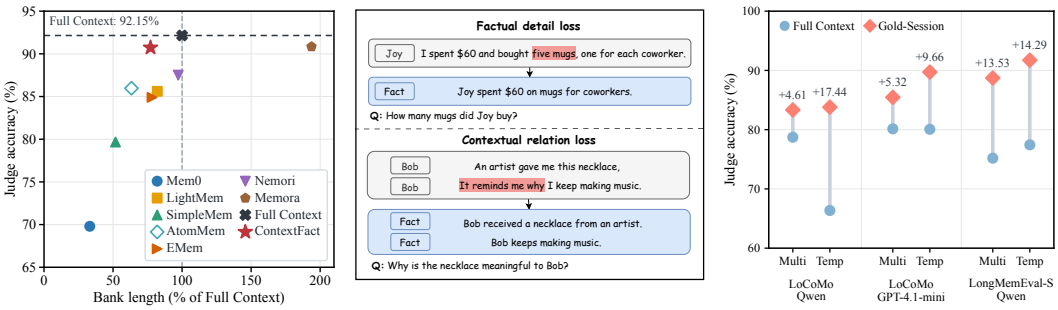


Figure 1: Accuracy versus evidence tokens on LoCoMo with Qwen3.5-27B.

1. We establish **a new design principle** for conversational memory: preserve the context needed to interpret each fact while keeping distinct evidence independently retrievable and selectively connectable across sessions (Sec. 3).
2. We propose **ContextFact**, which gathers the necessary interpretive context across nonadjacent turns while keeping distinct facts independently retrievable (Sec. 4.1).
3. We introduce **source-grounded entity representations** that consolidate scattered evidence about recurring referents across sessions, enabling both entity-linked fact discovery and direct entity-level evidence readout (Secs. 4.1 and 4.2).
4. ContextFact achieves **state-of-the-art performance** on both benchmarks with approximately 21% of the strongest baseline's evidence tokens, supported by controlled ablations (Sec. 5).

2 Related Work

Recent conversational-memory systems increasingly move beyond flat summaries or isolated facts. EMem constructs self-contained event-centric units with normalized entities, time, and source provenance (Zhou & Han, 2025), while SimpleMem compresses and consolidates interactions into compact semantic memories (Liu et al., 2026). AtomMem organizes atomic facts into event structures and temporal profiles (Yao et al., 2026). Memora connects concrete memories through primary abstractions and cue anchors (Xia et al., 2026). TriMem combines raw dialogue, atomic facts, and synthesized profiles (Sun et al., 2026). AnchorMem uses atomic facts as retrieval anchors while preserving associated source contexts (Shen et al., 2026). ContextFact combines context-preserving



(a) Single-hop answerability. (b) Construction-time loss. (c) Evidence-scope comparison.

Figure 2: Answerability after construction and selective evidence access. (a) Single-hop judge accuracy under full-bank reading on LoCoMo with Qwen3.5-27B. Bank length is normalized by Full Context, and ContextFact is shown for reference. (b) Constructed illustrations of two possible construction failures: omitting a queried factual detail and losing a relation needed for interpretation. (c) Full Context versus Gold-Session, which retains only complete sessions containing annotated supporting evidence from the same history. Annotations report the Gold-Session gain in percentage points. "Multi" refers to multi-hop/multi-session and "Temp" to temporal questions.

fact retrieval with cross-session entity organization: fact boundaries follow the context needed for interpretation, while source dialogue is revisited to resolve recurring referents and reconstruct entity-centered accounts. The two views support both entity-linked expansion and direct entity readout. See Appx. A for an extended discussion of the broader literature.

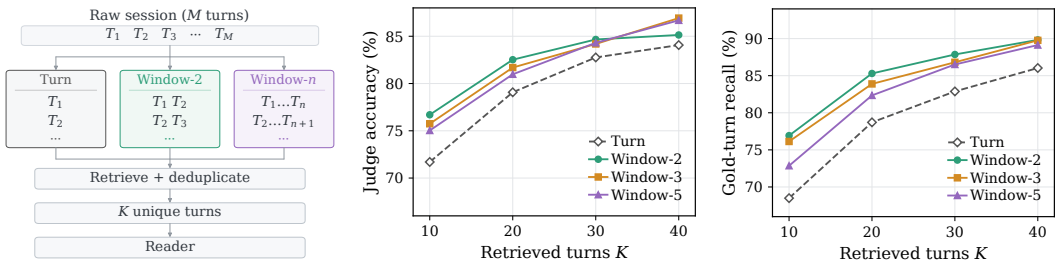
3 Diagnosing Conversational Memory

End-to-end QA performance conflates several factors: what memory construction preserves, which evidence retrieval selects, and how effectively the reader uses the resulting context. To disentangle these effects, we design two controlled diagnostics. First, we eliminate retrieval selection by evaluating complete memory banks directly, comparing them against raw dialogue history and an oracle selection of supporting sessions. Second, we hold the source dialogue and unique-turn budget fixed while varying retrieval-unit boundaries, isolating how contextual grouping affects evidence access. Full diagnostic details and complete results are reported in Appx. E.

Answerability after construction. We first remove retrieval selection by supplying each complete memory bank to the same reader. On LoCoMo with Qwen3.5-27B, all seven evaluated baseline banks fall below Full Context on single-hop questions (Fig. 2a), despite large differences in bank length. Thus, retrieval failure alone cannot account for the observed answerability gaps. Full-bank performance does not identify their cause—it reflects both what the constructed representation retains and how effectively the reader can use it. Fig. 2b illustrates two ways construction can make an answer harder or impossible to recover: omitting a queried detail, or preserving individual statements while losing the relation needed to interpret them.

Selective access to retained evidence. Complete source dialogue avoids construction-time rewriting, yet making the entire history available does not guarantee effective evidence use. As in Fig. 2c, Gold-Session—an oracle reference that retains only complete sessions containing supporting turns from the same history—improves over Full Context on every reported multi-hop/multi-session and temporal setting. The gains range from 4.61 to 17.44 percentage points across the two benchmarks and readers. Since Gold-Session introduces no additional source information, this comparison shows that selectively exposing relevant evidence can be more effective than reading the complete history.

Retrieval-unit boundaries. We next isolate the effect of contextual grouping while holding the underlying dialogue constant. As illustrated in Fig. 3a, we index each session as individual turns or overlapping windows, rank these units, and expand and deduplicate their turns to maintain the same unique-turn budget for the reader. Across all budgets, window-based retrieval improves single-hop accuracy over individual turns (Fig. 3b) and boosts gold-turn recall (Fig. 3c, defined in Eq. (8)). This improvement demonstrates that gains stem from retrieving related evidence jointly rather than sim-



(a) Retrieval-unit comparison. (b) Single-hop QA accuracy. (c) Single-hop gold-turn recall.

Figure 3: How retrieval-unit boundaries affect evidence access. LoCoMo single-hop results using dense retrieval and Qwen3.5-27B. (a) The same dialogue is indexed as individual turns or overlapping windows. Retrieved windows are expanded into turns and deduplicated, ensuring the reader receives the same number of unique turns across settings. (b) Judge accuracy at different retrieval budgets. (c) Gold-turn recall: the proportion of supporting turns included in the reader input.

ply reading more turns. However, larger windows are not uniformly superior: while a 2-turn window yields the highest gold-turn recall across all budgets, the optimal window width for QA accuracy varies with the allocated budget. Together, these diagnostics motivate the following principle:

> **Design principle.** Memory boundaries should follow the dependencies required for interpretation, not a fixed contextual span, while preserving separately retrievable units that can be selectively connected at query time.

4 Method

Guided by the design principle, we propose *ContextFact*, a two-view memory framework. The *fact view* preserves facts together with the context needed to interpret them, while the *entity view* reconstructs related evidence about recurring referents across sessions. As illustrated in Fig. 4, ContextFact first constructs these context-preserving fact memories and source-grounded entity representations (Sec. 4.1), then uses their links to support selective retrieval and readout (Sec. 4.2).

Notation and memory objects. Let $\mathcal{H} = (S_1, \ldots, S_T)$ be a conversation history of T sessions, where each S_t is an ordered sequence of conversational turns. ContextFact constructs a fact bank $\mathcal{F}$, an entity bank $\mathcal{E}$, and a provenance relation $\mathcal{L}$ linking records in the entity view to fact memories. Each fact memory also retains links to its source turns. Further notation is introduced as needed and summarized in Tab. 4.

4.1 Context-Preserving Memory Construction

ContextFact constructs its memory in three main stages: (i) extracting context-preserving fact memories and session-level entity entries, (ii) aligning recurring referents across sessions, and (iii) revisiting source dialogue to reconstruct the entity view. The procedure is formalized in Algo. 1, with construction details in Appx. B (including the LLM instructions in Appx. B.1).

Context-preserving fact extraction. ContextFact organizes each session around *focal facts* rather than fixed numbers of turns or pre-segmented topic blocks. Here, a focal fact serves as the organizing center that determines which supporting context is grouped with it. It concerns a specific occurrence, state, or intention described in the session, and it need not correspond to a single sentence or atomic proposition. For each focal fact, the extractor gathers the context needed to make it interpretable on its own, such as its circumstances, reasons, progress, or outcome, even when this context appears in nonadjacent turns. Consequently, each focal fact and its context forms a single fact memory, while distinct focal facts remain independently retrievable (Appx. G.1).

Before extraction, deterministic rules annotate supported relative-time expressions with absolute dates or intervals while preserving their original wording and granularity. A single LLM call then reads the complete annotated session $\widetilde{S}_t$ and jointly produces its fact memories and *session-level*

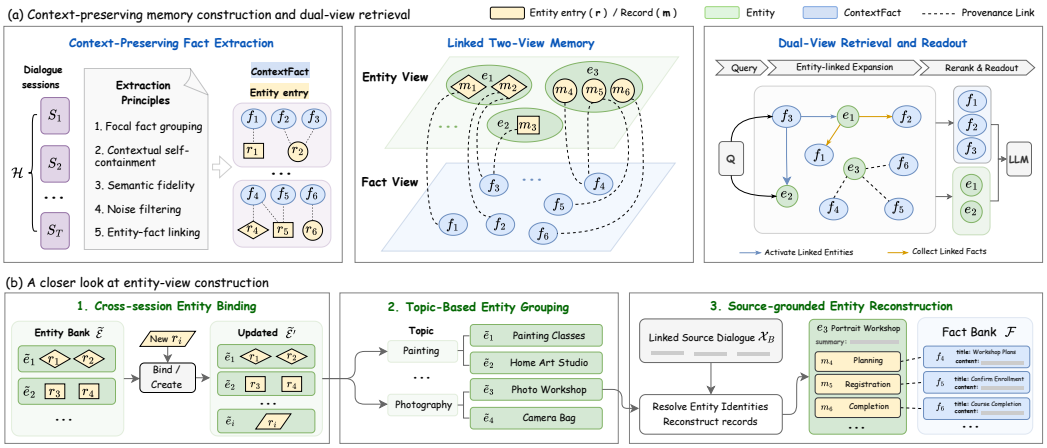


Figure 4: Overview of ContextFact. (a) ContextFact first extracts context-preserving fact memories and linked session-level entity entries from each dialogue session, followed by using provenace links to link the resulting fact and entities. At query time, both views are retrieved. A single entity-linked expansion allows retrieved facts to activate linked entities and expose additional related facts. Fact and entity candidates are then reranked separately, and those selected will be supplied to the reader. (b) Entity-view construction in detail. Session-level entity entries are incrementally aligned across sessions into provisional entities, organized into topic-based review groups, and reconstructed by revisiting their linked source dialogue. The resulting entities contain summaries and records linked to their supporting fact memories. See Appx. G for case studies.

entity entries. Each entry identifies a specific referent (such as a person, object, project, event, or place) with a name and a short description, linking it to the fact memories that substantively describe it. In addition, each fact memory retains its source-turn provenance.

Cross-session entity binding. The extraction stage associates each session with entity entries describing referents mentioned and linking them to the corresponding fact memories. Because the same underlying referent may be described differently across sessions, ContextFact incrementally aligns new entries with previously observed ones. Candidate matches are retrieved using dense similarity and BM25 (Robertson & Zaragoza, 2009) over entry names and descriptions, after which a LLM determines whether a new entry refers to an existing entity or introduces a new one. Entries judged to share the same referent are grouped into a *provisional entity*, providing a first-pass cross-session alignment while preserving the original entries and their fact links.

Source-grounded entity reconstruction. Incremental binding can miss valid matches or incorrectly combine distinct referents. After all sessions have been processed, ContextFact therefore performs a source-grounded reconstruction stage. Provisional entities are first grouped by topic to create local review groups; this grouping determines which candidates are reconsidered together but does not itself merge them. For each group, an LLM returns to the linked *source dialogue* (i.e., the linked conversational turns with deterministic temporal annotations) to verify referent identities and reconstruct the final entities. It may merge entries referring to the same entity, separate entries referring to distinct referents, and rebuild their summaries and records from source evidence.

Each final entity is represented as $e = (n_e, \sigma_e, \mathcal{M}(e))$, where n_e is its name, σ_e is a concise summary, and $\mathcal{M}(e)$ is a set of records. Each record $m \in \mathcal{M}(e)$ organizes a coherent aspect, event, state, or development of the entity, consolidating repeated or complementary evidence while keeping distinct occurrences and unresolved conflicts explicit. Let $\mathrm{Sources}(m) \subseteq \mathcal{F}$ denote the fact memories linked to the source evidence supporting m. These links define the provenance relation

$$\mathcal{L} = \big\{(e, m, f) \mid e \in \mathcal{E},\ m \in \mathcal{M}(e),\ f \in \mathrm{Sources}(m)\big\}. \tag{1}$$

Hence, the entity view organizes cross-session evidence without modifying the fact bank, while preserving provenance from reconstructed records to associated fact memories and source dialogue.

4.2 Dual-View Retrieval and Readout

At query time, ContextFact proceeds in two stages: dual-view retrieval with a single entity-linked expansion, followed by independent reranking and readout. We describe the two stages below. The complete procedure is formalized in Algo. 2, with further details in Appx. C and illustrative cases in Appxs. G.2 and G.3.

Dual-view retrieval and entity-linked expansion. Given a query q, hybrid dense and BM25 retrieval with reciprocal rank fusion (RRF; Cormack et al., 2009) independently returns initial fact candidates from $\mathcal{F}$ and entity candidates from $\mathcal{E}$. Fact memories are indexed using their memory text, while entity retrieval uses only their names n_e and summaries σ_e. We then expand these candidates through the existing provenance links in $\mathcal{L}$. Retrieved facts can activate linked entities, and the resulting entity pool—including both directly retrieved and activated entities—can expose additional facts through their records. This expansion is applied once, allowing direct matches to surface related cross-session evidence without recursive traversal.

Independent reranking and readout. The expanded fact and entity candidates serve complementary roles: fact memories provide focused evidence, whereas entity representations organize related evidence across sessions. We therefore rerank the two pools independently with a cross-encoder and retain the highest-ranked items from each view. Unlike seed retrieval, entity reranking uses the complete representation, including its name, summary, and all records. The selected complete entities are also supplied directly to the reader, following the selected fact memories and preserving reranking order within each view. Thus, the entity view contributes in two ways: its links expose facts beyond direct query matches, while its complete representations provide organized cross-session evidence directly to the reader.

5 Experiments

We evaluate ContextFact on LoCoMo and LongMemEval-S for QA accuracy and computational efficiency. Our analyses examine fact-memory boundaries, temporal normalization and entity reconstruction during memory building, and entity-linked expansion and entity readout at query time.

Datasets and baselines. We evaluate on LoCoMo (Maharana et al., 2024) and LongMemEval-S (Wu et al., 2025). We compare against Full Context and RAG, and ① SeCom (Pan et al., 2025), ② Mem0 (Chhikara et al., 2025), ③ A-Mem (Xu et al., 2025), ④ MemoryOS (Kang et al., 2025), ⑤ LightMem (Fang et al., 2026), ⑥ AtomMem (Yao et al., 2026), ⑦ SimpleMem (Liu et al., 2026), ⑧ EMem (Zhou & Han, 2025), ⑨ Nemori (Ma et al., 2026), and ⑩ Memora (Xia et al., 2026).

Models, variants, and evaluation. We report LoCoMo results with GPT-4.1-mini (OpenAI, 2025) and Qwen3.5-27B (Qwen Team, 2026), and LongMemEval-S results with Qwen3.5-27B. ContextFact uses Qwen3-Embedding-8B (Zhang et al., 2025) for dense retrieval and MemReranker-4B (Li et al., 2026) for reranking. We evaluate two variants of our method. *ContextFact-F* uses only the fact view, omitting entity-linked expansion and entity readout, while *ContextFact-E* denotes the full method using both fact and entity views. Unless otherwise stated, both use the retrieval and reranking configuration described in Sec. 4.2. The fixed-budget study uses RRF retrieval alone to isolate the quality of the fact representation. Our primary metric is LLM-judged accuracy (J), evaluated by GPT-4o-mini. On LoCoMo, we also report token-set F_1 and unigram BLEU (B_1). Evidence length counts only memory text supplied to the reader. Main benchmark comparisons use each method's native retrieval pipeline, whereas controlled budget experiments impose the same reader-evidence cap. The full settings, including evaluation prompts, appear in Appx. D.

5.1 Overall QA Performance and Evidence Efficiency

Tabs. 1 and 2 compare the evaluated systems under their native retrieval pipelines. Our method achieves the highest overall judge accuracy on both benchmarks. On LoCoMo, ContextFact-E reaches 87.27% with GPT-4.1-mini and 86.49% with Qwen3.5-27B, exceeding Memora (P), the strongest baseline, by 1.95 and 2.20 percentage points, respectively. On LongMemEval-S, it reaches 91.00%, a 1.80-point improvement. The largest gains from the entity view appear on questions

Table 1: LoCoMo QA under native retrieval pipelines. ContextFact-E and ContextFact-F achieve the best scores in most settings. "Tokens" is the mean reader-evidence length per question. **Best**/second-best scores within each backbone are **bold**/underlined; n denotes the number of questions. Full Context is excluded from ranking.

Method	Tokens	Single-hop ($n = 841$)			Temporal ($n = 321$)			Multi-hop ($n = 282$)			Open-domain ($n = 96$)			Overall ($n = 1{,}540$)		
		$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$	$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$	$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$	$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$	$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$
GPT-4.1-mini																
Full Context	20,441	64.36	56.73	89.77	58.98	48.29	80.06	44.46	33.38	80.14	27.29	22.00	58.33	57.28	48.53	84.03
RAG	1,945	49.64	43.45	70.04	48.74	42.43	57.32	36.09	23.03	58.51	24.14	19.65	47.92	45.38	38.01	63.90
SeCom	1,707	54.61	48.12	79.07	50.20	40.50	62.93	37.64	27.15	64.18	28.20	21.28	56.25	48.94	41.02	71.56
Mem0	1,408	41.70	36.63	63.85	49.24	42.24	65.11	34.64	25.17	60.64	20.96	15.33	41.67	40.68	34.37	62.14
A-Mem	1,174	54.20	48.46	75.98	46.60	38.85	70.09	31.78	22.21	54.61	17.33	14.88	37.50	46.21	39.55	68.44
MemoryOS	4,941	51.51	45.29	79.43	46.86	35.73	55.45	35.28	27.21	70.57	28.37	24.45	**64.58**	46.13	38.69	71.88
LightMem	2,484	53.74	46.83	83.95	57.14	44.54	80.69	41.28	32.86	74.47	25.54	20.32	53.12	50.41	42.14	79.61
AtomMem†	1,959	55.57	48.59	82.76	61.13	54.49	74.14	42.45	32.95	74.82	30.23	24.08	53.12	52.75	45.43	77.66
SimpleMem	2,016	56.96	49.87	86.80	37.73	30.12	78.19	40.06	30.43	72.34	24.64	19.46	50.00	47.84	40.29	80.06
EMem	1,660	55.47	49.31	84.78	57.04	46.69	77.57	40.52	31.94	74.47	27.95	23.44	53.12	51.35	43.97	79.42
Nemori	4,662	55.72	49.31	89.54	58.82	49.98	78.50	42.99	33.18	76.95	27.21	21.17	50.00	52.26	44.74	82.47
Memora (P)*	12,262	**60.88**	**53.49**	90.73	62.09	49.09	83.49	43.79	34.00	80.14	26.55	21.25	59.38	55.86	46.99	85.32
ContextFact-F	932	59.45	52.42	90.37	62.47	53.47	85.67	43.01	32.61	78.37	**32.90**	**27.24**	**64.58**	55.41	47.44	85.58
ContextFact-E	4,139	59.36	51.81	**91.44**	**64.31**	**55.28**	**89.72**	**45.06**	**35.43**	**81.56**	29.82	24.05	59.38	**55.93**	**47.80**	**87.27**
Qwen3.5-27B																
Full Context	20,441	65.67	55.22	92.15	52.07	42.69	66.36	45.18	34.26	78.72	26.27	23.63	50.00	56.63	46.80	81.69
RAG	1,945	48.72	39.61	68.73	41.50	33.07	52.34	32.87	22.20	54.96	23.17	19.52	44.79	42.72	33.81	61.30
SeCom	2,055	53.99	44.18	77.76	46.41	36.68	55.45	35.39	26.24	60.99	22.91	20.07	43.75	47.07	37.83	67.92
Mem0	1,265	46.22	38.79	65.99	49.58	37.79	58.57	39.19	30.14	66.67	23.49	19.43	38.54	44.22	35.79	62.86
A-Mem	1,384	46.59	41.69	64.21	35.25	29.21	49.22	30.77	22.92	53.19	17.12	13.94	36.46	39.49	33.92	57.34
MemoryOS	4,220	52.03	46.88	71.82	38.80	28.39	44.24	35.83	28.13	64.89	23.44	19.73	43.75	44.52	37.90	63.05
LightMem	1,830	53.48	42.48	81.81	61.04	47.29	79.44	40.59	31.58	73.40	22.44	19.84	46.88	50.76	40.07	77.60
AtomMem†	1,901	56.03	48.73	82.16	60.65	53.59	73.52	40.70	30.77	72.70	25.36	19.84	48.96	52.27	44.65	76.56
SimpleMem	1,529	50.42	41.16	76.69	47.01	37.76	83.49	39.54	31.78	71.63	19.27	15.97	45.83	45.78	37.16	75.26
EMem	1,449	52.65	42.12	82.40	**65.62**	53.92	77.26	38.81	28.94	68.44	25.93	22.22	50.00	51.15	40.92	76.75
Nemori	3,421	56.83	47.04	87.75	58.94	50.14	74.45	39.98	31.27	73.76	26.18	22.05	52.08	52.27	43.24	80.19
Memora (P)*	14,156	61.70	**51.60**	90.37	64.95	52.10	81.62	44.97	35.42	80.85	25.22	21.64	50.00	**57.04**	46.88	84.29
ContextFact-F	1,153	**61.71**	50.27	**90.49**	65.54	54.53	88.47	43.00	33.80	76.95	25.51	20.80	**54.17**	56.83	46.30	85.32
ContextFact-E	2,907	60.65	50.29	**90.49**	64.96	**54.65**	**88.79**	**45.33**	**36.25**	**84.04**	**26.96**	**23.37**	51.04	56.64	**46.95**	**86.49**

* Memora uses its higher-performing policy variant (P). † AtomMem uses the authors' released pre-extracted facts.

requiring evidence to be connected across memories. With Qwen on LoCoMo, ContextFact-E raises multi-hop accuracy from 76.95% for ContextFact-F to 84.04%, while single-hop accuracy remains 90.49%. On LongMemEval-S, adding the entity view raises overall accuracy from 87.60% to 91.00%, including gains on multi-session and temporal questions. Notably, these improvements are obtained with substantially shorter evidence than Memora (P): ContextFact-E uses an average of $2{,}907$ tokens on LoCoMo and $2{,}854$ on LongMemEval-S, approximately 79% fewer on both benchmarks. ContextFact-F provides a more compact alternative while retaining strong overall accuracy.

5.2 Fact Representation under Fixed Evidence Budgets

To assess the effectiveness of context-preserving fact memories, we conduct two complementary evaluations on LoCoMo with Qwen3.5-27B. We first benchmark ContextFact-F against existing methods, then evaluate complete ContextFacts against sentence-level and atomic decompositions of the same bank. Category-level breakdowns across question types are in Appxs. F.1 and F.2.

Budgeted retrieval. ContextFact-F achieves the highest overall judge accuracy among the compared systems at every plotted budget from 0.5k to 4k tokens (Fig. 5a). Notably, ContextFact-F uses RRF retrieval alone, without reranking or entity augmentation, while the competing systems retain their evaluated retrieval pipelines subject to the same reader-evidence cap. The advantage of the fact view therefore appears before introducing the entity mechanisms or final reranker.

Memory decomposition. We next derive verbatim sentence splits and rewritten atomic facts from the same ContextFact bank. Under RRF retrieval, complete ContextFacts achieve higher overall judge accuracy than both decomposed alternatives at every tested budget (Fig. 5b). They also outperform both alternatives on single-hop questions across the evaluated retrievers. These results

Table 2: LongMemEval-S QA with Qwen3.5-27B under native retrieval pipelines. ContextFact-E achieves the best scores in 6 of 8 settings, including overall accuracy. Tokens is the mean reader-evidence length per question. **Best**/second-best scores are **bold**/underlined; n denotes the number of questions. Full Context is excluded from ranking.

Method	Tokens	Single-User ($n = 70$)	Single-Assistant ($n = 56$)	Multi-Session ($n = 133$)	Temporal ($n = 133$)	Knowledge-Update ($n = 78$)	Single-Preference ($n = 30$)	Overall ($n = 500$)
Full Context	103,726	94.29	98.21	75.19	77.44	91.03	70.00	83.20
RAG	1,965	77.14	92.86	24.06	28.57	60.26	73.33	49.00
SeCom	6,537	88.57	**100.00**	65.41	68.42	87.18	83.33	77.80
Mem0	1,192	94.29	14.29	78.20	82.71	91.03	83.33	76.80
LightMem	2,896	94.29	14.29	78.95	87.97	92.31	80.00	78.40
SimpleMem	1,231	90.00	50.00	74.44	77.44	84.62	80.00	76.60
EMem	1,168	94.29	**100.00**	77.44	85.71	92.31	83.33	87.20
Nemori	2,931	98.57	80.36	74.44	73.68	83.33	**96.67**	81.00
Memora (P)	13,588	98.57	98.21	81.20	84.21	**94.87**	93.33	89.20
ContextFact-F	974	97.14	96.43	80.45	84.21	91.03	86.67	87.60
ContextFact-E	2,854	**100.00**	**100.00**	**83.46**	**88.72**	93.59	90.00	**91.00**

support retaining the dependencies needed to interpret a fact within one retrievable unit, rather than assuming that finer decomposition is uniformly beneficial.

5.3 Component and Mechanism Analysis

We next isolate three design choices in ContextFact: temporal normalization during memory construction, source-grounded entity reconstruction, and the two query-time roles of the entity view. The first two ablations vary memory construction while holding the downstream pipeline fixed; the third separates entity-linked expansion from direct entity readout. Full category-level results and controlled variants are reported in Appxs. F.3 to F.5.

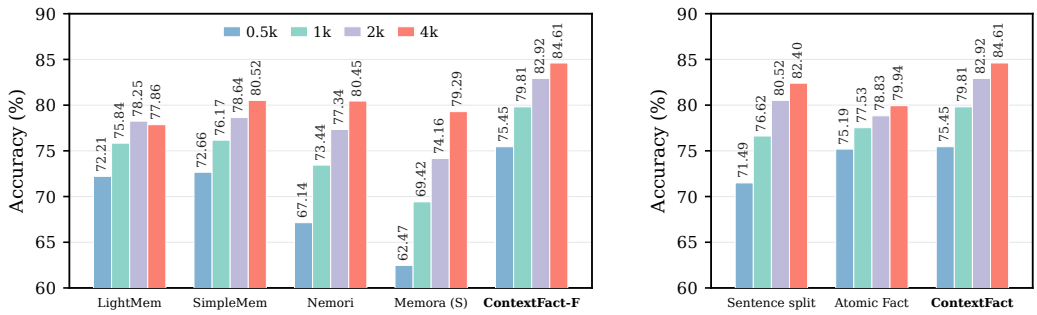
Temporal normalization. ContextFact resolves supported relative-time expressions with deterministic rules before fact extraction, reducing reliance on joint LLM normalization. We compare rule-guided normalization with joint normalization and no temporal guidance. Rule-guided normalization gives the highest temporal J (Fig. 6a), improving over joint normalization by 3.74 points with both backbones. Inspection of the generated facts shows that joint normalization can introduce incorrect absolute dates, supporting rule-guided normalization before fact extraction.

Source-grounded entity reconstruction. We next test whether the entity view should be reconstructed from earlier generated memories or from the underlying dialogue. Starting from the same provisional entity bank, we compare using the provisional entities directly, reconstructing from stored ContextFacts, and reconstructing from source dialogue. Source-grounded reconstruction gives the highest overall J (Fig. 6b), improving over provisional entities by 1.17 points with both backbones and over ContextFact-based reconstruction by 0.91 and 0.65 points with GPT-4.1-mini and Qwen3.5-27B, respectively. This supports revisiting source evidence when consolidating cross-session information.

Entity-linked expansion and readout. We separate the two query-time roles of the entity view by exposing linked facts and providing organized evidence directly to the reader. Holding the selected facts fixed, complete entity readout improves multi-hop J by 5.32 points without fact expansion and 5.67 points with fact expansion (Tab. 3(a)). The gain is not explained by simply supplying more facts: replacing the entities with twenty additional ContextFacts lowers multi-hop J by 2.83 points despite higher linked-turn recall and a longer reader input (Tab. 3(b)). Entity links also improve evidence access without entity readout: entity-seeded and fact-seeded expansion raise multi-hop J by 2.13 and 0.71 points, respectively (Tab. 3(c)). Together, these results show that the entity view contributes both through linked evidence access and through organized cross-session readout.

5.4 Computational Efficiency

Finally, we evaluate whether ContextFact can maintain strong QA performance while reducing the computational cost of memory construction and query-time inference, and examine the additional



(a) Memory-system comparison.

(b) Memory decomposition.

Figure 5: Budgeted QA with context-preserving fact memories. Overall judge accuracy on LoCoMo with Qwen3.5-27B under reader-evidence caps of 0.5k, 1k, 2k, and 4k tokens. (a) Comparison with existing memory systems under the same evidence caps. (b) Comparison of complete ContextFacts with verbatim sentence splits and rewritten atomic facts derived from the same bank, using RRF retrieval. Memora (S) uses the semantic retriever.

Table 3: Entity-linked expansion and entity readout ablations on LoCoMo with Qwen3.5-27B. CFn denotes n ContextFacts, and E10 denotes ten entities. Recall denotes *linked-turn recall*: the mean coverage of annotated supporting turns through the source-turn provenance of selected items.

Variant	Input	Tokens	Multi-hop				Overall
			J	Recall	F_1	B_1	J
(a) Disable fact expansion and/or entity readout; Δ vs. Full method							
Full method	CF10+E10	2,907	84.04	92.02	45.33	36.25	86.49
w/o fact expansion	CF10+E10	2,889	82.27 (↓1.77)	91.43 (↓0.59)	45.20 (↓0.13)	36.35 (↑0.10)	86.04 (↓0.45)
w/o entity readout	CF10	1,171	78.37 (↓5.67)	86.14 (↓5.88)	44.19 (↓1.14)	34.52 (↓1.73)	85.84 (↓0.65)
w/o both	CF10	1,153	76.95 (↓7.09)	80.03 (↓11.99)	43.00 (↓2.33)	33.80 (↓2.45)	85.32 (↓1.17)
(b) Additional-fact controls; Δ vs. Full method							
Replace E10 with 20 CFs	CF30	3,501	81.21 (↓2.83)	93.69 (↑1.67)	44.43 (↓0.90)	34.81 (↓1.44)	85.65 (↓0.84)
Direct CF retrieval (60→30)	CF30	3,433	80.14 (↓3.90)	92.03 (↑0.01)	44.38 (↓0.95)	34.31 (↓1.94)	85.39 (↓1.10)
(c) Fact expansion paths without entity readout; Δ vs. w/o both							
w/o both (reference)	CF10	1,153	76.95	80.03	43.00	33.80	85.32
Fact-seeded expansion	CF10	1,168	77.66 (↑0.71)	84.18 (↑4.15)	43.42 (↑0.42)	34.08 (↑0.28)	85.39 (↑0.07)
Entity-seeded expansion	CF10	1,170	79.08 (↑2.13)	85.60 (↑5.57)	44.12 (↑1.12)	34.59 (↑0.79)	86.17 (↑0.85)
Dual-path expansion	CF10	1,171	78.37 (↑1.42)	86.14 (↑6.11)	44.19 (↑1.19)	34.52 (↑0.72)	85.84 (↑0.52)

cost introduced by the entity view. ContextFact also reduces query-time cost relative to the strongest baseline. On LoCoMo with GPT-4.1-mini, ContextFact-F slightly exceeds Memora (P) in overall accuracy while reducing mean end-to-end latency from 8.33 s to 0.95 s (8.8×). ContextFact-E raises accuracy to 87.27% while remaining 5.6× faster at 1.49 s. Construction-token, search-latency, and tail-latency results are reported in Appx. F.7.

6 Conclusion

We study how to represent and organize history for selective long-term conversational memory. Our diagnostics show that answerability gaps can persist even when constructed memory banks are read in full, while retrieval-unit boundaries materially affect which supporting evidence reaches the reader. These findings motivate memory boundaries that follow the dependencies needed for interpretation, together with mechanisms for selectively reconnecting related evidence across sessions. ContextFact instantiates this principle through context-preserving fact memories and source-grounded entity representations, using the entity view for both entity-linked expansion and entity readout. Across LoCoMo and LongMemEval-S, ContextFact outperforms strong baselines while requiring substantially shorter evidence inputs, with controlled analyses confirming the individual contributions of fact construction, source reconstruction, and entity readout. Together, our results suggest that effective conversational memory requires not merely storing more information, but preserving the right context locally and reconnecting it selectively when needed.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish writing, support literature retrieval and discovery, assist with research ideation and execution, and draft preliminary sections of the paper. In particular, these tools were used for language editing, literature search, preliminary drafting, and refining experimental interpretations. We did not use generative AI tools to generate synthetic datasets; other required-disclosure uses are not applicable to this work. The authors reviewed and verified all AI-assisted content, including relevant literature, technical claims, experimental results, and interpretations. We take full responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We evaluate ContextFact on publicly released LoCoMo and LongMemEval-S benchmarks without recruiting participants or collecting new conversations. LongMemEval-S includes sessions drawn from public chat corpora (Wu et al., 2025), so we report aggregate results and do not attempt to link speakers to real-world identities. Because ContextFact stores and links facts about people, use on real conversations could expose sensitive details or preserve outdated information. Deployment would require consent, access controls, retention limits, and mechanisms to correct or delete memories. These safeguards are beyond the scope of our benchmark evaluation.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, we provide comprehensive algorithmic and experimental details throughout the paper and appendices. Appxs. B and C provide formal pseudocode for memory construction and retrieval. Construction prompt templates appear in Appx. B.1, and reader and judge prompt templates appear in Appx. D.4. Complete implementation settings, evaluation protocols, and evidence accounting appear in Appx. D. Diagnostic setups, controlled ablations, and category-level breakdowns are reported in Appxs. E and F.

REFERENCES

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

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 758–759, 2009. URL https://dl.acm.org/doi/10.1145/1571941.1572114.

Xingbo Du, Loka Li, Duzhen Zhang, and Le Song. MemR^3: Memory retrieval via reflective reasoning for LLM agents. *arXiv preprint arXiv:2512.20237*, 2025.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A Graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024. URL https://arxiv.org/abs/2404.16130.

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.

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. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6ddc001d07ca4f319af96a3024f6dbd1-Abstract-Conference.html.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21497–21515. PMLR, 2025. URL https://proceedings.mlr.press/v267/gutierrez25a.html.

Shuo Ji, Yibo Li, and Bryan Hooi. Memory is reconstructed, not retrieved: Graph memory for LLM agents. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=xRVWftS3ES.

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, 2025. doi: 10.18653/v1/2025.emnlp-main.1318. URL https://aclanthology.org/2025.emnlp-main.1318/.

Chunyu Li, Mengyuan Zhang, Jingyi Kang, Ding Chen, Jiajun Shen, Bo Tang, Xuanhe Zhou, Feiyu Xiong, and Zhiyu Li. MemReranker: Reasoning-aware reranking for agent memory retrieval. *arXiv preprint arXiv:2605.06132*, 2026. URL https://arxiv.org/abs/2605.06132v2.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. SimpleMem: Efficient lifelong memory for LLM agents. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=oBgLvd5YC6.

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

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, 2026.

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

OpenAI. GPT-4.1 mini. OpenAI API model documentation, 2025. URL https://developers.openai.com/api/docs/models/gpt-4.1-mini. Accessed September 14, 2026.

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

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.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

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

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and trends® in information retrieval*, 4(1-2):1–174, 2009.

Zhanyu Shen, Sijie Cheng, Zhicheng Guo, Weiqin Wang, Yile Wang, and Hui Huang. AnchorMem: Anchored facts with associative contexts for building memory in large language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 34784–34798, 2026. URL https://aclanthology.org/2026.findings-acl.1736/.

Jingwei Sun, Jianing Zhu, Jiangchao Yao, Tongliang Liu, and Bo Han. Rethinking how to remember: Beyond atomic facts in lifelong LLM agent memory. *arXiv preprint arXiv:2605.19952*, 2026.

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*, 2025. URL https://openreview.net/forum?id=pZiyCaVuti.

Menglin Xia, Xuchao Zhang, Shantanu Dixit, Paramaguru Harimurugan, Rujia Wang, Victor Rühle, Robert Sim, Chetan Bansal, and Saravan Rajmohan. Memora: A harmonic memory representation balancing abstraction and specificity. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=zSrvkj0ers.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-Mem: Agentic memory for LLM agents. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=FiM0M8gcct.

B. Y. Yan, Chaofan Li, Hongjin Qian, Shuqi Lu, and Zheng Liu. General agentic memory via deep research. *arXiv preprint arXiv:2511.18423*, 2025.

Yanyu Yao, Shangze Li, Zhi Zheng, Hui Zheng, Qi Liu, Tong Xu, and Enhong Chen. AtomMem: Building simple and effective memory system for LLM agents via atomic facts. *arXiv preprint arXiv:2606.19847*, 2026. URL https://arxiv.org/abs/2606.19847.

Jiangnan Yu, Kisson Songqi Lin, and Jilong Wu. WhenLoss: Diagnosing write and retrieval bottlenecks in long-context memory systems. *arXiv preprint arXiv:2605.24579*, 2026.

Boqin Yuan, Yue Su, and Kun Yao. Diagnosing retrieval vs. utilization bottlenecks in LLM agent memory. *arXiv preprint arXiv:2603.02473*, 2026.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

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

Sizhe Zhou and Jiawei Han. A simple yet strong baseline for long-term conversational memory of LLM agents. *arXiv preprint arXiv:2511.17208*, 2025.

Luyao Zhuang, Shengyuan Chen, Yilin Xiao, Huachi Zhou, Yujing Zhang, Hao Chen, Qinggang Zhang, and Xiao Huang. LinearRAG: Linear graph retrieval augmented generation on large-scale corpora. In *International Conference on Learning Representations*, volume 2026, pp. 147053–147075, 2026.

Contents of Appendix

A Related Work **14**

B Memory Construction Details **15**

B.1 Prompt Templates for Memory Construction 17

C Retrieval and Readout Details **22**

D Experimental Setup **23**

D.1 Datasets 23

D.2 Baselines 23

D.3 Implementation Details 24

D.4 Evaluation Protocol and Prompts 25

E Diagnostic Experiments **29**

E.1 Full-bank Reading 29

E.2 Retrieval-unit Boundaries 29

F Additional Experimental Results **31**

F.1 Memory-Unit Retrieval under Evidence Budgets 31

F.2 Memory Decomposition 31

F.3 Temporal Normalization 31

F.4 Entity Reconstruction 34

F.5 Entity-Linked Expansion and Entity Readout 35

F.6 Choice of Reranker 35

F.7 Computational Efficiency 36

G Case Studies **37**

G.1 Memory-Unit Boundaries 37

G.2 Entity-Linked Expansion 39

G.3 Cross-Session Entity Readout 40

A Related Work

Conversational memory construction and granularity. Long-term memory systems vary in what they preserve and the units they expose for retrieval. MemoryBank combines dialogue history with event summaries and user profiles (Zhong et al., 2024), MemoryOS organizes short-, mid-, and long-term stores (Kang et al., 2025), and Mem0 extracts and reconciles salient facts across interactions (Chhikara et al., 2025). LightMem combines compression with topic-aware consolidation (Fang et al., 2026), while Nemori maintains episodic and semantic memories (Ma et al., 2026). Closer to our focus on memory boundaries, SeCom partitions dialogue into contiguous topical segments (Pan et al., 2025), EMem constructs self-contained event-centric units with normalized entities, time, and source attribution (Zhou & Han, 2025), and SimpleMem constructs compact semantic memories and consolidates related information (Liu et al., 2026). ContextFact instead organizes memory around focal facts: information that explains, qualifies, or develops the same fact may be gathered across nonadjacent turns, while distinct facts remain separately retrievable. Its boundaries are therefore determined by the context needed to interpret each fact rather than by fixed windows or contiguous topics.

Linked memory and cross-session organization. Other work organizes memories beyond a flat collection. A-Mem links and updates contextualized notes (Xu et al., 2025), while Zep represents episodes, entities, relations, and temporal facts in a knowledge graph (Rasmussen et al., 2025). Memora connects concrete memories through primary abstractions and cue anchors (Xia et al., 2026); AtomMem organizes atomic facts into event structures and temporal profiles (Yao et al., 2026); TriMem combines raw dialogue, atomic facts, and synthesized profiles (Sun et al., 2026); and AnchorMem uses atomic facts as retrieval anchors while retaining associated source context (Shen et al., 2026). ContextFact likewise combines fine-grained and organized views, but constructs its entity view by aligning recurring referents across sessions and then revisiting source dialogue to verify identities and reconstruct entity-centered accounts. These accounts organize distinct aspects and developments, retain links to supporting fact memories, and are provided directly to the reader.

Evidence retrieval and selection. Structured memories can also recover evidence beyond direct semantic matches. GraphRAG organizes entity and relation descriptions with community summaries (Edge et al., 2024); HippoRAG and HippoRAG 2 propagate relevance through graph associations (Gutiérrez et al., 2024; 2025); LinearRAG operates over an entity–sentence–passage graph (Zhuang et al., 2026); and EMem-G retrieves over its session–event–argument structure (Zhou & Han, 2025). Other systems use more adaptive query-time retrieval: Memora exploits memory connections with a policy (Xia et al., 2026), GAM performs query-time research over retained information (Yan et al., 2025), MemR^3 alternates retrieval and reflection (Du et al., 2025), and MRAgent iteratively traverses an associative memory graph (Ji et al., 2026). ContextFact instead uses a lightweight query-time procedure: independent fact and entity retrieval, a single entity-linked expansion, and separate reranking of the two candidate pools with MemReranker (Li et al., 2026). Selected facts provide focused evidence, while complete entity representations provide an organized cross-session view.

Evaluating and diagnosing conversational memory. LoCoMo and LongMemEval evaluate memory over extended multi-session interactions (Maharana et al., 2024; Wu et al., 2025), while long-context studies show that available evidence is not always used reliably (Liu et al., 2024). Recent work further separates sources of memory failure: Yuan et al. (2026) study retrieval and utilization bottlenecks across memory-writing and retrieval choices, while WhenLoss compares oracle evidence, complete stored memory, and retrieved memory to characterize write- and retrieval-side gaps (Yu et al., 2026). We complement these analyses with diagnostics focused on memory-unit design. Full-bank reading tests whether answerability gaps persist after removing retrieval selection, while controlled retrieval-window interventions vary unit boundaries over unchanged dialogue at matched unique-turn budgets. We also derive sentence-level and atomic representations from the same ContextFact bank to study how decomposition affects evidence access under fixed budgets.

Table 4: Notation used in ContextFact.

Symbol	Definition
Conversation and fact memories	
$\mathcal{H} = (S_1, \ldots, S_T)$	Conversation history of T ordered sessions.
S_t	The t-th session, represented as an ordered sequence of conversational turns.
$\widetilde{\mathcal{H}} = (\widetilde{S}_1, \ldots, \widetilde{S}_T)$	Temporally annotated history produced before memory extraction.
$f,\ \mathcal{F}_t,\ \mathcal{F}$	A fact memory, the fact memories extracted from session S_t, and the complete fact bank.
Entity construction	
$r \in \mathcal{R}_t$	A session-level entity entry extracted jointly with the facts from session S_t.
$\mathrm{Sources}(r)$	Fact memories in $\mathcal{F}_t$ that substantively describe the referent of entry r.
$\widetilde{\mathcal{E}}^{(t)}$	Provisional entity bank after processing sessions $S_1, \ldots, S_t$.
$\tilde{e},\ \mathcal{R}(\tilde{e})$	A provisional entity and the session-level entries currently bound to it.
$\mathcal{C}_t(r)$	Candidate provisional entities considered when binding entry $r \in \mathcal{R}_t$.
$s_d(r, \tilde{e}),\ s_b(r, \tilde{e})$	Dense-similarity and BM25 scores used for candidate generation.
τ_d, τ_b	Candidate-selection thresholds for dense similarity and BM25.
$\mathcal{T}$	Topic hierarchy over the provisional entity bank after processing all sessions.
$\mathcal{B},\ B$	Collection of review groups defined by the leaf nodes, and one such group.
$\mathcal{X}_B$	Source-dialogue evidence collected for review group B.
$\mathcal{E}_B$	Final entities reconstructed from review group B.
Final entity representation	
$e = (n_e, \sigma_e, \mathcal{M}(e))$	A final entity with name n_e, summary σ_e, and records $\mathcal{M}(e)$.
$m \in \mathcal{M}(e)$	An record describing a coherent aspect, event, state, or development of entity e.
$\mathrm{Sources}(m)$	Fact memories whose linked source evidence supports reconstructed record m.
$\mathcal{E}$	Final reconstructed entity bank.
$\mathcal{L}$	Provenance relation containing triples (e, m, f) with $m \in \mathcal{M}(e)$ and $f \in \mathrm{Sources}(m)$.
Retrieval and readout	
q	Query presented to the memory system.
$\mathcal{F}_0, \mathcal{E}_0$	Initial fact and entity seeds returned by hybrid retrieval.
k_f, k_e	Maximum numbers of initial fact and entity seeds.
$\mathcal{E}_+$	Entity candidate pool after activating entities linked to retrieved facts.
$\mathcal{F}_+$	Fact candidate pool after expanding through the entities in $\mathcal{E}_+$.
$\widehat{\mathcal{F}}, \widehat{\mathcal{E}}$	Final reranked fact and entity selections supplied to the reader.
k	Maximum number of selected items retained from each view.
$\mathrm{Hybrid}(\cdot)$	Dense–BM25 retrieval combined by reciprocal rank fusion.
$\mathrm{Rerank}(\cdot)$	Cross-encoder reranking within one memory view.

B Memory Construction Details

Algo. 1 formalizes the construction procedure described in Sec. 4.1. Starting from the ordered history $\mathcal{H}$, ContextFact annotates supported temporal expressions, jointly extracts fact memories and session-level entity entries, incrementally binds recurring referents across sessions, and finally reconstructs the entity bank from source dialogue. Exact instructions for the prompt-based operators are provided in Appx. B.1 and cross-referenced below.

Temporal annotation. Before memory extraction, AnnotateTime resolves supported relative time expressions against the corresponding session date and appends the resulting absolute date or interval while preserving the original expression and temporal granularity. Ambiguous or unsupported expressions are left unchanged. We denote the resulting history by $\widetilde{\mathcal{H}} = (\widetilde{S}_1, \ldots, \widetilde{S}_T)$. The extraction and reconstruction stages operate on this annotated history, whose conversational content otherwise remains unchanged.

Joint extraction and candidate generation. For each annotated session $\widetilde{S}_t$, JointExtract produces fact memories $\mathcal{F}_t$ and session-level entity entries $\mathcal{R}_t$ in one LLM call (Appx. B.1.1).

Algorithm 1 ContextFact memory construction

Require: Ordered sessions $\mathcal{H} = (S_1, \ldots, S_T)$
Require: Candidate thresholds τ_d, τ_b
Ensure: Fact bank $\mathcal{F}$, entity bank $\mathcal{E}$, and provenance relation $\mathcal{L}$

$\widetilde{\mathcal{H}} \leftarrow \textsc{AnnotateTime}(\mathcal{H})$ ▷ $\widetilde{\mathcal{H}} = (\widetilde{S}_1, \ldots, \widetilde{S}_T)$
$\mathcal{F} \leftarrow \varnothing$; $\widetilde{\mathcal{E}}^{(0)} \leftarrow \varnothing$
Joint extraction and incremental entity binding
for $t = 1, \ldots, T$ **do**
  $(\mathcal{F}_t, \mathcal{R}_t) \leftarrow \textsc{JointExtract}(\widetilde{S}_t)$
  $\mathcal{F} \leftarrow \mathcal{F} \cup \mathcal{F}_t$
  $\mathcal{C}_t \leftarrow \textsc{Candidates}(\mathcal{R}_t, \widetilde{\mathcal{E}}^{(t-1)}, \tau_d, \tau_b)$
  $\widetilde{\mathcal{E}}^{(t)} \leftarrow \textsc{Bind}(\widetilde{\mathcal{E}}^{(t-1)}, \mathcal{R}_t, \mathcal{C}_t)$
end for
Global topic organization
$\mathcal{T} \leftarrow \textsc{TopicGroup}(\widetilde{\mathcal{E}}^{(T)})$ ▷ LLM-generated hierarchy with entity IDs at leaves
$\mathcal{B} \leftarrow \textsc{LeafGroups}(\mathcal{T})$ ▷ Deterministically extract leaf entity sets
Source-grounded entity reconstruction
$\mathcal{E} \leftarrow \varnothing$
for all $B \in \mathcal{B}$ **do**
  $\mathcal{X}_B \leftarrow \textsc{SourceEvidence}(B, \mathcal{F}, \widetilde{\mathcal{H}})$
  $\mathcal{E}_B \leftarrow \textsc{Reconstruct}(B, \mathcal{X}_B, \mathcal{T})$
  $\mathcal{E} \leftarrow \mathcal{E} \cup \mathcal{E}_B$
end for
Record-level provenance
$\mathcal{L} \leftarrow \{(e, m, f) \mid e \in \mathcal{E},\ m \in \mathcal{M}(e),\ f \in \mathrm{Sources}(m)\}$
return $(\mathcal{F}, \mathcal{E}, \mathcal{L})$

Each entry $r \in \mathcal{R}_t$ contains a name and short description of a specific referent together with links $\mathrm{Sources}(r) \subseteq \mathcal{F}_t$ to fact memories that describe it. To align an entry with previously observed referents, we compare it against the current provisional bank $\widetilde{\mathcal{E}}^{(t-1)}$ using dense similarity and BM25. For each entry r, candidate generation produces

$$\mathcal{C}_t(r) = \left\{ \tilde{e} \in \widetilde{\mathcal{E}}^{(t-1)} \;\middle|\; s_d(r, \tilde{e}) \geq \tau_d \ \vee \ s_b(r, \tilde{e}) \geq \tau_b \right\}, \tag{2}$$

where matching is performed over entity names and descriptions. The candidate thresholds and implementation-specific preprocessing are given in Appx. D.

Incremental entity binding. BIND considers each new session-level entry together with its candidate provisional entities. An LLM determines whether the entry refers to one of these existing referents or introduces a distinct one, following the identity and continuity criteria in Appx. B.1.2. A matched entry is added to the corresponding provisional entity. Otherwise, a new provisional entity is created. The resulting bank $\widetilde{\mathcal{E}}^{(t)}$ therefore provides a first-pass alignment of referents observed through session t while retaining the member entries $\mathcal{R}(\tilde{e})$ and their fact links. Binding is intentionally provisional: later reconstruction can correct missed or erroneous identity decisions.

Topic grouping and source evidence. After all sessions have been processed, TOPICGROUP organizes $\widetilde{\mathcal{E}}^{(T)}$ into a hierarchy $\mathcal{T}$, assigning each provisional entity to a leaf node; the organization criteria and output structure are specified in Appx. B.1.3. LEAFGROUPS then deterministically extracts the entity sets at the leaf nodes to form $\mathcal{B}$. Each $B \in \mathcal{B}$ is a review group of provisional entities that are jointly reconsidered during reconstruction. Topic grouping does not itself merge or rewrite entities; it only determines which provisional entities are jointly reviewed during reconstruction.

For each $B \in \mathcal{B}$, SOURCEEVIDENCE follows the retained fact and source-turn links to collect the associated dialogue from $\widetilde{\mathcal{H}}$, yielding $\mathcal{X}_B$. RECONSTRUCT then compares the provisional representations with this source evidence and produces a set $\mathcal{E}_B$ of final entities, following the instructions in Appx. B.1.4. It may merge entries referring to the same underlying referent, separate mixed referents, correct earlier descriptions, and organize the resulting information into records.

Record-level provenance. For each reconstructed entity e and record $m \in \mathcal{M}(e)$, reconstruction retains the fact IDs associated with the source evidence used for that record. Their union defines

$$\mathcal{L} = \{(e, m, f) : e \in \mathcal{E},\ m \in \mathcal{M}(e),\ f \in \text{Sources}(m)\}.$$

Because reconstruction reads the source dialogue, a record may preserve a supported detail that is not explicitly present in the text of its linked fact memory. $\mathcal{L}$ should therefore be interpreted as a provenance relation through the associated source evidence, rather than as a textual entailment relation between f and m.

B.1 Prompt Templates for Memory Construction

We provide the prompts used by the four LLM-based construction operators in Algo. 1: JointExtract, Bind, TopicGroup, and Reconstruct. Deterministic preprocessing, candidate generation, review-group extraction, and source-evidence assembly are described above.

B.1.1 Joint Fact and Entry Extraction

JointExtract: fact memories and session-level entity entries

````
# Task
Convert one raw conversation session into a chronological list of context-
    preserving facts and a linked entity index in one output.

# Input
```text
CONVERSATION DATE: <date, including the weekday>

EVIDENCE:
D<session>:<turn> <Speaker>: <text> [IMG blip="<caption>" query="<query>"]
...
```
The `IMG` part is optional and belongs to the same source turn.

# Principles

1. **Identify and regroup core facts**
   - Read the whole session before forming any ContextFacts. First identify
     the core facts in the conversation; do not create one fact from each turn.
      A core fact is a specific event, action, state, plan, experience,
     preference, attribute, personal detail, belief, realization, decision,
     outcome, or change--not a broad topic or everything said about one
     participant.
   - For each new clause, decide whether it:
      (a) further develops an existing core fact by identifying, describing,
     explaining, clarifying, or evaluating that same fact; providing its
     circumstances, reasons, progress, outcome, or event-specific feelings; or
     directly reacting to it through encouragement, gratitude, agreement, or
     another immediate response; or
      (b) introduces another independently meaningful memory focus, such as a
     distinct occurrence, plan, state, habitual practice, preference, general
     belief, realization, decision, or lasting change that is not limited to
     describing or reacting to the current fact.
      Attach (a) to the ContextFact it supports. Create or resume a separate
     ContextFact for (b). Classifying a clause as supporting context does not
     permit omitting it or replacing its specific meaning with a broader
     summary.
   - After identifying the focuses, regroup non-adjacent clauses that belong
     to the same concrete fact. If one utterance reacts to one core fact and
     introduces or resumes another, divide it by meaning. One source turn may
     support multiple ContextFacts, and one ContextFact may use non-consecutive
      turns.

2. **Self-contained references and attribution**
   - Make each ContextFact understandable on its own by explicitly identifying
      the relevant people, objects, events, and activities when needed.
````

JointExtract: fact memories and session-level entity entries (continued)

```
   - Resolve omitted subjects, pronouns, and vague references such as "this"
    or "that one" when their referents are clear. Pronouns may remain when the
     referent is explicit and unambiguous within the same ContextFact. If a
    reference cannot be resolved from the evidence, preserve the original
    wording rather than guessing.
   - Preserve participant roles and perspective. Every action, experience,
    feeling, belief, evaluation, and reaction must remain attributed to its
    original speaker or subject.

3. **Preserve the original expression**
   - Keep the original wording, tone, specific descriptions, intensifiers,
    negation, modality, and uncertainty as much as possible. Make only the
    grammatical changes needed for attribution, pronoun resolution, and
    readability. Never turn a plan into a completed action, a possibility into
     certainty, or one participant's statement into the other's experience.
   - Preserve specific actions, identifying details, quoted wording, and the
    scope of personal statements; do not replace them with broader activities,
     interpretations, or summaries.

4. **Remove non-memory-bearing content**
   - Omit greetings, farewells, filler, content-free acknowledgements, exact
    repetition, and routine closing remarks that add no substantive
    information. Preserve substantive personal and relationship information
    wherever it appears.
   - Interpret each answer together with the question it addresses. Preserve
    the target, requested attribute or relationship, and any comparison,
    contrast, negation, or time scope needed to retain the answer's meaning as
     a self-contained fact.
   - Omit purely information-seeking questions and the narration that they
    were asked, while retaining their answers and substantive factual content.
     State only what the exchange supports: an unconfirmed question premise or
     proposed alternative must not become an established fact.

5. **Image rule**
   `blip` provides a visual description of the shared photo; `query` is only
    the phrase used to search for it and may not match the photo.
   - Use a photo when it identifies or clarifies the concrete subject
    presented in its source turn, or when the surrounding conversation
    discusses it. Sharing a broad topic with the text is not sufficient. Omit
    generic, decorative, unrelated, or conflicting image information.
   - For a relevant photo, retain concrete visual features and any depicted
    wording explicitly provided in blip when they distinguish the subject or
    contribute to understanding the discussion. Integrate relevant visual
    information into the ContextFact for the pictured subject, using wording
    such as "shared a photo of ...". A photo alone does not require a separate
     ContextFact.
   - Use the conversation to establish personal attribution and factual
    connections. A visible feature or inscription does not by itself establish
     who owns or created the subject, or which real-world event it depicts.
    Use a query-only detail only when independently supported by the
    conversation and not contradicted by blip.

6. **Preserve dual-format time references**
   - Preserve time expressions in the format: relative time (absolute date or
    range), retaining the original precision.
   - If a time expression is clearly resolvable from the source context and
    session date but lacks an absolute value, append it. If an existing
    absolute value is clearly incorrect, correct it.

7. **Extract and link entities**
   - Extract specific, context-grounded entities from the raw session as
    retrieval entry points, including referents embedded in descriptions. Link
     each entity to all ContextFacts that provide substantive information
    about it.
   - Include people (excluding the speakers themselves), animals, places,
    organizations, works, objects, foods, recipes, projects, and events.
```

JointExtract: fact memories and session-level entity entries (continued)

```
- Do not extract either speaker as an entity. A speaker-related entry must
  identify a specific object, event, activity, or project associated with
  that speaker, not the person as a whole.
- Use stated names or concise identifying labels, adding ownership or
  distinguishing context when needed. Keep each entry centered on one
  referent rather than a generic keyword, broad theme, or separate aspect of
   the same entity.
```

B.1.2 CROSS-SESSION ENTITY BINDING

Bind: cross-session entity binding

```
You are an expert in entity resolution and temporal memory organization.

# Task

For each new record, determine whether it refers to the same specific object,
person, project, activity, or occurrence as one of its candidate entities.

If so, add it to that entity and update the entity's name and description.
Otherwise, create a new entity from the record.

# Principles

1. Resolve the underlying referent

Compare names, distinguishing features, participants, ownership,
relationships, and temporal context. Candidate session times show when its
supporting information was discussed; occurrence times remain in the
    descriptions.

Different names may refer to the same thing, while similar descriptions may
refer to different things. An entity remains distinct from its participants,
products, tools, locations, associated events, and individual members of a
collection.

2. Distinguish continuity from a new referent

Use explicit references, conversational continuity, identifying details,
and temporal development.

Plans, work in progress, completion, and later feedback may be stages of one
persistent project or object. Interpret "new", "another", ordinals, changes
of participants or locations, and temporal differences according to whether
they indicate continuity or a distinct referent.

Keep separate occurrences of recurring activities distinct.

3. Do not confuse association with identity

A shared topic, owner, purpose, activity, or temporal proximity does not by
itself establish identity.

Prefer the candidate with the strongest identifying continuity. Recency may
support a compatible match but is not sufficient on its own. If the supplied
information cannot reliably identify one candidate, create a new entity at
the record's supported specificity.

4. Update the matched entity

Preserve the entity's specific identity while incorporating the new record.

Retain useful identifying details and distinct temporal stages, including
planned, ongoing, and completed states. Do not broaden the entity into a
general topic or collection.
```

Bind: cross-session entity binding (continued)

```
Process records in input order. If multiple records bind to the same entity,
each later update must incorporate earlier updates from this response.
```

B.1.3 TOPIC-BASED ENTITY GROUPING

TopicGroup: hierarchical review groups

```
# Task

Build a hierarchical topic directory for the supplied entities from one
    conversation. The directory will support later joint review of their
    evidence.

# Organization

Start with broad aspects and introduce more specific subtopics where useful.
    Use at most MAX_TOPIC_DEPTH topic levels; branches may have different
    depths.

Keep possible duplicate entries and closely connected objects in the same
    specific branch, including a collection and its members, or an activity
    and its projects. Stop subdividing when the entities form a coherent
    review scope. Do not merge or rewrite entities.

Assign every supplied entity exactly once. Preserve its name and ID.

# Output

Return the complete hierarchy as JSON.

Each group has:

- name: A concise topic label.
- description: One sentence describing the scope of its descendants.
- entities: Exact supplied entity names.
- entity_ids: Corresponding IDs in the same order.
- children: Subgroups with the same structure.

A group contains either subgroups or entities:

- With subgroups, entities and entity_ids are empty.
- With entities, children is empty.

{ "groups": [ { "name": "...", "description": "...", "entities": [], "
    entity_ids": [], "children": [ { "name": "...", "description": "...", "
    entities": ["...", "..."], "entity_ids": ["E01", "E07"], "children": [] },
     { "name": "...", "description": "...", "entities": ["...", "..."], "
    entity_ids": ["E02", "E11"], "children": [] } ] } ] }

# Input

MAX_TOPIC_DEPTH: 3

ENTITIES:

{ENTITIES}
```

B.1.4 SOURCE-GROUNDED ENTITY RECONSTRUCTION

Reconstruct: source-grounded identities and records

```
You are an expert in entity resolution and conversational memory organization.

# Task

Reconstruct the referents represented by the supplied entity inventory
from the original evidence, then organize their information into concise,
aspect-based records.

Entities preserve independently retrievable subjects. Compactness applies
to their descriptions, not to minimizing the number of entities.

# Principles

1. Verify and correct the existing representation

Compare the supplied entity and member-record descriptions with their
associated dialogue before using them. Verify referents, speaker attribution,
ownership, relationships, status, and dates.

Correct claims introduced or distorted by earlier extraction and binding.
An error found only in a generated description is not an alternative
source account.

ContextFact titles guide evidence lookup. Use image descriptions with
their attribution; image queries are search cues.

2. Preserve supported referents

Represent each distinct, evidence-supported referent in the supplied
entities and member records as an output entity, separately or through
a confirmed same-referent merge.

Mentioning a referent in another entity's record, or listing its input ID
under that entity, does not preserve its identity.

Split mixed entries. Keep related objects and separate occurrences distinct,
while retaining stages of the same persistent referent within one entity.
Express unresolved references at the specificity supported by the dialogue.

3. Organize aspects without losing defining information

Reconstruct records around coherent aspects, combining repeated mentions
and complementary information. Preserve the relationships, circumstances,
motivations, and outcomes needed to identify and follow the subject.

Keep qualifiers that determine identity, event boundaries, counts, or
status explicit in the relevant records, even when the linked Facts retain
the full details. Preserve what each number or ordinal counts, along with
its subject and scope.

Related entities may describe a shared relationship from their respective
perspectives. Attach the unique supplied ContextFact IDs supporting each
record's account.

Give each entity a clear name and a concise summary consistent with its
corrected records and any material qualifications.

4. Distinguish development from conflicting accounts

Organize supported changes of state as development, distinguishing plans,
ongoing activities, and completed outcomes.

For differing source statements, first align the referent, property, and
scope. Preserve incompatible accounts briefly with their attribution or
reporting time, unless the dialogue explicitly resolves or corrects them.
```

Reconstruct: source-grounded identities and records (continued)

```
A later statement alone does not establish either a real-world change
or a correction of the earlier statement.

Order events by supported occurrence time. When only reporting time is
known, describe when the information was reported rather than assigning
that date to the event or its beginning.
```

C Retrieval and Readout Details

Algo. 2 formalizes the two-stage query-time procedure from Sec. 4.2: dual-view retrieval with a single entity-linked expansion, followed by independent reranking and readout.

Hybrid seed retrieval. Given a query q, HYBRID independently retrieves initial fact seeds $\mathcal{F}_0 \subseteq \mathcal{F}$ and entity seeds $\mathcal{E}_0 \subseteq \mathcal{E}$. For each bank, dense and BM25 rankings are combined by reciprocal rank fusion (RRF; Cormack et al., 2009). For an item x, the fusion score is

$$s_{\mathrm{RRF}}(x) = \frac{1}{c + r_d(x)} + \frac{1}{c + r_b(x)}, \tag{3}$$

where $r_d(x)$ and $r_b(x)$ are its dense and BM25 ranks and c is the RRF constant. Fact memories are indexed using their memory text. For entity seed retrieval, only the entity name n_e and summary σ_e are indexed. Record–fact provenance links support entity-linked expansion, while complete entity representations are used for entity reranking and readout. The implementation-specific retriever, value of c, and seed limits appear in Appx. D.

Single entity-linked expansion. The initial seeds are expanded through the provenance relation constructed during memory construction. Define

$$\textsc{LinkedEntities}(\mathcal{F}_0, \mathcal{L}) = \{e \in \mathcal{E} \mid \exists\, m \in \mathcal{M}(e),\ f \in \mathcal{F}_0 : (e, m, f) \in \mathcal{L}\}, \tag{4}$$

and

$$\textsc{LinkedFacts}(\mathcal{E}', \mathcal{L}) = \{f \in \mathcal{F} \mid \exists\, e \in \mathcal{E}',\ m \in \mathcal{M}(e) : (e, m, f) \in \mathcal{L}\}. \tag{5}$$

The expanded candidate pools are then

$$\mathcal{E}_+ = \mathcal{E}_0 \cup \textsc{LinkedEntities}(\mathcal{F}_0, \mathcal{L}), \tag{6}$$

$$\mathcal{F}_+ = \mathcal{F}_0 \cup \textsc{LinkedFacts}(\mathcal{E}_+, \mathcal{L}). \tag{7}$$

This ordering captures both routes used by ContextFact: an initially retrieved fact may activate a linked entity and thereby expose other facts, while an independently retrieved entity can directly expose its linked facts. Expansion stops after this step, and newly added facts do not trigger another round of entity activation.

Independent reranking and readout. Because the two views serve different roles, their scores are not compared across views. A cross-encoder independently reranks $\mathcal{F}_+$ and $\mathcal{E}_+$, yielding

$$\widehat{\mathcal{F}} = \textsc{Rerank}(q, \mathcal{F}_+, k), \qquad \widehat{\mathcal{E}} = \textsc{Rerank}(q, \mathcal{E}_+, k).$$

Facts are reranked using their fact-memory text. Entities are reranked using their complete representation $(n_e, \sigma_e, \mathcal{M}(e))$, rather than only the name and summary used for seed retrieval. The reader input is then

$$\textsc{Format}(\widehat{\mathcal{F}}) \parallel \textsc{Format}(\widehat{\mathcal{E}}),$$

where facts precede entities and the reranking order within each view is preserved. Hence, the entity view affects answering in two distinct ways: its provenance links alter which fact memories enter the candidate pool, while its complete representations also serve directly as organized reader evidence.

Algorithm 2 Dual-view retrieval and readout

Require: Query q; banks $\mathcal{F}, \mathcal{E}$; provenance relation $\mathcal{L}$
Require: Seed limits k_f, k_e; selection limit k per view
Ensure: Ordered reader input of selected facts and complete entities

Dual-view seed retrieval
$\mathcal{F}_0 \leftarrow \textsc{Hybrid}(q, \mathcal{F}, k_f)$
$\mathcal{E}_0 \leftarrow \textsc{Hybrid}(q, \mathcal{E}, k_e)$ ▷ Name and summary only
Single entity-linked expansion
$\mathcal{E}_+ \leftarrow \mathcal{E}_0 \cup \textsc{LinkedEntities}(\mathcal{F}_0, \mathcal{L})$
$\mathcal{F}_+ \leftarrow \mathcal{F}_0 \cup \textsc{LinkedFacts}(\mathcal{E}_+, \mathcal{L})$ ▷ Across all records
Independent reranking
$\widehat{\mathcal{F}} \leftarrow \textsc{Rerank}(q, \mathcal{F}_+, k)$
$\widehat{\mathcal{E}} \leftarrow \textsc{Rerank}(q, \mathcal{E}_+, k)$ ▷ Complete entity representations
Readout
return $\textsc{Format}(\widehat{\mathcal{F}}) \parallel \textsc{Format}(\widehat{\mathcal{E}})$

D Experimental Setup

This section provides the experimental details supporting Sec. 5. We describe the datasets and baseline systems, the implementation settings used for ContextFact, and the reader and judge protocols. Unless otherwise stated, the main QA comparisons use each method's evaluated retrieval pipeline without a shared evidence-token cap. Controlled budget experiments instead impose the same reader-evidence cap across compared methods. Evidence length counts only the memory text supplied to the reader.

D.1 Datasets

LoCoMo (Maharana et al., 2024). LoCoMo contains extended conversations between two speakers, generated from personas and temporal event graphs and subsequently verified by human annotators. Its question-answering task requires recalling details and connecting evidence across sessions. We evaluate the ten-conversation subset, excluding adversarial questions, for a total of 1,540 questions: 841 single-hop, 321 temporal, 282 multi-hop, and 96 open-domain. Single-hop questions target individual facts, multi-hop questions combine multiple pieces of evidence, temporal questions concern dates and event order, and open-domain questions connect conversational information with background knowledge.

LongMemEval-S (Wu et al., 2025). LongMemEval evaluates memory over timestamped user–assistant histories, covering information extraction, cross-session reasoning, temporal reasoning, knowledge updates, and abstention when the history does not support an answer. We evaluate all 500 questions in LongMemEval-S: 70 single-user, 56 single-assistant, 133 multi-session, 133 temporal, 78 knowledge-update, and 30 single-preference. Unlike LoCoMo's recurring speaker pairs, these histories contain accumulated user interactions, including assistant responses and revisions to previously stated facts.

D.2 Baselines

We compare ContextFact with a full-history reference (Full Context), a retrieval baseline (RAG), and representative conversational-memory systems spanning segmented, fact-based, episodic, and structured memory representations. The descriptions below emphasize the memory representation and retrieval mechanism relevant to our comparison:

- *SeCom (Pan et al., 2025).* SeCom partitions conversations into topically coherent segments and applies compression-based denoising to construct memory units. It retrieves relevant segments using their compressed representations to support response generation.
- *Mem0 (Chhikara et al., 2025).* Mem0 extracts salient facts from incoming dialogue using recent exchanges and historical summaries. It reconciles these facts with stored memories through addition, update, deletion, or no-change operations, and retrieves relevant facts for answering.

- *A-Mem (Xu et al., 2025).* A-Mem represents interactions as structured notes enriched with contextual descriptions, keywords, and tags. It links related notes and updates existing memory as new information arrives, supporting retrieval over contextualized memory units.
- *MemoryOS (Kang et al., 2025).* MemoryOS organizes conversations into short-term, mid-term, and long-term memory, with consolidation and updating across levels. Its reader input combines retrieved dialogue memories with longer-term user knowledge and profile information.
- *LightMem (Fang et al., 2026).* LightMem compresses incoming dialogue and groups related content through topic segmentation before extracting memories. It separates online memory accumulation from offline consolidation and retrieves stored memories by semantic similarity.
- *AtomMem (Yao et al., 2026).* AtomMem introduces a high-quality atomic fact extraction dataset produced through model-based extraction and human refinement. Its memory system organizes atomic facts into hierarchical event structures and temporal profiles, with an associative memory graph connecting related information.
- *SimpleMem (Liu et al., 2026).* SimpleMem compresses dialogue into self-contained memory units by resolving references, anchoring temporal expressions, and synthesizing related information. These units are indexed for query-dependent retrieval to construct the answering context.
- *EMem (Zhou & Han, 2025).* EMem decomposes conversations into enriched elementary discourse units: self-contained event statements with normalized entities, temporal cues, and source-turn attributions. It combines dense retrieval with recall-oriented LLM filtering to select relevant units for answering.
- *Nemori (Ma et al., 2026).* Nemori segments conversations into coherent episodes and maintains complementary episodic and semantic memories. Its construction framework uses prediction and calibration to derive knowledge from episodes, while retrieval draws on both memory types.
- *Memora (Xia et al., 2026).* Memora organizes factual and episodic memories under primary abstractions and connects them through cue anchors. Its semantic retriever (S) selects memories by similarity, while its policy retriever (P) uses an LLM to navigate the memory structure and gather related evidence.

D.3 Implementation Details

Model configurations. On LoCoMo, we evaluate separate GPT-4.1-mini and Qwen3.5-27B settings, using the corresponding model for memory construction and answer generation. On LongMemEval-S, we report Qwen3.5-27B results. ContextFact uses Qwen3-Embedding-8B for dense retrieval and MemReranker-4B for final reranking.

Entity-binding settings. For entity binding on LoCoMo, we set the dense-similarity threshold to $\tau_d = 0.60$ and the BM25 threshold to $\tau_b = 6$. A provisional entity is retained as a candidate when either its dense cosine similarity is at least 0.60 or its BM25 score is at least 6; no additional top-k restriction is applied. Dense and BM25 matching use concatenated entity names and descriptions. For BM25 matching, the two conversation participants' names are removed from both the incoming entry and candidate text. The candidate-generation procedure is defined in Appx. B.

Retrieval settings. Dense retrieval uses Qwen3-Embedding-8B with normalized 4,096-dimensional embeddings, while BM25Okapi operates on lowercased alphanumeric tokens. Dense and BM25 rankings are combined by equal-weight reciprocal rank fusion as defined in Appx. C, with RRF constant $c = 60$. Retrieval is restricted to the conversation associated with the current question. The main ContextFact-E configuration retrieves $k_f = k_e = 30$ initial fact and entity seeds and retains up to $k = 10$ items from each view after expansion and independent reranking. Entity seed retrieval indexes only the entity name and summary, whereas reranking and reader input use the complete entity representation, including all records. The final reader input places selected fact memories before selected entities, preserving reranking order within each view. The main configuration uses fixed unit limits rather than an explicit evidence-token cap.

Evaluation and evidence accounting. Our primary metric is LLM-judged answer accuracy (J), with GPT-4o-mini as the judge. On LoCoMo, we additionally report token-set F_1 and unigram BLEU (B_1). Overall scores are averaged over questions. Reported evidence-token counts include only the memory text supplied to the answering model, rather than the full prompt or retrieval computation. Main-system comparisons use each method's evaluated retrieval pipeline, whereas fixed-budget experiments cap the final reader evidence consistently across methods.

Hardware and deployment. Our experiments run on servers equipped with four NVIDIA A100-SXM4 GPUs (40 GB each), two AMD EPYC 7313 processors (32 physical cores and 64 threads in total), and approximately 504 GiB of OS-reported system memory. For the efficiency measurements, Qwen3-Embedding-8B and MemReranker-4B are deployed locally for embedding and reranking, respectively, while GPT-4.1-mini is accessed through an API as the reader.

D.4 Evaluation Protocol and Prompts

For LoCoMo, we use the same reader and judge prompts as Mem0 (Chhikara et al., 2025).[1] For LongMemEval-S, we follow Memora (Xia et al., 2026) and use its reader and judge prompts.[2] Within each dataset, all evaluated methods use the same reader and judge templates. We use GPT-4o-mini as the judge across all experiments to ensure a consistent evaluation.

D.4.1 LoCoMo Reader

LoCoMo reader

```
You are an intelligent memory assistant tasked with retrieving accurate
information from conversation memories.

# CONTEXT:
You have access to memories from a conversation between two speakers.
These memories contain timestamped information that may be relevant to
answering the question.

# INSTRUCTIONS:
1. Carefully analyze all provided memories
2. Pay special attention to the timestamps to determine the answer
3. If the question asks about a specific event or fact, look for direct
   evidence in the memories
4. If the memories contain contradictory information, prioritize the
   most recent memory
5. If there is a question about time references (like "last year", "two
   months ago", etc.), calculate the actual date based on the memory
   timestamp.
6. Always convert relative time references to specific dates, months, or
   years. Ignore the reference while answering the question.
7. Focus only on the content of the memories. Do not confuse character
   names mentioned in memories with the actual users who created those
   memories.
8. The answer should be less than 5-6 words.

# APPROACH (Think step by step):
1. First, examine all memories that contain information related to the
   question
2. Examine the timestamps and content of these memories carefully
3. Look for explicit mentions of dates, times, locations, or events that
   answer the question
4. If the answer requires calculation (e.g., converting relative time
   references), show your work
5. Formulate a precise, concise answer based solely on the evidence in
   the memories
6. Double-check that your answer directly addresses the question asked
7. Ensure your final answer is specific and avoids vague time references
```

[1] Mem0 evaluation prompts: `reader` and `judge`.
[2] Memora evaluation prompts: `reader` and `judge`.

D.4.2 LoCoMo Judge

LoCoMo 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/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 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 (like "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.
```

D.4.3 LongMemEval-S Reader

LongMemEval-S reader

```
You are an intelligent Memory Assistant. Your task is to answer the user's
    question accurately by reasoning from conversational memories.

# TASK
Your are provided with a user's question and a set of timestamped memories.
You must synthesize information from the memories to provide a accurate and
    concise answer to the user's question.
This may require multi-hop reasoning, connecting dots across different
    memories, and performing time calculations when necessary.
If the memories do not contain sufficient information to answer the question,
    explicitly state that.

# GUIDELINES
1. **Answer Questions:** Answer questions accordingly to the memories provided
    .
2. **Chain-of-Thought:** Use a structured Chain-of-Thought approach to ensure
    no details are missed.
3. **Complete Entity Retention:** Always include full names, locations, brands
    , and other proper nouns as they appear in the memories. Never use generic
     terms like "a colleague" or "the company."
4. **Absolute Precision:** Retain all exact numbers, prices, dates (calculate
    relative ones), frequencies, and proper nouns.
5. **Multi-Hop Reasoning:** You must actively connect dots. If Memory A says "
    User works at Acme Corp" and Memory B says "Acme Corp announced layoffs,"
    you must infer the user might be affected. Do not treat memories in
    isolation.
6. **Time Calculations:** Always calculate the actual dates from relative
    references.

# REASONING PROCESS (Think step-by-step):

Please follow the following Chain-of-Thought process step-by-step to ensure a
    comprehensive and accurate answer:

## STEP 1: EVIDENCE EXTRACTION
Identify all memories that contain information relevant to the question.
   - **Identify:** Quote the specific memory content and its timestamp.
   - **Extract Entities:** List all specific entities (People, Locations,
    Objects) found in these snippets.
```

LongMemEval-S reader (continued)

```
## STEP 2: LOGICAL SYNTHESIS & MULTI-HOP REASONING
Analyze the evidence to establish connections and derive new insights.
- **Link:** Identify connections between different memories (e.g., "Memory 1
    mentions X, Memory 2 links X to Y, therefore...").
- **Cross-reference:** Cross-reference entities appearing in multiple memories
     to uncover implicit relationships.
- **Infer:** Perform logical inferences where direct evidence is lacking but
    strongly suggested by the data.

## STEP 3: RESOLVE ENTITIES & TIME CALCULATIONS
Resolve any ambiguous reference to entities or timeframes to ensure clarity.
- **Resolve:** For any ambiguous entities (e.g., "the colleague," "the project
    "), clarify who/what they refer to based on context. For example, instead
    of "the project" specify "Project Phoenix"; instead of "a gift," specify "
    the birthday gift from Sarah" if that is clear from the memories.
- **Calculate Time:** For any relative time references (e.g., "last month," "
    in two weeks"), compute the absolute date based on the memory timestamp
    and the question date. For example, if a memory from 15 March 2023
    mentions "tomorrow," then the date is 16 March 2023.

## STEP 4: REFUTATION OF CONTRADICTIONS
If conflicting information arises from different memories, identify and
    resolve these contradictions.
- **Identify Conflicts:** Highlight any discrepancies in the data (e.g., "
    Memory A states X, while Memory B states not X").
- **Resolution Strategy:** Determine which memory is more reliable based on
    recency, source credibility, or corroboration from other memories.
- **Resolve Conflicts:** Clearly state which information is accepted and which
     is discarded, along with the rationale.

## STEP 5: COMPLETENESS CHECK
Review your findings against the "Critical Requirements":
- Did I include all specific names (people, places, organizations)?
- Are all numbers and amounts exact?
- Did I convert all relative time references to specific dates?
- Is the answer derived strictly from the provided memories + logical
    inference?

## STEP 6: FINAL ANSWER
Provide the final, concise answer to the user.
- Write naturally but maintain high information density.
- Provide your final answer after "FINAL ANSWER:". The final answer should be
    concise yet include all necessary details.

---

**INPUT DATA:**

Memories:
{{ memories }}

Question: {{ question }}
Question Date: {{ question_date }}

**OUTPUT:**
```

D.4.4 LONGMEMEVAL-S JUDGE

LongMemEval-S judge

```
# System message
You are an expert grader that determines if answers to questions match a
gold standard answer
```

LongMemEval-S judge (continued)

```
# User message: Default: single-session-user, single-session-assistant, multi-
    session, and other question types
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>
B: {question}
</QUESTION>
<CORRECT ANSWER>
{gold_answer}
</CORRECT ANSWER>
<RESPONSE>
A: {response}
</RESPONSE>

Please answer 'yes' or 'no':

# User message: Temporal reasoning
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/months, etc., and the model makes
off-by-one errors (e.g., predicting 19 days when the answer is 18), the
model's response is still correct.

<QUESTION>
B: {question}
</QUESTION>
<CORRECT ANSWER>
{gold_answer}
</CORRECT ANSWER>
<RESPONSE>
A: {response}
</RESPONSE>

Please answer 'yes' or 'no':

# User message: Knowledge update
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>
B: {question}
</QUESTION>
<CORRECT ANSWER>
{gold_answer}
</CORRECT ANSWER>
<RESPONSE>
A: {response}
</RESPONSE>

Please answer 'yes' or 'no':

# User message: Single-session preference
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
```

LongMemEval-S judge (continued)

```
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>
B: {question}
</QUESTION>
<RUBRIC>
{gold_answer}
</RUBRIC>
<RESPONSE>
A: {response}
</RESPONSE>

Please answer 'yes' or 'no':
```

E Diagnostic Experiments

E.1 Full-bank Reading

We evaluate full-bank answerability by providing each method's complete memory bank to the reader without retrieval. For ContextFact, we supply the complete fact bank $\mathcal{F}$ only and exclude the entity view. GPT-4o-mini judges answer correctness, and Tokens denotes the mean evidence length per question.

LoCoMo. We evaluate all 1,540 questions using Qwen3.5-27B and GPT-4.1-mini as readers (Tab. 5). Full Context provides the complete dialogue, whereas oracle Gold-Session provides only sessions containing annotated supporting turns. For AtomMem, we use the authors' released atomic facts, generated by a Qwen3-14B extractor trained with LoRA supervised fine-tuning on their high-quality, human-refined fact-extraction dataset (Yao et al., 2026). ContextFact achieves the highest overall judge accuracy among the evaluated memory methods with both readers, at 85.32% and 85.06%, respectively.

LongMemEval-S. We evaluate all 500 questions using Qwen3.5-27B as the reader (Tab. 6). ContextFact achieves the highest overall accuracy among the evaluated memory methods, at 90.60%.

E.2 Retrieval-unit Boundaries

We index the same LoCoMo dialogue as individual turns or overlapping windows of two, three, or five turns. Windows use stride one within sessions; shorter sessions form a single unit. Complete windows are ranked by the same dense retriever. We expand ranked windows in their original turn order, deduplicate turns, and stop at exactly $K \in \{10, 20, 30, 40\}$ unique turns. Selected turns are presented chronologically. We report judge accuracy (J) and gold-turn recall:

$$\text{GoldTurnRecall} = \frac{100}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \frac{|G_q \cap R_q|}{|G_q|}, \tag{8}$$

where $\mathcal{Q}$ is the set of questions with valid supporting-turn annotations in the evaluated category, and G_q and R_q denote the sets of annotated supporting turns and retrieved turns for question q, respectively.

Table 5: Full-bank QA performance (%) on LoCoMo. Bold and underlined scores indicate the best and second-best results among memory methods within each backbone, respectively; ties share the same rank. Full Context and Gold-Session are excluded from ranking.

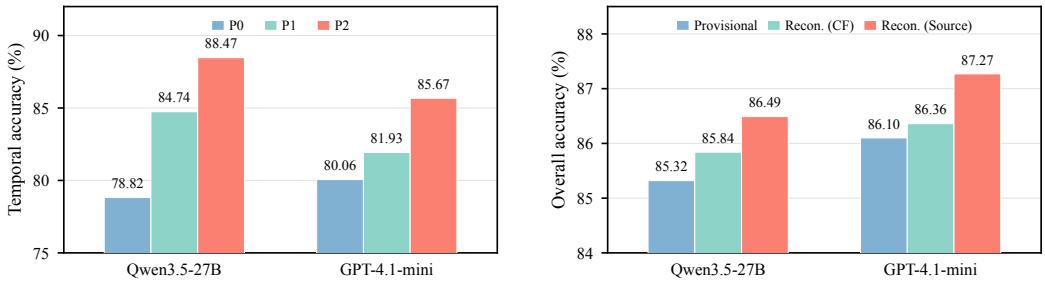
Method	Tokens	Single-hop			Temporal			Multi-hop			Open-domain			Overall		
		$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$	$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$	$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$	$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$	$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$
Qwen3.5-27B																
Full Context	20,441	65.67	55.22	92.15	52.07	42.69	66.36	45.18	34.26	78.72	26.27	23.63	50.00	56.63	46.80	81.69
Gold-Session	-	66.92	53.60	93.22	61.71	47.90	83.80	50.93	39.08	83.33	30.70	25.89	57.29	60.65	48.03	87.21
Mem0	6,774	47.26	41.16	69.80	34.85	27.02	49.84	39.43	30.20	68.44	20.58	18.15	37.50	41.58	34.77	63.38
LightMem	16,771	57.16	48.01	85.61	54.88	44.92	79.75	42.80	33.35	76.24	20.81	18.63	44.79	51.79	42.85	80.13
SimpleMem	10,581	53.81	45.99	79.67	60.58	48.82	84.74	44.37	34.33	76.95	21.95	18.01	45.83	51.50	42.70	78.12
AtomMem	12,945	59.56	49.80	85.97	60.28	51.77	77.88	44.72	33.93	76.24	23.25	19.66	47.92	54.73	45.43	80.13
EMem	15,992	56.25	47.58	84.90	**66.02**	55.09	76.95	42.71	32.31	76.95	25.19	**22.57**	51.04	53.87	44.79	79.68
Nemori	19,875	54.54	46.81	87.51	62.36	53.64	79.44	42.58	32.88	75.18	25.13	21.93	50.00	52.15	44.13	81.23
Memora	39,601	63.42	54.14	**90.84**	64.16	53.20	79.44	**47.56**	**37.27**	**80.14**	**25.54**	22.29	**52.08**	**58.31**	48.87	84.09
ContextFact	15,768	**64.24**	**55.13**	90.73	63.08	**55.97**	**87.85**	45.55	35.99	79.08	23.74	20.63	47.92	58.05	**49.65**	**85.32**
GPT-4.1-mini																
Full Context	20,441	64.36	56.73	89.77	58.98	48.29	80.06	44.46	33.38	80.14	27.29	22.00	58.33	57.28	48.53	84.03
Gold-Session	-	68.21	60.62	94.89	67.69	55.88	89.72	50.38	39.51	85.46	34.26	27.71	63.54	62.72	53.71	90.13
Mem0	6,197	45.66	40.09	68.61	46.91	39.24	62.93	36.50	27.15	63.48	21.18	15.36	43.75	42.72	36.01	64.94
LightMem	14,235	54.24	47.56	83.59	49.91	40.72	78.82	42.89	31.89	76.60	22.64	18.17	54.17	49.29	41.43	79.48
SimpleMem	25,306	59.12	52.12	88.23	37.71	30.48	77.88	43.11	32.40	74.47	27.77	21.01	55.21	49.77	42.06	81.49
AtomMem	12,945	57.36	50.51	85.14	61.45	**54.16**	80.06	43.99	32.93	80.50	23.44	18.13	53.12	53.65	46.03	81.23
EMem	20,070	58.46	52.07	88.35	**62.74**	53.56	83.49	44.18	**34.72**	78.37	**27.96**	**21.80**	54.17	54.83	**47.31**	83.38
Nemori	21,065	51.36	44.88	86.92	58.55	49.84	78.19	41.98	31.84	**81.56**	25.83	20.76	53.12	49.55	42.02	82.01
Memora	38,887	**60.17**	**52.98**	**90.49**	62.17	48.67	80.37	43.91	33.67	78.01	26.45	21.61	51.04	**55.51**	46.59	83.64
ContextFact	13,761	59.75	52.53	89.54	62.38	53.79	**85.67**	**44.44**	33.46	80.50	23.47	17.62	**57.29**	55.23	47.13	**85.06**

Table 6: Full-bank QA accuracy (%) on LongMemEval-S. Bold and underlined scores indicate the best and second-best results among memory methods, respectively.

Method	Tokens	Single-User ($n=70$)	Single-Assistant ($n=56$)	Multi-Session ($n=133$)	Temporal ($n=133$)	Knowledge-Update ($n=78$)	Single-Preference ($n=30$)	Overall ($n=500$)
Full Context	103,726	94.29	98.21	75.19	77.44	91.03	70.00	83.20
Gold-Session	-	97.14	100.00	88.72	91.73	92.31	90.00	92.60
Mem0	6,687	92.86	14.29	79.70	78.20	88.46	86.67	75.60
LightMem	16,065	94.29	16.07	**84.96**	90.98	87.18	83.33	80.40
SimpleMem	18,425	92.86	51.79	75.19	75.94	87.18	**93.33**	78.20
EMem	97,648	95.71	**98.21**	75.94	87.22	92.31	76.67	86.80
Nemori	29,117	94.29	80.36	77.44	73.68	85.90	86.67	81.00
Memora	127,726	98.57	**98.21**	77.44	81.95	91.03	83.33	86.40
ContextFact	34,891	**100.00**	94.64	78.95	**92.48**	**96.15**	90.00	**90.60**

Table 7: Effect of retrieval-unit granularity on LoCoMo. Judge accuracy (J), gold-turn recall, token-set F_1, and BLEU-1 (B_1) are reported as percentages. Each setting supplies exactly K unique turns to the reader. Bold marks the highest score within each column and budget, including ties.

K	Unit	Single-hop				Temporal		Multi-hop		Open-domain		Overall			
		J	Recall	F_1	B_1	J	Recall	J	Recall	J	Recall	J	Recall	F_1	B_1
10	Turn	71.70	68.49	51.62	44.10	**61.06**	**66.93**	55.67	**35.76**	**44.79**	33.77	64.87	60.10	46.54	37.87
10	Window-2	**76.69**	**76.92**	**55.35**	**47.42**	56.39	64.43	**57.09**	33.71	41.67	34.57	**66.69**	**63.87**	**47.85**	**39.29**
10	Window-3	75.74	76.13	54.42	47.08	55.14	59.94	50.71	30.38	39.58	34.87	64.61	61.91	46.27	38.04
10	Window-5	75.03	72.86	53.28	45.69	45.48	52.10	48.94	28.08	41.67	**34.88**	62.01	58.05	43.89	35.59
20	Turn	79.07	78.71	57.93	49.79	**67.29**	**75.49**	62.06	**45.73**	**47.92**	38.06	71.56	69.57	51.41	42.44
20	Window-2	**82.52**	**85.30**	**59.80**	**51.41**	65.42	71.81	**63.12**	44.96	**47.92**	39.02	**73.25**	**72.33**	**51.99**	**43.15**
20	Window-3	81.69	83.89	58.43	50.00	62.93	70.69	**63.12**	43.74	44.79	40.80	72.08	71.21	51.23	42.27
20	Window-5	80.98	82.34	57.52	49.41	58.57	63.16	58.51	40.67	45.83	**43.37**	70.00	68.38	49.76	40.79
30	Turn	82.76	82.88	60.41	51.81	**69.16**	78.56	67.38	52.49	**50.00**	40.39	75.06	73.87	53.70	44.44
30	Window-2	**84.66**	**87.86**	**61.26**	**52.89**	68.85	**78.82**	67.02	**53.28**	47.92	43.19	**75.84**	**76.97**	**54.30**	**45.22**
30	Window-3	84.19	86.81	60.85	52.36	65.73	74.87	**68.79**	52.84	46.88	43.24	75.19	75.49	53.58	44.62
30	Window-5	84.30	86.51	60.16	51.76	64.17	68.25	64.18	48.40	47.92	**46.40**	74.16	73.32	52.33	43.30
40	Turn	84.07	86.03	60.94	52.53	**71.96**	**81.91**	68.79	58.08	46.88	43.08	76.43	77.49	54.78	45.76
40	Window-2	85.14	**89.82**	61.80	53.04	69.16	81.36	**70.21**	**59.92**	**50.00**	44.98	76.88	**79.90**	**55.08**	**46.01**
40	Window-3	**86.92**	89.76	**61.92**	**53.12**	69.47	78.92	68.79	58.70	48.96	46.23	**77.60**	79.21	54.85	45.80
40	Window-5	86.68	89.13	61.67	52.79	66.98	75.13	68.79	55.07	48.96	**48.36**	76.95	77.53	53.81	44.77



(a) Temporal normalization.

(b) Entity reconstruction.

Figure 6: Construction ablations on LoCoMo. (a) Temporal judge accuracy for No temporal guidance (P0), Joint normalization (P1), and Rule-guided normalization (P2) in ContextFact-F. (b) Overall judge accuracy using provisional entities, reconstruction from stored ContextFacts, or source-grounded reconstruction. Within each backbone, the variants share the same downstream configuration.

F Additional Experimental Results

This section supplements the analyses in Sec. 5 with evidence-budget and memory-decomposition results, temporal normalization during construction, entity reconstruction, and entity-linked expansion and entity readout, followed by reranker comparisons.

F.1 Memory-Unit Retrieval under Evidence Budgets

Tab. 8 reports category-level results on LoCoMo with Qwen3.5-27B under evidence budgets of $B \in \{0.5, 1, 2, 4, 8\}$k tokens, following the token accounting in Appx. D. All baselines retain their original retrieval pipelines and are evaluated under the same reader-evidence cap. ContextFact-F retrieves RRF-ranked fact memories within each budget, without reranking, entity-linked expansion, or entity readout. ContextFact-F achieves the highest Overall judge accuracy at every budget and the highest single-hop judge accuracy from 0.5k to 4k, while ranking second at 8k. Rankings on the other question types vary across budgets.

F.2 Memory Decomposition

Tab. 9 extends Fig. 5b to Dense and RRF retrieval across all question types, under evidence budgets of $B \in \{0.5, 1, 2, 3, 4\}$k tokens. Evidence accounting follows Appx. D. We compare complete ContextFact units with two representations derived from the same fact content: verbatim sentence splits and atomic facts produced by rewriting the content of each fact independently.

Complete ContextFact units outperform both alternatives on single-hop questions with either retriever, whereas atomic facts can perform better on multi-hop questions. The relative performance of atomic facts and sentence splits also depends on the retriever, indicating that the benefits of memory granularity vary across retrieval settings and question types.

F.3 Temporal Normalization

Construction variants. We compare three ways of handling time during memory construction. *No temporal guidance* (P0) uses the original timestamped dialogue and removes the temporal-normalization instruction. *Joint normalization* (P1) uses the same unannotated dialogue but instructs the extractor to retain relative expressions and append absolute dates or intervals while rewriting facts. *Rule-guided normalization (P2)*, the default ContextFact-F configuration, first annotates supported expressions with deterministic rules and then extracts facts from the annotated dialogue, retaining the original wording alongside the resolved dates or intervals. The extractor preserves these annotations and may correct them only when the source context provides clear contradictory evidence.

Table 8: QA performance under evidence-token budgets on LoCoMo. For Memora, we evaluate the semantic-retrieval variant (S), which provides ranked memories for budget-constrained packing, rather than the policy-retrieval variant (P), which gathers evidence through adaptive multi-step search. "Units" denotes the mean number of memory units supplied to the reader per question. **Bold** and underlined scores indicate the **best** and second-best results within each budget, respectively, including ties.

Budget	Method	Units	Single-hop			Temporal			Multi-hop			Open-domain			Overall		
			$F_1\uparrow$	$B_1\uparrow$	J ↑	$F_1\uparrow$	$B_1\uparrow$	J ↑	$F_1\uparrow$	$B_1\uparrow$	J ↑	$F_1\uparrow$	$B_1\uparrow$	J ↑	$F_1\uparrow$	$B_1\uparrow$	J ↑
0.5k	A-Mem	3.2	35.65	31.17	50.54	27.41	23.59	35.83	21.00	12.47	34.75	14.57	12.20	31.25	29.93	24.98	43.38
	MemoryOS	8.4	41.92	37.00	58.98	30.76	23.00	32.71	24.41	15.77	39.01	22.15	18.02	41.67	35.15	29.01	48.77
	SeCom	1.4	43.57	34.88	63.73	35.11	26.93	40.81	21.86	13.80	42.20	20.35	17.60	36.46	36.38	28.29	53.31
	Mem0	23.7	40.46	33.65	58.98	49.19	37.05	55.76	33.82	24.02	55.32	19.54	16.17	36.46	39.76	31.51	56.23
	LightMem	15.9	51.36	40.36	77.29	58.07	44.62	75.08	34.77	26.14	63.48	22.45	18.11	43.75	47.92	37.26	72.21
	SimpleMem	8.3	51.67	**46.37**	74.91	60.76	47.09	82.24	35.93	26.47	**65.25**	22.65	18.93	42.71	48.87	41.16	72.66
	AtomMem	16.4	52.68	42.34	75.86	60.79	51.32	73.83	**36.01**	**26.63**	63.12	**22.74**	18.41	40.62	49.45	39.84	70.91
	EMem	16.7	53.33	43.13	80.74	**64.44**	**53.21**	73.52	35.41	24.30	63.12	21.94	18.47	**45.83**	50.41	40.25	73.83
	Nemori	4.0	46.44	37.11	75.51	49.56	41.85	63.24	30.24	20.60	53.90	22.38	19.11	**45.83**	42.62	33.95	67.14
	Memora (S)	5.2	46.14	36.84	68.61	51.88	40.57	61.37	30.39	21.25	53.55	17.55	14.85	38.54	42.67	33.39	62.47
	ContextFact-F	4.2	**56.15**	46.06	**82.76**	62.16	52.35	**82.87**	29.64	21.42	56.03	22.57	**19.18**	43.75	**50.46**	**41.18**	**75.45**
1k	A-Mem	6.9	43.30	38.64	59.81	32.66	27.53	45.79	26.84	18.36	47.52	17.31	13.81	34.38	36.45	31.06	53.05
	MemoryOS	13.8	47.79	42.14	66.59	35.43	26.08	40.81	28.99	20.58	44.68	22.91	17.89	43.75	40.22	33.33	55.78
	SeCom	3.2	54.46	43.75	78.95	48.67	37.82	57.94	29.55	20.68	53.19	22.38	19.66	41.67	46.69	36.79	67.53
	Mem0	47.3	45.57	38.36	65.99	49.18	37.30	57.32	38.10	28.87	66.31	23.64	19.83	38.54	43.59	35.25	62.53
	LightMem	32.5	52.90	41.55	80.86	59.71	46.01	76.64	37.24	28.56	70.57	21.57	19.02	44.79	49.50	38.70	75.84
	SimpleMem	16.9	53.51	**48.29**	77.65	62.80	48.98	**84.74**	38.18	28.67	**73.05**	21.94	17.93	43.75	50.67	42.95	76.17
	AtomMem	33.3	54.77	44.43	80.38	61.93	52.45	73.83	**40.83**	**30.65**	71.28	22.55	19.25	43.75	51.70	42.01	75.06
	EMem	33.8	54.21	43.79	83.83	**65.45**	**54.73**	75.39	38.29	27.98	68.09	21.63	18.32	45.83	51.61	41.59	76.82
	Nemori	6.5	51.89	41.59	81.45	54.24	45.95	68.54	34.09	24.87	64.18	**24.75**	**21.20**	46.88	47.43	38.17	73.44
	Memora (S)	7.6	51.87	41.19	76.22	54.33	42.86	67.91	33.86	24.17	59.93	23.29	19.88	42.71	47.30	37.09	69.42
	ContextFact-F	8.9	**58.68**	47.69	**86.09**	63.64	53.61	84.42	37.03	28.58	66.67	24.24	20.91	**47.92**	**53.60**	**43.75**	**79.81**
2k	A-Mem	14.3	48.86	43.70	67.18	38.11	31.85	50.16	33.89	26.30	58.87	14.70	11.55	37.50	41.75	36.04	60.26
	MemoryOS	23.7	53.44	47.56	74.20	39.90	29.45	46.73	34.51	25.72	57.09	22.30	17.41	45.83	45.21	37.90	63.57
	SeCom	6.7	**60.35**	49.23	86.56	54.77	42.56	66.04	35.91	26.22	63.48	21.72	18.91	44.79	52.30	41.74	75.45
	Mem0	94.8	46.84	39.36	68.01	49.16	37.79	58.88	40.34	31.45	70.57	23.99	19.83	39.58	44.71	36.36	64.81
	LightMem	65.4	53.56	42.54	82.40	61.06	47.16	78.82	40.81	31.62	75.18	23.17	20.83	48.96	50.89	40.15	78.25
	SimpleMem	34.0	54.89	**49.69**	79.79	64.34	50.27	**87.54**	41.09	32.18	76.24	22.88	18.81	45.83	52.34	44.68	78.64
	AtomMem	66.9	56.82	46.41	83.71	63.17	53.38	77.88	**42.02**	**32.90**	**76.95**	**24.48**	**20.86**	47.92	53.42	43.80	79.03
	EMem	67.8	54.80	44.82	84.07	64.97	54.04	75.39	41.32	31.17	74.47	22.14	19.49	46.88	52.42	42.66	78.18
	Nemori	14.7	54.42	44.30	84.66	56.06	47.14	71.96	38.55	28.33	70.57	**24.48**	20.46	**51.04**	49.99	40.48	77.34
	Memora (S)	12.4	55.84	44.70	81.21	57.48	45.21	73.21	36.02	26.42	65.60	21.14	18.41	40.62	50.39	39.82	74.16
	ContextFact-F	18.2	60.24	49.44	**88.82**	**65.15**	**54.72**	86.92	40.42	31.32	72.34	24.45	20.62	48.96	**55.40**	**45.43**	**82.92**
4k	A-Mem	29.2	54.45	49.24	75.62	37.70	31.80	53.27	36.72	28.89	63.48	17.55	13.42	40.62	45.41	39.65	66.56
	MemoryOS	43.1	57.95	52.00	79.43	44.95	33.45	51.40	41.47	31.85	64.89	25.96	21.29	50.00	50.23	42.53	69.09
	SeCom	13.8	**63.21**	**52.15**	89.42	57.54	45.36	69.16	41.10	31.04	73.05	22.93	19.71	46.88	55.47	44.85	79.55
	Mem0	189.9	47.06	39.92	69.68	50.50	39.55	60.44	42.85	33.22	72.70	21.64	17.88	42.71	45.42	37.24	66.62
	LightMem	131.0	53.94	43.50	82.52	60.58	47.73	77.88	42.24	33.06	75.53	21.38	19.54	43.75	51.15	40.98	77.86
	SimpleMem	67.9	55.99	50.55	80.98	64.54	50.73	**86.60**	43.64	**34.75**	**82.98**	24.37	20.71	48.96	53.54	45.84	80.52
	AtomMem	133.5	57.93	48.19	85.49	62.89	54.22	76.01	43.80	33.91	79.79	23.89	20.18	44.79	54.26	45.09	79.94
	EMem	135.5	55.67	46.06	85.14	**65.12**	54.26	76.95	41.88	31.73	76.60	24.54	21.33	48.96	53.18	43.60	79.61
	Nemori	36.2	56.69	46.96	87.16	59.83	51.27	75.08	40.20	30.98	75.53	**26.49**	**21.90**	**54.17**	52.44	43.37	80.45
	Memora (S)	22.1	58.41	47.39	85.26	61.51	47.75	78.19	39.66	29.77	75.18	22.83	20.03	42.71	53.40	42.53	79.29
	ContextFact-F	36.8	61.21	51.22	**90.37**	64.11	**54.35**	85.05	**44.21**	34.53	78.37	23.31	18.68	51.04	**56.34**	**46.79**	**84.61**
8k	A-Mem	59.1	57.09	51.51	78.12	40.60	33.86	56.39	42.17	34.14	74.47	18.35	15.02	40.62	48.51	42.38	70.58
	MemoryOS	80.5	61.52	**55.47**	85.26	46.89	34.69	53.58	44.59	**36.77**	74.47	24.77	19.87	48.96	53.08	45.50	74.42
	SeCom	29.1	**64.46**	53.79	**91.20**	59.49	47.56	71.34	42.66	32.86	78.72	23.65	20.97	50.00	**56.89**	46.61	82.21
	Mem0	369.4	47.76	41.03	70.87	49.37	38.01	60.44	42.59	33.02	70.21	22.30	17.96	42.71	45.56	37.50	66.82
	LightMem	260.1	54.65	44.61	83.59	61.05	48.38	80.06	42.13	33.15	77.30	22.94	20.53	44.79	51.71	41.80	79.29
	SimpleMem	136.0	56.07	50.56	81.57	65.39	51.23	**87.23**	43.77	35.09	**83.69**	**26.33**	**22.00**	48.96	53.91	46.09	81.10
	AtomMem	263.6	58.89	49.55	85.85	62.70	54.41	78.82	**44.66**	34.69	81.56	23.64	20.25	44.79	54.88	46.01	81.04
	EMem	269.3	56.22	47.35	85.14	**66.80**	55.56	77.88	43.28	33.26	78.37	24.43	20.84	48.96	54.08	44.83	80.13
	Nemori	80.0	56.36	47.16	88.35	61.91	53.64	77.57	42.28	32.71	79.08	25.36	21.09	**51.04**	53.01	44.24	82.08
	Memora (S)	42.6	60.36	49.58	87.75	64.02	50.83	79.75	43.34	33.09	77.66	23.85	20.39	44.79	55.73	45.00	81.56
	ContextFact-F	73.8	61.66	52.29	90.49	65.12	**55.95**	86.60	44.47	34.80	79.08	22.55	19.72	45.83	56.80	**47.82**	**84.81**

Table 9: Memory decomposition under evidence-token budgets on LoCoMo with Qwen3.5-27B. Units denotes the mean number of memory units supplied to the reader per question. Bold and underlined scores indicate the best and second-best results within each retriever and budget, respectively, including ties.

Budget	Method	Units	Single-hop			Temporal			Multi-hop			Open-domain			Overall		
			$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$	$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$	$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$	$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$	$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$
Dense																	
0.5k	ContextFact	4.7	**54.42**	**44.56**	**80.86**	58.89	49.40	**81.62**	35.60	27.22	64.89	21.60	17.85	**43.75**	49.86	**40.73**	75.78
	Sentence split	15.9	51.78	42.35	74.67	60.43	50.24	79.13	36.88	**28.75**	67.73	21.12	17.78	41.67	48.94	39.97	72.27
	Atomic Fact	20.8	53.47	42.76	80.26	**64.01**	**53.51**	**81.62**	**37.30**	26.94	**69.15**	**22.50**	**18.96**	41.67	**50.77**	40.62	**76.10**
1k	ContextFact	9.8	**56.72**	**46.43**	**85.02**	62.51	52.86	**84.11**	38.92	**30.96**	69.15	**24.69**	**20.54**	**54.17**	**52.67**	**43.32**	**80.00**
	Sentence split	32.5	55.15	45.33	78.95	61.62	51.67	83.18	39.22	29.69	71.99	20.75	17.57	43.75	51.44	42.06	76.36
	Atomic Fact	42.3	55.60	44.53	82.76	**63.82**	**53.92**	82.55	**40.32**	30.46	**75.18**	22.52	18.97	42.71	52.45	42.32	78.83
2k	ContextFact	20.0	**58.91**	**48.42**	**87.16**	62.74	53.20	**85.05**	**41.39**	**33.63**	**77.66**	**24.61**	**21.20**	**51.04**	**54.36**	**45.01**	**82.73**
	Sentence split	65.8	57.27	47.24	81.69	63.34	52.99	83.80	39.64	30.46	74.82	23.66	19.84	48.96	53.21	43.66	78.83
	Atomic Fact	85.4	56.68	46.33	84.30	**64.47**	**54.62**	83.18	41.38	31.65	77.30	22.89	19.37	44.79	53.40	43.69	80.32
3k	ContextFact	30.0	**59.49**	**49.52**	**87.99**	64.51	**55.05**	**87.23**	**43.94**	**35.24**	79.79	24.32	19.96	**50.00**	**55.50**	**46.22**	**83.96**
	Sentence split	98.9	58.18	48.54	82.40	**64.87**	54.44	85.36	40.86	31.70	74.82	23.48	19.96	46.88	54.24	44.90	79.42
	Atomic Fact	128.6	56.86	46.88	83.59	64.58	54.79	84.11	42.75	32.71	**81.21**	**24.75**	**21.27**	46.88	53.88	44.34	80.97
4k	ContextFact	39.8	**60.54**	**50.64**	**89.54**	63.58	54.26	85.05	**44.28**	**35.24**	80.50	**24.93**	20.94	46.88	**55.98**	**46.72**	**84.29**
	Sentence split	131.8	59.41	49.76	83.59	64.19	54.16	**85.67**	41.97	32.68	78.37	24.91	**21.09**	**51.04**	55.06	45.76	81.04
	Atomic Fact	171.8	57.80	47.94	84.54	**64.35**	**54.76**	82.87	43.34	33.22	**81.21**	23.01	20.14	44.79	54.35	44.93	81.10
RRF																	
0.5k	ContextFact	4.2	**56.15**	**46.06**	**82.76**	**62.16**	**52.35**	**82.87**	29.64	21.42	56.03	22.57	19.18	**43.75**	**50.46**	**41.18**	**75.45**
	Sentence split	15.4	53.41	43.23	76.81	60.87	50.99	80.06	33.48	23.71	55.67	20.68	17.61	42.71	49.28	39.68	71.49
	Atomic Fact	20.3	53.56	42.68	80.74	61.67	51.69	79.44	**35.92**	**26.21**	**65.96**	**23.50**	**20.22**	39.58	50.15	40.15	75.19
1k	ContextFact	8.9	**58.68**	**47.69**	**86.09**	**63.64**	**53.61**	**84.42**	37.03	28.58	66.67	**24.24**	**20.91**	**47.92**	**53.60**	**43.75**	**79.81**
	Sentence split	31.4	57.04	46.43	82.76	62.55	52.50	81.93	37.39	28.60	63.48	20.70	17.28	43.75	52.33	42.61	76.62
	Atomic Fact	41.2	55.80	44.69	82.76	62.58	51.96	81.31	**38.83**	**28.68**	**69.50**	22.33	19.49	42.71	52.02	41.70	77.53
2k	ContextFact	18.2	**60.24**	**49.44**	**88.82**	**65.15**	**54.72**	**86.92**	40.42	**31.32**	72.34	**24.45**	**20.62**	**48.96**	**55.40**	**45.43**	**82.92**
	Sentence split	63.5	59.30	48.58	85.26	64.29	54.10	85.36	**41.08**	31.24	71.63	22.18	19.35	**48.96**	54.69	44.73	80.52
	Atomic Fact	83.1	56.65	46.36	83.71	63.07	53.29	81.62	40.24	30.44	**73.76**	22.00	18.77	41.67	52.82	43.17	78.83
3k	ContextFact	27.6	**61.12**	**50.57**	**89.77**	**64.68**	**55.36**	**85.36**	**42.66**	**33.30**	**76.60**	**26.14**	**22.16**	**52.08**	**56.30**	**46.63**	**84.09**
	Sentence split	95.5	60.27	50.05	86.33	64.47	54.31	**85.36**	41.54	31.15	73.40	24.18	20.71	50.00	55.46	45.65	81.49
	Atomic Fact	125.1	56.86	46.76	84.19	63.42	53.83	82.24	41.72	32.29	75.89	20.98	17.83	41.67	53.22	43.78	79.61
4k	ContextFact	36.8	**61.21**	**51.22**	**90.37**	64.11	54.35	85.05	**44.21**	**34.53**	78.37	**23.31**	**18.68**	**51.04**	**56.34**	**46.79**	**84.61**
	Sentence split	127.5	60.58	50.55	86.09	**64.44**	54.52	**86.29**	43.34	32.76	**78.72**	22.20	18.48	47.92	55.84	46.12	82.40
	Atomic Fact	167.4	58.01	47.87	84.90	63.84	**54.81**	81.93	41.59	31.92	75.18	20.89	17.85	43.75	53.91	44.53	79.94

Table 10: Temporal normalization ablations on LoCoMo. All variants use the same retrieval and readout pipeline.

Method	Tokens	Single-hop ($n=841$)			Temporal ($n=321$)			Multi-hop ($n=282$)			Open-domain ($n=96$)			Overall ($n=1{,}540$)		
		$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$	$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$	$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$	$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$	$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$
GPT-4.1-mini																
P0: No temporal guidance	833	59.44	**52.44**	89.42	59.94	47.35	80.06	**43.74**	**33.97**	76.60	29.64	23.28	58.33	54.81	46.18	83.18
P1: Joint normalization	855	58.80	51.98	88.59	61.90	51.55	81.93	43.56	33.92	**78.72**	31.66	24.87	61.46	54.96	46.89	83.70
P2: Rule-guided normalization	932	**59.45**	52.42	**90.37**	**62.47**	**53.47**	**85.67**	43.01	32.61	78.37	**32.90**	**27.24**	**64.58**	**55.41**	**47.44**	**85.58**
Qwen3.5-27B																
P0: No temporal guidance	1,061	**61.92**	50.20	**91.20**	63.85	48.99	78.82	42.24	32.61	73.76	**26.66**	**23.13**	53.12	56.52	45.04	83.05
P1: Joint normalization	1,091	61.35	50.06	90.84	**65.97**	**55.33**	84.74	**43.56**	**33.84**	**76.95**	25.72	21.34	52.08	**56.84**	**46.40**	84.61
P2: Rule-guided normalization	1,153	61.71	**50.27**	90.49	65.54	54.53	**88.47**	43.00	33.80	**76.95**	25.51	20.80	**54.17**	56.83	46.30	**85.32**

Table 11: General conventions for rule-guided temporal normalization.

Type	Rule
Reference time	Use the session date by default.
Day offsets	Resolve expressions such as "yesterday" to individual dates.
Week offsets	Resolve expressions such as "last week" to the corresponding start–end intervals.
Month/year offsets	Resolve the corresponding month or year while retaining the original granularity.
Weekday references	Interpret "last" and "next" as the nearest previous and future occurrences.
Unresolved expressions	Leave vague or unsupported expressions unchanged.

Table 12: Entity reconstruction ablation on LoCoMo. *Provisional entities* are obtained after incremental cross-session binding. From the same provisional bank, *Recon. (CF)* uses stored ContextFacts for reconstruction, while *Recon. (Source)* revisits the linked source dialogue. All variants share the fact bank and downstream retrieval/readout pipeline.

Variant	Tokens	Single-hop ($n = 841$)			Temporal ($n = 321$)			Multi-hop ($n = 282$)			Open-domain ($n = 96$)			Overall ($n = 1{,}540$)		
		$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$	$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$	$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$	$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$	$F_1\uparrow$	$B_1\uparrow$	$J\uparrow$
GPT-4.1-mini																
Provisional entities	4,568	**60.23**	**52.62**	90.37	61.68	53.32	86.60	**45.16**	**35.52**	**81.56**	**31.88**	**26.03**	**60.42**	**56.00**	**47.98**	86.10
+ Recon. (CF)	4,144	58.52	51.31	90.73	63.22	54.59	88.16	44.60	34.92	**81.56**	26.73	21.96	56.25	54.97	47.16	86.36
+ Recon. (Source)	4,139	59.36	51.81	**91.44**	**64.31**	**55.28**	**89.72**	45.06	35.43	**81.56**	29.82	24.05	59.38	55.93	47.80	**87.27**
Qwen3.5-27B																
Provisional entities	2,222	**61.05**	**50.45**	90.01	**65.62**	54.36	87.54	45.12	35.53	81.21	24.83	20.88	48.96	**56.83**	46.69	85.32
+ Recon. (CF)	2,989	60.28	50.27	90.13	65.25	**54.65**	**89.41**	**45.58**	36.04	80.14	**27.41**	**24.09**	**53.12**	56.58	**46.95**	85.84
+ Recon. (Source)	2,907	60.65	50.29	**90.49**	64.96	**54.65**	88.79	45.33	**36.25**	**84.04**	26.96	23.37	51.04	56.64	**46.95**	**86.49**

Analysis. Joint normalization can introduce incorrect absolute dates during fact rewriting. Comparing generated facts against source dialogues and session dates reveals recurring failure modes: erroneous day offsets, misassigned months or years, past events projected into the future, and dates contradicting explicit weekdays. For instance, in an August 14, 2023 session, GPT-4.1-mini normalized "next month" to August rather than September; similarly, in an August 3, 2023 session, Qwen3.5-27B mapped "last week" to August 27, 2022. We also observe distortions in temporal granularity, such as multi-day or week-level expressions collapsed into arbitrary dates, and weekends expanded into full weeks. Rule-guided pre-annotation provides explicit temporal anchors before fact extraction, reducing the need for the extractor to infer absolute dates or intervals.

Joint normalization: temporal instruction (P1)

```
- Dual-format time references
    When the evidence gives a time relative to the conversation date, preserve
     the relative expression and append its resolved absolute date or date
    range: `relative time (absolute date)`. Resolve a day-level expression to
    one date, a week-level expression to a date range, and a year-level
    expression to the year. Write dates as `Month D, YYYY` without a weekday or
     time. Keep vague expressions such as `recently`, `a few weeks ago`, and `
    around three years ago` unchanged rather than inventing precision.
```

F.4 Entity Reconstruction

We compare three variants of entity-bank construction. *Provisional entities* are obtained through incremental cross-session binding, which updates entity descriptions while retaining the original session-level entity entries; no subsequent reconstruction is performed. *Recon. (CF)* reconstructs this bank using linked ContextFacts as supporting evidence. *Recon. (Source)*, the default ContextFact configuration, instead revisits the linked source dialogue to verify entity identities and descriptions and organize cross-session information into records. The two reconstruction variants start independently from the same provisional bank. Within each backbone, all variants share the fact bank and downstream retrieval and readout configuration.

Table 13: Entity-linked expansion and entity readout on LoCoMo with Qwen3.5-27B. A and B denote fact-seeded and entity-seeded expansion, respectively; E denotes complete-entity readout. CFn and En denote n selected ContextFacts and complete entities. Recall denotes linked-turn recall. All reported scores are percentages; **bold** marks the maximum in each column.

Setting	Path		E	Reader Input	Tokens	Single	Temp.	Multi-hop		Open	Overall
	A	B				J	J	J	Recall↑	J	J
Expansion without entity readout											
R0: No expansion	✗	✗	✗	CF10	1,153	90.49	88.47	76.95	80.03	54.17	85.32
R1: Fact-seeded expansion	✓	✗	✗	CF10	1,168	90.37	89.10	77.66	84.18	52.08	85.39
R2: Entity-seeded expansion	✗	✓	✗	CF10	1,170	**90.84**	**89.41**	79.08	85.60	**55.21**	86.17
R3: Dual-path expansion	✓	✓	✗	CF10	1,171	**90.84**	88.79	78.37	86.14	54.17	85.84
Entity readout											
R4: w/o fact expansion	✗	✗	✓	CF10+E10	2,889	90.73	87.85	82.27	91.43	50.00	86.04
R5: Full method	✓	✓	✓	CF10+E10	2,907	90.49	88.79	**84.04**	92.02	51.04	**86.49**
Additional-fact controls											
C1: Direct CF retrieval	✗	✗	✗	CF30	3,433	90.13	86.92	80.14	92.03	54.17	85.39
C2: Replace E10 with 20 CFs	✓	✓	✗	CF30	3,501	90.73	85.36	81.21	**93.69**	**55.21**	85.65

Table 14: Reranker comparison on LoCoMo with Qwen3.5-27B. All rerankers receive the same 30 RRF candidates and select ten fact memories; RRF-10 and RRF-30 are direct-retrieval references. All scores are percentages, and **bold** denotes the best result among the rerankers in each column.

Setting	Avg. tokens	Single-hop			Temporal			Multi-hop			Open-domain			Overall		
		F_1	B_1	J	F_1	B_1	J	F_1	B_1	J	F_1	B_1	J	F_1	B_1	J
Direct-retrieval references																
RRF-10	1,132	59.18	48.52	87.04	64.44	54.29	86.60	37.01	28.42	67.02	24.07	20.46	46.88	54.03	44.29	80.78
RRF-30	3,383	60.97	50.64	90.37	64.57	55.34	85.36	42.91	34.08	76.60	24.56	20.74	47.92	56.14	46.72	84.16
RRF-30 → rerank-10																
MemReranker-4B	1,153	**61.71**	**50.27**	**90.49**	**65.54**	**54.53**	**88.47**	**43.00**	**33.80**	**76.95**	**25.51**	20.80	**54.17**	**56.83**	**46.30**	**85.32**
Qwen3-Reranker-4B	1,076	60.77	49.94	89.54	64.42	54.18	87.54	41.78	32.66	73.76	24.64	**20.98**	52.08	55.80	45.86	83.90
BGE-reranker-v2-m3	1,216	61.15	50.25	90.13	64.47	53.90	86.92	42.71	33.02	75.89	23.38	20.76	51.04	56.11	46.02	84.42

F.5 Entity-Linked Expansion and Entity Readout

Tab. 13 expands the entity-mechanism ablation in Tab. 3 into its controlled variants. We additionally report linked-turn recall, defined as the mean fraction of annotated supporting turns covered by the provenance of selected ContextFacts and selected entities when applicable. Here, A denotes fact-seeded expansion, B entity-seeded expansion, and E complete-entity readout. R0/R4 and R3/R5 use identical selected ContextFacts and ordering, isolating the effect of entity readout without changing the fact evidence. R4/R5 use the same ten complete entities, isolating the additional effect of fact expansion. The two fact-only controls test whether the gains can instead be obtained by reading more ContextFacts: C1 reranks 60 directly retrieved facts to 30, whereas C2 preserves R3's top ten expanded facts and appends the next twenty from the same expanded ranking in place of the entity readout.

F.6 Choice of Reranker

To isolate the effect of reranking, we hold the fact bank, initial candidate pool, and final selection size fixed. Each reranker receives the same 30 RRF candidates and selects ten fact memories, without entity-linked expansion or entity readout. RRF-10 and RRF-30 provide direct-retrieval references without reranking. Among the rerankers, MemReranker-4B achieves the highest overall judge accuracy (85.32%), outperforming BGE-reranker-v2-m3 and Qwen3-Reranker-4B by 0.90 and 1.42 points, respectively. It also improves over reading all 30 RRF candidates by 1.16 points while using roughly one third as many evidence tokens. We therefore use MemReranker-4B for the main ContextFact experiments. Qwen3-Reranker-4B uses the same task instruction and scoring template as MemReranker, whereas BGE uses its native query–document format.

Table 15: Efficiency comparison on LoCoMo using GPT-4.1-mini. Memory construction (Build) tokens are averaged per conversation and reported in thousands. Search and end-to-end (E2E) times are measured per query in seconds. Acc. denotes overall judge accuracy.

Method	Build (k) ↓		Search (s) ↓			E2E (s) ↓			Acc. (%) ↑
	Input	Output	Mean	P50	P95	Mean	P50	P95	
Mem0	1122.9	123.9	0.06	0.06	0.07	0.93	0.84	1.13	62.14
MemoryOS	305.3	102.0	1.64	1.44	2.50	2.50	2.30	3.44	71.88
LightMem	71.6	20.0	0.02	0.02	0.03	0.86	0.82	1.06	79.61
SimpleMem	40.7	77.2	0.06	0.06	0.07	1.50	1.42	2.07	80.06
EMem	56.8	17.7	2.27	1.58	6.24	3.09	2.40	7.12	79.42
Nemori	171.5	28.5	0.42	0.39	0.56	1.29	1.22	1.66	82.47
Memora (P)	609.4	68.5	6.69	5.84	13.87	8.33	7.48	15.86	85.32
ContextFact-F	100.9	20.3	0.18	0.18	0.20	0.95	0.91	1.11	85.58
ContextFact-E	273.5	71.1	0.70	0.66	1.04	1.49	1.45	1.90	87.27

F.7 Computational Efficiency

Tab. 15 reports memory-construction token use and per-query latency on LoCoMo with GPT-4.1-mini under the deployment described in Appx. D.3.

Accuracy–efficiency trade-off. ContextFact-F reaches 85.58% overall judge accuracy, slightly exceeding Memora (P) at 85.32%, while using substantially fewer construction tokens (100.9k input and 20.3k output versus 609.4k and 68.5k). Its mean end-to-end latency is 0.95 s, compared with 8.33 s for Memora (P), corresponding to an $8.8\times$ speedup. Adding the entity view increases the cost of ContextFact-E to 273.5k input and 71.1k output construction tokens and 1.49 s mean end-to-end latency, while raising overall accuracy to 87.27%. Thus, the entity view introduces additional construction and query-time cost, but the full system remains $5.6\times$ faster than Memora (P) in mean end-to-end latency while achieving higher accuracy.

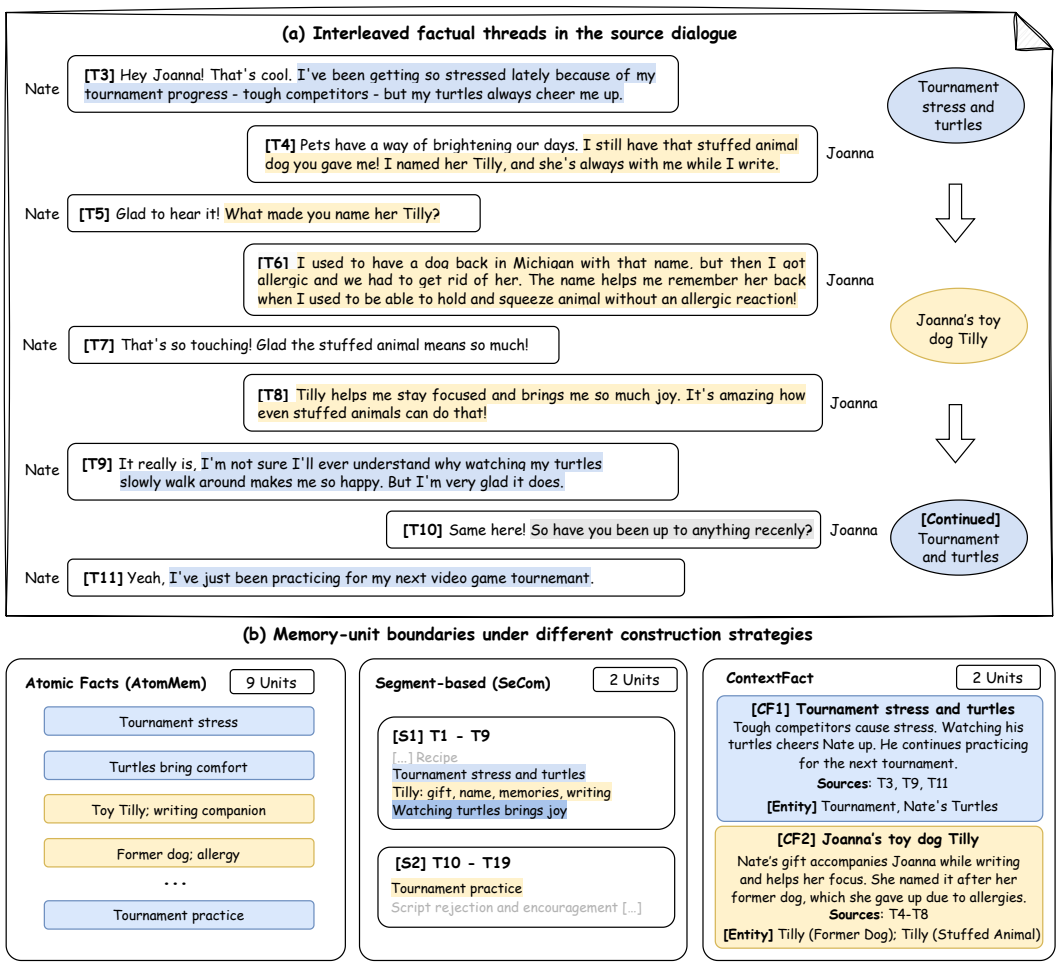


Figure 7: Preserving context across nonadjacent turns.

G Case Studies

In this section, we provide five case studies illustrating how ContextFact constructs and uses conversational memory. We first examine how fact memories preserve context across nonadjacent turns and separate distinct facts within a turn. We then illustrate how the entity view provides access to information missed by direct fact retrieval, resolves cross-session references, and supports counting distinct objects from retrieved evidence.

G.1 Memory-Unit Boundaries

The following two examples compare ContextFact's context preservation against atomic-fact-based methods (represented by AtomMem) and segment-based methods (represented by SeCom).

Context across nonadjacent turns. Fig. 7 illustrates how related context can be interleaved across nonadjacent turns: Nate first describes his tournament stress and the comfort his turtles bring him in T3, engages with Joanna's discussion of her stuffed dog from T4 to T8, returns to his turtles in T9, and finally shifts to tournament practice in T11.

AtomMem stores the related details in separate atomic units, while SeCom's contiguous segments mix the two discussions and separate Nate's later update from the earlier context. ContextFact groups T3, T9, and T11 into one memory, connecting tournament stress, emotional support, and continued practice; a separate memory preserves Tilly's role as a writing companion together with

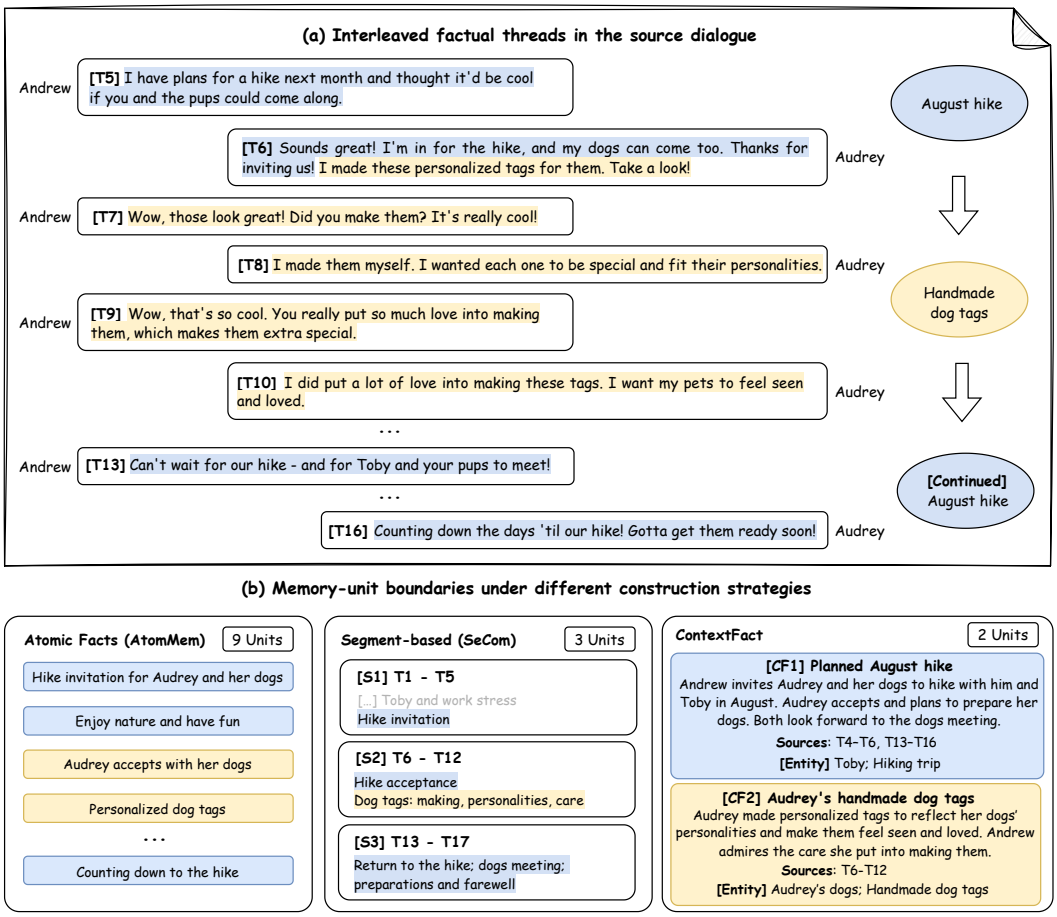


Figure 8: Separating distinct facts within a turn.

the connection to Joanna's former dog that explains its name. These boundaries keep each focal fact together with its supporting context, even when another discussion intervenes.

Distinct facts within a turn. Fig. 8 shows how a single turn can contribute to two distinct memories. In T6, Audrey accepts Andrew's hike invitation and introduces the personalized tags she made for her dogs. The conversation then develops the tag-making discussion before returning to the hike and preparations for the dogs.

AtomMem stores the invitation and acceptance as separate atomic units. SeCom's contiguous segments separate the invitation from the acceptance, group the acceptance with the dog-tag discussion, and place the later hike preparations in another segment. ContextFact groups the invitation, acceptance, and preparations into one memory about the planned hike. A separate memory connects the handmade tags with their design and Audrey's motivation for making them. Both memories cite T6, but each retains the information relevant to its own focal fact. Memory boundaries can therefore separate distinct facts within a turn while gathering their supporting context from other turns.

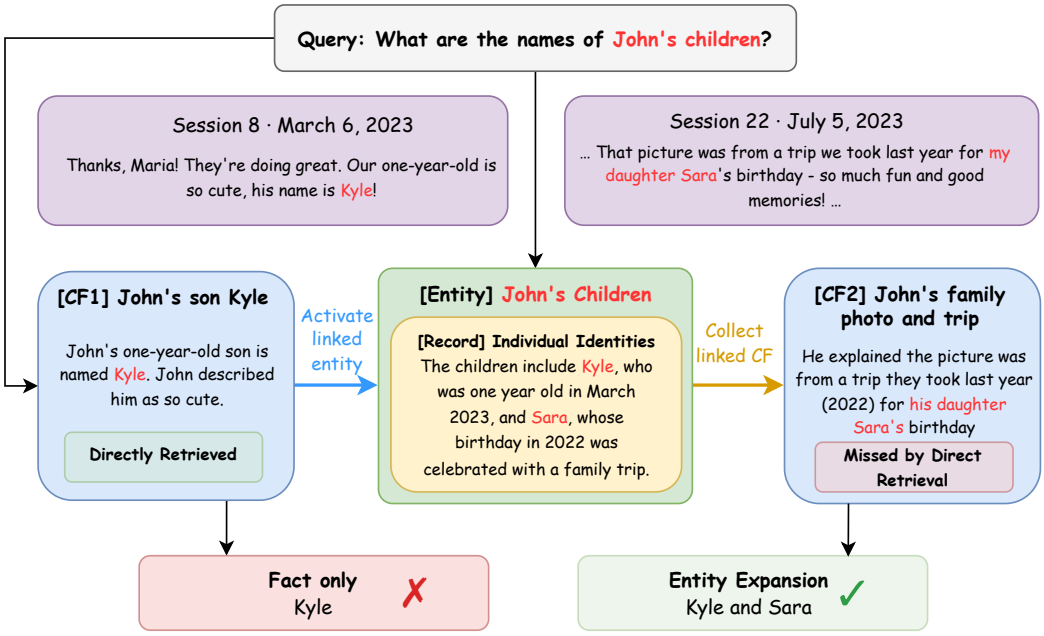


Figure 9: Entity-linked expansion recovers evidence missed by direct fact retrieval.

G.2 Entity-Linked Expansion

The question in Fig. 9 requires combining information from different sessions to identify John's children. Direct fact retrieval finds the memory naming his son Kyle but misses the memory mentioning his daughter Sara in the context of a family birthday trip.

The two memories are connected through the entity *John's Children*, providing two routes to the missing information. ContextFact can follow the entity's links to retrieve the birthday-trip memory, or the reader can obtain Sara's name directly from the entity record. Either route supplies the missing name and supports the complete answer "Kyle and Sara", whereas direct fact retrieval alone provides evidence for Kyle.

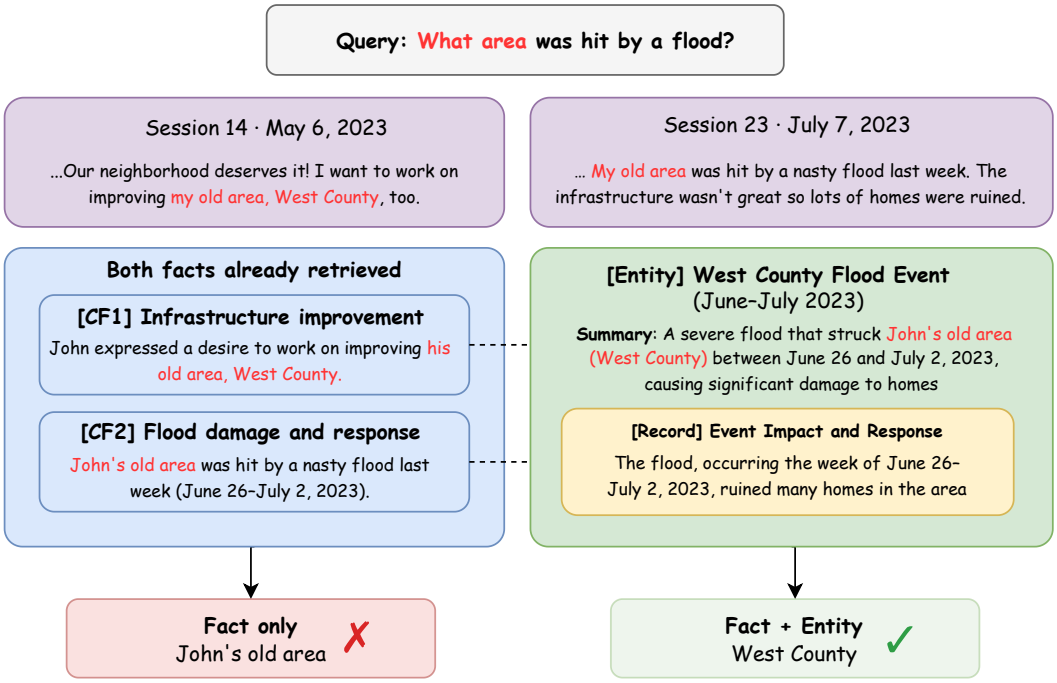


Figure 10: Entity readout resolves cross-session references.

G.3 Cross-Session Entity Readout

The following two examples illustrate how the entity view helps resolve references and count distinct objects across sessions.

Resolving cross-session references. In Fig. 10, John identifies his "old area" as West County in one session and reports a flood in his old area in a later session. Both fact memories are retrieved, but the fact-only reader answers "John's old area", without resolving the reference to a named location.

The entity *West County Flood Event* explicitly associates the flood with West County in its name and summary. Adding entity readout while keeping the selected fact memories fixed yields the answer "West County". The explicit connection between the location and the event helps the reader interpret the evidence already retrieved.

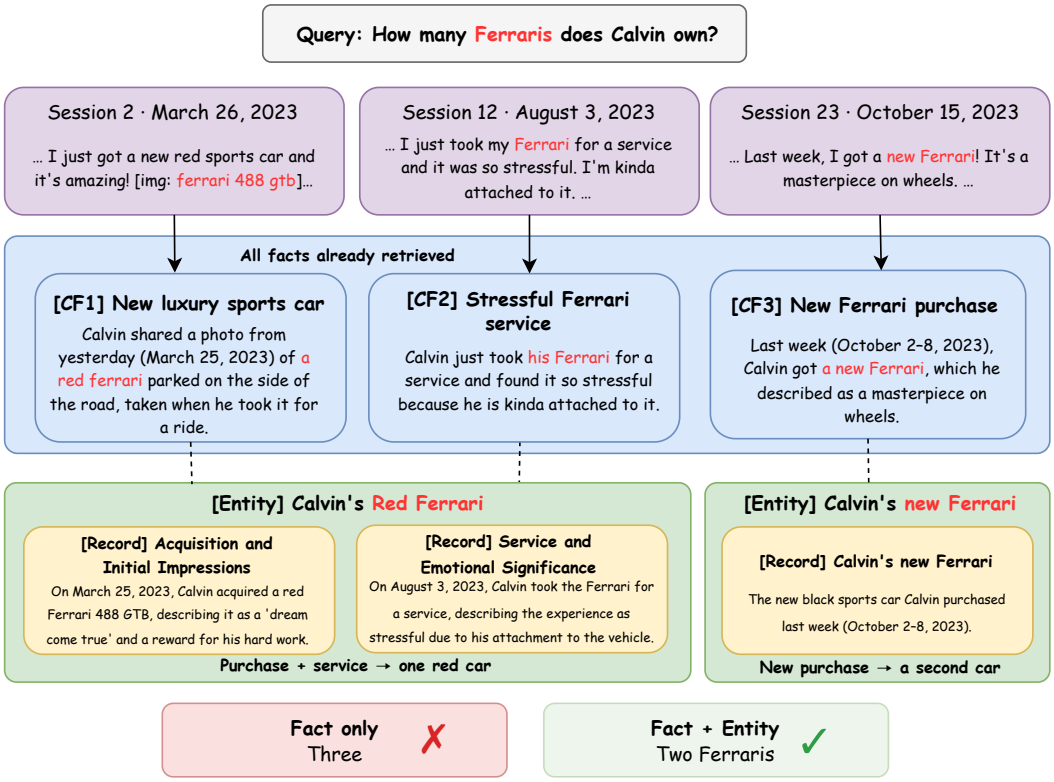


Figure 11: Entity readout supports counting distinct objects across sessions.

Counting distinct objects across sessions. In Fig. 11, three retrieved fact memories describe Calvin's purchase of a red sports car, a subsequent Ferrari service, and a later Ferrari purchase. Counting the cars requires determining which mentions refer to the same vehicle. Treating each memory as a separate car would incorrectly give a count of three.

The entity view groups the initial purchase and service as records of the same red Ferrari, while representing the later purchase as a separate vehicle. Presenting these entities alongside the facts makes the two cars explicit and supports the answer "Two Ferraris".