INFINI MEMORY: MAINTAINABLE TOPIC DOCUMENTS FOR LONG-TERM LLM AGENT MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) enable personal assistant agents that work with users over long-term interactions. These assistants need memory that retains information across conversations and updates it when that information changes. Existing memory systems typically extract and summarize conversations into retrievable memory records. However, this process can fragment related information across records, causing incomplete retrieval and conflicting updates, while repeated summarization progressively discards details. We propose **Infini Memory**, a long-term memory system inspired by organizing notes around topics while preserving links to their source conversations. It organizes extracted entries into topic documents that serve as persistent units of revision. When new information arrives, the topic-document pipeline revises each selected document using its complete body and the incoming entries. A separate Source-Evidence Layer archives dialogue before extraction and preserves links from entries to their sources as documents evolve. At query time, the system retrieves relevant topic documents and follows source links to recover missing details. Experiments on MemoryAgentBench show that Infini Memory reaches an overall score of 64.7, 19.2 points above the strongest evaluated baseline. On the original LongMemEval-S benchmark, it achieves 96.4% accuracy across six question types.

1 INTRODUCTION

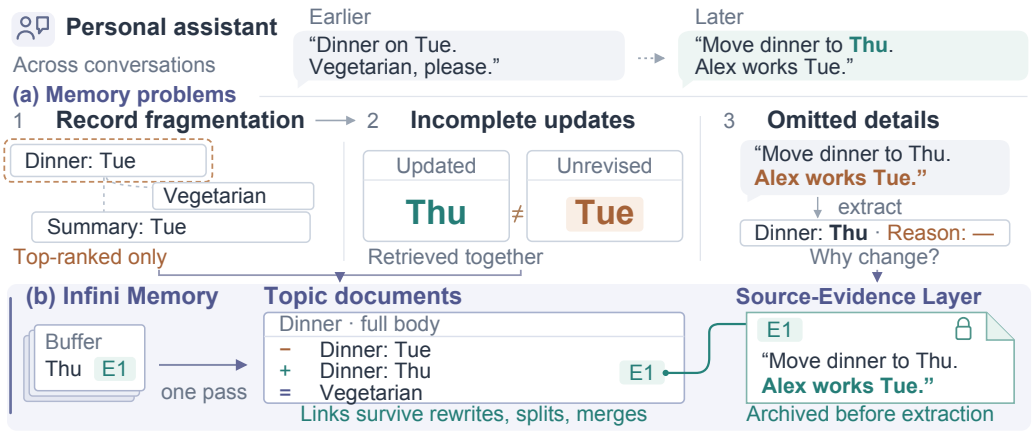

Figure 1: (a) Three memory system problems. (b) The corresponding designs in Infini Memory.

Large language models (LLMs) have enabled agents to plan, use tools, and act in their environments (Yao et al., 2023; Sumers et al., 2024). These capabilities support LLM-based personal assistants that work with users over long-term interactions, helping with ongoing plans and everyday requests (Zhong et al., 2024). To sustain that assistance, agents need to remember users' preferences, prior decisions, and task details that may no longer be available in the current context (Maharana et al., 2024).

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.

Memory systems support this persistence by storing selected information outside the context window, and agents can retrieve it for later requests (Packer et al., 2023). As conversations accumulate, users may correct earlier statements, revise their plans, or clarify when a preference applies (Yuan et al., 2025). A useful memory connects new information to relevant earlier material, and distinguishes what currently holds from the history behind it. Existing memory systems address this evolving history by extracting and updating records (Chhikara et al., 2025), linking related records (Xu et al., 2025), or organizing them into thematic or graph structures (Hu et al., 2026; Rasmussen et al., 2025).

However, these approaches can leave stored information incomplete or inconsistent as conversations grow. Figure 1(a) illustrates three ways this continuity can break. *First, record fragmentation separates related information.* Statements about the same subject may be stored across records created from different conversations. An update that considers only the top-ranked records can miss an earlier condition or a later correction (Pan et al., 2025; Chhikara et al., 2025). *Second, incomplete memory updates leave conflicting records.* Records outside the revised set remain retrievable, allowing older information to appear beside its replacement (Rasmussen et al., 2025). *Third, concise records can lose supporting details.* Extraction and summarization produce concise records that are easier to organize and reuse. These records may omit dates, conditions, or surrounding dialogue needed by later questions (Wu et al., 2025). Subsequent updates over those records carry the omission forward.

In this paper, we propose Infini Memory, a long-term memory system inspired by notes organized around topics and connected to their sources (Figure 1(b)). We implement this idea through topic documents linked to a separate Source-Evidence Layer. Its topic-document path extracts new information into entries and holds them in a buffer. During consolidation, each selected topic document is revised using its complete body together with the incoming entries, making the document a persistent unit of revision. A complementary Source-Evidence path archives dialogue before extraction, preserving message roles and order in separate Evidence objects. Entries retain links to those objects as documents are reorganized. At query time, a source-aware reader searches maintained memory and archived dialogue to assemble the context for answering.

Our contributions are:

- We organize memory into topic documents and revise each selected document with its complete body and incoming entries. The document provides a persistent unit for updating related information together.
- We introduce a Source-Evidence Layer that archives dialogue before extraction and preserves entry-to-source links through document reorganization, keeping the original turns available to retrieval.
- We evaluate Infini Memory on MemoryAgentBench and on the original LongMemEval-S. It scores 64.7 on the four-capability aggregate, 19.2 points above the strongest of seven evaluated baselines. On the LongMemEval-S, its source-aware configuration achieves 96.4% accuracy across six question types.

2 Related Work

2.1 Memory Demands of Long-Term Assistants

Long-term assistants build persistent representations of users from interaction history, so that later responses can reflect earlier statements, the conditions under which they apply, and subsequent feedback (Zhong et al., 2024; Yuan et al., 2025; Huang et al., 2026). As conversations accumulate, the information needed to represent a user changes.

Benchmarks make these demands concrete. LoCoMo evaluates question answering and event summarization over extended dialogue (Maharana et al., 2024), while LongMemEval tests information extraction, multi-session and temporal reasoning, knowledge updates, and abstention (Wu et al., 2025). MemoryAgentBench groups incremental memory demands into accurate retrieval, test-time learning, long-range understanding, and selective forgetting (Hu et al., 2025). Preference benchmarks examine whether responses follow explicit or implicit preferences (Zhao et al., 2025) and adapt to evolving user profiles (Jiang et al., 2025). Together, these benchmarks connect information retention with revision and appropriate use in context.

2.2 AGENTIC MEMORY SYSTEMS

Agentic memory systems expose storage and retrieval as operations an LLM can invoke (Packer et al., 2023), or learn policies for choosing memory operations (Yu et al., 2026).

Memory organization and updating. A central design choice is the content considered during one update. Mem0 compares each extracted fact with semantically similar stored facts before choosing an action (Chhikara et al., 2025). A-MEM creates linked notes and evolves the contextual attributes of retrieved neighbors (Xu et al., 2025). LightMem groups information by topic and moves long-term updates to an offline stage (Fang et al., 2026). Other systems maintain broader structures. EverMemOS consolidates episodic MemCells into thematic MemScenes (Hu et al., 2026); Zep maintains temporal relations in a knowledge graph (Rasmussen et al., 2025); and Hindsight separates facts, experiences, summaries, and evolving beliefs (Latimer et al., 2026). In Infini Memory, the selected topic document is the unit of revision: its complete body is considered together with buffered entries, while splitting and merging adjust the scope of future revisions (Section 4).

Source preservation. Derived memory also needs a route back to the interactions from which it was formed. PGMem connects persona signals to conversational context through provenance and evidence edges (Choi et al., 2026). LycheeMemory V2 retrieves typed records alongside raw dialogue turns and handles competing records when assembling an answer (Li et al., 2026). The Source-Evidence path of Infini Memory archives dialogue before extraction in separate Evidence objects and retains source identifiers on entries as documents are reorganized. Its reader can search the archived dialogue alongside maintained memory, giving revisable entries and their original turns distinct roles at retrieval time.

3 PRELIMINARY

Infini Memory separates the arrival of new entries from the revision of their organization. We describe this process through a memory state and three memory operations.

Memory State. Let $t \in \mathbb{N}$ index successive memory states. At step t, the memory state consists of a buffer C_t of recent entries, a collection D_t of topic documents, and an Evidence archive $\mathcal{E}_t$ of source dialogue:

$$\mathcal{M}_t \triangleq \langle C_t, D_t, \mathcal{E}_t \rangle. \tag{1}$$

We denote an entry by f, consisting of text and associated metadata. When source evidence is enabled, its `origin` metadata defines a partial source mapping $\rho_t(f) = f.\texttt{origin}$, where $\mathcal{E}_t[\rho_t(f)]$ denotes the corresponding archived dialogue. The index t counts state transitions, separately from an entry's write order and event time.

Memory Operations. For an incoming interaction z and a query q, we define three memory operations:

$$\mathcal{M}_{t+1} \leftarrow \mathrm{Ingest}(\mathcal{M}_t, z), \qquad \mathcal{M}_{t+1} \leftarrow \mathrm{Consolidate}(\mathcal{M}_t), \qquad Z \leftarrow \mathrm{Retrieve}(\mathcal{M}_t, q). \tag{2}$$

Ingest places extracted entries in the buffer and may extend the source archive. Consolidate organizes buffered entries into topic documents while leaving the archive unchanged. Retrieve constructs answer context Z from the state it retrieves. Consolidation is scheduled separately from ingestion, so retrieval can also act on $\mathcal{M}'$ while its new entries remain in the buffer.

4 INFINI MEMORY DESIGN

Figure 2 illustrates how Infini Memory stores information from interactions and uses it in later responses. Recent entries remain searchable in `CURRENT`, topic documents organize them for revision, and `Evidence` preserves the original dialogue (Section 4.1). Consolidation assigns entries to topics, revises complete document bodies, and adjusts topic boundaries through Split and Merge (Section 4.2). Interactions are archived as Evidence objects before extraction, and entries can carry identifiers that point to them (Section 4.3). At query time, the readers construct answer context from maintained memory and, in the source-aware path, relevant Evidence (Section 4.4).

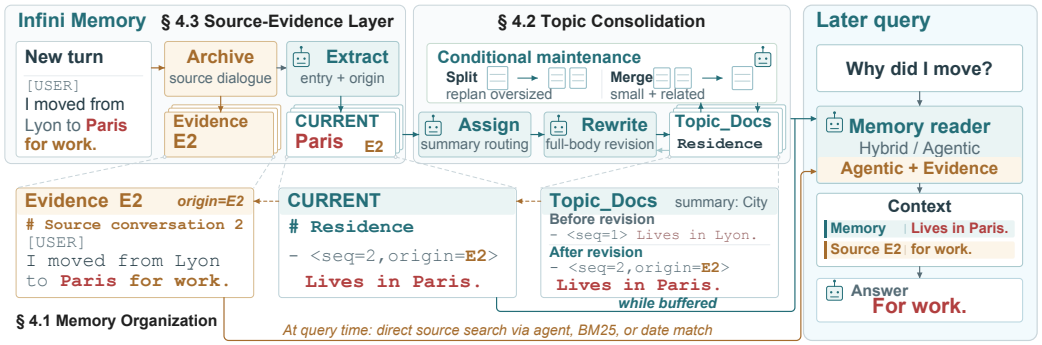

Figure 2: Conceptual overview of memory update and source-aware reading.

4.1 Memory Organization

As defined in Section 3, Infini Memory organizes memory into three spaces with different roles. `CURRENT` realizes the searchable buffer C_t, keeping recent entries available before consolidation. Each document $d_{i,t} \in D_t$ contains an identifier, a compact summary, and a complete Markdown body:

$$d_{i,t} \triangleq \langle i, s_{i,t}, B_{i,t} \rangle. \tag{3}$$

Its summary serves as a lightweight address for topic selection, whereas its body defines the entries jointly considered during revision. Within the body, each first-level heading and its entries form a *heading block* for finer-grained retrieval.

`Evidence`, corresponding to the archive $\mathcal{E}_t$, preserves source dialogue with its message roles and original order. It remains separate from `Topic_Docs`, so document revision does not alter the dialogue from which its entries were derived.

4.2 Topic Consolidation

`Consolidate` organizes buffered entries into topic documents. It consists of three stages: *topic assignment*, *document revision*, and *structural maintenance*.

Topic Assignment. On each write, consolidation checks whether `CURRENT` has remained unchanged for the configured interval. It also checks the buffer size after appending new entries. In the staged pipeline, a separate rewrite groups buffered entries by topic and removes duplicates before assignment. The topic planner then reads the batch together with the identifier-summary catalog:

$$S_t = \{(i, s_{i,t}) \mid d_{i,t} \in D_t,\ |B_{i,t}| \leq L_{\text{doc}}\}, \tag{4}$$

where S_t contains the identifier and summary of each eligible topic document, and L_{doc} is the document-length threshold. The planner uses summaries to determine which existing topic documents should receive the incoming content and which content should instead form new topics. For each existing document i, let $\Delta_{i,t}$ denote the incoming content assigned to $d_{i,t}$. The full content of a selected document is loaded only after this assignment is determined.

Document Revision. Each selected document is revised using its complete stored body together with its assigned content:

$$d'_{i,t} \leftarrow \text{RewriteDoc}_{\text{LLM}}\left(d_{i,t}, \Delta_{i,t}\right). \tag{5}$$

The rewriter updates the body and summary while retaining the document identifier; a newly created topic applies the same operation with an empty previous body. Conflicts are resolved using entry metadata: larger `seq` values take precedence across writes, later content takes precedence within the same write, and user-authored content takes precedence over assistant-authored content at equal `seq`. Thus, each revision considers the complete maintained state of the target topic.

Structural Maintenance. Repeated revision can gradually make some documents remain small and closely related, while others grow broad. Infini Memory therefore applies two maintenance operators, *Merge* and *Split*. *Merge* combines small, related documents selected from their summaries and update times. *Split* handles an oversized document by running topic assignment over its full content. Its entries are then reassigned to existing topic documents or used to form new topics. Both operators act on complete document content and modify the boundaries of future revision units.

A completed consolidation consumes its processed buffer and produces the revised document collection D_{t+1}. For this consolidation step, the source archive remains unchanged: $\mathcal{E}_{t+1} = \mathcal{E}_t$. Appendix C.4 gives the corresponding procedures.

4.3 Source-Evidence Layer

The Source-Evidence Layer preserves the dialogue from which entries are extracted and makes it available to the reader.

Source Archiving. Each archived interaction has an Evidence identifier. When the same interaction identity is encountered again, the existing object is reused with its stored dialogue intact. Archiving precedes extraction, so source dialogue remains available even when that interaction produces no retained entry.

Origin Preservation. An extracted entry associated with Evidence object e stores that identifier in `origin`, instantiating the mapping from Section 3 as $\rho_{t+1}(f) = e$. For inferred entries from multiple archived source groups, source-text matching and a uniquely identifying observation date guide this assignment. The mapping remains partial when neither identifies a group. Because the identifier belongs to the entry, its document placement does not determine which archived object it addresses. The linked dialogue can then be consulted during retrieval, including after the entry has been assigned to another topic.

4.4 Memory Retrieval

At query time, Retrieve selects relevant content from the searchable buffer C_t and topic documents D_t. It can return a complete document or *heading blocks* that focus the answer context on particular entries. As shown in Figure 3, we instantiate this interface with two readers. The *hybrid* reader selects complete documents from their summaries and supplements them with BM25-ranked heading blocks. The *agentic* reader searches and inspects documents iteratively within a fixed round budget, then selects documents or passages.

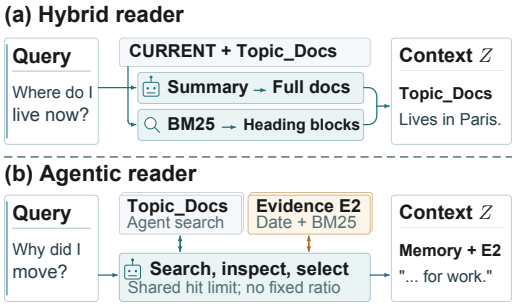

Figure 3: Hybrid and agentic readers construct answer context Z from maintained memory.

With the Source-Evidence Layer, the agentic reader can also search $\mathcal{E}_t$ directly, with date and lexical matches contributing source candidates. Selected memory entries and source passages are assembled into a focused evidence ledger and source-ordered excerpts. Together they form the answer context Z; the excerpts retain neighboring turns, speaker roles, and their order for interpreting the selected entries.

5 Experiments

5.1 Experimental Setup

Benchmarks and Metrics. We evaluate on MemoryAgentBench (Hu et al., 2025), which covers Accurate Retrieval (AR), Test-Time Learning (TTL), Long-Range Understanding (LRU), and Selective Forgetting (SF). We use its task-specific metrics and report the mean of the four capability scores as the overall score. We also evaluate on the original LongMemEval-S (Wu et al., 2025), and conduct

Table 1: MemoryAgentBench performance across four capabilities.

	Accurate Retrieval					Test-Time Learning			Long-Range Underst.			Selective Forgetting			
Method	SH-QA	MH-QA	LME	Event	Avg.	MCC	Rec.	Avg.	Summ.	DetQA	Avg.	FC-SH	FC-MH	Avg.	Overall
Baselines															
RAPTOR	32.0	39.0	38.2	48.2	39.4	56.8	14.0	35.4	25.3	45.6	35.5	16.0	2.0	9.0	29.8
MemoRAG	35.0	36.0	25.0	54.4	37.6	75.0	15.7	45.4	15.3	53.5	34.4	25.0	8.0	16.5	33.5
HippoRAG-v2	78.0	70.0	54.7	72.2	68.7	65.2	12.3	38.8	32.1	54.3	43.2	58.0	5.0	31.5	45.5
Mem0	28.0	36.0	42.0	35.5	35.4	35.2	11.2	23.2	12.4	42.3	27.4	25.0	3.0	14.0	25.0
MemGPT	49.0	30.0	42.7	45.6	41.8	65.6	14.3	40.0	10.9	40.2	25.6	32.0	3.0	17.5	31.2
LightMem	57.0	34.0	69.2	39.2	49.9	58.0	12.7	35.4	27.5	41.0	34.2	29.0	6.0	17.5	34.2
REMem	62.0	42.0	62.3	41.2	51.9	80.0	13.2	46.6	23.4	47.6	35.5	32.0	9.0	20.5	38.6
Infini Memory variants															
Infini Memory-H	77.0	76.0	76.0	83.0	78.0	81.0	15.8	48.4	57.2	**78.4**	67.8	80.0	22.0	51.0	61.3
Infini Memory-A	**83.0**	**79.0**	**79.3**	**83.6**	**81.2**	**84.0**	**18.0**	**51.0**	**59.9**	77.2	**68.6**	**81.0**	**35.0**	**58.0**	**64.7**

Note. Scores use a 0–100 scale; Rec. is Recall@5 and Summ. is Fluency × F1. Shaded columns show unweighted capability means and their overall mean. Bold marks the best score in each column.

a separate ablation on the FactConsolidation task (Zhong et al., 2023) to examine how structural maintenance affects the use of updated facts in single- and multi-hop reasoning.

Baselines. We compare Infini Memory with seven memory systems through MemoryAgentBench's official integrations: RAPTOR (Sarthi et al., 2024), MemoRAG (Qian et al., 2025), HippoRAG-v2 (Gutiérrez et al., 2025), Mem0 (Chhikara et al., 2025), MemGPT (Packer et al., 2023), LightMem (Fang et al., 2026), and REMem (Shu et al., 2026).

Configuration. The MemoryAgentBench comparison uses `gpt-5-mini` with 4096-token input chunks and `gpt-5` for evaluation. On the original LongMemEval-S, we use `deepseek-v4-flash-0731` for memory construction and answering, and `deepseek-v4-pro-0813` for judging. The FC study uses the same model configuration. We denote the hybrid and agentic reader variants as Infini Memory-H and Infini Memory-A, respectively. Appendix A provides the memory thresholds, retrieval budgets, and detailed evaluation settings.

5.2 Overall Performance

Table 1 shows the result across the four capabilities of MemoryAgentBench. Infini Memory-A reaches an overall score of 64.7, exceeding the strongest of the seven evaluated baselines by 19.2 points. Notably, the hybrid reader configuration (Infini Memory-H) reaches 61.3, close to the 64.7 of Infini Memory-A, suggesting that strong overall performance carries across both reader configurations. Within Selective Forgetting, Infini Memory-A scores 81.0 on FC-SH and 35.0 on FC-MH, exceeding the strongest evaluated FC-MH baseline at 9.0 while highlighting the challenge of reasoning over multiple updated facts. Section 5.3 examines both patterns through the structural-maintenance and retrieval ablations.

5.3 Ablation Experiments

We examine three choices along the path from stored facts to answers: memory maintenance, source-aware versus topic-only reading, and retrieval strategy.

5.3.1 Memory Maintenance Ablation

On the original LongMemEval-S, Infini Memory-H scores 82.2% with maintenance and 70.4% without it; Infini Memory-A scores 96.4% and 91.8%, respectively (Table 2). For both readers, the largest separation occurs on multi-session questions, which require facts from different conversations (Appendix B.1). On FactConsolidation, with the reader held fixed, maintenance scores 94.0% versus 88.0% on FC-SH and 45.0% versus 21.0% on FC-MH. The larger multi-hop difference highlights the role of organizing updated facts that must be used together.

Table 2: Memory maintenance and retrieval ablations.

Maint.	Reader	*LongMemEval-S*							*FactConsolidation*		
		KU	MS	SS-A	SS-P	SS-U	TR	Avg. (%)	FC-SH	FC-MH	Avg. (%)
Summary-guided		80.8	72.9	94.6	93.3	71.4	79.7	79.4	80.0	14.0	47.0
w/o maint.	Hybrid	84.6	57.9	96.4	60.0	78.6	61.7	70.4	88.0	21.0	54.5
	Agentic	93.6	83.5	**100.0**	**100.0**	94.3	**92.5**	91.8 (+21.4)	88.0	35.0	61.5 (+7.0)
Full	Hybrid	92.3	79.7	98.2	90.0	77.1	72.9	82.2	**94.0**	45.0	69.5
	Agentic	**97.4**	**99.2**	**100.0**	96.7	**97.1**	91.0	**96.4** (+14.2)	**94.0**	**61.0**	**77.5** (+8.0)

Note. w/o maint. omits topic-document formation, plan-driven updates, Split, and Merge on LongMemEval-S, and structural maintenance on FC. KU/MS: knowledge update/multi-session; SS-A/P/U: single-session assistant/preference/user; TR: temporal reasoning. Avg. pools LongMemEval-S questions or averages FC-SH/MH; parentheses give Agentic–Hybrid gaps (pp).

5.3.2 Source-Aware Reading

On the same frozen memory, source-aware Agentic reading scores 96.4% on the original LongMemEval-S, compared with 84.0% for topic-only reading (Table 3). Preference and assistant-detail questions show the widest gaps, at 36.7 and 25.0 points; knowledge updates and multi-session questions follow at 16.7 and 15.8 points. These questions place particular weight on speaker identity, revised facts, and context distributed across conversations. Single-session user and temporal reasoning questions show smaller gaps of 1.4 and 1.5 points (Appendix B.1).

Table 3: Source-aware and topic-only reading (%).

Type	Topic-only	Source-aware	Δ
KU	80.8	97.4	+16.7
MS	83.5	99.2	+15.8
SS-A	75.0	100.0	+25.0
SS-P	60.0	96.7	+36.7
SS-U	95.7	97.1	+1.4
TR	89.5	91.0	+1.5
All	**84.0**	**96.4**	**+12.4**

5.3.3 Retrieval Strategy Ablation

As shown in Table 2, the summary-guided selection scores 79.4% on the original LongMemEval-S, while Infini Memory-H reaches 82.2%. On maintained FactConsolidation memory, Infini Memory-H and Infini Memory-A both score 94.0% on FC-SH, while Infini Memory-A reaches 61.0% on FC-MH versus 45.0% for Infini Memory-H. The 16-point FC-MH gap and the 19.5-point multi-session gap occur in tasks that assemble evidence across updates or conversations. Together, they motivate a closer look at the granularity of retrievable records (Section 5.4); Appendix A.4 identifies where the LongMemEval-S reader rows differ by question type.

5.4 Effects of Memory Organization

Memory organization sets the units available to the reader and the context each unit carries. We examine how supporting information is distributed across records and how changing the split threshold affects answer accuracy.

5.4.1 Memory Fragmentation

We define record fragmentation as the number of independently retrievable records associated with a question's supporting sessions. In the t-SNE projections of Figure 4, Infini Memory retrieves the dates needed for Case 1's 18-day answer, while Mem0 misses the dates and MemGPT retains an unanchored "today"; in Case 2, MemGPT chooses an outdated stock count despite retrieving the current one. Across 20 matched original LongMemEval-S memories, the median is 4 source-associated topic documents for Infini Memory, versus 20–242 system-specific records for the four baselines (Appendix A.6). Together, these traces show why retrieved memory must preserve temporal anchors and distinguish current facts from outdated ones.

Table 5: Model study on LongMemEval-S (%).

	Reader model	
Memory build	DS-V4-Flash	Qwen3-4B
DS-V4-Flash	**96.4**	88.4
Qwen3-4B	89.4	90.0

Table 6: Build time and query footprint.

	Full	No excerpts	No evidence
Reads	47.1	47.1	0
Prompt tokens	15,918	12,373	16,683
Build (500 users)		16.8 s/user	

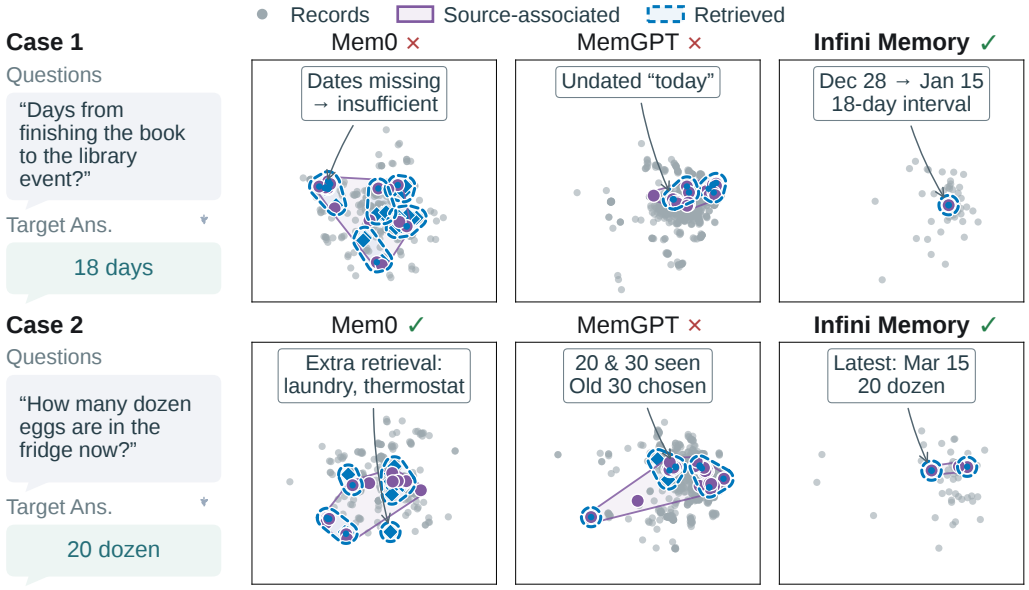

Figure 4: Two LongMemEval-S retrieval cases projected with t-SNE.

5.4.2 Split-Threshold Sensitivity

Table 4 evaluates the document split threshold using the hybrid reader. Reducing the threshold to 1000 or 3000 tokens lowers accuracy by at most 2.0 points, whereas increasing it to 7000 or 9000 tokens causes larger drops of 5.3 and 6.3 points, respectively. This asymmetry suggests that oversized mixed-topic documents are more harmful than moderate fragmentation, supporting the use of an upper bound on document size to preserve topical coherence. Appendix B.3 provides the corresponding document-length distributions.

Table 4: Split-threshold sensitivity.

Threshold	#Docs	Acc.	Δ
≥ 1000	762	74.0	−2.0
≥ 3000	319	74.3	−1.7
≥ 5000	322	**76.0**	—
≥ 7000	280	70.7	−5.3
≥ 9000	255	69.7	−6.3

5.5 Model and Efficiency Study

Model Choice. In Table 5, DeepSeek-V4-Flash-0731 (DS-V4-Flash) and Qwen3-4B-Instruct-2507 (Qwen3-4B) are abbreviated for space. On the original LongMemEval-S, the DeepSeek-V4-Flash-0731 build-and-reader configuration of Infini Memory-A reaches 96.4%. The other three combinations score 88.4–90.0%; on memory built by Qwen3-4B-Instruct-2507, the DeepSeek-V4-Flash-0731 and Qwen3-4B-Instruct-2507 readers reach 89.4% and 90.0%, respectively. On the shared Qwen3-4B-Instruct-2507-built memory, changing the reader shifts accuracy by only 0.6 points; among the four reported configurations, the DeepSeek-V4-Flash-0731 build-and-reader pairing achieves the highest score.

Efficiency. Table 6 profiles memory construction and answer-context assembly. Building memory for the original LongMemEval-S takes 16.8 seconds per user on average, comprising 4.4 seconds for extraction and 12.3 seconds for maintenance. On 50 matched questions from the same benchmark, the full query configuration reads 47.1 Evidence objects and assembles a 15,918-token answer prompt. Omitting primary excerpts shortens this constructed prompt by 22.3% to 12,373 tokens, whereas removing Source-Evidence access yields 16,683 tokens. Evidence reads remain at 47.1 without primary excerpts because the reader scans each user's archive before shaping the answer context.

6 CONCLUSION

We presented Infini Memory, a long-term memory system that organizes extracted entries into topic documents and archives source dialogue in a separate Source-Evidence Layer. Its source-aware reader searches maintained memory and archived dialogue to assemble answer context. Infini Memory scores 64.7 overall on MemoryAgentBench and achieves 96.4% accuracy on the original LongMemEval-S. As information changes across conversations, the ablations suggest a broader principle: revise related facts as a coherent unit while keeping their source context retrievable.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language editing and polishing, literature retrieval and discovery, code implementation, and experimental design; all AI-assisted text, retrieved references and code were reviewed and verified by the authors. We did not use generative AI tools for synthetic data generation or mathematical claims. We take responsibility for the final content of this work, including text, claims, code, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. We evaluate on existing benchmark datasets and collect no new user conversations. Because the Source-Evidence Layer retains original dialogue that may contain sensitive personal information, deployments should obtain appropriate consent and apply access controls, retention limits, and deletion policies to both archived dialogue and derived memory.

REPRODUCIBILITY STATEMENT

We have made efforts to ensure the reproducibility of our results. The code and data used in this study will be made available upon publication. We encourage other researchers to replicate our experiments and validate our findings.

REFERENCES

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

Wonjun Choi, Yerim Kim, Yukyung Lee, and Susik Yoon. PGMem: Tightly coupled persona-memory graph for lifelong personalized agents, 2026. URL https://arxiv.org/abs/2608.01708. arXiv preprint arXiv:2608.01708.

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 *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a05b72653ec5b473732129829ae04195-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, 2025. URL https://proceedings.mlr.press/v267/gutierrez25a.html.

Chuanrui Hu, Xingze Gao, Zuyi Zhou, Dannong Xu, Yi Bai, Xintong Li, Hui Zhang, Tong Li, Chong Zhang, Lidong Bing, and Yafeng Deng. EverMemOS: A self-organizing memory operating system for structured long-horizon reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 45836–45853, 2026. doi: 10.18653/v1/2026.acl-long.2125.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in LLM agents via incremental multi-turn interactions, 2025.

Zhaopei Huang, Qifeng Dai, Guozheng Wu, Xiaopeng Wu, Xubin Li, Tiezheng Ge, Wenxuan Wang, and Qin Jin. Mem-PAL: Towards memory-based personalized dialogue assistants for long-term user-agent interaction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(37): 31229–31237, 2026. doi: 10.1609/aaai.v40i37.40385. URL https://ojs.aaai.org/index.php/AAAI/article/view/40385.

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

Christopher Latimer, Nicolò Boschi, Andrew Neeser, Chris Bartholomew, Gaurav Srivastava, Xuan Wang, and Naren Ramakrishnan. Hindsight: Structured agent memory that retains, recalls, and reflects. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 275–285, 2026. doi: 10.18653/v1/2026.acl-demo.27.

Dongfang Li, Zixuan Liu, Junmai Wang, Jiahe Huang, Fuhao Li, Bonian Jia, Baotian Hu, and Min Zhang. LycheeMemory V2: Efficient long-term memory for LLM agents via semantic segment-level consolidation, 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 *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, 2024. doi: 10.18653/v1/2024.acl-long.747.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems, 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 *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/e56f394bbd4f0ec81393d767caa5a31b-Abstract-Conference.html.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. MemoRAG: Boosting long context processing with global memory-enhanced retrieval augmentation. In *Proceedings of the ACM on Web Conference 2025*, pp. 2366–2377, 2025. doi: 10.1145/3696410.3714805.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory, 2025.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=GN921JHCRw.

Yiheng Shu, Padmaja Jonnalagedda, Xiang Gao, Bernal Jimenez Gutierrez, Weijian Qi, Kamalika Das, Huan Sun, and Yu Su. Remem: Reasoning with episodic memory in language agent. In *International Conference on Learning Representations*, volume 2026, pp. 55958–55987, 2026.

Theodore R. Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. Cognitive architectures for language agents. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=1i6ZCvflQJ.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=pZiyCaVuti.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Yi Yu, Liuyi Yao, Yuexiang Xie, Qingquan Tan, Jiaqi Feng, Yaliang Li, and Libing Wu. Agentic memory: Learning unified long-term and short-term memory management for large language model agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21457–21483, 2026. doi: 10.18653/v1/2026.acl-long.981.

Ruifeng Yuan, Shichao Sun, Yongqi Li, Zili Wang, Ziqiang Cao, and Wenjie Li. Personalized large language model assistant with evolving conditional memory. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 3764–3777, 2025. URL https://aclanthology.org/2025.coling-main.254/.

Siyan Zhao, Mingyi Hong, Yang Liu, Devamanyu Hazarika, and Kaixiang Lin. Do llms recognize your preferences? evaluating personalized preference following in llms. In *International Conference on Learning Representations*, volume 2025, pp. 15888–15931, 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. doi: 10.1609/aaai.v38i17.29946.

Zexuan Zhong, Zhengxuan Wu, Christopher Manning, Christopher Potts, and Danqi Chen. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 15686–15702, 2023. doi: 10.18653/v1/2023.emnlp-main.971.

Appendix Contents

A Experimental Details . . . 12
A.1 Benchmarks and Metrics . . . 12
A.2 Common Configurations . . . 12
A.3 Ablation Configurations . . . 13
A.4 Retrieval Strategy Diagnostics . . . 14
A.5 Model Study Protocol . . . 14
A.6 Fragmentation Measurement and Retrieval Cases . . . 14
B Additional Experimental Diagnostics . . . 16
B.1 Per-Type Correct Counts . . . 16
B.2 Qualitative Cases from the Paired Ablation . . . 16
B.3 Split-Threshold Diagnostics . . . 17
C Prompt Templates and Algorithms . . . 18
C.1 Normalized English Prompt Templates . . . 18
C.2 Storage Layout . . . 21
C.3 Source-Evidence Format and Identifiers . . . 21
C.4 Algorithmic Pseudocode . . . 22
D Discussion . . . 24
D.1 Strategy Selection and Design Rationale . . . 24
D.2 Deployment and Extensibility . . . 24
D.3 Limitations . . . 24

A Experimental Details

A.1 Benchmarks and Metrics

MemoryAgentBench (Hu et al., 2025) evaluates four capabilities through the ten tasks in Table 7. Each task is scored on a 0–100 scale, and the overall score averages the four capability scores. Each capability score is the unweighted mean of its task scores, and the overall score is the unweighted mean of the four capability scores. Movie recommendation uses Recall@5, novel summarization uses Fluency $\times$ F1, and the remaining tasks use accuracy.

The reconstructed LongMemEval subset contains 45 knowledge-update, 75 multi-session, 30 single-session assistant, 30 single-session preference, 45 single-session user, and 75 temporal-reasoning questions. The original LongMemEval-S contains 500 questions and is used for the structural, Source-Evidence, retrieval, and model studies below. Its per-type counts appear in Table 10. The five classification tasks contribute 100 questions each to the 500-question MCC score.

A.2 Common Configurations

Models and input. The MemoryAgentBench comparison uses `gpt-5-mini` as the backbone and `gpt-5` for LLM-based evaluation. All benchmark integrations process 4096-token input chunks. The reference LongMemEval-S and FC configurations use `deepseek-v4-flash-0731` for memory construction and answering, and `deepseek-v4-pro-0813` for judging. Table 8 lists the shared topic-document settings used with MemoryAgentBench.

Table 7: MemoryAgentBench task sizes and metrics used ($N = 2{,}071$).

Abbrev.	Task	N	Metric
Accurate Retrieval			
SH-QA	Single-hop document QA	100	Accuracy
MH-QA	Multi-hop document QA	100	Accuracy
LME	Reconstructed LongMemEval ($S^\star$)	300	LLM-judged accuracy
Event	EventQA	500	Accuracy
Test-Time Learning			
MCC	Multi-class classification	500	Accuracy
Rec.	Movie recommendation	200	Recall@5
Long-Range Understanding			
Summ.	Novel summarization	100	Fluency $\times$ F1
DetQA	Detective reasoning QA	71	Accuracy
Selective Forgetting			
FC-SH	FactConsolidation, single-hop	100	Accuracy
FC-MH	FactConsolidation, multi-hop	100	Accuracy

Table 8: Topic-document memory settings for MemoryAgentBench.

Setting	Value
CURRENT size threshold	Approximately 5000 tokens
Document length threshold	5000 tokens
Merge candidate size	Below 1000 tokens
Agentic retrieval budget	Up to 7 rounds
BM25 parameters	$k_1 = 1.5, b = 0.75$

Writing and retrieval configuration. In the staged path, buffered entries are grouped before assignment to topic documents; the online path checks an active buffer when a later write arrives. Structural maintenance considers document summaries and update times when selecting small documents for merging and reassigns oversized documents. The hybrid reader (Infini Memory-H) selects documents from an `id + summary` catalog and supplements them with BM25-ranked heading blocks. The agentic reader (Infini Memory-A) combines search, matching, catalog browsing, and line reading within its round budget; BM25 supplies candidates when its selected context is empty or short. Appendix C.4 details the retrieval interface and result assembly.

A.3 Ablation Configurations

A.3.1 MemoryAgentBench Protocol

The split-threshold sweep uses MemoryAgentBench's reconstructed 300-question LongMemEval subset and rebuilds memory for each threshold while keeping the hybrid reader fixed. Its thresholds are 1000, 3000, 5000, 7000, and 9000 tokens (Table 4). The structural and Source-Evidence results in Section 5.3 instead use the original LongMemEval-S.

A.3.2 LongMemEval-S Protocol

Source-Evidence. The source-aware Agentic and topic-only readers answer the same 500 questions over one frozen memory. The former can inspect archived Evidence; the latter reads topic documents without Evidence objects, origin hydration, or primary source excerpts.

Maintenance ablation. The `w/o maint.` configurations use a rebuilt memory for the same 500 questions. They retain extraction, `CURRENT` rewriting, and Evidence archiving, while disabling topic-document formation, plan-driven updates, Split, and Merge.

Table 9: Source-associated records and record lengths across 20 matched histories. Values are median [Q1, Q3] across histories; length uses the within-history median estimated token count.

System	Record unit	Associated records	Length (tokens)
A-MEM	Note	29 [27, 29]	111 [103, 125]
Mem0	Fact	20 [14.75, 28.25]	14 [14, 15]
Mem0-g	Graph relation	242 [214.75, 352.75]	10 [10, 11]
MemGPT	Recall or archival	48 [44.75, 51]	369 [327, 390]
Infini Memory	Topic document	4 [2, 4]	1,365 [1,227, 1,478]

A.3.3 FACTCONSOLIDATION PROTOCOL

For FC-SH and FC-MH, `w/o maint.` disables structural maintenance while holding the FC reader setting fixed. FC-SH asks for a single latest fact, whereas FC-MH requires a relation chain whose constituent facts may each have been updated.

A.4 RETRIEVAL STRATEGY DIAGNOSTICS

Summary-guided selection supplies the full bodies of documents chosen by their summaries; Hybrid adds BM25-ranked heading blocks, while Agentic can inspect documents and archived Evidence. Table 2 reports their results by question type.

Among the maintained rows, multi-session and temporal-reasoning questions account for 50 of the 71 net additional answers in the Agentic configuration (26 and 24, respectively). This locates the difference in questions that assemble facts across conversations or order them in time.

A.5 MODEL STUDY PROTOCOL

Table 5 presents four memory-build and reader combinations on the original 500-question LongMemEval-S. The DeepSeek-V4-Flash-0731 build with a DeepSeek-V4-Flash-0731 reader scores 482/500 (96.4%); the other combinations score 442/500, 447/500, and 450/500. The two Qwen3-4B-Instruct-2507-built combinations use the same memory and differ by three correct answers between readers.

A.6 FRAGMENTATION MEASUREMENT AND RETRIEVAL CASES

The fragmentation study uses 20 original LongMemEval-S question-history pairs with completed exports for A-MEM, Mem0, Mem0-g, MemGPT, and Infini Memory: eight multi-session, seven temporal-reasoning, and five knowledge-update questions. We selected histories within these question types by systematic midpoint sampling and retained one anchor case for inspection. Each system received the same complete history for a pair.

For each question, we take the union of its annotated supporting sessions and count each surviving exported record once if its recorded source turns intersect one of those sessions. We use A-MEM notes, Mem0 facts, Mem0-g Neo4j relations, MemGPT recall and archival records, and Infini Memory topic documents. This count measures source association, not whether a record contains every fact needed to answer. All 100 system–history pairs have at least one provenance-associated record for every supporting session. This coverage check does not imply semantic evidence preservation or successful retrieval.

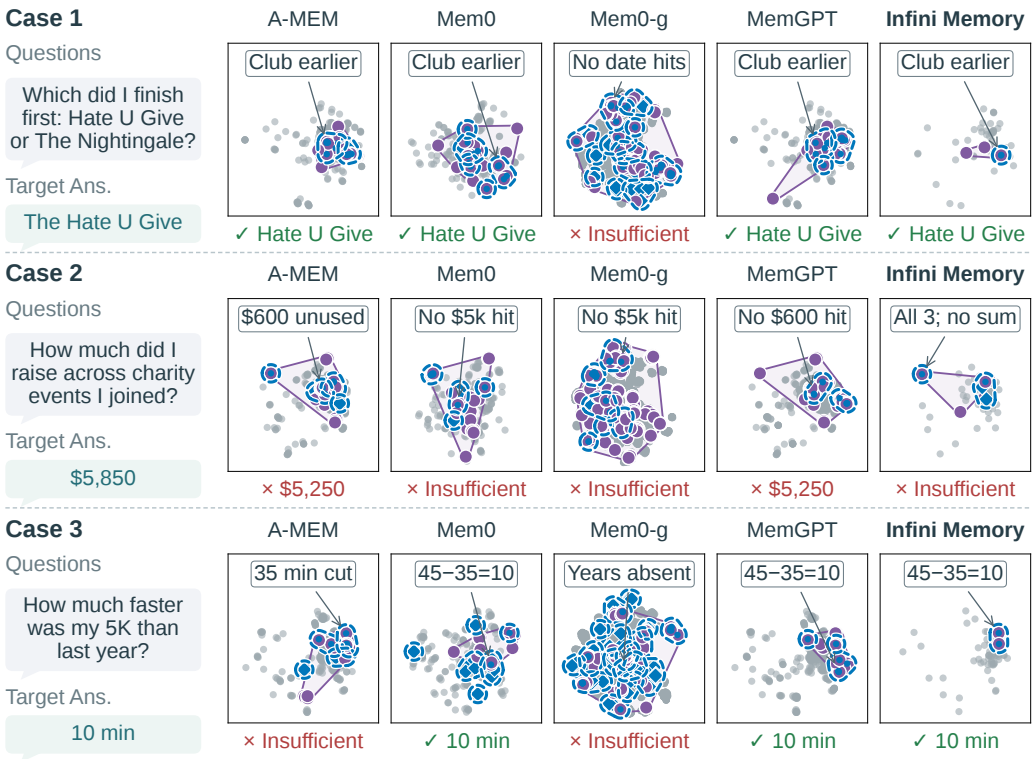

Figure 5: Three retrieval cases in t-SNE space.

	Multi-session								Temporal reasoning							Knowledge update				
Sample ID	*78*	*93*	*108*	*123*	*175*	*190*	*205*	*220*	*246*	*267*	*295*	*315*	*324*	*334*	*352*	*374*	*388*	*403*	*417*	*431*
A-MEM	41	65	39	39	27	29	29	29	29	27	27	29	29	29	25	29	29	29	27	27
Mem0	34	48	28	29	24	12	13	12	21	25	8	15	29	22	14	29	19	16	19	18
Mem0-g	350	729	410	248	231	352	211	546	334	222	216	211	236	185	201	338	355	356	233	189
MemGPT	74	120	68	71	44	48	48	57	48	45	44	48	49	48	40	48	48	49	43	44
Infini Memory	3	5	4	4	4	4	2	11	2	2	2	4	4	2	7	2	4	5	2	2

Figure 6: Source-associated record counts across 20 matched histories.

Figure 5 adds three cases, including one where Infini Memory also fails. Figure 6 shows the associated-record count for every system and history. The two-dimensional projections are visual aids only: neither distance nor contour area enters the count. Purple contours indicate source association, not verified answer evidence; blue contours indicate retrieved records. Retrieval outside purple may be irrelevant or useful, so it is not automatically labeled noise. Orange outlines mark seven MemGPT histories with logged ingestion errors; they remain in the reported cohort.

The case probes use the same BM25 retrieval and answer model across systems; they are not the systems' native-agent benchmark results. Case selection and the differing record granularity limit generalization from the probe to end-to-end accuracy. The larger median record length of Infini Memory also means that a smaller record count need not imply a smaller retrieval context.

B Additional Experimental Diagnostics

B.1 Per-Type Correct Counts

Table 10 gives the correct counts for the four reported configurations on the original LongMemEval-S. All four answer the same 500 question IDs; the column totals reproduce the accuracies quoted in Section 5.3. Protocol differences between configurations are specified in Appendix A.3.2.

Table 10: Per-type correct counts on the original LongMemEval-S, with accuracy in parentheses. The final column is the full configuration.

Type	N	w/o both	w/o maint.	Topic-only	Full
KU	78	64 (82.1)	73 (93.6)	63 (80.8)	76 (97.4)
MS	133	79 (59.4)	111 (83.5)	111 (83.5)	132 (99.2)
SS-A	56	52 (92.9)	56 (100.0)	42 (75.0)	56 (100.0)
SS-P	30	20 (66.7)	30 (100.0)	18 (60.0)	29 (96.7)
SS-U	70	48 (68.6)	66 (94.3)	67 (95.7)	68 (97.1)
TR	133	104 (78.2)	123 (92.5)	119 (89.5)	121 (91.0)
All	500	367 (**73.4**)	459 (**91.8**)	420 (**84.0**)	482 (**96.4**)
Δ vs. full (pp)		23.0	4.6	12.4	—

B.2 Qualitative Cases from the Paired Ablation

To make the Source-Evidence comparison in Table 10 concrete, we examine three questions, two answered correctly only by the full configuration and one only by the ablated configuration. They share the same question set and memory. In each case the selected derived documents match, while archived Evidence is available only to the full configuration. The examples illustrate how the evaluated readers assemble different answer contexts. Quoted answers are verbatim model output and quoted sources are verbatim archived conversation.

Case 1. Single-session assistant: both arms hold the answer, only one finds it

Question	*"I'm planning to visit Bandung again and I was wondering if you could remind me of the name of that restaurant in Cihampelas Walk that serves a great Nasi Goreng?"* (asked 2023/05/30, over a 44-session history)
Gold answer	Miss Bee Providore
Infini Memory	Retrieves 8 keys: 6 documents and 2 Evidence objects. The top-ranked key is the Evidence object for the gold session. Answers *"Miss Bee Providore"*. **correct**
w/o Source Evidence	Retrieves the same 6 documents and no Evidence. Answers *"Stored memory is insufficient; I cannot recall the restaurant name."* **incorrect**
Archived source	2023/05/30, assistant: *"1. Miss Bee Providore: This restaurant serves a mix of western and Indonesian cuisine and has a cozy and stylish interior."* A later turn in the same session adds *"Miss Bee's Nasi Goreng: Their take on the classic Indonesian fried rice dish is a must-try!"*
Reading	The ablated arm is not missing the information: one of its six documents is 90,843 bytes and holds the answer string at byte 54,139, roughly 13.5K tokens into a single retrieved item. The Evidence object carrying the same session is 12 KB and was ranked first for the full arm.

Case 2. Knowledge update: two dated sessions fix a before-and-after order

Question	*"Before I purchased the gravel bike, do I have other bikes in addition to my mountain bike and my commuter bike?"* (asked 2023/06/21, over a 46-session history)
Gold answer	Yes, a road bike as well

Infini Memory	Retrieves 20 keys: 6 documents and 14 Evidence objects, among them both gold sessions (2023/05/25 and 2023/05/28). Answers *"Yes. I also had a road bike."* **correct**
w/o Source Evidence	Retrieves the same 6 documents and no Evidence. Answers *"Insufficient information: stored memory has no gravel bike purchase or date, so I cannot determine whether other bikes existed before that purchase."* **incorrect**
Reading	Answering requires two facts that were stated three days apart and an ordering between them and a third event. Each Evidence object is dated and bounded at its own session, so the two can be placed in order; the ablated arm declines on the date rather than on the bicycle.

Case 3. Temporal reasoning: verbatim source reintroduces a relative date

Question	*"How many weeks ago did I attend the 'Summer Nights' festival at Universal Studios Hollywood?"* (asked 2023/08/05, over a 53-session history)
Gold answer	3 weeks ago
Infini Memory	Retrieves 20 keys: 7 documents and 13 Evidence objects, including the gold object for the session of 2023/07/15. Answers *"6 weeks ago"*. **incorrect**
w/o Source Evidence	Retrieves the same 7 documents and no Evidence. Answers *"3 weeks ago"*. **correct**
Archived source	2023/07/15, user: *"I just got back from an amazing day at Universal Studios Hollywood, we had so much fun at the 'Summer Nights' festival. . . "*; assistant, same session: *"I didn't know you went to Universal Studios Hollywood three weeks ago."*
Reading	The gold session is itself three weeks before the question date, and its own text says "three weeks ago", so composing the two offsets gives six. Verbatim source carries relative expressions anchored at the session rather than at the question, and the extra Evidence does not correct that: none of the other twelve objects mentions the festival.

The two positive cases share a shape. The ablated arm holds documents that do contain the relevant turns and still refuses, so what the layer supplies is addressing and context budget rather than information that extraction had discarded. That matches the refusal counts, where removing the layer raises refusals on questions not labeled for abstention from five to 50. Case 3 shows the cost side of the same property: because archived Evidence is verbatim conversation, relative time expressions inside it stay anchored at their session, and five of the eight questions that the ablation improves are temporal reasoning. The cases also mark a scope limit of the qualitative evidence. In this store, the derived documents hold clustered raw turns rather than rewritten prose, so these three questions speak to the Source-Evidence Layer and not to what topic consolidation contributes.

B.3 Split-Threshold Diagnostics

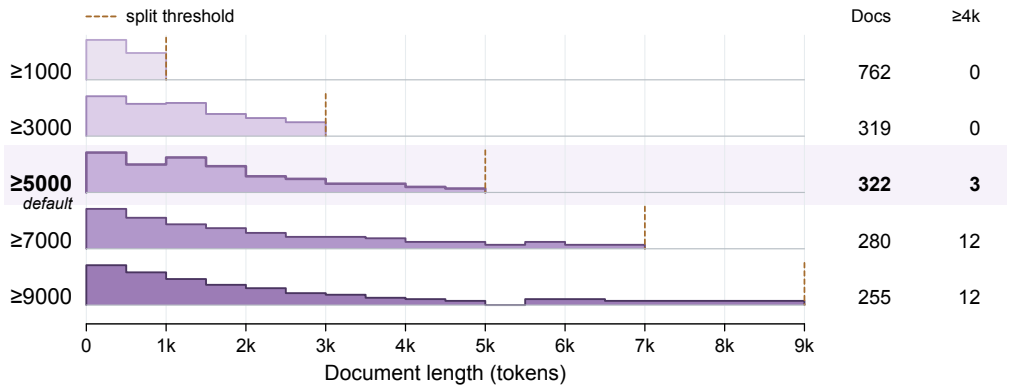

Figure 7: Document lengths under five split thresholds (500-token bins). Ridges are independently normalized after square-root scaling, so heights compare shapes within each row.

Table 4 shows an asymmetric accuracy curve. Lowering the threshold to $\geq$3000 or $\geq$1000 costs only 1.7 and 2.0 points, even though the document collection expands to 762 documents at the most aggressive setting. The resulting documents stay topically clustered at this granularity, so the hybrid reader can recover from moderate document fragmentation through summary and BM25 matching. Raising the threshold to $\geq$7000 or $\geq$9000 costs 5.3 and 6.3 points while the document count moves only from 322 to 280 and 255. The drop therefore reflects the cost of oversized documents mixing unrelated subtopics rather than document count alone.

The default $\geq$5000 threshold sits just above the 4096-token chunking budget (Section 5.1), so splitting primarily acts on documents that absorbed more than one chunk. Figure 7 shows that $\geq$1000 concentrates documents below 1000 tokens, $\geq$9000 produces a long tail beyond 7000 tokens, and the default keeps the bulk below 4000 tokens with only a thin tail near the threshold: three documents reach 4000 tokens under the default, against twelve under $\geq$7000 and twelve under $\geq$9000. Each sweep also stops strictly below its own threshold, so the rule acts as a ceiling on document size rather than a preference.

C Prompt Templates and Algorithms

The templates below condense the operative instructions and output schemas in the released source. Prompts A–E describe the topic-document pipeline; Prompt F describes source-context answering under the Source-Evidence Layer. Preprocessing and result assembly are described separately.

C.1 Normalized English Prompt Templates

Writing conventions. The prompts request faithful recording, preservation of entry metadata, and output in the specified Markdown or JSON format. Rewriting groups and revises textual entries according to the rules below. Metadata preservation and semantic conflict resolution are instructions to the model; the returned document or plan is then parsed.

Prompt A: Memory Extraction

```
Extract facts, preferences, personal information, plans, and other
useful memories from the conversation.  Include user information
stated, confirmed, or summarized by the assistant, marking
assistant-derived entries with source=AI. Preserve supplied labels
and other metadata.

Stay faithful to the source.  Do not correct, annotate, or supplement
its claims.  Keep the literal sequence placeholder @@SEQ@@ unchanged.
Use the source timestamp at its available precision; if absent, write
time=empty.  Preserve the input language.

Return Markdown only:
- begin with YAML frontmatter containing summary:  <topic keywords;
factual summary>, within {summary_length} tokens
- group entries under first-level headings
- use the entry format:
- <seq=@@SEQ@@,time=TIMESTAMP[,label=...]
   [,source=AI]> fact
- if no facts are extractable, return a document with summary and
heading No memorable facts and a matching bullet.
```

Evidence extraction extension. With the Source-Evidence Layer enabled, the supplied source dialogue is archived before extraction, preserving message roles and order. The extraction prompt uses the supplied numeric sequence, distinguishes `observed` from event `time`, retains concrete assistant-provided content with `source=AI`, and forbids inventing `origin`. After extraction, sequence and available observation dates are normalized, source identifiers are inserted, and entry identifiers are assigned. This layer uses H1-grouped bullets and `NO_MEMORY` for empty output.

Prompt B: CURRENT Rewrite

```
You will receive the append-only CURRENT document.  Reorganize its
facts into a topic-structured Markdown document.

- group semantically related facts under first-level headings
- preserve each retained fact's meaning, sequence number, timestamp,
source, label, and other metadata
- for semantically identical duplicates, retain one entry, preferring
the larger seq
- for contradictory facts with different sequence numbers, retain the
larger seq
- add no facts, commentary, corrections, or explanations
- preserve the input language

Return only the final Markdown document.
```

Prompt C: Update Planning

```
You will receive new Markdown content and a document catalog of (id,
summary) pairs.  Assign content to existing documents or create new
topic documents.

- aggregate update fragments by destination document id
- use only ids listed in the input catalog; otherwise create a new
document
- group new content by topic and avoid both excessive fragmentation
and overly broad documents
- keep each new document within {markdown_length} tokens
- preserve sequence numbers, timestamps, source annotations, labels,
and other metadata
- assign each fact to one update or new document without overlapping
content
- if the catalog is empty, organize the content into new topic
documents

Return JSON only:
{
"updates":  [{"id":  string, "new_content":  string}],
"new_docs":  [{"title":  string, "content":  string}]
}
```

Prompt D: Document Rewrite

```
You will receive an old Markdown document and new content assigned to
it.  Produce the updated complete document using facts from these two
inputs.

- organize related facts under first-level headings
- preserve all entry metadata and the document language
- within one add operation, later contradictory content takes
precedence
- across add operations, retain the contradictory fact with the
larger seq
- when competing facts share a sequence number, user input takes
precedence over source=AI content
- add no facts, annotations, or explanations

Return Markdown only, with a refreshed YAML summary of at most
{summary_length} tokens and metadata-prefixed fact bullets.
```

Prompt E: Agentic Retrieval

```
Given a query and document catalog, locate relevant information
through:
- grep: corpus-wide regular-expression matching
- grep_doc: matching within one document
- search: BM25 search over H1 partitions
- list_docs: paginated catalog browsing
- read_lines: reading a bounded document line range

Start with search and grep calls. Use their results to guide
document-local matching and further reading. For aggregation, search
for additional relevant mentions across documents. For temporal
or update questions, inspect sequence and timestamp metadata. Stop
when sufficient information is found or search options are exhausted,
within {max_iterations} rounds.

To request tools, return JSON:
{"tool_calls": [{"tool": string, "args": object}],
"reasoning": string}

To finish, return JSON:
{"done": true,
"relevant_snippets": [{"doc_id": string, "start_line": int,
"end_line": int}],
"relevant_docs": [string, ...],
"reasoning": string}

Prefer line ranges with surrounding context. Select a whole document
when most of it is relevant. Return no more than {search_limit}
selections.
```

Retrieval result assembly. Retrieval expands selected line ranges to overlapping heading blocks and materializes whole-document selections. It applies BM25 fallback or supplementation when the selected content is empty or short. With the Source-Evidence Layer enabled, retrieval additionally constructs a focused ledger and bounded, source-ordered excerpts from selected source objects. The ledger omits a derived entry when its `origin` identifies a selected Evidence object, and filters repeated normalized text. Prompt F then requests task-specific interpretation of this context.

Prompt F: Evidence-Grounded Answer

```
Answer QUERY from FOCUSED_EVIDENCE_LEDGER and PRIMARY_SOURCE_EXCERPTS.
Ledger entries retain source identifiers, roles, headings, metadata,
and text. Source excerpts supply adjacent turns for resolving
pronouns, omitted attributes, and list ownership.

Identify the requested scope and enumerate distinct matching
records before calculation or comparison. Count repeated reports
of the same event once; keep different subjects, dates, states, and
events separate. Use user facts by default and assistant facts for
questions about assistant-provided content. Keep event time separate
from observation time. Resolve omitted entities from unambiguous
turns within the same source conversation.

If a required input is missing or multiple answers remain unresolved,
return an explicit insufficient-information answer. Use no outside
knowledge. FINAL_TASK_GUIDANCE specifies the operation to perform
and cannot override the evidence.

Return one compact JSON object:
{"evidence": ["distinct event or value", ...],
"calculation": "calculation or relation",
"answer": "concise answer in the query language"}

QUERY: {query}
FOCUSED_EVIDENCE_LEDGER: {evidence_ledger}
PRIMARY_SOURCE_EXCERPTS: {primary_source_context}
FINAL_TASK_GUIDANCE: {task_guidance}
```

Merge-related prompts. The group-selection prompt requests non-overlapping groups of two to four topically similar documents and returns JSON containing `groups` with `doc_ids` and a grouping reason. The merge prompt receives their bodies ordered by update time and requests a single Markdown document with a refreshed summary. It preserves entry metadata, favors larger sequence numbers for duplicate or contradictory entries, and prefers the user version of otherwise identical user/assistant entries.

C.2 Storage Layout

The three stores of Equation 1 share one file format: a YAML header followed by a Markdown body. The header carries the identifier, a `kind` field naming the store, the summary used for selection, and creation and update times. The body carries the text that consolidation revises and retrieval reads. `CURRENT` is a single file with `kind: current`, each topic document is one file with `kind: memory`, and each archived conversation is one file with `kind: evidence`. A common format lets one selection and reading interface serve all three, so a reader can mix maintained documents with archived dialogue in a single pass.

A topic document body groups entries under first-level headings, where a heading and the entries beneath it form the heading block of Section 4.1. Each entry is one bullet that opens with an inline metadata tag, so entry metadata is carried alongside the entry text rather than in a separate table:

Topic_Docs/<document id>.md

```
---
id: <document id>
kind: memory
summary: <one-line topic summary used for selection>
title: <topic title>
store_id: <store>
user_id: <user>
created_at: <timestamp>
updated_at: <timestamp>
---
# <heading>

- <seq=...,observed=...,origin=EVIDENCE_...,fid=f_...> entry text
- <seq=...,time=...> entry text
```

`CURRENT` uses the same body convention for entries that are not yet assigned to a topic. The `origin` and `fid` fields appear only when the Source-Evidence Layer is enabled; Table 11 gives the meaning of each tag field.

C.3 Source-Evidence Format and Identifiers

An Evidence object stores dialogue before entry extraction. When all messages carry a session identifier and the input spans multiple sessions, each session group is archived separately; otherwise, the supplied messages form one group. Its Markdown body preserves message roles and content in input order, with surrounding whitespace trimmed. An abbreviated object is shown below:

Evidence/EVIDENCE_<24-hex digest>.md

```
---
id: EVIDENCE_<24-hex digest>
kind: evidence
summary: <short preview>
title: Source conversation
store_id: <store>
user_id: <user>
created_at: <timestamp>
updated_at: <timestamp>
---
# Source conversation <observed date or sequence>
```

```
[USER]
<user message>

[ASSISTANT]
<assistant message>
```

To construct the object ID, the group's message records are serialized as JSON with sorted keys and unescaped Unicode, omitting absent optional fields. The instance ID, extraction sequence, and serialized records are concatenated with NUL separators and hashed with SHA-256, and the first 24 hexadecimal characters receive the prefix `EVIDENCE_`. Repeating these inputs reuses the existing object without rewriting its payload.

For inferred entries, `origin` is supplied after extraction, using available source-observation dates to select a source object and otherwise falling back to the request's default object. This field identifies the stored dialogue object from which reading selects passages. Source creation precedes the extraction result, so an object can remain stored even when extraction returns `NO_MEMORY`.

For an entry missing `fid`, the identifier is formed by hashing its origin or fallback source ID, sequence, one-based position among parsed entry lines, and case-folded text with whitespace collapsed, again using NUL separators. The first 24 hexadecimal characters receive the prefix `f_`. An existing `fid` is retained. This identifier follows the stored entry. Independently extracted paraphrases can receive different identifiers. Table 11 summarizes the metadata used to distinguish entry identity, source association, write order, and event time.

Table 11: Assignment and meaning of evidence-related entry metadata.

Field	Assigned by	Meaning
`seq`	write request, else system	Logical write coordinate
`origin`	system	Supporting Evidence object ID
`fid`	system	Entry identifier preserved when already present
`observed`	write request, else system	Date on which the source was observed
`time`	extractor, from the dialogue	Time of the event described by the entry
`source=AI`	extractor	Content authored by an assistant turn

C.4 Algorithmic Pseudocode

The procedures use $\mathcal{M} = (C, D, \mathcal{E})$ from Equation 1 and omit time subscripts. Entry-level `origin` metadata supplies ρ (Section 3). The first algorithm inlines ingestion within online scheduling; staged writing uses the same assignment and revision steps on rewritten batches.

Algorithm 1 Ingestion and Online Scheduling

procedure WriteAndConsolidate($\mathcal{M}, z, cfg$)
  if StaleBuffer($\mathcal{M}, cfg$) **then**
    $\mathcal{M} \leftarrow$ Consolidate($\mathcal{M}, cfg$)
  end if
  $(C, D, \mathcal{E}) \leftarrow \mathcal{M}$
  if Evidence is enabled **then**
    $\mathcal{E} \leftarrow$ ArchiveSourceGroupsIfAbsent($\mathcal{E}, z$)
  end if
  $F \leftarrow$ ExtractEntries(z, cfg)
  if Evidence is enabled **then**
    $F \leftarrow$ NormalizeAndAttachOrigins($F, z, \mathcal{E}$)
  end if
  $a \leftarrow (C \neq \varnothing) \wedge$ HasRetainedEntries(F)
  if HasRetainedEntries(F) **then**
    $C \leftarrow$ CreateOrAppendCurrent(C, F)
  end if
  $\mathcal{M}' \leftarrow (C, D, \mathcal{E})$
  if $a \wedge$ BufferOverLimit($\mathcal{M}', cfg$) **then**
    $\mathcal{M}' \leftarrow$ Consolidate($\mathcal{M}', cfg$)
  end if
  return $\mathcal{M}'$
end procedure

Here a records whether retained entries extend an existing buffer, so initial buffer creation does not trigger the size check. Eligible summaries S_D address candidate topics; U and N contain existing-document updates and new-topic proposals. REWRITEDOC revises the complete body and summary, while MERGEDUE captures the scheduled merge triggers. Staged runs call oversized-document reassignment separately.

Algorithm 2 Topic Consolidation

procedure CONSOLIDATE($\mathcal{M}$, *cfg*)
  $(C, D, \mathcal{E}) \leftarrow \mathcal{M}$
  if C is empty **then**
    return $\mathcal{M}$
  end if
  $R \leftarrow$ BODY(C); $S_D \leftarrow$ ELIGIBLESUMMARIES(D, cfg)
  $(U, N) \leftarrow$ SAFEPLANUPDATE(R, S_D, cfg)
  $(U, N) \leftarrow$ REDIRECTINVALIDTARGETS(U, N, S_D)
  for all $n \in N$ with nonempty content **do**
    $x \leftarrow$ TITLEANDCONTENT(n); $d \leftarrow$ REWRITEDOC$(\varnothing, x)$
    $D \leftarrow$ INSERTWITHNEWID(D, d)
  end for
  for all $u \in U$ **do**
    $D[u.\text{id}] \leftarrow$ REWRITEDOC$(D[u.\text{id}], u.\text{new_content})$
  end for
  $C \leftarrow$ CLEARCONSUMEDBUFFER(C)
  INCREMENTCONSOLIDATIONCOUNT
  if MERGEDUE(D, cfg) **then**
    $D \leftarrow$ MERGESMALLRELATED(D, cfg)
    $D \leftarrow$ REASSIGNOVERSIZEDDOCS(D, cfg)
  end if
  REFRESHLIBRARYMETADATA(D)
  return $(C, D, \mathcal{E})$
end procedure

SAFEPLANUPDATE splits long batches by heading or newline and merges assignments in order. If a plan is unusable, it recursively splits without overlap; new-topic proposals from the first half extend the catalog seen by the second.

Retrieval returns answer context Z. The agentic reader materializes selected documents or heading blocks. COALESCESUPPLEMENTANDCAP removes duplicate selections, adds BM25 blocks when the context is empty or short, and enforces the result limit. With Evidence enabled, maintained and archived content are selected together before context assembly; Hybrid uses summary-guided selection followed by BM25 heading blocks.

Algorithm 3 Retrieval and Source-Context Assembly

procedure RETRIEVE($\mathcal{M}, q, cfg$)
  $(C, D, \mathcal{E}) \leftarrow \mathcal{M}$
  if Evidence is enabled **then**
    $(V, E_q) \leftarrow$ SELECTMEMORYANDSOURCES$(q, C, D, \mathcal{E}, cfg)$
    return ASSEMBLESOURCECONTEXT(q, V, E_q, cfg)
  end if
  $L \leftarrow$ SEARCHABLECONTENT(C, D)
  $H \leftarrow [(q, \text{LIBRARYSUMMARIES}(L))]$; $Z \leftarrow [\,]$
  for $i = 1$ **to** *cfg*.agentic_max_iterations **do**
    $o \leftarrow$ AGENTSTEP(H)
    if o.done **then**
      $Z \leftarrow$ MATERIALIZESELECTIONS(o, L)
      break
    else if $o.\text{tool_calls} = [\,]$ **then**
      break
    end if
    $H \leftarrow$ ADVANCEHISTORY(H, o, L, cfg)
  end for
  return COALESCESUPPLEMENTANDCAP(q, L, Z, cfg)
end procedure

D DISCUSSION

Organization and answerability. Topic organization changes the unit the reader can address. In the original LongMemEval-S maintenance comparison, 21 of the 23-question net gain come from the multi-session category. On FactConsolidation, the reported maintenance comparison shows a larger improvement on multi-hop than on single-hop updates (Table 2). The split-threshold sweep adds a scale condition: in the tested range, allowing longer documents beyond the default hurts more than choosing shorter ones (Appendix B.3). Together, these observations make document granularity an answer-time design choice.

Source context at reading time. Archived dialogue retains speaker roles, dates, and adjacent turns after extracted entries have been reorganized. The source-aware comparison shows the largest separations on preference, assistant-detail, knowledge-update, and multi-session questions (Table 3). The paired cases show how source-local wording helps locate and interpret evidence already present in a derived document; they also show why a relative-time expression needs an explicit temporal anchor before it becomes an answer (Appendix B.2). Provenance is most useful when the reader can consult it while interpreting a retrieved fact.

The cost of keeping context available. On the matched 50-question profiles, omitting primary excerpts shortens the mean constructed answer prompt by 22.3%, while the reader still scans 47.1 archived objects per question on average (Table 6). Prompt length and source-read cost are thus separate targets for improvement. The next efficiency opportunity is source selection before archive reads.

D.1 STRATEGY SELECTION AND DESIGN RATIONALE

Topic documents keep related entries in coherent retrievable units while preserving text's flexibility for preferences, temporal revision, and source-linked details. Vector and graph indexes can serve as additional access paths over the same stored content.

The buffer separates frequent appends from document revision. Consolidation amortizes topic assignment, rewriting, and boundary changes, making maintenance predictable.

D.2 DEPLOYMENT AND EXTENSIBILITY

Topic documents and archived dialogue are readable text and can be held in a local filesystem or S3-compatible object store. This representation supports inspection and export, while summaries and headings can serve as addresses for additional retrieval indexes. The evaluated readers use summary selection, lexical matching, and iterative inspection. Because the archive includes conversation text, deployments should apply authorization, redaction, retention, export, and deletion policies to both source objects and derived documents. The two have independent lifecycles: deleting a topic leaves its source objects, and deleting a source may leave an entry with an unresolved provenance link.

D.3 LIMITATIONS

These findings reflect the evaluated tasks and retrieval settings. Fragmentation counts use system-specific record units, and efficiency measurements cover constructed answer prompts, source reads, and memory-build stage times. Request-level token usage and end-to-end cost remain unmeasured. Archived dialogue resides on writable storage, so reuse without overwriting does not establish tamper resistance.