HANDLEMEM: SOURCE-LINKED COMPLEMENTARY HANDLES FOR ADAPTIVE EVIDENCE ACQUISITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon agents often compress past interactions into compact memories. These representations play two roles: they guide the retrieval of relevant prior episodes and preserve evidence for later question answering. The difficulty is that compression must decide what to retain before the future use of the memory is known. As a result, it may discard cues needed to locate a relevant interaction, creating an *access gap*; it may also discard details needed to support an answer, creating an *evidence gap*. Since richer compression alone cannot eliminate these gaps, we argue that compact memory should instead provide access to retained evidence rather than replace it. We introduce HandleMem, a long-term memory framework built around this separation. It constructs source-linked complementary event- and detail-level handles that expose different retrieval cues for the same source segment, improving access coverage while preserving links to the original interaction for recovering omitted evidence. At query time, adaptive source resolution performs multi-round retrieval driven by the current information need and newly retrieved cues, using each observation to refine what to search for next, before resolving selected memories back to their sources for final answering. On LoCoMo, HandleMem achieves 57.40 F1 and 81.69% LLM-as-a-Judge accuracy. On LongMemEval-S, it achieves 80.20% LLM-as-a-Judge accuracy, obtaining the highest overall scores among the compared methods on both benchmarks. Our code is available in an anonymous repository at this link.

1 INTRODUCTION

Long-horizon agents often need to use information from interactions that no longer fit within the active context window (Packer et al., 2023). To address this, long-term memory systems compress past interactions into compact memories, such as summaries, facts, events, or structured records (Chhikara et al., 2025; Pan et al., 2025; Xu et al., 2026b). At inference time, these memories are retrieved and inserted into the active context window, where they serve as the basis for the model's response. In this common design, compact memories are asked to serve two roles: as retrieval targets for accessing long histories and as evidence for generating responses.

The problem is that compression must support both retrieval and answering without knowing what future queries will require. If compression omits a cue needed to find the relevant interaction, it creates an *access gap*. If it omits information needed to answer the query, it creates an *evidence gap*. Figure 1 illustrates both failures. The original interaction states that the user started pottery while recovering from a knee injury and records the exact date. Compression may preserve the pottery fact but lose its association with recovery, hindering retrieval; even when the relevant memory is found, the exact date may be missing, leaving insufficient evidence to answer. Thus, the access gap affects source retrieval, while the evidence gap affects answer support.

A natural way to mitigate these gaps is to make compact memories more informative, for example by retaining more facts, imposing richer structure, or constructing more granular representations (Wu et al., 2024; Xu et al., 2026a; Tang et al., 2026). Increasing memory precision can reduce both access and evidence gaps, but it does not eliminate their shared problem. When the same compressed representation serves as both the retrieval target and the answer evidence, it must anticipate at write time both how an unknown future query will refer to the relevant interaction and which details that

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

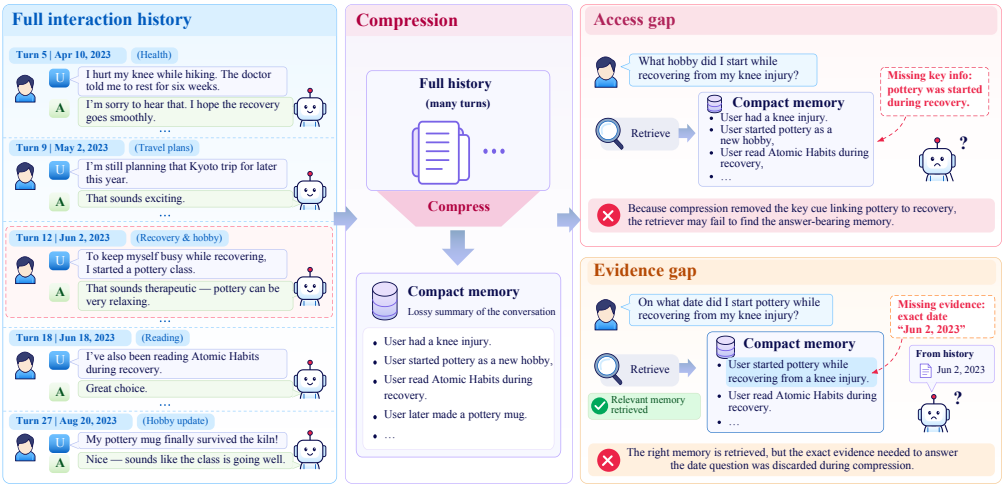


Figure 1: **Why compressed memory can fail.** Write-time compression may discard information needed by future queries. **Access gap:** a missing cue makes the relevant interaction hard to locate. **Evidence gap:** even when found, the compressed memory may lack the detail needed to answer.

query will require. We instead argue that compact memory should provide access to retained evidence rather than replace it: compression is used to make past interactions retrievable, while the original source segments remain the evidence for answering.

We introduce HandleMem, a memory framework that uses compact memories as source-linked retrieval handles rather than as complete answer evidence. In HandleMem, the original source segments remain the evidence for answering, while compressed handles expose cues for accessing them. The framework constructs **source-linked complementary handles** for each retained source segment: event-level handles capture the broader interaction or episode, while detail-level handles expose specific facts, entities, constraints, relations, preferences, and temporal information. Because an initial query may reveal only partial cues, **adaptive source resolution** performs iterative retrieval over these handles using the current information need and newly observed cues, deciding whether to search more broadly, search for finer-grained details, or inspect a linked source segment. After retrieval, selected handles are resolved to their original source segments, and the final answer is generated from those sources rather than from the compressed handles alone.

We evaluate HandleMem on LoCoMo (Maharana et al., 2024) and LongMemEval-S (Wu et al., 2024). HandleMem achieves 57.40 F1 and 81.69% LLM-as-a-Judge (LLJ) accuracy on LoCoMo and 80.20% LLJ accuracy on LongMemEval-S. Experiments show HandleMem outperforms baselines on LoCoMo and LongMemEval-S, while further analyses show that it mitigates both access and evidence gaps by improving source access and recovering answer-supporting evidence.

Our contributions are:

- We characterize access and evidence gaps as two failure modes that arise when compact memories must support both retrieval and answering before future queries are known.
- We propose HandleMem, which changes the role of compressed memory rather than using compressed memory as a substitute for the source: source-linked handles support access, while original interactions provide answer evidence.
- Experiments show HandleMem outperforms baselines on LoCoMo and LongMemEval-S. Further analyses show complementary handles improve source access, adaptive retrieval improves evidence acquisition, and source recovery benefits retrieval and final answering.

2 HandleMem

2.1 Overview

To address the access and evidence gaps described in Section 1, HandleMem separates memory access from answer evidence. Let an interaction history be divided into n retained source segments

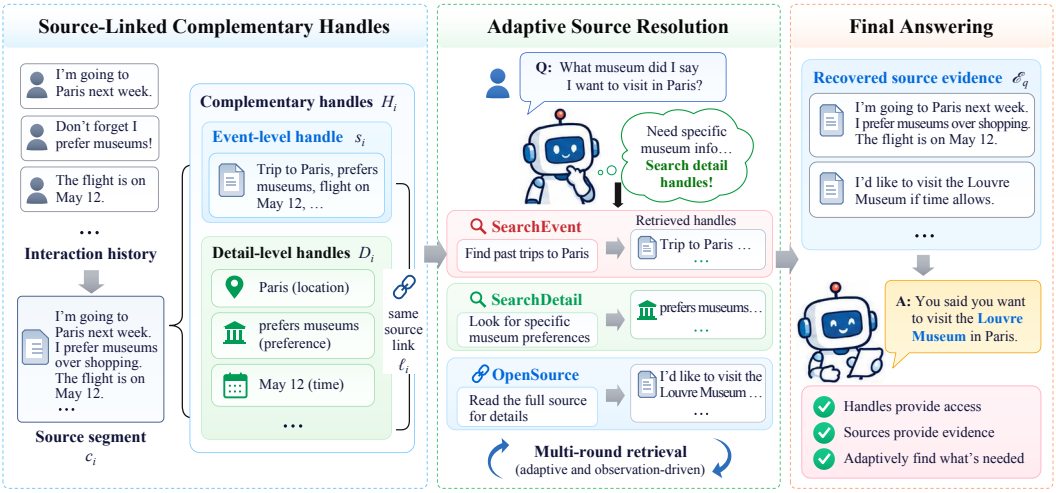


Figure 2: Overview of HandleMem. The framework builds source-linked complementary event- and detail-level handles that link back to retained source segments. Adaptive source resolution uses these handles for multi-round retrieval, then resolves selected memories to their sources for final answering.

$\mathcal{C} = \{c_i\}_{i=1}^{n}$. 2. We keep $\mathcal{C}$ as the evidence layer and build a compact access layer $\mathcal{M}$ on top of it:

$$\mathcal{M} = \{m_i\}_{i=1}^{n}, \qquad m_i = (H_i, \ell_i), \qquad \ell_i \rightsquigarrow c_i, \tag{1}$$

where H_i is a set of compact handles for source c_i, and ℓ_i is the source link retained by that memory record. The access layer is used to find relevant past interactions; the linked source segment remains the evidence used for answering.

This design addresses the two gaps separately. Complementary handles can expose different ways of reaching the same source, improving access coverage without forcing one compressed view to preserve every possible cue. The source link preserves a path to information omitted by compression, so missing answer evidence can still be recovered after the relevant source is found. Given a query q, the resulting flow is

$$q \xrightarrow{\text{search } \mathcal{M}} m_i \xrightarrow{\ell_i} c_i. \tag{2}$$

The key invariant is simple: **handles provide access; sources provide evidence**.

2.2 Source-Linked Complementary Handles

To instantiate the access layer in Eq. (1), HandleMem constructs source-linked complementary handles. A future query may refer to the same source via its broader experience or a specific fact, relation, condition, preference, or time. A single compact summary is unlikely to expose all of these references equally well. Let W_{evt} and W_{det} denote the event- and detail-level handle constructors, respectively. We then construct complementary event- and detail-level handles for each source segment:

$$s_i = W_{\text{evt}}(c_i), \qquad D_i = W_{\text{det}}(c_i, s_i), \qquad H_i = \{s_i\} \cup D_i. \tag{3}$$

The conditioning of D_i on s_i is important: detail handles are generated to complement the event view rather than form another independent summary of the source.

Event-level handles. The event-level handle s_i provides a broad contextual route to c_i. It organizes the source around its main entities, actions, relations, and state changes, so that the source can be found when a query refers to the overall experience. Its goal is not to preserve every detail, but to provide a compact contextual anchor. Here, "event-level" describes how the source is represented, not how the conversation is segmented. HandleMem does not require explicit event-boundary detection or assume that one source segment contains exactly one complete event.

Detail-level handles. Since the event-level handle s_i is intentionally compact, some fine-grained retrieval cues may be omitted or weakly expressed. We therefore construct

$$D_i = \{d_{ij}\}_{j=1}^{k_i} = W_{\text{det}}(c_i, s_i), \tag{4}$$

where k_i is the number of detail-level handles constructed for source c_i, and each d_{ij} exposes a concise, source-supported cue that is not sufficiently represented by the broad event view, such as a specific entity, relation, condition, preference, quantity, temporal cue, cause, or outcome. Conditioning on s_i encourages these handles to complement, rather than duplicate, the event-level representation. Each detail handle can be retrieved independently, providing additional routes for fine-grained queries to reach the same source c_i.

Every handle in H_i retains the same source link:

$$h \in H_i \quad \Longrightarrow \quad h \xrightarrow{\ell_i} c_i. \tag{5}$$

The handles are therefore alternative access routes to one retained source, not separate evidence units. This shared link lets the complementary handles improve access coverage while leaving omitted evidence recoverable from c_i.

2.3 Adaptive Source Resolution

Source-linked complementary handles provide multiple cues to reach a source, but the evidence needed by a query may still be unclear before retrieval begins. An initial result can reveal a new entity, relation, condition, or temporal cue, which changes what should be searched next. Adaptive source resolution therefore makes retrieval observation-driven rather than fixing all retrieval decisions from the original query.

Let $\mathcal{T}_{<t}$ denote the resolver trajectory before step t, consisting of the intermediate resolution states, tool calls, and resulting observations accumulated so far. The observations include retrieved handles and any source text inspected during resolution. At each step, the resolver uses the query and the accumulated trajectory to form an intermediate resolution state r_t and select the next action a_t:

$$(r_t, a_t) = \pi(q, \mathcal{T}_{<t}), \tag{6}$$

where π denotes the resolver's action-selection policy, and r_t captures its current assessment of what evidence remains needed and which retrieval cue should be pursued next. An action may search event handles, search detail handles, inspect a linked source, or terminate.

Executing a non-terminating action a_t produces an observation o_t, after which the trajectory is extended as

$$\mathcal{T}_{\leq t} = \mathcal{T}_{<t} \oplus (r_t, a_t, o_t), \tag{7}$$

where $\oplus$ denotes trajectory concatenation.

Adaptive retrieval. The resolver accesses the two complementary views through `SearchEvent` and `SearchDetail`. Event search is useful for broader contextual references, while detail search targets specific facts, relations, conditions, preferences, or temporal cues. The view and search query are selected adaptively based on the current information need reflected in the accumulated trajectory.

A retrieved handle can either satisfy part of the query or reveal a new cue for subsequent retrieval. Multi-round retrieval therefore follows an evolving interaction loop:

$$(q, \mathcal{T}_{<t}) \longrightarrow (r_t, a_t) \longrightarrow o_t \longrightarrow \mathcal{T}_{\leq t}, \tag{8}$$

where $\mathcal{T}_{\leq t}$ denotes the updated trajectory after incorporating observation o_t. Thus, information discovered at one step, together with the resolver's intermediate state, can directly reshape the next retrieval decision.

Source-guided resolution. The same source link ℓ_i supports both intermediate inspection and final evidence recovery. During exploration, a retrieved handle may locate the right interaction but still omit a condition or detail needed to continue. `OpenSource` follows ℓ_i to inspect c_i; the recovered source text becomes part of the resulting observation and is appended to the trajectory, where it can guide later retrieval or selection. This inspection is selective and used only when the compact view is insufficient.

Within a fixed action budget, the resolver continues until the required evidence is sufficiently covered or further search is unlikely to help. Let V_q denote the memory records encountered along the retrieval trajectory, and let $S_q \subseteq V_q$ denote the records selected when the resolver terminates.

The final evidence context is obtained by resolving all selected records to their original sources, removing duplicate sources and ordering the remaining segments chronologically:

$$\mathcal{E}_q = \text{Chronological}\left(\text{Dedup}\{c_i \mid m_i \in S_q\}\right). \tag{9}$$

The answer model receives q and $\mathcal{E}_q$, rather than the compressed handles or intermediate resolver trajectory. Since multiple selected handles may resolve to the same source, DEDUP removes repeated source records, while CHRONOLOGICAL restores the retained sources to their original temporal order. In this way, source-linked complementary handles broaden how a source can be reached, while adaptive source resolution decides which routes to follow as information needs evolve and ultimately recovers the original source evidence needed for the final answer.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

Implementation details. We use GPT-4o-mini (OpenAI, 2024) throughout HandleMem: for memory construction, query-time evidence acquisition, and final answer generation.Event- and detail-level handles are indexed separately using BM25 and BGE-M3 dense retrieval with reciprocal-rank fusion. For LoCoMo, each original session is retained as one source segment; LongMemEval-S, dialogues are segmented into chunks of up to 1,024 tokens. Unless otherwise specified, all HandleMem variants use the same experimental settings. Additional implementation details are provided in Appendix A.

Benchmarks and metrics. We evaluate HandleMem on two widely used long-term conversational memory benchmarks: LoCoMo (Maharana et al., 2024) and LongMemEval-S (Wu et al., 2024). Following prior work, we evaluate 1,540 non-adversarial questions from four categories: Single-Hop, Multi-Hop, Temporal, and Open-Domain. LongMemEval-S contains 500 questions over conversation histories averaging approximately 115K tokens, covering single-session-user (SSU), multi-session (MS), single-session-preference (SSP), temporal reasoning (TR), knowledge update (KU), and single-session-assistant (SSA).

For effectiveness, we report token-level F1 and LLM-as-a-Judge (LLJ) accuracy on LoCoMo, and LLJ accuracy on LongMemEval-S following its benchmark protocol. F1 measures lexical overlap with reference answers, while LLJ evaluates semantic correctness using a binary correctness decision. We use GPT-4o-mini as the judge and report all effectiveness scores as percentages. For efficiency, we separately measure offline token consumption during memory construction and online token consumption during query-time evidence acquisition and final answering.

Baselines. We compare HandleMem with Full Context, RAG (Lewis et al., 2020), and representative long-term memory systems. Full Context provides the answer model with the complete interaction history, while RAG retrieves relevant segments directly from the retained raw interaction history. The long-term memory baselines include A-Mem (Xu et al., 2026b), Mem0 (Chhikara et al., 2025), Nemori (Nan et al., 2025), MemoryOS (Kang et al., 2025), ES-Mem (Zou et al., 2026), LightMem (Fang et al., 2026), and LangMem (LangChain, 2026).We use GPT-4o-mini as the LLM backbone for HandleMem and RAG; for Full Context and the long-term memory baselines, we use previously reported results obtained with GPT-4o-mini.

3.2 MAIN RESULTS

We first evaluate the overall effectiveness of HandleMem. Table 1 reports the results on LoCoMo under token-level F1 and LLJ evaluation, respectively, while Table 2 presents LLJ accuracy on LongMemEval-S.

LoCoMo F1. As shown in Table 1, HandleMem achieves 57.40 overall F1 and ranks first in all four categories. **(1) Comparison with Full Context and RAG.** HandleMem improves overall F1 over Full Context and RAG by 11.20 and 10.88 points, respectively. Since both baselines already use original conversation text, retaining uncompressed evidence alone does not explain this advantage. The comparison highlights the importance of how relevant evidence is reached: making the full

Table 1: LoCoMo results (%) on non-adversarial questions. Each category reports F1 (left) and LLJ accuracy (right). Best results are bolded; "–" denotes unavailable results.

	Single-Hop		Multi-Hop		Temporal		Open-Domain		Overall	
Method	F1	LLJ	F1	LLJ	F1	LLJ	F1	LLJ	F1	LLJ
Full Context	53.10	83.00	35.40	66.80	44.10	56.20	24.50	48.60	46.20	72.30
RAG	51.22	74.67	35.22	60.64	50.06	60.75	26.68	54.17	46.52	67.92
A-Mem	44.65	58.20	27.02	43.60	45.85	54.20	12.14	22.90	39.65	52.50
Mem0	47.65	68.10	38.72	60.30	48.93	50.40	28.64	40.60	45.09	61.30
Nemori	46.33	81.90	32.36	61.70	55.99	67.60	29.19	45.80	44.72	73.00
MemoryOS	48.62	62.50	35.27	52.50	41.15	38.00	20.02	45.80	42.84	54.50
ES-Mem	50.07	–	36.52	–	47.90	–	24.77	–	45.56	–
LightMem	47.64	76.81	32.11	67.02	53.79	76.32	26.14	45.83	44.73	72.99
LangMem	40.91	61.40	35.51	52.40	30.75	24.90	26.04	47.60	36.87	51.30
HandleMem	**63.06**	**87.63**	**43.22**	**73.40**	**62.47**	**79.75**	**32.53**	**60.42**	**57.40**	**81.69**

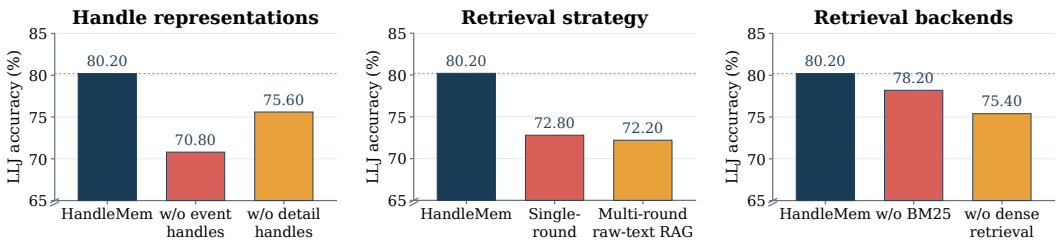
Table 2: LongMemEval-S LLJ accuracy (%). Best results are bolded.

Method	SSU	MS	SSP	TR	KU	SSA	Overall
Full Context	78.60	38.30	6.70	42.10	78.20	89.30	55.00
RAG	77.14	45.11	63.33	51.88	65.38	98.21	61.80
A-Mem	**92.86**	48.87	46.67	47.36	64.11	96.43	62.20
Mem0	81.43	46.21	60.00	40.15	70.12	41.07	53.51
Nemori	88.60	51.10	46.70	61.70	61.50	83.90	64.20
MemoryOS	80.00	31.06	30.00	32.33	48.72	64.29	44.66
ES-Mem	84.29	66.17	**73.33**	64.66	78.21	82.14	72.40
LightMem	87.14	71.74	68.18	67.18	**83.12**	32.14	69.81
LangMem	60.00	20.30	60.00	15.79	66.67	46.43	37.20
HandleMem	91.43	**72.93**	70.00	**77.44**	76.92	**100.00**	**80.20**

history available or retrieving directly from it is less effective here than using complementary handles and observation-driven retrieval to locate relevant sources. **(2) Comparison with existing memory systems.** HandleMem exceeds the strongest overall memory baseline, ES-Mem, by 11.84 points. The gains extend beyond Single-Hop questions, with improvements of 4.50 and 6.48 points over the strongest Multi-Hop and Temporal baselines, respectively. This pattern is consistent with the benefit of separating access from evidence: once a relevant source is located, answering is not restricted to the facts, relations, or temporal details preserved in its compressed handles. Together, these comparisons support a design that combines compact retrieval cues with recoverable source evidence, rather than relying on either direct access to raw history or compressed memories alone.

LoCoMo LLJ. The same advantage holds under semantic evaluation (Table 1). HandleMem achieves 81.69% overall LLJ accuracy, exceeding Full Context and RAG by 9.39 and 13.77 percentage points, respectively. It also outperforms Nemori, the strongest reported memory baseline, by 8.69 percentage points and ranks first in all four categories. Thus, the advantage over both groups of baselines is not limited to lexical overlap: it also appears in the judge's assessment of answer correctness. The controlled experiments below examine how complementary handles, iterative retrieval, and source-grounded answering contribute to this overall improvement.

LongMemEval-S. As shown in Table 2, HandleMem achieves the highest overall LLJ accuracy of 80.20%. It outperforms Full Context and RAG by 25.20 and 18.40 percentage points, respectively, and exceeds ES-Mem, the strongest memory baseline, by 7.80 points. Across the six categories, HandleMem ranks first in MS, TR, and SSA and second in SSU and SSP, while ranking fourth in KU. This shows that the improvement is not limited to a particular type of query. The largest gain appears on temporal reasoning (TR), where HandleMem reaches 77.44%, 10.26 points higher than the strongest baseline. This strong temporal performance is consistent with our design of



(a) Handle representations. (b) Retrieval strategy. (c) Retrieval backends.

Figure 3: Component ablations on LongMemEval-S. Each panel reports LLJ accuracy (%), with the full system shown as the dark reference bar.

recovering original source segments for answering, which preserves temporal expressions and contextual relations that may be omitted during compression. Overall, HandleMem achieves the best aggregate performance, while the KU results indicate remaining room for improvement.

3.3 Ablation Studies

We conduct ablation and diagnostic experiments on LongMemEval-S to examine the contributions of complementary handles, adaptive retrieval, and the hybrid retrieval backend. All variants follow the same evaluation protocol as the full system, with the specified component or retrieval strategy changed. Figure 3 summarizes the overall results.

Complementary handle representations. Removing event or detail handles lowers overall accuracy by 9.40 and 4.60 percentage points, respectively. The larger decrease without event handles suggests that fine-grained cues alone do not fully substitute for broader interaction context, while the decrease without detail handles shows the additional value of more specific retrieval cues. Importantly, both variants retain original source segments for final answering: the difference lies in the representations used to locate those sources, not in replacing source evidence with compressed text. These results support complementary event- and detail-level views rather than treating either representation as a sufficient access interface on its own.

Adaptive retrieval. The single-round and multi-round raw-text RAG controls test whether the full-system gain can be explained by the handle representation alone or by iterative retrieval alone. Single-round retrieval retains the handles but removes subsequent rounds of observation-driven search, whereas multi-round raw-text RAG retains iterative retrieval under the same action budget but searches original interactions directly. Their respective accuracies of 72.80% and 72.20% remain below the full system by 7.40 and 8.00 percentage points. Thus, neither one-shot access through handles nor repeated raw-text retrieval recovers the full performance. The results support combining complementary retrieval cues with a process that can revise its search as intermediate observations reveal new information needs.

Retrieval backends. Removing BM25 or dense retrieval lowers overall accuracy by 2.00 and 4.80 percentage points, respectively. The hybrid configuration therefore performs better than either backend alone under the evaluated setting, consistent with lexical and semantic matching providing complementary retrieval signals. These results support retaining both backends alongside the complementary handle representations and adaptive retrieval process.

3.4 Access and Evidence Gap Analysis

Table 3 examines two complementary questions on LongMemEval-S: whether the retrieval process can reach the annotated evidence sources, and whether the compact handles remain sufficient once those sources have been identified.

Access gap. Table 3(a) evaluates evidence coverage in the final source set, with session recall measuring the fraction of gold session segments annotated as evidence for each question that are

Table 3: Access and evidence diagnostics on LongMemEval-S. (a) Final evidence-session coverage evaluates whether relevant sources are reached. (b) Source-use controls evaluate whether original source information remains necessary during evidence acquisition and answering.

(a) Source coverage

Metric	Value
Session recall (%)	91.80
Selected segments	2.35

(b) Source recovery

Setting	LLJ (%)	Δ
HandleMem	**80.20**	0.00
w/o `OpenSource`	71.20	−9.00
Handles-only answer	73.40	−6.80

Table 4: LongMemEval-S efficiency and answer quality. Token counts are mean tokens per case/question, reported in K; time is reported in seconds.

		A-MEM	**Mem0**	**LightMem**	**LangMem**	**HandleMem**
Tokens (K)	Offline tokens	1497.1	1144.2	40.7	3003.9	317.8
	Online tokens	3.7	0.7	1.0	1.1	24.0
	All tokens	1500.8	1144.9	41.7	3005.0	341.8
Time (s)	Offline time	3035.0	1082.3	211.0	2836.2	210.4
	Online time	6.1	5.1	6.8	6.2	14.6
	Total time	3041.1	1087.4	217.8	2842.4	225.0
LongMemEval-S LLJ (%)		62.2	53.5	69.8	37.2	80.2

covered by the selection. HandleMem achieves 91.80% mean session recall while selecting only 2.35 source segments per question on average. These results show that high coverage of the annotated evidence is achieved within a small final source set, rather than by passing a large number of segments to the answer model. This finding supports the intended role of handles as access interfaces for locating relevant source evidence and assembling a compact context for answer generation.

Evidence gap. Table 3 (b) examines what happens after relevant memories have been accessed. Removing `OpenSource` reduces overall accuracy from 80.20% to 71.20%, even though selected memories are still resolved to their original sources before final answering. This suggests that source information is useful during evidence acquisition itself. Inspecting a linked source can recover conditions, relations, temporal information, or other context omitted from a compact handle, which can in turn guide later retrieval and selection decisions.

The handles-only control isolates a different role of source evidence. Here, the selected memory records are kept fixed, but the answer model receives their compact handles instead of the corresponding source segments. Accuracy decreases to 73.40%, a 6.80-point drop that cannot be explained by differences in memory selection. This result is consistent with the evidence gap: a handle may contain enough information to locate a relevant interaction while still omitting details needed to support the final answer. Together, the two controls show that retained sources are useful both during evidence acquisition and after retrieval is complete, supporting the separation between compact representations for access and original interactions for evidence recovery.

3.5 Efficiency Analysis

Table 4 compares efficiency and answer quality on LongMemEval-S, showing that HandleMem keeps offline costs relatively low by using compressed memories primarily as access handles rather than as complete answer evidence. Memory construction therefore focuses on exposing useful retrieval cues without requiring complex memory structures or encoding every potentially useful answer detail. As a result, HandleMem uses 317.8K offline tokens and 210.4 seconds, substantially less than A-MEM, Mem0, and LangMem. Adaptive source resolution introduces additional query-time computation, increasing the online cost to 24.0K tokens and 14.6 seconds per question. However, the online cost is still much smaller than the savings from offline construction. Overall, HandleMem reduces total token usage by 70.2–88.6% and total runtime by 79.3–92.6% compared with these three methods, while achieving higher answer accuracy. LightMem remains more efficient overall, but HandleMem

improves LLJ accuracy by 10.4 percentage points with only 3.3% more total runtime. These results show that HandleMem uses some additional computation at query time to recover evidence, while keeping the overall cost low and achieving the highest accuracy among the compared methods.

4 Related Work

Memory Compression for LLM Agents. Long-term memory systems for LLM agents compress interaction histories into compact representations for efficient reuse over extended interactions. Early approaches maintain persistent external memories or summarize past interactions into reusable records (Park et al., 2023; Zhong et al., 2024; Packer et al., 2023). Recent methods improve memory construction through fact extraction, structured notes, hierarchical organization, and progressive consolidation (Chhikara et al., 2025; Xu et al., 2026b; Kang et al., 2025; Nan et al., 2025; Fang et al., 2026). These approaches enable more compact and scalable use of interaction history by retaining important information in increasingly structured forms. However, compression requires deciding what to preserve before future information needs are known, often without knowing which details will later become relevant. Consequently, details that seem secondary during memory construction may be discarded even when later needed to answer a specific query.

Memory Organization and Access. Beyond compression, prior work organizes memory into richer structures to improve access over large information spaces. Hierarchical and associative retrieval methods use intermediate representations to locate relevant content at different levels or through semantic associations (Sarthi et al., 2024; Gutiérrez et al., 2024; Qian et al., 2025). In long-term agent memory, related approaches organize interactions using event-based and hierarchical memory structures (Zou et al., 2026; Sun et al., 2026). These structures provide more effective access points than flat retrieval and allow memory to be searched at different semantic granularities. However, the representations used to organize and retrieve memory are often also used as the content provided for downstream reasoning. Such coupling makes retrieval dependent on abstractions that may omit fine-grained source details, particularly when the information required by a query does not align with the representation chosen during memory construction.

Adaptive Evidence Retrieval. Retrieval-augmented reasoning has increasingly moved from one-shot retrieval toward iterative and adaptive information acquisition. Early agentic methods interleave reasoning with external search or decompose questions into intermediate information needs (Yao et al., 2022; Press et al., 2023). Subsequent approaches refine retrieval using intermediate reasoning or generated content (Trivedi et al., 2023; Jiang et al., 2023; Shao et al., 2023), while adaptive retrieval methods further determine when external information is needed and how retrieval should proceed (Asai et al., 2024; Jeong et al., 2024; Schick et al., 2023). These methods demonstrate that information revealed during reasoning can guide subsequent retrieval. However, they mainly study retrieval from external knowledge sources, where the query is progressively refined against a relatively independent corpus. In long-term conversational memory, relevant evidence may instead be distributed across related interactions, and an initially retrieved memory may expose entities, relations, or temporal cues required to locate additional evidence. This dependency makes fixed, one-shot retrieval particularly vulnerable to incomplete evidence acquisition.

5 Conclusion

We introduced HandleMem to address a limitation of long-term memory systems that use compressed representations for both memory access and answer evidence. Because future information needs are unknown at write time, compression can omit either cues needed to locate relevant interactions or evidence needed to answer from them. HandleMem separates these roles: source-linked handles provide complementary access paths, while adaptive source resolution recovers the retained source evidence needed for answering. Experiments on LoCoMo and LongMemEval-S show strong overall performance, while further analysis and ablations confirm the contributions of source-linked complementary handles and adaptive source resolution to the two failure modes. These results support a simple design principle: long-term memory can remain compact for access without requiring compression to preserve every future-useful detail.

6 AI USE STATEMENT

Generative AI tools were used to assist with translation and language polishing, including improving readability and correcting grammatical and typographical errors. All AI-assisted text was reviewed by the authors, who take full responsibility for the final content.

7 ETHICS STATEMENT

Our method retains original conversations as evidence for answering questions, which improves evidence recovery but also introduces privacy and data-security risks. Therefore, practical deployment should include appropriate safeguards, such as user consent, access controls, and mechanisms for deleting both original conversations and their corresponding memory handles. Moreover, access to original evidence does not guarantee the correctness of model outputs, which should still be validated in downstream applications.

8 REPRODUCIBILITY STATEMENT

The methodology and experimental setup are described in Sections 2 and 3.1, respectively, with additional implementation details provided in Appendix A. We also make our code available through an anonymous GitHub repository linked in the abstract.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pp. 9112–9141, 2024.

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.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, et al. Lightmem: Lightweight and efficient memory-augmented generation. In *International Conference on Learning Representations*, volume 2026, pp. 98706–98729, 2026.

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

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 conference of the north american chapter of the association for computational linguistics: Human language technologies (volume 1: Long papers)*, pp. 7036–7050, 2024.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 7969–7992, 2023.

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. 25972–25981, 2025.

LangChain. LangMem: Long-term agent memory tools, 2026. URL `https://langchain-ai.github.io/langmem/`. Software documentation. Accessed September 6, 2026.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

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.

Jiayan Nan, Wenquan Ma, Wenlong Wu, and Yize Chen. Nemori: Self-organizing agent memory inspired by cognitive science. *arXiv e-prints*, pp. arXiv–2508, 2025.

OpenAI. GPT-4o mini: advancing cost-efficient intelligence, 2024. URL `https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/`. Model announcement, July 18, 2024.

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, et al. Secom: On memory construction and retrieval for personalized conversational agents. In *International Conference on Learning Representations*, volume 2025, pp. 91851–91885, 2025.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, 2023.

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.

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*, volume 2024, pp. 32628–32649, 2024.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9248–9274, 2023.

Haoran Sun, Shaoning Zeng, and Bob Zhang. H-mem: Hierarchical memory for high-efficiency long-term reasoning in llm agents. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 341–350, 2026.

Zihao Tang, Xin Yu, Ziyu Xiao, Zengxuan Wen, Zelin Li, Jiaxi Zhou, Hualei Wang, Haohua Wang, Haizhen Huang, Weiwei Deng, et al. Mnemis: Dual-route retrieval on hierarchical graphs for long-term llm memory. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23914–23928, 2026.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 10014–10037, 2023.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*, 2024.

Derong Xu, Yi Wen, Pengyue Jia, Yingyi Zhang, Wenlin Zhang, Yichao Wang, Huifeng Guo, Ruiming Tang, Xiangyu Zhao, Enhong Chen, et al. From single to multi-granularity: Toward long-term memory association and selection of conversational agents. In *International Conference on Learning Representations*, volume 2026, pp. 91581–91617, 2026a.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604, 2026b.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

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.

Huhai Zou, Tianhao Sun, Chuanjiang He, Yu Tian, Zhenyang Li, Li Jin, Nayu Liu, Jiang Zhong, and Kaiwen Wei. Es-mem: Event segmentation-based memory for long-term dialogue agents. *arXiv preprint arXiv:2601.07582*, 2026.

A Implementation Details

Source segments and handles. For each segment, HandleMem generates one event-level handle and zero to eight detail-level handles.

Retrieval. `SearchEvent` and `SearchDetail` search separate event-level and detail-level indices, respectively. Each search action accepts one to three queries. BM25 and dense retrieval each return the top-36 handle candidates from the selected view. BM25 uses $k_1 = 1.5$ and $b = 0.75$. We combine the ranked lists using reciprocal-rank fusion (RRF), with rank contributions of $1/(60+\mathrm{rank})$. We then group candidates by their underlying memory record and return the top six distinct records to the resolver. No additional reranker is used.

Action budget. Adaptive source resolution allows at most 12 actions per question. Each call to `SearchEvent`, `SearchDetail`, `OpenSource`, or `Finish` counts as one action. If the resolver has not produced a valid `Finish` before the final action, it is forced to make a `Finish` call as the 12th and final action.

Source inspection. Each `OpenSource` call opens one previously retrieved source. The resolver may inspect at most ten distinct sources per question. Each call returns the corresponding source segment, which remain in the resolver's history for later steps.

Evidence selection. `Finish` selects at most ten previously retrieved memory records. Empty selections are disallowed once valid search results are available. Selections containing unretrieved memory IDs are rejected. The selected records are resolved to their original source segments, deduplicated by source identity, and ordered chronologically for final answering.