Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Metamemory-guided retrieval and memory revision for long-term conversational memory

Shiwei Wu
Shandong University
Yafei Ning
ningyafei@sdu.edu.cn
Shandong University
Chao Zhang
Nanjing University of Science and Technology
Linghui Min
Shandong University
Shuo Chen
Shandong Extreme Vision Technology Co., Ltd.

Research Article

Keywords:

Posted Date: November 2nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11343428/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Metamemory-guided retrieval and memory revision for long-term conversational memory

Abstract

Relevant information in long-term conversations is often scattered across sessions. Even when a system finds evidence that supports its answer, it may omit other information needed to address the question. To address this problem, we propose a framework for metamemory-guided active retrieval and memory revision (MARR). The framework organizes historical conversations into a heterogeneous memory graph, connecting records across sessions through entities and topics while retaining original messages and temporal information for later inspection. When answering a question, the system checks whether the evidence is sufficient in light of what it has already answered and uses the actual retrieval trajectory to determine where to search next. Information obtained through supplementary retrieval informs revision proposals, which the Writer checks against the original text before deciding whether to accept them, reducing unsupported changes. Evaluations on LoCoMo and LongMemEval-S show that the framework recovers some omitted evidence and improves question-answering performance. Further analysis finds that some answer errors can be corrected by reexamining material already available, while continued retrieval yields more pronounced benefits for multi-hop questions with incomplete evidence. These findings suggest that the effectiveness of long-term conversational memory depends both on finding relevant information and on the system's ability to understand and use it adequately.

Keywords: metamemory; knowledge graph; long-term conversational memory; large language model

1 Introduction

Intelligent assistants in long-term interactions need to use past conversations to answer current questions, such as which activities a user has participated in, how their preferences have changed, and when similar events occurred. As interactions accumulate, relevant facts become scattered across sessions, and the experiences and states of the same person may change. These tasks require systems to recover cross-session connections and understand the subject, time, and state of each fact. LoCoMo (Maharana et al. 2024) and LongMemEval (Wu et al. 2025) incorporate cross-session information extraction and reasoning into memory evaluation, advancing research on long-term memory for sustained interactions.

However, retrieving evidence that supports an answer does not mean that the question has been answered adequately. As shown in Figure 1, the user asks which volunteer activities they participated in before moving. The initial answer about a park cleanup is supported by the original text but omits a food bank experience. Confirming the latter requires combining the activity information in u_3, which has already been read, with the participation record in u_4, which has not been retrieved. Verifying existing statements alone cannot reveal this omission; cross-session evidence must also be found and integrated. We use **insufficient recall** to describe cases in which historical information is not fully recovered or correctly used in answering the current question, and distinguish factual support for existing statements from coverage of the information required by the question.

Prior research provides foundations in memory organization and feedback-based revision. HippoRAG (Jiménez Gutiérrez et al. 2024) uses knowledge graphs to support associative retrieval, and A-Mem (Xu et al. 2025) organizes memory through structured notes and dynamic links. Self-RAG (Asai et al. 2024) coordinates retrieval and generation, while CRITIC (Gou et al. 2024) uses

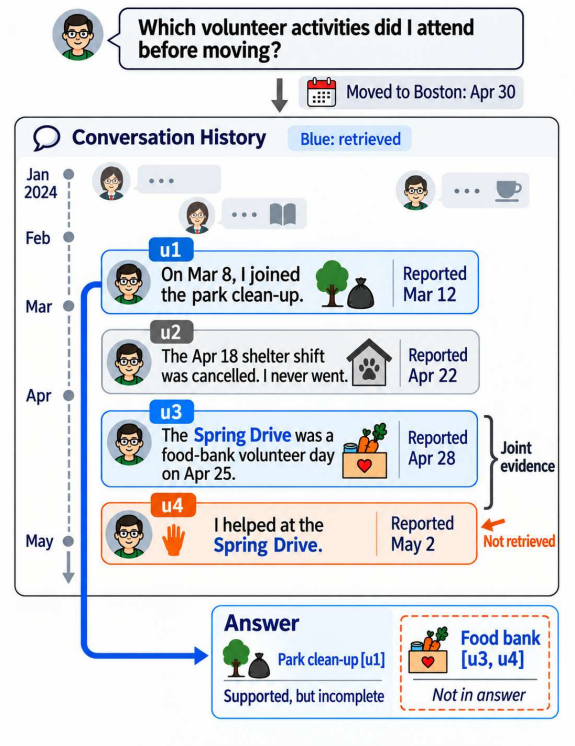


Fig. 1 The distinction between factual support and answer coverage. A hypothetical cross-session example. Blue messages are retrieved sources; the orange message u_4 has not yet been retrieved. u_1 supports the initial answer about the park cleanup. The food bank experience must be confirmed jointly by the activity type and date in u_3 and the record of the user's own participation in u_4. Although u_4 was reported after the move, the activity still took place before it; the canceled activity in u_2 is excluded from the answer.

tool feedback to verify and revise outputs. Building on these advances, we focus on regulating recall after an initial answer has been produced from shared conversational memory. At this point, the information not yet covered is not known in advance. The system needs to determine subsequent retrieval directions from what it has already answered and searched for, and judge whether new findings justify a revision. Supported conclusions may conceal omissions, while semantically related new records may concern a different person, event, or time range. Continued evidence acquisition therefore needs to work together with judgments about whether that evidence applies.

Research on human memory offers a monitoring-and-control perspective on this process. The metamemory framework of Nelson and Narens (1990) distinguishes monitoring of memory activity from control processes that regulate subsequent activity using monitoring information. Koriat and Goldsmith (1996) further discuss how reporting decisions regulate the accuracy and quantity of information in memory reports. These studies motivate us to consider both how information is acquired and how it is used. In this study, monitoring concerns the current answer and its evidence acquisition process, while control takes the form of continued retrieval, checks against the original text, and answer adjustment. We therefore ask: **When initial recall may be insufficient, how can the current recall state guide further evidence acquisition and verification to fill omissions, correct errors, and preserve as much existing correct content as possible?**

To this end, we propose the **Metamemory-Guided Active Retrieval and Revision (MARR) framework**. Its central idea is to regulate subsequent retrieval using the current answer and its evidence acquisition process, and to judge whether to revise the answer based on the original text. The initial answer presents the conclusions already reported, the evidence already read provides a basis for verification, and the actual retrieval trajectory records the queries attempted, access paths, and results. Together, these form an observable recall state that enables the system to identify leads for further examination based on what it has already answered, read, and searched for.

This process is implemented through cooperation between a Writer and a Challenger. The Writer first retrieves evidence and generates an initial answer. The Challenger receives this answer and inherits the existing retrieval state, conducts supplementary retrieval for potential omissions, event confusion, and temporal conditions, and adjusts the next retrieval round according to the results actually returned. Supplementary retrieval may seek relevant records that have not yet been read or recheck original text already available. The Challenger then provides review comments with source citations. The Writer judges whether the original text supports the proposed changes and whether the corresponding facts match the subject, event, and time range required by the question. The framework separates proposing changes from accepting them, retaining content that remains valid while completing or correcting the answer

and preserving necessary uncertainty about details that cannot be verified. We call this process of adjusting the recall result for the current question based on evidence memory revision.

To support supplementary retrieval across sessions and source verification, MARR organizes event and state records into a heterogeneous memory graph through shared entities and topics, retaining links to original messages as well as event times and report times. Connections in the graph help locate candidates, while original messages are used to check specific facts. Both roles share the same read-only memory graph. Revision affects only the current answer, with at most one final revised answer produced after the initial answer.

We evaluate MARR on LoCoMo and LongMemEval-S. In a developmental evaluation of 1,540 LoCoMo questions, supplementary retrieval recovered 125 of the 410 question/annotated supporting-message pairs missing at the initial-answer stage, or 30.5%. Across all 500 questions in LongMemEval-S, the Challenger increased Judge accuracy from 81.80% to 85.00%. Ablation experiments show that supplementary retrieval has a more pronounced effect on LoCoMo multi-hop questions with incomplete initial evidence. These results support continued evidence acquisition, examination, and integration after the initial answer, while also showing that revision gains need to be evaluated alongside the preservation of correct initial answers.

The main contributions of this work are as follows:

1. We propose MARR to address cases in long-term conversational question answering where an initial answer has factual support but may still omit necessary information. MARR uses the initial answer, evidence already read, and actual retrieval trajectory jointly to make subsequent retrieval decisions, directing supplementary retrieval toward potential omissions, event confusion, and temporal conditions in the current answer.
2. We design an answer revision mechanism grounded in original sources that separates proposing changes from accepting them. The Writer verifies the Challenger's review comments against the visible original text and considers the subject, event, and temporal conditions to determine which content should be retained, completed, or corrected.
3. We conduct question-answering evaluations, component ablations, and evidence coverage analyses on LoCoMo and LongMemEval-S to examine the roles of rechecking existing evidence, supplementary retrieval, and actual retrieval trajectories. We assess the net benefit of revision through paired counts of error corrections and harmful revisions. The results show that rechecking existing material can correct some initial-answer errors, while supplementary retrieval yields more pronounced benefits for multi-hop questions with incomplete initial evidence.

2 Related work

2.1 Long-term memory for large language model agents

Research on long-term memory for large language model (LLM) agents examines how to retain experiences, user information, and task knowledge from past interactions and retrieve this information for subsequent tasks. In sustained multi-turn interactions spanning multiple sessions, memory systems need to decide what to store, how to update it as new information arrives, and how to place relevant experiences in the current context. Existing methods address experience storage, hierarchical management, and information integration. Generative Agents (Park et al. 2023) records experiences in natural language and combines retrieval, reflection, and planning to support agent behavior. MemGPT (Packer et al. 2023) draws on hierarchical storage mechanisms in operating systems, allowing models to call memory functions to manage information inside and outside the context window. MemoryOS (Kang et al. 2025) organizes short-term, mid-term, and long-term memory to support storage, updating, retrieval, and generation. Mem0 (Chhikara et al. 2025) extracts important information from ongoing conversations and reduces redundancy through consolidation or updating to maintain reusable memories.

Long-term conversation benchmarks evaluate memory through the recovery and use of information across sessions. LoCoMo (Maharana et al. 2024) tests factual question answering, temporal reasoning, and multi-hop associations in long conversations. LongMemEval (Wu et al. 2025) further distinguishes information extraction, multi-session

reasoning, temporal reasoning, knowledge updating, and abstention, and analyzes performance at the indexing, retrieval, and reading stages. These studies show that memory design must account for both information retention and task-oriented use. We focus on the question-answering stage after historical memory has been constructed, studying how a system can review its initial answer and continue searching for the cross-session evidence required by the question.

2.2 Structured memory and associative retrieval

To locate and integrate scattered information, existing methods explicitly organize memory content and its relationships. A-Mem (Xu et al. 2025) generates notes with semantic attributes, establishes links between notes, and updates existing representations as new memories arrive. MemInsight (Salama et al. 2025) autonomously augments historical interactions with semantic attributes that aid retrieval. These methods improve the accessibility of historical information through memory associations and semantic descriptions, respectively.

Graphs are also used to connect entities and facts across records. Inspired by hippocampal indexing theory, HippoRAG (Jiménez Gutiérrez et al. 2024) combines knowledge graphs with personalized PageRank to support associative retrieval. HippoRAG 2 (Jiménez Gutiérrez et al. 2025) integrates passages more deeply and introduces online model involvement to support both factual retrieval and associative reasoning. For conversational memory, the graph memory variant of Mem0 (Chhikara et al. 2025) organizes interaction information through entities and their relationships. These methods provide different mechanisms for representing and accessing cross-record associations.

Our framework uses structured memory records and links established through entities and topics as the basis for memory access, retains references to original messages, and keeps memory read-only during question answering. On this basis, it uses the initial answer and retrieval state to continue locating cross-session evidence and verifies whether that evidence supports revision.

2.3 Memory regulation and feedback-based revision

Beyond organizing and retrieving memory, systems must decide when to continue searching, what to search for, and how to use the information obtained. Metamemory research provides a monitoring-and-control framework for these decisions (Nelson and Narens 1990). Research on memory reports further examines how people's judgments about their recall affect which information they report or temporarily withhold (Koriat and Goldsmith 1996). Recent work also directly studies memory control in large language model systems. AutoMem (Wu et al. 2026) treats memory management as a learnable cognitive skill, optimizing memory structure through trajectory analysis and training models to perform memory operations. MemCon (Jiang et al. 2026) learns adaptive policies from task feedback to select retrieval timing, content, and quantity, as well as memory consolidation and forgetting operations. These two recent preprints extend the focus from memory representations to decisions about using and managing memory.

Active retrieval and feedback-based revision offer related technical approaches. IRCoT (Trivedi et al. 2023) retrieves more information based on intermediate reasoning; ReAct (Yao et al. 2023) alternates between reasoning and tool actions; Self-RAG (Asai et al. 2024) learns reflection tokens to coordinate retrieval, generation, and critique. Feedback can also guide the current output or subsequent attempts. Self-Refine (Madaan et al. 2023) iteratively improves outputs through self-feedback, while Reflexion (Shinn et al. 2023) stores verbal feedback in episodic memory to guide subsequent attempts. For answers that depend on evidence, CRITIC (Gou et al. 2024) obtains verification feedback through external tools, and RARR (Gao et al. 2023) retrieves supporting evidence to revise unsupported content. Together, these studies establish a basic approach: acquire more information based on intermediate states and revise outputs accordingly.

3 Method

3.1 Method overview

Relevant information in long-term conversational question answering is often scattered across multiple sessions, and a person's experiences, preferences, and states may change over time. Retrieving evidence that supports a statement does not mean that all the information needed to answer the current question has been obtained. For example, a record of an activity may support the statement that someone "participated in this activity," yet be insufficient to answer "Which activities has this person participated in?" Answer review must therefore assess whether existing statements are supported and check for other relevant facts that may have been omitted.

MARR examines and adjusts the current recall result through retrieval-trajectory-aware review and evidence-constrained revision. Figure 2 shows the overall MARR framework. First, historical conversations are organized into a heterogeneous memory graph that preserves links to the original source messages. The Writer then actively retrieves evidence and generates an initial answer. Using the initial answer, the evidence already read, and the actual retrieval trajectory, the Challenger conducts supplementary retrieval to investigate potential omissions, event confusion, and temporal conditions. Finally, the Writer verifies the review feedback against the original messages and generates the final answer.

The initial answer, the evidence already read, and the actual retrieval trajectory together constitute the observable recall state for the current question. This state provides a basis for monitoring answer adequacy and controlling subsequent retrieval. MARR separates the proposal of revisions from their acceptance: the Challenger searches for evidence that may affect the initial answer, while the Writer determines whether that evidence is sufficient to support a revision. In this paper, memory revision refers to the process of completing or correcting the current recall result based on evidence. Both roles share the same read-only memory graph and tool interface, and revisions apply only to the answer to the current question.

3.2 Problem definition: factual support and answer coverage

Let the conversation history consist of M sessions. The j-th session s_j contains a report time τ_j^{report} and a message sequence $\mathcal{U}_j$ in its original order. Given a question q, MARR constructs a memory graph and generates an answer with source citations:

$$\begin{aligned} H &= \{s_j\}_{j=1}^{M}, \qquad s_j = (\tau_j^{\text{report}}, \mathcal{U}_j), \\ G &= \mathcal{M}(H), \qquad (a^*, \mathcal{C}^*) = \mathcal{F}(q, G). \end{aligned} \tag{1}$$

Here, $\mathcal{M}$ and $\mathcal{F}$ denote MARR's memory construction and question-answering procedures, respectively, and $\mathcal{C}^*$ is the set of original message IDs cited in the final answer. Each message retains its speaker, text, message ID, and position within the session for source tracing and context recovery.

This paper distinguishes factual support from answer coverage. Let $\mathcal{E}$ denote the original messages already read, $\mathcal{R}_{\text{unsupported}}(a; \mathcal{E})$ denote statements in the answer that are not yet supported by these messages, and $\mathcal{R}_{\text{missing}}(q, a; H)$ denote facts in the conversation history that are relevant to the question but not covered by the answer. Then:

$$\begin{aligned} &\mathcal{R}_{\text{unsupported}}(a; \mathcal{E}) = \varnothing \\ &\quad \not\Rightarrow \quad \mathcal{R}_{\text{missing}}(q, a; H) = \varnothing. \end{aligned} \tag{2}$$

Equation (2) captures the limitation of verifying only the initial answer: all existing statements may be supported by evidence, while other necessary information remains omitted. Since relevant facts in the unread history are not known in advance, MARR searches for omissions through supplementary retrieval within a fixed budget; it does not treat the sets in Equation (2) as completeness labels known at runtime.

3.3 Heterogeneous memory graph with preserved source links

3.3.1 Memory representation and source preservation

The memory graph contains five node types: sessions, original messages, memory records, entities, and topics. Events and states are distinguished by the type field of each memory record. MARR extracts structured records session by session

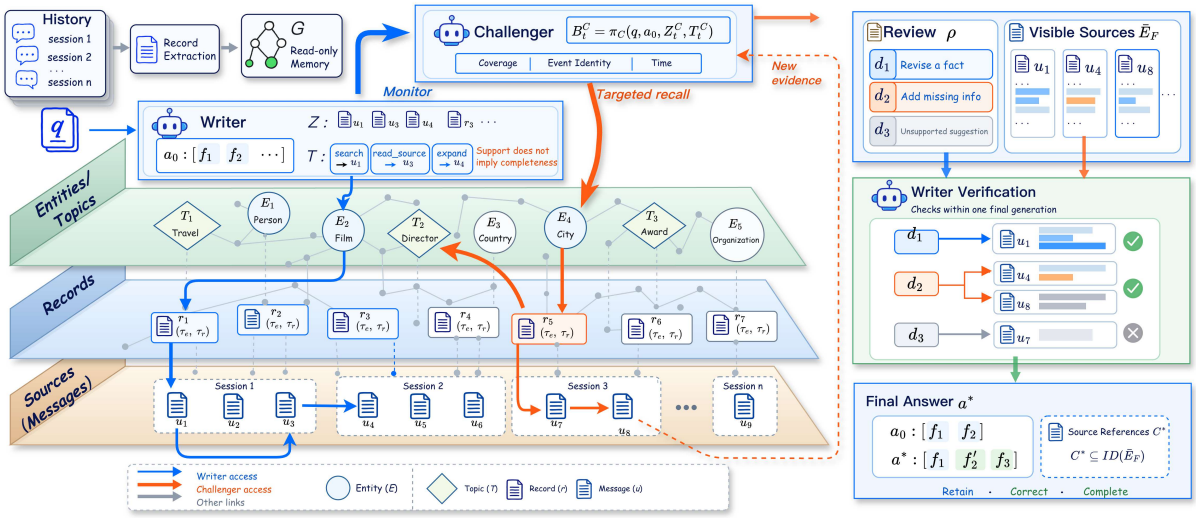


Fig. 2 Overall MARR framework. Historical conversations are organized into a heterogeneous memory graph that preserves links to the original source messages. The Writer retrieves information for the question and produces an initial answer a_0. The Challenger inherits the cumulative retrieval state Z and actual execution trajectory T, uses the initial answer to check potential omissions, event identity, and temporal conditions, and conducts targeted supplementary retrieval. The blue and orange paths illustrate memory access by the two roles. After supplementary retrieval, the Writer uses the original messages visible in the final request, $\widetilde{\mathcal{E}}_F$, to verify the review feedback ρ. In a single final-answer generation step, the Writer retains supported content, corrects errors or fills omissions, and provides source citations $\mathcal{C}^*$. The factual entries and access paths in the figure are illustrative. The memory graph remains read-only during question answering, and revisions apply only to the current answer.

and retains their corresponding original messages. The memory graph and an individual record are represented as:

$$\begin{aligned} G &= (V, \mathcal{L}), \\ r &= (x, \kappa, \mathcal{X}, \mathcal{T}, \tau^{\text{event}}, \tau^{\text{report}}, \mathcal{P}), \\ \kappa &\in \{\text{Event}, \text{State}\}. \end{aligned} \tag{3}$$

V is the node set, and $\mathcal{L}$ is the set of typed edges. To simplify notation, session and record subscripts are omitted for an individual record. Table 1 describes the fields.

Memory construction does not take evaluation questions, reference answers, or answer locations as inputs. The question-answering stage likewise uses neither reference answers nor Judge results. Each record is linked to at least one source message in the current session. Event records cover completed, ongoing, and planned activities, while state records describe relatively stable information such as preferences and relationships. Extraction preserves the correct subject attribution, negation, uncertainty, and distinction between planned and completed activities, without adding details absent from the original text. Original messages are stored separately, so content not covered by structured records remains accessible through message-level retrieval.

3.3.2 Cross-session associations and source lookup

Memory records are connected through entity and topic nodes. Entity and topic names are normalized for whitespace and case so that access nodes can be reused; this processing does not perform full alias disambiguation. For a record r, its set of associated candidates is defined as:

$$\begin{aligned} \mathcal{N}_{\text{assoc}}(r) = \big\{ r' \neq r : \\ \mathcal{X}(r) \cap \mathcal{X}(r') \neq \varnothing \vee \\ \mathcal{T}(r) \cap \mathcal{T}(r') \neq \varnothing \big\}. \end{aligned} \tag{4}$$

Equation (4) corresponds to an access path from a record through an entity or topic to an associated record. No separate direct edges between records are created in the graph. After locating candidates, MARR follows their source links to read the original messages and may include adjacent

Table 1 Fields of event and state records in MARR

Field	Content and purpose
x	Event or state description, used as the content representation of the record
κ	Record type, distinguishing event experiences from state information such as preferences and relationships
$\mathcal{X}$	Related entities, connecting records that involve the same people, places, or organizations
$\mathcal{T}$	Related topics, providing topic-based access across sessions
τ^{event}	Event time expressed in the original text, used for temporal ordering and comparison
τ^{report}	Report time of the session or message, preserving the temporal context of the statement
$\mathcal{P}$	Original messages supporting the record, used for factual verification and context recovery

messages from the same session to check referents, omitted context, and speaker attribution.

Shared entities or topics provide retrieval clues only. A person may participate in similar activities on multiple occasions, and a topic may involve different experiences. Whether associated candidates refer to the same event and can complete or correct the initial answer must therefore still be determined from the original text.

3.3.3 Event time and report time

The memory graph stores the time an event occurred separately from the time it was reported. For relative time expressions, the report time may provide context for interpretation; when the original text provides no basis for the event time, the report time is not used as a substitute.

The timeline tool organizes candidates with identifiable event times in chronological order and lists records with unknown or ambiguous times separately. The time-comparison tool computes only the temporal order or interval between two selected records. Both operations depend on event selection: candidates must first be confirmed to match the subject and activity specified in the question before their temporal information is used.

3.4 Retrieval-trajectory-aware evidence acquisition

3.4.1 Dual-granularity retrieval and tool interface

MARR builds retrieval indices at two granularities: memory records and original messages. The record level provides summarized entry points for retrieval, while the message level preserves specific details from the original text. At each granularity, BM25 (Robertson and Zaragoza 2009) keyword retrieval and BGE (Xiao et al. 2024) vector retrieval are used, and candidate scores are computed through reciprocal rank fusion (RRF) (Cormack et al. 2009):

$$s_\ell(d, q') = \sum_{\substack{m \in \mathcal{M}_{\text{ret}} \\ d \in L_{\ell,m}(q')}} \frac{1}{k_0 + \text{rank}_{\ell,m}(d; q')}, \quad \ell \in \{\text{record}, \text{message}\}. \tag{5}$$

$\mathcal{M}_{\text{ret}}$ is the set of enabled retrievers, and d is a candidate memory record or original message at granularity ℓ. q' is the current retrieval query, $L_{\ell,m}(q')$ is the ranked list returned by retriever m at granularity ℓ, and $\text{rank}_{\ell,m}$ is the candidate's rank. The sum includes only lists in which the candidate actually appears, and the smoothing constant k_0 is set to 60. When the vector index is disabled, only keyword ranking is used. Candidates are selected separately at the two granularities, after which their corresponding source messages are expanded.

The Writer and Challenger use the same six types of tools, as shown in Table 2.

3.4.2 Actual trajectory and retrieval state

MARR records actual tool calls and their returned results, then updates the retrieval state accordingly. Let $R \in \{W, C\}$ denote the Writer and Challenger, respectively, and let T_t^R and Z_t^R denote the cumulative trajectory and retrieval state before planning round t. After the operation batch for this round is executed:

$$\begin{aligned} T_{t+1}^R &= T_t^R \oplus \left[(o_{t,i}^R, y_{t,i}^R)\right]_{i=1}^{m_t}, \\ Z_{t+1}^R &= \text{Merge}(T_{t+1}^R). \end{aligned} \tag{6}$$

Table 2 Memory access tools in MARR

Tool	Access method and purpose
`search`	Searches globally at the record and message levels to establish an entry point for the question or search again based on new clues
`expand`	Follows entity or topic links to find associated records, with at most two hops from a record and at most one hop from an entity or topic
`read_source`	Reads original messages and adjacent context within the same session to check subjects, referents, negation, and event status
`similar`	When the vector index is enabled, finds semantic neighbors in other sessions from an existing record or message, adding candidates beyond explicit graph links
`timeline`	Organizes related records by event time and lists unknown or ambiguous times separately to check event order and time ranges
`compare_time`	Compares the event times of two selected records to compute temporal order or intervals

$o^R_{t,i}$ and $y^R_{t,i}$ are the i-th tool operation and its returned result, respectively, m_t is the number of operations in this round, and $\oplus$ denotes appending logs in execution order. Merge accumulates and deduplicates the returned memory records and original messages while retaining access paths and temporal results. The trajectory is produced by program execution and is not replaced by the model's retrospective description of retrieval.

Operations within a batch are jointly determined in a single planning step and then executed sequentially. The model can use newly returned results only in the next planning round, so retrieval decisions are adjusted between rounds, rather than replanned after each tool returns.

3.4.3 Initial-answer construction and targeted supplementary retrieval

The planning processes of the two roles in MARR are expressed as:

$$\begin{aligned} B^W_t &= \pi_W(q, Z^W_t, T^W_t), \\ B^C_t &= \pi_C(q, a_0, Z^C_t, T^C_t). \end{aligned} \tag{7}$$

B^R_t is the batch of tool operations for the current round, and π_W and π_C are implemented by the language model with the corresponding role prompts. Equation (7) describes the information on which the two stages depend; actual model inputs are also subject to the context organization rules in Section 3.4.4.

The Writer first performs a global `search` using the question, then plans subsequent operations based on the retrieval results, with at most two planning rounds and at most three actions per round. An empty batch ends the process early; the initial `search` does not count toward these two planning rounds. After retrieval, the Writer produces the initial answer a_0 from the original messages, includes source citations, and retains necessary qualifications for content that cannot be confirmed.

The Challenger inherits the Writer's cumulative trajectory and retrieval state and also receives the initial answer. The initial answer contains the conclusions already reported, the evidence already read provides a basis for verification, and the trajectory records the queries attempted, access paths, and their results. The Challenger uses this information to determine targets for supplementary retrieval, such as clues to activities not yet covered, events that may have been confused, or other records that satisfy the question's temporal conditions.

The Challenger also has at most two planning rounds with at most three actions per round and may stop early. Logs from supplementary retrieval are appended to the inherited trajectory, and subsequent review uses the cumulative material from both stages. Newly accessed and previously accessed messages are marked separately, but a valid revision need not introduce a new source: rechecking negation, referents, or temporal qualifications in text already read may also correct the initial answer.

3.4.4 Context organization and citation scope

The cumulative retrieval state is distinct from the context actually included in each model request. MARR organizes separate inputs for planning,

initial answering, review, and final answering; complete execution logs are archived, while the model makes judgments only from the material supplied in the current request.

When the context exceeds the window, MARR first compresses whitespace in serialized content and removes redundant notes, catalogs, or trajectory fields. The LoCoMo final-answer configuration in this paper retains all original messages and temporal evidence in the cumulative retrieval material and reports failure if they still cannot fit. For long histories in LongMemEval-S, if the context still exceeds the window after redundancy removal, complete messages cited by the current answer and those read most recently receive priority, and omitted sources are recorded. Messages excluded from the request remain in the history and execution logs.

For the k-th model request, let $\widetilde{\mathcal{E}}^{(k)}$ denote the original messages actually included, and let $\mathcal{C}^{(k)}$ denote the message citations output by that request. The following requirement applies throughout:

$$\mathcal{C}^{(k)} \subseteq \mathrm{ID}\Big(\widetilde{\mathcal{E}}^{(k)}\Big). \tag{8}$$

ID extracts the original message IDs. This constraint applies to the initial answer, review feedback, and final answer, requiring citations to come from original text visible in the current request. Output constraints and programmatic checks enforce source ID validity; whether the source content supports the stated conclusion still requires semantic verification.

3.5 Source-constrained review and a single revision

3.5.1 Review feedback

During MARR's review stage, the Challenger combines the initial answer with the cumulative material after supplementary retrieval to provide review feedback:

$$\rho = \mathrm{Review}(q, a_0, \mathcal{I}_C) = (v, \eta, \mathcal{C}_\rho), \tag{9}$$

$\mathcal{I}_C$ is the material actually included in the review request, v is the recommended disposition, η is the issue description, and $\mathcal{C}_\rho$ is the set of source citations. The disposition is `keep`, `revise`, or `insufficient_evidence`, indicating, respectively, that there is not yet sufficient reason to revise, that a proposed revision requires verification, or that the relevant judgment still lacks evidence. Source citations follow Equation (8).

For feedback involving time, the review must identify the target event required by the question, the event used in the initial answer, and the candidate event proposed for use, with corresponding sources. Verification distinguishes three cases: a correction to the description of the same event, an error in the initial event selection, and an unresolved relationship between events. Event selection is checked first, followed by whether the temporal calculation supports a revision. The temporal audit results provide clues for the final check.

3.5.2 Writer verification and final answering

The Writer treats the review feedback as judgments to be tested. It checks, in order, whether sources are available, whether the original text supports the suggestion, and whether the people, event status, and temporal conditions meet the question's requirements. Final answering is expressed as:

$$(a^*, \mathcal{C}^*) = \mathrm{Finalize}_W\big(q, a_0, \rho, \mathcal{I}_F\big), \tag{10}$$

$\mathcal{I}_F$ is the cumulative material and verification information actually supplied in the final-answer request. Based on this material, the Writer may retain the initial answer, fill omissions, or correct errors, preserving uncertainty where evidence is insufficient. Added or substituted facts require support from the original text; content that remains valid and relevant to the question should be retained. Final source citations also follow Equation (8).

MARR performs the final-answer step in Equation (10) once for each question, with the review labels providing a basis for the Writer's verification. No further retrieval loop begins after final answering. The initial answer, review feedback, final answer, source citations, and actual execution trajectory are saved.

3.6 Algorithm

Algorithm 1 presents the pseudocode for MARR. The memory graph and the record- and message-level retrieval indices are built before question answering, and both the Writer and Challenger access the same read-only memory graph. Each role starts at $t = 0$ and increments t only after executing a nonempty batch. Thus, when the loop terminates, T_t^R and Z_t^R always represent the final cumulative trajectory and retrieval state for that stage.

Here, $o_{t,i}^R$ is the operation in batch B_t^R at position i in execution order; the initial retrieval operation and its returned result are denoted by o and y, respectively. Execute executes the given tool operation.

To represent the initial-answer step, let $\mathcal{I}_W$ additionally denote the material actually included in the initial-answer request, and let $\mathcal{C}_0$ denote its set of source citations. Answer_W represents the Writer's initial-answer procedure in Section 3.4.3. BuildContext organizes the material actually visible in the current request according to Section 3.4.4, while BuildFinalContext also includes the initial-answer citations $\mathcal{C}_0$, source citations in the review feedback, and temporal verification information.

Including the initial retrieval, at most $1 + 3 \times (2 + 2) = 13$ tool actions are executed per question. MARR has at most seven model stages: two Writer planning steps, one initial-answer step, two Challenger planning steps, one review step, and one final-answer step. Early stopping of retrieval can reduce the actual number of calls.

4 Experiments

This section reports the main results of MARR on LoCoMo and LongMemEval-S, compares configurations for the Writer's initial answer construction, supplementary retrieval, and actual retrieval trajectories, and analyzes the corrections and harmful revisions produced by MARR's subsequent supplementary retrieval and revision process.

4.1 Experimental setup

Datasets. We use two benchmarks for long-term conversational memory: LoCoMo (Maharana et al. 2024) and LongMemEval-S (Wu et al. 2025). LoCoMo contains 10 long-term conversations, 272 sessions, and 5882 original messages. The evaluation covers 1540 questions in categories 1-4, comprising 841 single-hop, 282 multi-hop, 321 temporal reasoning, and 96 open-domain questions; adversarial questions in category 5 are excluded. Memory is constructed separately for each conversation and remains read-only during question answering. As these conversations were used during the early development of MARR, the LoCoMo results constitute a development evaluation.

The LongMemEval-S experiments use all 500 questions in the official cleaned-S dataset, which contains 23,867 historical sessions and 246,750 messages. The six question types comprise 70 single-session user information questions, 56 assistant information questions, 30 preference questions, 133 multi-session questions, 133 temporal reasoning questions, and 78 knowledge update questions. The 30 abstention questions form an overlapping subset of these types and are not added to the total.

Comparison methods. We compare MARR with the following six long-term memory methods:

- **A-Mem (Xu et al. 2025):** Drawing on the Zettelkasten method, it organizes interactions into notes containing content, context, keywords, and tags. When new memories are added, it creates semantic links and updates existing notes. During question answering, it retrieves relevant notes and their associated content.
- **Mem0 (Chhikara et al. 2025):** It extracts facts from recent messages and historical summaries, retrieves similar memories, and uses a large language model to decide whether to add, update, delete, or leave memories unchanged. During question answering, it retrieves consolidated memory entries to provide historical facts.
- **LangMem (LangChain Team 2025):** It uses a large language model to extract persistent information such as facts and preferences, creating or updating memory entries that can be accessed across sessions. It provides memory management and search as tools for agents to maintain memory and retrieve the context needed to answer questions.
- **MemoryOS (Kang et al. 2025):** It maintains short-term, mid-term, and long-term personal memories. Short-term conversations move into mid-term memory on a first-in, first-out basis, while long-term memory is updated according

Algorithm 1 Retrieval and revision procedure of MARR

Input: Question q; read-only memory graph G and retrieval indices built offline
Budget: At most two planning rounds each for Writer and Challenger, with at most three actions per round
Output: Final answer a^* and source citations $\mathcal{C}^*$

$o \leftarrow (\text{search}, q)$; $y \leftarrow \text{Execute}(G, o)$ ▷ Initial retrieval does not count toward the planning budget
$T_0^W \leftarrow [(o, y)]$; $Z_0^W \leftarrow \text{Merge}(T_0^W)$
$t \leftarrow 0$
while $t < 2$ **do** ▷ Writer retrieval
  $B_t^W \leftarrow \pi_W(q, Z_t^W, T_t^W)$
  if $B_t^W = \varnothing$ **then break**
  $m_t \leftarrow |B_t^W|$ ▷ $m_t \leq 3$
  for $i \leftarrow 1$ **to** m_t **do**
    $y_{t,i}^W \leftarrow \text{Execute}(G, o_{t,i}^W)$
  end for
  $T_{t+1}^W \leftarrow T_t^W \oplus [(o_{t,i}^W, y_{t,i}^W)]_{i=1}^{m_t}$
  $Z_{t+1}^W \leftarrow \text{Merge}(T_{t+1}^W)$
  $t \leftarrow t + 1$
end while
$\mathcal{I}_W \leftarrow \text{BuildContext}(\texttt{initial_answer}, q, Z_t^W, T_t^W)$
$(a_0, \mathcal{C}_0) \leftarrow Answer_W(q, \mathcal{I}_W)$
$(T_0^C, Z_0^C) \leftarrow (T_t^W, Z_t^W)$ ▷ Inherit the Writer's cumulative material
$t \leftarrow 0$
while $t < 2$ **do** ▷ Challenger targeted supplementary retrieval
  $B_t^C \leftarrow \pi_C(q, a_0, Z_t^C, T_t^C)$
  if $B_t^C = \varnothing$ **then break**
  $m_t \leftarrow |B_t^C|$ ▷ $m_t \leq 3$
  for $i \leftarrow 1$ **to** m_t **do**
    $y_{t,i}^C \leftarrow \text{Execute}(G, o_{t,i}^C)$
  end for
  $T_{t+1}^C \leftarrow T_t^C \oplus [(o_{t,i}^C, y_{t,i}^C)]_{i=1}^{m_t}$
  $Z_{t+1}^C \leftarrow \text{Merge}(T_{t+1}^C)$
  $t \leftarrow t + 1$
end while
$\mathcal{I}_C \leftarrow \text{BuildContext}(\texttt{review}, q, a_0, Z_t^C, T_t^C)$
$\rho \leftarrow \text{Review}(q, a_0, \mathcal{I}_C)$ ▷ $\rho = (v, \eta, \mathcal{C}_\rho)$
$\mathcal{I}_F \leftarrow \text{BuildFinalContext}(q, a_0, \mathcal{C}_0, \rho, Z_t^C, T_t^C)$
$(a^*, \mathcal{C}^*) \leftarrow Finalize_W(q, a_0, \rho, \mathcal{I}_F)$ ▷ Answer once after verifying the original text
$\text{Save}(a_0, \mathcal{C}_0, \rho, a^*, \mathcal{C}^*, T_t^C)$
return $(a^*, \mathcal{C}^*)$

to heat scores. During question answering, it retrieves topic segments and dialogue pages and combines recent and long-term information to generate answers.

- **TiMem (Li et al. 2026):** It organizes memory into five levels using a temporal memory tree: segments, sessions, days, weeks, and user profiles. It summarizes facts, behavioral patterns, and preferences level by level. During retrieval, it matches bottom-level segments and collects ancestor memories, filtering them for relevance and temporal consistency before using them to answer questions.
- **MRAgent (Ji et al. 2026):** It models recall as evidence reconstruction over an associative graph, connecting events, facts, and topics through cues and tags. Starting from question cues, it selects tags and reads content, adjusts the traversal direction based on the evidence, prunes irrelevant branches, and alternates between retrieval and reasoning.

Evaluation metrics. We use F1 and LLM-as-Judge accuracy, both reported as percentages. F1 measures token overlap between the predicted and reference answers. Precision P, recall R, and F1 are defined as follows:

$$P = \frac{M}{N_p}, \qquad R = \frac{M}{N_r}, \qquad \text{F1} = \frac{2PR}{P+R} \times 100\%. \tag{11}$$

Here, N_p and N_r are the numbers of tokens in the predicted and reference answers, respectively, and M is the number of matching tokens, with repeated tokens matched by their occurrence counts; F1 is 0 when there are no matches. LLM-as-Judge determines whether an answer is correct based on the question and reference answer, and accuracy is the proportion judged correct. For MARR on LoCoMo, the Qwen generation group uses DeepSeek Flash Judge, and the DeepSeek generation group uses Qwen3.8-27B-FP8 Judge; on LongMemEval-S, the Qwen and DeepSeek generation groups use Qwen3.8-27B-FP8 and DeepSeek Flash Judge, respectively. External baselines retain the scoring configurations of their respective sources. Judge configurations for the ablation experiments are specified separately.

Implementation details. MARR's Writer and Challenger use Qwen3.8-27B-FP8 or DeepSeek Flash according to the experimental group, with temperature set to 0 and thinking mode disabled. The requested DeepSeek model is deepseek-v4-flash. Retrieval uses BGE-base-en-v1.5 and BM25, combined through RRF; the BM25 parameters are $k_1 = 1.2$ and $b = 0.75$, and the RRF constant is 60. Each agent performs at most two planning rounds, with at most three tool actions per round.

4.2 Main results

Tables 3 and 4 report the overall results and results by question type on LoCoMo and LongMemEval-S, respectively. The following comparisons use the configurations reported for each method.

LoCoMo. As shown in Table 3, MARR achieves overall Judge accuracies of 88.64% and 86.56% in the Qwen and DeepSeek groups, respectively, ranking first in both groups and exceeding the best baseline, TiMem, by 7.99 and 9.66 percentage points. MARR also achieves the highest Judge accuracy in each group across all four question types: single-hop, multi-hop, temporal reasoning, and open-domain. Its overall F1 scores are 56.83% and 54.15%, respectively, ranking second in each group, below TiMem's 58.55% and 62.49%. Thus, MARR performs well in the overall assessment of answer correctness, although it still trails the best baseline on the token-matching metric.

For multi-hop question answering, MARR achieves the highest F1 and Judge scores in both groups. Its multi-hop Judge accuracies are 82.27% and 77.66% in the Qwen and DeepSeek groups, exceeding TiMem, the best baseline for this question type, by 10.28 and 9.83 percentage points, respectively; its multi-hop F1 scores are 45.69% and 56.90%, exceeding TiMem's 44.74% and 44.15% by 0.95 and 12.75 percentage points, respectively. By comparison, MARR does not achieve the highest F1 in either group for single-hop, temporal reasoning, or open-domain questions. These results show that MARR's advantage in Judge accuracy extends across all four question types, while multi-hop question answering is the type on which it leads in both F1 and Judge in both groups, demonstrating its performance advantage on tasks that require integrating multiple facts.

LongMemEval-S. In Table 4, MARR achieves overall Judge accuracies of 85.00% and 87.00% under the Qwen and DeepSeek configurations, exceeding the best baseline in each group, TiMem, by 4.00 and 6.04 percentage points, respectively. Its overall F1 scores are 54.10% and 51.66%, exceeding MemoryOS, the best F1 baseline in each group, by 0.86 and 0.33 percentage points, respectively. MARR therefore ranks first in its corresponding group on both overall metrics, with a larger reported margin in Judge accuracy.

The results by question type further show a relatively consistent Judge advantage on questions that require integrating multiple pieces of information. MARR's Judge accuracies on multi-session questions in LongMemEval-S are 79.70% and 82.71%, exceeding the best baselines for this question type by 6.02 and 6.73 percentage points, respectively; its temporal reasoning Judge accuracies are 81.20% and 82.71%, with corresponding margins of 0.75 and 3.76 percentage points. Together with the multi-hop results on LoCoMo, these observations are consistent with MARR's design goal of continuing to retrieve dispersed information to complete and check answers. However, the main tables reflect MARR's overall performance; the individual contributions of supplementary retrieval and actual retrieval trajectories must be assessed using the ablation results in Section 4.3.

MARR still has limitations in preference question answering and abstention. Its preference Judge accuracies in the Qwen and DeepSeek groups are 63.33% and 66.67%, below MemoryOS's 73.33% and 76.55% in the corresponding groups; its Judge accuracies on the abstention subset are only 66.67% and 70.00%, also below several baselines in the

Table 3 Main results on LoCoMo (%). The full results for MARR and the Qwen baselines cover 1540 questions in categories 1-4, while results from the literature cover 1985 questions.

Model	Method	Overall		Single-hop		Multi-hop		Temporal		Open-domain	
		F1	Judge	F1	Judge	F1	Judge	F1	Judge	F1	Judge
qwen-3.8-27b	A-Mem	40.94	69.09	55.96	72.65	30.11	65.96	17.53	70.41	19.47	42.71
	Mem0	42.54	66.51	52.40	72.38	42.01	64.18	22.35	57.48	25.19	52.08
	LangMem	8.63	52.21	7.00	55.17	19.32	43.97	5.22	49.84	2.87	58.33
	MemoryOS	45.63	66.49	58.36	74.20	40.71	64.89	21.58	52.34	29.02	51.04
	TiMem	**58.55**	80.65	**64.26**	87.51	44.74	71.99	63.01	78.50	**34.23**	53.13
	MRAgent v1	50.30	63.31	51.74	66.83	36.36	51.77	**69.44**	75.39	14.63	26.04
	MARR	56.83	**88.64**	61.98	**94.05**	**45.69**	**82.27**	60.50	**87.23**	32.13	**64.58**
deepseek-v4-flash	A-Mem[a]	40.91	25.04	54.67	37.90	31.94	13.10	18.66	2.80	21.13	21.90
	Mem0[a]	44.05	54.87	55.15	69.00	41.79	40.10	22.72	33.60	24.71	45.80
	LangMem	9.30	50.80	7.54	53.13	21.74	41.60	4.95	50.94	2.72	56.95
	MemoryOS	48.23	64.61	63.54	75.15	38.77	57.45	21.82	48.60	30.19	46.88
	TiMem	**62.49**	76.90	**68.71**	81.72	44.15	67.83	70.64	80.11	**34.59**	50.59
	MRAgent v1	53.77	63.41	53.60	67.37	40.92	52.71	**77.41**	73.60	13.98	26.07
	MARR	54.15	**86.56**	59.22	**94.29**	**56.90**	**77.66**	47.09	**82.55**	25.31	**58.33**

[a] Published DeepSeek-V4-Flash results are taken from Fan et al. (2026).

Table 4 Main results on LongMemEval-S (%). Results from our runs and from the literature are marked separately.

Model	Method	KU		MS		SSA		SSP		SSU		TR		Abs.		Overall	
		F1	Judge	F1	Judge	F1	Judge	F1	Judge	F1	Judge	F1	Judge	F1	Judge	F1	Judge
qwen-3.8-27b	A-Mem	44.67	80.77	13.22	61.65	79.55	98.21	**18.08**	56.67	69.60	85.71	20.73	51.88	**28.59**	**96.67**	35.74	69.20
	Mem0	62.74	76.92	38.35	58.65	15.27	16.07	8.32	46.67	76.94	87.14	33.41	60.15	5.87	93.33	41.86	60.40
	LangMem	11.57	84.62	5.46	49.62	15.04	62.50	12.57	63.33	19.12	87.14	7.97	60.15	20.41	86.67	10.49	65.40
	MemoryOS	**64.79**	87.18	47.33	72.93	**83.92**	94.64	17.09	**73.33**	**80.14**	**98.57**	33.45	56.39	26.21	90.00	53.24	76.80
	TiMem	8.85	**92.31**	5.87	73.68	19.70	80.36	13.42	60.00	23.14	92.86	9.34	80.45	21.47	86.67	11.68	81.00
	MRAgent v1	59.29	73.08	**52.55**	65.41	24.34	30.36	7.66	53.33	68.12	81.43	**44.81**	66.17	13.25	76.67	47.87	64.40
	MARR	60.28	89.74	48.12	**79.70**	80.79	**100.00**	16.16	63.33	74.55	94.29	43.01	**81.20**	25.32	66.67	**54.10**	**85.00**
deepseek-v4-flash	A-Mem[a]	43.90	67.90	13.21	39.10	73.81	87.50	15.72	33.30	61.23	88.60	20.74	21.80	**29.76**	**100.00**	33.66	51.00
	Mem0[a]	55.79	69.20	35.46	61.70	14.39	58.90	8.42	36.70	74.85	92.90	33.92	39.10	5.26	94.70	39.75	59.40
	LangMem	11.63	82.83	5.46	50.35	15.08	62.11	12.99	64.14	16.84	85.49	8.24	62.33	19.71	84.73	10.28	65.67
	MemoryOS	**59.38**	87.14	45.58	75.98	79.39	**100.00**	**17.61**	**76.55**	**83.14**	96.70	31.43	57.17	24.27	96.29	51.33	78.35
	TiMem	8.45	90.60	5.55	73.62	19.91	81.97	13.90	63.61	21.71	94.63	8.82	78.95	20.39	86.53	11.24	80.96
	MRAgent v1	59.11	73.11	**50.98**	69.82	23.98	30.07	7.32	52.62	70.41	85.97	**46.35**	64.66	13.36	76.25	48.09	65.74
	MARR	59.17	**91.03**	36.02	**82.71**	**85.02**	**100.00**	14.17	66.67	76.82	**97.14**	44.09	**82.71**	23.00	70.00	**51.66**	**87.00**

[a] Published DeepSeek-V4-Flash results are taken from Fan et al. (2026). KU: knowledge update; MS: multi-session reasoning; SSA/SSP/SSU: single-session assistant information/preferences/user information; TR: temporal reasoning; Abs.: the overlapping subset of 30 abstention questions, reported separately and not counted again in the overall results.

table. Higher overall accuracy does not yet imply better judgments of user preferences or more reliable abstention when evidence is insufficient. The abstention subset overlaps with the other question types, and its results are used separately to analyze this ability without being counted again in the overall results.

4.3 Ablation experiments

MARR's ablation experiments are divided into two groups: the Writer's initial answer construction, and subsequent supplementary retrieval and revision. This separation avoids treating results with different starting points as a single controlled

comparison. The former reruns the corresponding retrieval processes and evaluates the resulting initial answers; the latter reuses the same frozen initial answers and their states, changing only the subsequent components. The evaluation covers all 1540 LoCoMo questions and all 500 LongMemEval-S questions, without selecting questions based on known errors. The generation model for the component ablations in Tables 5 and 6(a) is qwen-3.8-27b; Table 6(b) also lists the DeepSeek Flash gated revision results on LoCoMo and the DeepSeek Flash online results with fixed memory on LongMemEval-S, with the scoring models specified separately. Completed ablations use the existing scoring records directly.

4.3.1 Effects of the Writer's active retrieval and graph navigation

To examine the contributions of active retrieval and graph navigation to initial answer construction, we compare MARR's Writer with three control configurations. **W1 Seed retrieval only** omits both rounds of active planning and generates answers directly from the initially retrieved material; **W2 Expanded fixed retrieval** performs no active planning and obtains more material through fixed question-based retrieval, with preset limits of up to 42 note nodes and 28 source-text nodes, subject to the model's context limit; **W3 No entity/topic graph navigation** disables navigation paths that depend on entity and topic adjacency, while retaining ordinary search, independent semantic retrieval, source-text reading, source citation, and date calculation. All conditions use the same memory content, generation model, and final answering prompt. Each generates its own initial answers without running the subsequent Challenger process.

In addition to F1 and Judge, we report supporting evidence recall and complete coverage to examine the relationship between answer performance and evidence acquisition. For questions with a nonempty support set S_i, evidence recall is the question-level average of $|E_i \cap S_i|/|S_i|$, and complete coverage is the proportion of questions satisfying $S_i \subseteq E_i$, where E_i denotes the set of supporting evidence actually read when generating the initial answer. LoCoMo uses message-level support annotations, whereas LongMemEval-S uses session-level support annotations, so the two are reported separately. Questions with inapplicable annotations or failed mappings are excluded from the evidence metrics; abstention questions in LongMemEval-S are also excluded from the standard supporting evidence metrics.

As shown in Table 5, compared with W1, which uses only seed retrieval, MARR's Writer improves Judge accuracy by 4.22 and 4.40 percentage points on LoCoMo and LongMemEval-S, respectively, and complete evidence coverage by 5.47 and 4.89 percentage points. The improvements in both evidence coverage and Judge scores on both datasets indicate that continued retrieval after the initial retrieval helps obtain the supporting material needed for answers. This provides experimental support for the Writer's use of subsequent retrieval to improve initial answers.

To further examine whether these gains can be obtained by expanding fixed retrieval, we compare MARR's Writer with W2. On LongMemEval-S, W2 achieves complete evidence coverage of 99.15%, above the 95.74% of MARR's Writer, yet MARR's Writer still achieves Judge and F1 scores that are higher by 5.60 and 3.05 percentage points, respectively. This indicates that, on this dataset, expanding fixed retrieval and increasing coverage of supporting sessions alone do not produce the same answer performance as active retrieval. Together with the W1 results, this suggests that the value of subsequent retrieval needs to be assessed in terms of both evidence acquisition and answer quality: session-level coverage records whether sessions containing supporting material were accessed, but does not adequately reflect whether the necessary details were fully read and used in the answer. On LoCoMo, W2 achieves a higher Judge score than MARR's Writer, showing that the benefits of active retrieval relative to expanded fixed retrieval differ between the two datasets.

When ordinary search, semantic retrieval, and source-text reading are retained, removing entity/topic graph navigation changes the Writer's initial answer performance only slightly, while complete evidence coverage also remains largely unchanged. These results provide relatively direct support for continued evidence acquisition after the initial retrieval.

Table 5 Ablations of MARR's Writer initial answer construction (%, Qwen generation group)

Configuration	LoCoMo				LongMemEval-S			
	F1	Judge	evidence recall	complete coverage	F1	Judge	evidence recall	complete coverage
MARR's Writer (Initial Writer)	43.89	84.35	91.11	85.84	52.65	81.80	98.35	95.74
W1 Seed retrieval only	43.37	80.13	86.13	80.37	53.17	77.40	96.35	90.85
W2 Expanded fixed retrieval	43.71	87.21	96.33	92.82	49.60	76.20	99.56	99.15
W3 No entity/topic graph navigation	43.30	84.09	90.99	85.52	52.90	82.20	98.45	95.74

Note: LoCoMo uses official F1, and LongMemEval-S uses Token F1. All Judge scores use the first-round scores from the existing experiments in this group and are compared only across configurations on the same dataset within the table. The LoCoMo results in this table use Qwen Judge and are reported separately from the DeepSeek Judge results in Table 6(a) and Figure 4. MARR's Writer and the ablation conditions were run on different dates, and service fluctuations were not isolated; the following analysis is a descriptive comparison.

4.3.2 Supplementary retrieval and actual retrieval trajectories

This section examines how the Challenger's supplementary retrieval and retrieval records affect subsequent review and revision after the Writer produces an initial answer in MARR. All conditions start from the same initial answers. Using Initial Writer, which outputs the initial answers directly, and MARR as references, we introduce two independent controls: **A1 Existing evidence only** prevents the Challenger from obtaining new material while retaining review of the evidence already read by the Writer and subsequent revision; **A2 No retrieval records** retains the existing evidence and the ability to continue retrieval, but removes historical action names, query parameters, and execution order from the inputs to the Challenger's supplementary retrieval and review and to the Writer's subsequent answering. The two controls examine the contributions of acquiring new evidence and providing information about the retrieval process, respectively.

Table 6 shows that Challenger review and subsequent Writer revision can further improve initial answer performance. MARR raises LoCoMo F1 from 43.89% to 56.83% and LongMemEval-S Judge accuracy from 81.80% to 85.00%. This indicates that there is still room to verify and complete answers after the Writer's initial retrieval and answering, providing experimental support for a review and revision stage after the initial answer. A1 and A2 further distinguish the roles of supplementary retrieval and retrieval records in this process.

Under A1, which uses only existing evidence, LongMemEval-S Judge accuracy still rises from the initial 81.80% to 84.40%. This shows that reexamining existing material and revising answers can correct some initial errors even when the evidence remains unchanged. When the Challenger is allowed to perform supplementary retrieval, MARR corrects 24 questions, compared with 21 for A1, while both produce 8 harmful revisions, yielding an additional Judge gain of 0.60 percentage points. With the same number of harmful revisions, supplementary retrieval is associated with more initial errors being corrected, supporting the design in which the Challenger continues to acquire evidence while reviewing existing material.

Figure 3 shows how the Challenger recovers support missing at the initial answer stage across different memories. Supporting evidence is recovered in all ten LoCoMo memories, with recovery rates ranging from 16.0% to 43.3%. In total, of the 410 question/annotated supporting-message pairs missing at the Writer stage, 125 are recovered at the Challenger stage, accounting for 30.5%. This result shows that supplementary retrieval can expand the supporting evidence visible to the model, and that this effect occurs across different memory instances. The accompanying per-question statistics further show that, among the 217 questions with incomplete supporting evidence at the Writer stage, 56 obtain all annotated support after supplementary retrieval, providing a fuller basis for subsequent verification and completion of the initial answers.

Table 6 Ablations of MARR's supplementary retrieval and retrieval records, and overall revision results.

(a) Answer quality (%, Qwen generation group).

Configuration	LoCoMo F1	LoCoMo Judge	LongMemEval-S F1	LongMemEval-S Judge
MARR	56.83	88.64	54.10	85.00
Subsequent regulation process removed (Initial Writer)	43.89	87.92	52.65	81.80
A1 Existing evidence only	56.25	87.21	54.05	84.40
A2 No retrieval records	56.76	88.44	53.50	84.00

(b) Revision results relative to the same initial answers.

Dataset and model	Configuration	C	R	Net gain	Judge Change (Percentage Points)
LoCoMo (DeepSeek)	MARR (gated revision)	12	8	+4	+0.26
LongMemEval-S (Qwen)	MARR	24	8	+16	+3.20
LongMemEval-S (DeepSeek)	MARR (fixed memory)	12	6	+6	+1.20
LongMemEval-S (Qwen)	A1 Existing evidence only	21	8	+13	+2.60
LongMemEval-S (Qwen)	A2 No retrieval records	17	6	+11	+2.20

Note: LoCoMo uses official F1, and LongMemEval-S uses Token F1. All LoCoMo configurations in Table 6(a) and Figure 4 use scores for frozen answers under a unified DeepSeek Flash Judge protocol, reusing verified historical scores; the Qwen group on LongMemEval-S uses Qwen Judge. C and R denote the numbers of questions with an incorrect initial answer and a correct final answer, and with a correct initial answer and an incorrect final answer, respectively. The net gain is $C - R$, and all are calculated from paired judgments by the same Judge.

With supplementary retrieval retained, A2 further examines how retrieval records help subsequent processing. On LongMemEval-S, MARR's Judge and F1 scores exceed A2 by 1.00 and 0.60 percentage points, respectively; the number of corrections increases from 17 to 24, while harmful revisions increase from 6 to 8, yielding a net gain of 5 correctly answered questions. When retrieval records are provided, the increase in corrections exceeds the increase in harmful revisions, producing a higher net benefit from revision. Retrieval records contain previously attempted queries, executed operations, and their order, giving the subsequent model context for understanding how the existing material was obtained. These results support using this process information together with the initial answer and already-read evidence as the basis for review and revision.

Results by question type further show that supplementary retrieval and retrieval records contribute differently across types. As shown in Figure 4, MARR gains 5.67 percentage points over A1 on multi-hop reasoning, and also shows a positive difference of 1.42 percentage points over A2, both larger than for the other question types. The proportions of unchanged judgments for multi-hop questions are 91.5% and 93.6%, respectively, compared with 96.9% to 99.0% for the other types, indicating that multi-hop answers are more sensitive to changes in these two components. This is consistent with the need to connect information from multiple locations in multi-hop questions: supplementary retrieval can expand the available evidence, while retrieval records provide context about the prior search process for subsequent processing. For temporal reasoning, MARR exceeds A1 by 1.56 percentage points and matches A2; for single-hop question answering, the net change relative to either ablation is zero. For open-domain question answering, MARR has a net gain of 1 correctly answered question over A1 and a net loss of 1 relative to A2. The component benefits in these ablations are concentrated in multi-hop reasoning; comparisons by question type thus reveal effects of the method that are not fully reflected in the overall average scores.

The evidence stratification at the bottom of Figure 4 further explains the benefits of supplementary retrieval on multi-hop questions. MARR's Judge accuracy exceeds A1 by 9.85 percentage points in the subset with incomplete Writer evidence and by 2.03 percentage points in the subset

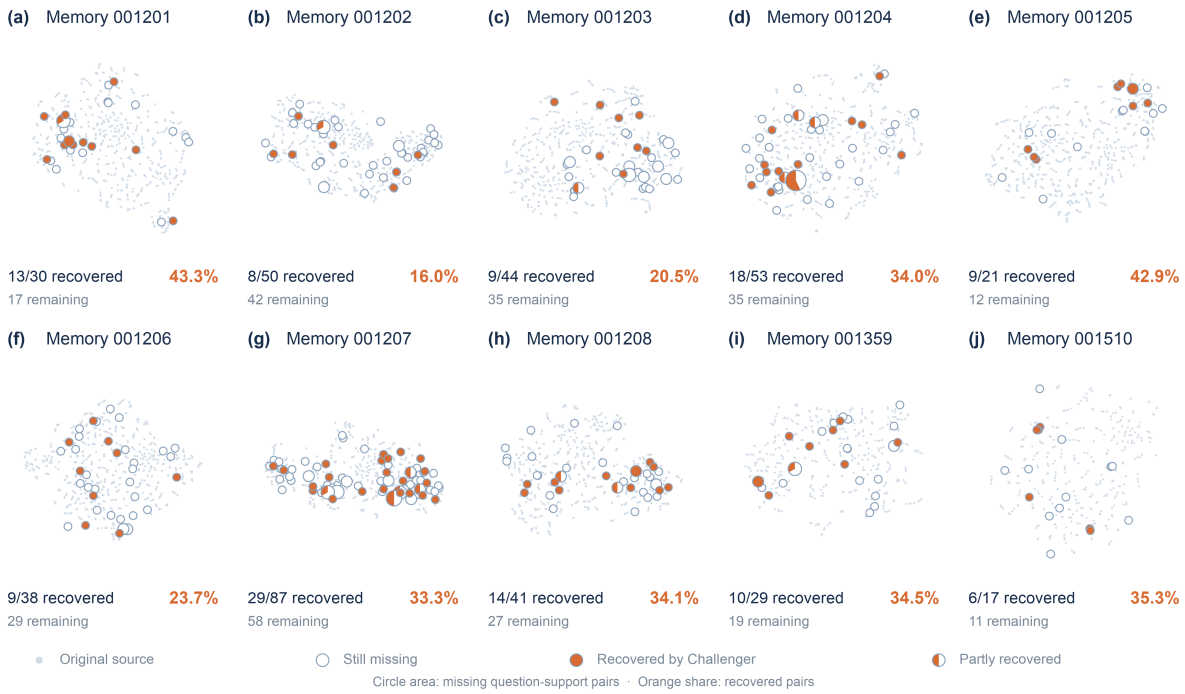


Fig. 3 Distribution of support missing from the Writer and recovered through MARR's supplementary retrieval. The ten panels correspond to the ten LoCoMo memories. The light gray-blue dots are fixed embedding projections of source texts, with dimensionality reduction performed independently for each memory; outlined circles indicate annotated supporting sources still missing for at least one question after the Writer stage, with circle area proportional to the number of question/supporting-message pairs involved. Orange sectors indicate the proportion recovered by the Challenger, and white sectors indicate the proportion still uncovered; the same source may have different outcomes across questions. Below each panel, recovered and remaining indicate the recovered count/original missing count and the count still missing, respectively, while the orange percentage indicates the recovery rate. Coverage status is reconstructed from the source texts cumulatively visible in the actual model requests. The evidence analysis includes 1533 questions with valid support annotations, excluding 4 questions without support annotations and 3 with failed support mappings; question-answering evaluation still covers all 1540 questions. Support coverage does not imply answer correctness.

with complete evidence, indicating that the benefits of supplementary retrieval on multi-hop questions are more pronounced when not all annotated support has been obtained at the initial answer stage. MARR's gains over A2 are −0.68 and 3.79 percentage points in the complete and incomplete evidence groups, respectively, likewise showing a larger positive difference in the incomplete evidence subset. For these multi-hop questions, the benefits of both supplementary retrieval and retrieval trajectories are more pronounced when the initial evidence is incomplete.

4.4 Revision gains and risks of harmful revision

MARR can correct additional errors even when the initial answers already have high accuracy. In the Qwen group on LongMemEval-S, 26.37% of initially incorrect answers are corrected, while 98.04% of initially correct answers remain correct, producing a net accuracy gain of 3.20 percentage points. This shows that subsequent review can resolve some questions that were not answered correctly on the first attempt while preserving the vast majority of already correct answers.

Corrections in both LongMemEval-S experimental groups are concentrated in three question types: multi-session reasoning, temporal reasoning, and knowledge update. These tasks require finding relevant content as well as connecting dispersed clues, clarifying temporal relationships, or distinguishing old and new states. The concentration of corrections is consistent with these tasks' demands for evidence integration, indicating that subsequent

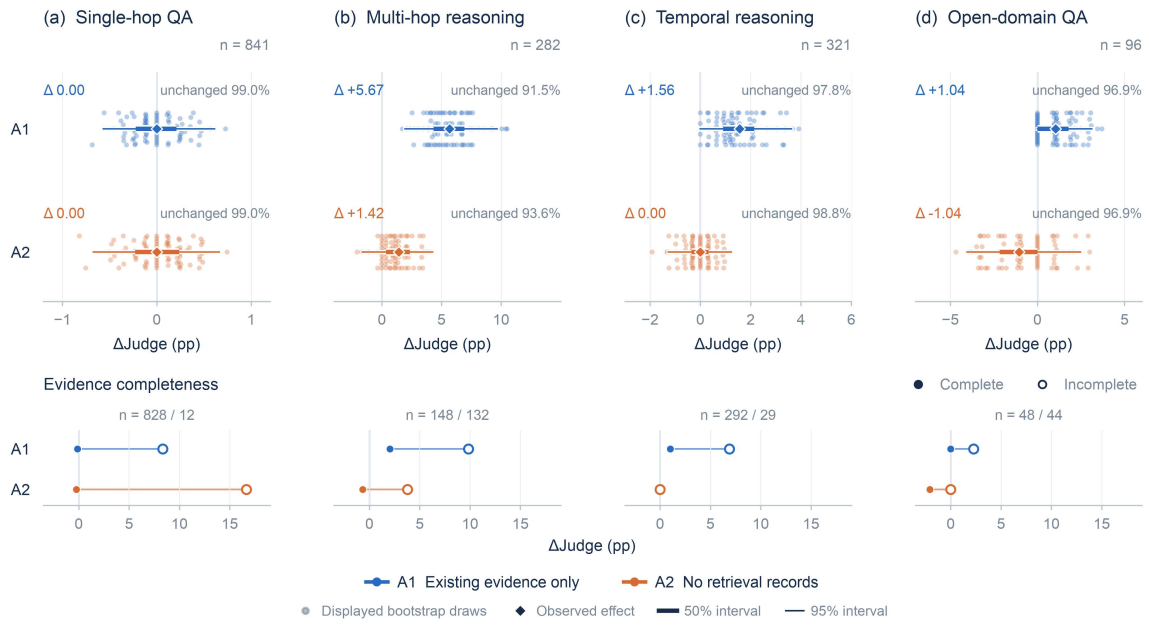


Fig. 4 MARR's ablation effects by question type on LoCoMo (DeepSeek Judge). A1 uses existing evidence only, and A2 provides no retrieval records, shown in blue and orange, respectively. Net differences are calculated from paired results for MARR and the corresponding ablation configuration under a unified DeepSeek Flash Judge protocol, consistent with the LoCoMo Judge evaluation in Table 6(a), and are measured in percentage points. Each row displays 80 randomly selected conversation-level paired resamples. Thick bars and thin lines show the central 50% and 95% intervals calculated from all 2000 resamples, respectively, and diamonds mark the original net differences. "unchanged" denotes the proportion of questions receiving the same Judge decision for MARR and the corresponding ablation configuration. The four main panels use different horizontal-axis ranges; the lower panels use a common scale to compare questions with complete and incomplete Writer evidence, shown as filled and open dots, respectively, with question counts listed in the same order. Evidence stratification includes 1533 questions; the remaining 7 questions are retained in the analysis above.

supplementary retrieval and review have a more pronounced effect on questions requiring information integration and judgments about relationships among evidence.

The net benefit also depends on preserving correct initial answers. In the Qwen group, 8 harmful revisions offset one-third of the gain from 24 corrections, showing that even a small number of harmful revisions can substantially reduce the benefit when most initial answers are correct.

Limitations. MARR's final revision can still introduce harmful revisions. More detailed evidence consistency checks and confidence calibration could improve decisions about accepting proposed changes. The Writer and Challenger currently use preset retrieval budget limits; future work could allocate budgets dynamically according to question complexity and the actual retrieval state to balance answer quality and call costs. The current framework also performs question answering over a preconstructed read-only memory graph. For continued interaction, it could be extended with incremental memory updates and mechanisms for maintaining outdated information.

5 Conclusion

This paper proposes MARR, a metamemory-guided active retrieval and memory revision framework, to address the problem that initial answers in long-term conversational question answering may have factual support yet omit necessary information. Built on a heterogeneous memory graph that preserves source and temporal information, the framework treats the initial answer, already-read evidence, and actual retrieval trajectory together as an observable recall state. This state guides the Challenger to perform supplementary retrieval and provide review feedback, after which the Writer checks the feedback against source texts to fill

omissions, correct errors, and preserve content that is already correct. Evaluation on LoCoMo and LongMemEval-S provides experimental support for review and revision after the initial answer, with the Qwen configuration on LongMemEval-S improving Judge accuracy from 81.80% to 85.00%. Ablations and evidence analysis further show that reexamining existing material can correct some initial errors, while supplementary retrieval helps recover missing support, with more pronounced benefits on multi-hop questions whose initial evidence is incomplete. These findings indicate that effective use of long-term conversational memory requires attention to both evidence acquisition and its interpretation and integration in answers.

Future work will further investigate decision mechanisms for supplementary retrieval and revision to improve the system's judgments about when to continue retrieval, when to revise, and when to retain the initial answer.

References

Asai A, Wu Z, Wang Y, et al (2024) Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In: International Conference on Learning Representations, URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/25f7be9694d7b32d5cc670927b8091e1-Abstract-Conference.html

Chhikara P, Khant D, Aryan S, et al (2025) Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. arXiv preprint arXiv:2504.19413, URL https://arxiv.org/abs/2504.19413, preprint, arXiv:2504.19413

Cormack GV, Clarke CLA, Buettcher S (2009) Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 758–759, https://doi.org/10.1145/1571941.1572114, URL https://doi.org/10.1145/1571941.1572114

Fan X, Liu M, Yang R, et al (2026) Can Agent Memory Systems Track Evolving State? arXiv preprint arXiv:2608.19652v1, https://doi.org/10.48550/arXiv.2608.19652, URL https://arxiv.org/abs/2608.19652v1, preprint; cited version 1, arXiv:2608.19652v1

Gao L, Dai Z, Pasupat P, et al (2023) RARR: Researching and Revising What Language Models Say, Using Language Models. In: Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), URL https://aclanthology.org/2023.acl-long.910/

Gou Z, Shao Z, Gong Y, et al (2024) CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing. In: International Conference on Learning Representations, URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/fef126561bbf9d4467dbb8d27334b8fe-Abstract-Conference.html

Ji S, Li Y, Hooi B (2026) Memory is Reconstructed, Not Retrieved: Graph Memory for LLM Agents. arXiv preprint arXiv:2606.06036v1, URL https://arxiv.org/abs/2606.06036v1, accepted at ICML 2026; cited public version: arXiv:2606.06036v1, arXiv:2606.06036v1

Jiang EH, Zhang Z, Wu Y, et al (2026) Memory as a Controlled Process: Learned Adaptive Memory Management for LLM Agents. arXiv preprint arXiv:2607.13591, URL https://arxiv.org/abs/2607.13591, preprint, arXiv:2607.13591

Jiménez Gutiérrez B, Shu Y, Gu Y, et al (2024) HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models. In: Advances in Neural Information Processing Systems, URL https://proceedings.nips.cc/paper_files/paper/2024/hash/6ddc001d07ca4f319af96a3024f6dbd1-Abstract-Conference.html

Jiménez Gutiérrez B, Shu Y, Qi W, et al (2025) From RAG to Memory: Non-Parametric Continual Learning for Large Language Models. In: International Conference on Machine Learning, URL https://proceedings.mlr.press/v267/gutierrez25a.html

Kang J, Ji M, Zhao Z, et al (2025) Memory OS of AI Agent. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, URL https://aclanthology.org/2025.emnlp-main.1318/

Koriat A, Goldsmith M (1996) Monitoring and Control Processes in the Strategic Regulation of Memory Accuracy. Psychological Review 103(3):490–517. https://doi.org/10.1037/0033-295X.103.3.490, URL https://cris.haifa.ac.il/en/publications/monitoring-and-control-processes-in-the-strategic-regulation-of-m/

LangChain Team (2025) LangMem SDK for agent long-term memory. Official software release announcement, URL https://www.langchain.com/blog/langmem-sdk-launch, project documentation: https://langchain-ai.github.io/langmem/

Li K, Yu X, Ni Z, et al (2026) TiMem: Temporal-Hierarchical Memory Consolidation for Long-Horizon Conversational Agents. In: Findings of the Association for Computational Linguistics: ACL 2026, pp 21700–21720, https://doi.org/10.18653/v1/2026.findings-acl.1091, URL https://aclanthology.org/2026.findings-acl.1091/

Madaan A, Tandon N, Gupta P, et al (2023) Self-Refine: Iterative Refinement with Self-Feedback. In: Advances in Neural Information Processing Systems, URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html

Maharana A, Lee DH, Tulyakov S, et al (2024) 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, https://doi.org/10.18653/v1/2024.acl-long.747, URL https://aclanthology.org/2024.acl-long.747/

Nelson TO, Narens L (1990) Metamemory: A Theoretical Framework and New Findings. In: Psychology of Learning and Motivation, vol 26. Academic Press, p 125–173, https://doi.org/10.1016/S0079-7421(08)60053-5, URL https://sites.socsci.uci.edu/~lnarens/1990/Nelson%26Narens_Book_Chapter_1990.pdf

Packer C, Wooders S, Lin K, et al (2023) MemGPT: Towards LLMs as Operating Systems. arXiv preprint arXiv:2310.08560, URL https://arxiv.org/abs/2310.08560, preprint; version 2 revised in February 2024, arXiv:2310.08560

Park JS, O'Brien J, Cai CJ, et al (2023) Generative Agents: Interactive Simulacra of Human Behavior. In: Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, pp 1–22, https://doi.org/10.1145/3586183.3606763, URL https://doi.org/10.1145/3586183.3606763

Robertson S, Zaragoza H (2009) The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends in Information Retrieval https://doi.org/10.1561/1500000019, URL https://doi.org/10.1561/1500000019

Salama R, Cai J, Yuan M, et al (2025) MemInsight: Autonomous Memory Augmentation for LLM Agents. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, URL https://aclanthology.org/2025.emnlp-main.1683/

Shinn N, Cassano F, Gopinath A, et al (2023) Reflexion: Language Agents with Verbal Reinforcement Learning. In: Advances in Neural Information Processing Systems, URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html

Trivedi H, Balasubramanian N, Khot T, et al (2023) Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions. In: Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), URL https://aclanthology.org/2023.acl-long.557/

Wu D, Wang H, Yu W, et al (2025) LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. In: International Conference on Learning Representations, URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/d813d324dbf0598bbdc9c8e79740ed01-Abstract-Conference.html

Wu S, Zhu H, Zhang Y, et al (2026) AutoMem: Automated Learning of Memory as a Cognitive Skill. arXiv preprint arXiv:2607.01224, URL https://arxiv.org/abs/2607.01224, preprint, arXiv:2607.01224

Xiao S, Liu Z, Zhang P, et al (2024) C-Pack: Packed Resources for General Chinese Embeddings. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 641–649, https://doi.org/10.1145/3626772.3657878, URL https://doi.org/10.1145/3626772.3657878

Xu W, Liang Z, Mei K, et al (2025) A-Mem: Agentic Memory for LLM Agents. In: Advances in Neural Information Processing Systems, URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/19909c36f51abc4856b4560aff3d36d6-Abstract-Conference.html

Yao S, Zhao J, Yu D, et al (2023) ReAct: Synergizing Reasoning and Acting in Language Models.

In: International Conference on Learning Representations, URL https://openreview.net/forum?id=WE_vluYUL-X