UNFOLDMEM: UNFOLDING MULTIMODAL CONVERSATIONAL MEMORY VIA ITERATIVE MULTI-AGENT INTERACTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term memory has become an increasingly important component of modern AI systems, enabling agents to leverage information from past interactions. More recent studies have shifted from passive retrieval toward active memory navigation, enabling agents to iteratively access relevant information from long-term memory. However, these approaches conduct both memory retrieval and evidence interpretation within the shared context, which entangles retrieval process with reasoning towards answer, making focused evidence interpretation increasingly difficult. To address this limitation, we propose UnfoldMem, a multi-agent framework that decouples memory exploration from evidence interpretation for multimodal conversational memory. Given a user query, the main agent progressively explores memory by selecting memory access tools and formulates focused sub-questions. Each sub-question is answered by a temporary sub-agent over question-specific context without further exploration, and its response guides main agent's subsequent memory access and final answer generation. Experiments on MemEye and Mem-Gallery demonstrate that UnfoldMem achieves state-of-the-art performance on multimodal long-term memory question answering. Extensive experiments and ablations further validate the effectiveness of each component of our framework and demonstrate its adaptive tool-use behavior across different questions.

1 INTRODUCTION

Long-term memory has become a key component of conversational and autonomous agents (Hu et al., 2025; Huang et al., 2026; Mei et al., 2025). By retaining information beyond the current context window, memory allows an agent to build on earlier interactions, maintain user-specific knowledge, and carry out tasks that span multiple sessions. As agent interactions become increasingly multimodal, remembering text alone is no longer sufficient (Liu et al., 2026b; Kang et al., 2025; Chhikara et al., 2025; Park et al., 2023). User histories may contain photographs, screenshots, and documents whose details cannot be fully preserved in textual summaries. This has motivated multimodal memory systems (Liu et al., 2026a; Long et al., 2026) and benchmarks (Xue et al., 2025; Liu et al., 2024) that require agents to recover and reason over the original visual evidence.

Many existing work addresses this challenge by improving memory retrieval (Gutiérrez et al., 2024; Chen et al., 2022). Past interactions are organized through embeddings, summaries, hierarchies, or relational structures so that relevant records can be retrieved for a query, as illustrated in Figure 1-(a). More recently, memory utilization has shifted from passive retrieval toward active navigation (Tian et al., 2026; Wang & Chen, 2025). As shown in Figure 1-(b), rather than issuing a single search, an agent can inspect intermediate results, reformulate its request, and access memory repeatedly until it gathers sufficient evidence to the question.

However, previous studies (Bei et al., 2026; Guo et al., 2026) have revealed that this paradigm has no advantage over RAG-based methods in most cases. We think that this is because existing active navigation methods conduct both memory retrieval and evidence interpretation within the shared context. This entangles retrieval process with reasoning towards answer, making focused evidence interpretation increasingly difficult.

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.

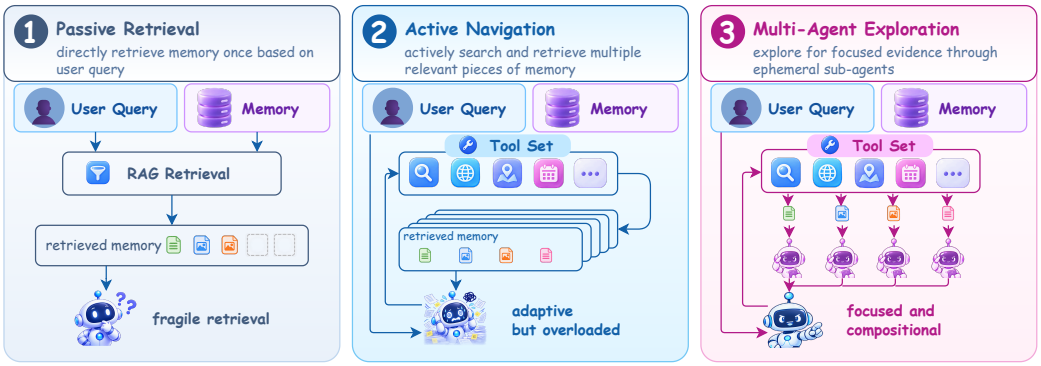

Figure 1: Comparison of three paradigms for multimodal long-term memory. Passive retrieval performs a single query-conditioned lookup and feeds the retrieved content directly to the answering model, making it vulnerable when relevant evidence is distributed across sessions or depends on context missing from the retrieved snippets. Active navigation iteratively searches memory to broaden coverage, but the same agent must plan the search and interpret an expanding collection of raw evidence within one context. Multi-agent exploration separates these roles: a main agent explores through targeted sub-questions, while temporary sub-agents interpret bounded, question-specific multimodal contexts and return intermediate results for composition.

We propose **UnfoldMem**, a multi-agent framework that separates memory exploration from evidence interpretation. The main agent retains a compact overview of the memory and determines what information should be examined next. For each step, it selects an appropriate memory-access tool to expose a relevant memory region and formulates a focused sub-question. UnfoldMem then instantiates a temporary sub-agent with a question-specific context containing the selected conversation rounds and their original images. The sub-agent examines this bounded context, returns its answer to the main agent, and is discarded. The main agent can subsequently ask a different question over a different context, using previous answers to guide its exploration. In this way, memory is progressively *unfolded* through targeted interactions rather than accumulated as raw content in a single agent window.

We evaluate UnfoldMem on two multimodal long-term memory benchmarks, *MemEye* (Guo et al., 2026) and *Mem-Gallery* (Bei et al., 2026). UnfoldMem achieves state-of-the-art performance on both benchmarks. Further ablations and behavioral analyses demonstrate the complementary roles of the memory overview, session-level unfolding, and fine-grained retrieval, and show how the framework adaptively invokes its tools.

Our contributions are threefold:

- We introduce UnfoldMem, a multi-agent framework for long-term memory that progressively exposes multimodal memory through focused, question-specific sub-agent contexts.
- We demonstrate state-of-the-art performance on multimodal memory benchmarks and systematically validate the contribution of each component.
- Through systematic ablations and behavioral analyses, we demonstrate that UnfoldMem exhibits adaptive tool-use behavior across questions.

2 Related Work

2.1 Memory Construction

Studies of memory construction started from systems (Lewis et al., 2020; Zhong et al., 2024; Chhikara et al., 2025) which represent experience or conversations as single-layer independently retrievable records, such as conversation chunks, facts, or memory notes. Building upon this, hierarchical approaches introduce multiple levels of abstraction or storage to support coarse-to-fine access: MemGPT (Packer et al., 2023) uses an operating-system-inspired virtual memory

hierarchy, while MemoryOS (Kang et al., 2025) and MemOS (Li et al., 2025) organize memories across distinct short- and long-term tiers. The same idea has been extended to multimodal memory: M3-Agent (Long et al., 2026) constructs modality-specific perceptual records, while MIRIX (Wang & Chen, 2025) partitions heterogeneous experience into specialized memory types. Another line of work organizes memory by explicitly modeling relations among memory units. HippoRAG (Gutiérrez et al., 2024) constructs an entity-centered graph for associative retrieval, A-MEM (Xu et al., 2025) links evolving memory notes, and M2A (Feng et al., 2026) maintains links between semantic records and raw evidence. However, these structures usually predefine which attributes and relations are salient before the query is known. They may therefore fail when a query depends on information or relations that were not captured during memory construction.

2.2 Memory Utilization

Memory utilization has evolved from passive retrieval to increasingly active memory exploration. Early memory-augmented systems (Lewis et al., 2020; Zhong et al., 2024; Chhikara et al., 2025) directly retrieve relevant records using the user query and inject the retrieved context into the generator within single step. More recent approaches improve retrieval by exploiting the structure of the constructed memory: HippoRAG (Gutiérrez et al., 2024) performs graph-based propagation to associate query-relevant entities with supporting passages, while SimpleMem (Liu et al., 2026b) uses intent-aware retrieval planning to adapt the retrieval scope and combine multiple retrieval views. Beyond fixed retrieval pipelines, agentic approaches allow the system to actively control memory access. MemOS (Li et al., 2025) schedules and activates heterogeneous memory resources according to the task. MIRIX (Wang & Chen, 2025) allows its chat agent to select memory types, search fields, and retrieval strategies for targeted retrieval. M2A (Feng et al., 2026) equips a chat agent with explicit memory query and update operations, and M3-Agent (Long et al., 2026) performs multi-turn reasoning with iterative memory access, repeatedly retrieving information as the reasoning process unfolds. Despite this progression toward active memory access, existing agentic approaches couple memory exploration with interpretation within the same agent context, causing retrieved content and intermediate reasoning to accumulate as exploration proceeds. This coupling entangles subsequent exploration with previously retrieved evidence, forcing the agent to carry potentially irrelevant content even when seeking new information or answering.

3 UnfoldMem

3.1 Problem Formulation

We consider a multimodal long-term conversational history $\mathcal{H} = \{\mathcal{S}_i\}_{i=1}^{N}$ consisting of N sessions. Each session $\mathcal{S}_i = \{x_{i,j}\}_{j=1}^{M_i}$ is an ordered sequence of M_i user–assistant dialogue rounds, where each round $x_{i,j} = (u_{i,j}, a_{i,j}, I_{i,j}, t_{i,j})$ contains the user input $u_{i,j}$, the assistant response $a_{i,j}$, a possibly empty set of associated images $I_{i,j}$, and the corresponding timestamp $t_{i,j}$.

A typical multimodal long-term memory system consists of two stages. In *memory construction* stage, history $\mathcal{H}$ is processed into a memory representation $\mathcal{M} = \mathrm{Build}(\mathcal{H})$. In *memory utilization* stage, given a query q, the system utilizes the constructed memory to retrieve and reason over relevant information and generates a response $a = \mathrm{Query}(q; M)$.

3.2 Overview

Following the formulation in sec 3.1, UnfoldMem consists of two major components, as illustrated in Figure 2. For *memory construction*, we first construct an explorable memory from the multimodal conversational history by generating session-level summaries and extracting round-level textual and visual representations. During *memory utilization*, UnfoldMem enables the main agent to actively explore the constructed memory by orchestrating iterative interactions with temporary sub-agents. At each interaction, the main agent determines what information to seek and selects one of three memory access tools, which differ in how they retrieve and expose relevant context to the sub-agent. The sub-agent inspects the exposed memory and returns focused evidence, which the main agent integrates to guide its subsequent exploration until sufficient information is gathered to answer the query. We detail these two components in the following sections.

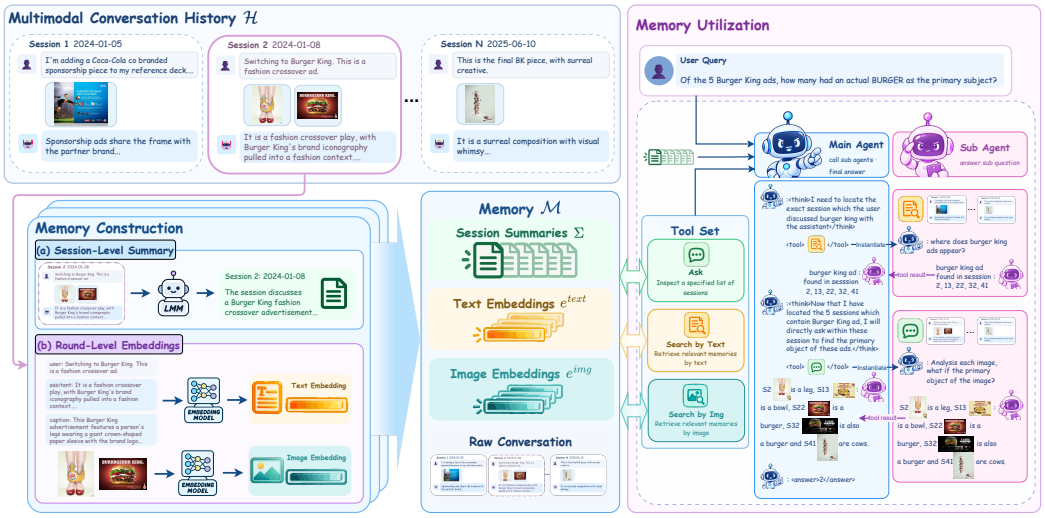

Figure 2: Overview of UnfoldMem. On the left, the memory construction stage transforms the multimodal conversational history $\mathcal{H}$ into an explorable memory $\mathcal{M}$ by generating session-level summaries and extracting round-level textual and visual representations, while preserving the original history for on-demand unfolding. On the right, the memory utilization stage enables a main agent to iteratively explore $\mathcal{M}$ through temporary sub-agents by calling three memory access tools: *ask*, *search by text*, and *search by img*. Each tool exposes a bounded portion of memory to a sub-agent, whose focused evidence is returned to the main agent to guide further exploration and final answer generation.

3.3 Memory Construction

Since future queries may require arbitrary information from the history $\mathcal{H}$, we avoid task-specific processing during memory construction. Instead, we construct two lightweight, general-purpose components that serve as reusable entry points for the main agent to explore the memory: a session-level summary for locating potentially relevant sessions, and round-level embeddings for retrieving fine-grained information. The original history $\mathcal{H}$ is preserved alongside these representations, so retrieved content can be unfolded on demand during utilization stage.

Round-level Embeddings. To enable fine-grained retrieval within the history $\mathcal{H}$, we encode each dialogue round into textual and visual embeddings. Specifically, images are first encoded into visual embeddings, and then converted into captions to generate text embeddings with the original dialogue. This process is defined as:

$$e^{\text{text}}_{i,j} = \text{Enc}_{\text{text}}\left(x^{\text{text}}_{i,j} \oplus \text{Caption}(I_{i,j})\right), \qquad e^{\text{img}}_{i,j} = \text{Enc}_{\text{img}}(I_{i,j}), \tag{1}$$

where $\oplus$ denotes concatenation. The resulting embeddings provide two retrieval modalities $\Theta = (\Theta^{\text{text}}, \Theta^{\text{img}})$, which allow the main agent to efficiently retrieve rounds relevant to either textual or visual information.

Session-level Summaries. We use an LMM to summarize each session $\mathcal{S}_i$ into a concise textual summary S_i:

$$S_i = \text{LMM}(\mathcal{S}i). \tag{2}$$

We organize these summaries into a session index $\Sigma = [(name_i, date_i, S_i)]_{i=1}^{N}$, which provides the main agent with a compact overview of the history and helps identify sessions that may contain information relevant to the current query.

3.4 Memory Utilization

As shown on the right part of Figure 2, given a user query q, UnfoldMem explores memory through iterative interactions between a main agent and temporary sub-agents. The main agent orchestrates

memory exploration, while each sub-agent inspects a bounded portion of memory exposed by one of three memory access tools and returns focused evidence to the main agent.

3.4.1 AGENT ROLE DEFINITION

The main agent maintains the user query q, the session-level memory index Σ, the interaction history, and determines whether additional memory evidence is needed. At step t, its policy can be defined as:

$$a_t \sim \pi_\theta^{\text{main}}\left(\cdot \mid \Sigma, q, a_{<t}, o_{<t}\right), \tag{3}$$

where $a_{<t}$ and $o_{<t}$ denote the actions taken and observations received in previous steps, respectively. a_t denotes the next action, which is either a memory access tool call or the final answer.

Whenever the main agent selects a memory access tool, it initiates a sub-agent and formulates a focused sub-question q'_t to specify the information it seeks. The selected tool determines the memory content $\mathcal{X}_t$ exposed to the sub-agent. Sub-agent's policy can be defined as:

$$y_t \sim \pi_\theta^{\text{sub}}\left(\cdot \mid q'_t, \mathcal{X}_t\right), \tag{4}$$

where y_t denotes the sub-agent's answer to q'_t. The sub-agent has access only to q'_t and the exposed memory $\mathcal{X}_t$. Thus, its role is therefore restricted to inspecting the provided memory content and extracting the evidence relevant to the sub-question.

This separation establishes a clear division of responsibilities: the main agent decides whether and what to explore, while the sub-agent inspects the exposed memory and extracts the evidence relevant to the sub-question.

3.4.2 MEMORY ACCESS TOOL DESIGN

Memory is exposed to the sub-agent through three complementary tool calls: *ask*, *search by text*, and *search by img*. Depending on the tool selected by the main agent, different portions of the constructed memory are materialized as $\mathcal{X}_t$, providing either session-level or targeted round-level access.

ask takes one or more session tags and a sub-question as input, and dispatches a sub-agent over the complete contents of the selected sessions, unfolding them into their underlying dialogue rounds in chronological order. This provides session-level access when the relevant sessions can be identified from Σ.

search by text and *search by img* take textual and image queries and retrieve the top-K most relevant rounds using Θ^{text} and Θ^{img}, respectively. For both tools, retrieved rounds are materialized together with their associated textual and visual content as $\mathcal{X}_t$ provided to a sub-agent.

These tools provide complementary access to the constructed memory: *ask* unfolds selected sessions in full, while *search by text* and *search by img* provide targeted round-level retrieval. The main agent can freely switch among these access modes throughout the exploration process.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate UnfoldMem on two multimodal agent-memory benchmarks, MemEye(Guo et al., 2026) and Mem-Gallery(Bei et al., 2026). MemEye focuses on preserving and utilizing fine-grained visual evidence, with questions organized by visual-evidence granularity (X_1–X_4, from scene-level to pixel-level) and memory-reasoning depth (Y_1–Y_3, from single-fact retrieval to reasoning over multiple evolving facts), and evaluated in both multiple-choice (MCQ) and open-ended (Open) formats. Mem-Gallery evaluates long-horizon conversational memory through nine tasks spanning three ability domains: extraction and adaptation, reasoning, and knowledge management.

Baselines. We compare against two groups of memory baselines. **RAG** includes Semantic-RAG(Guo et al., 2026) and UniversalRAG(Yeo et al., 2026). **Agentic memory** includes A-Mem(Xu et al., 2025), M2A(Feng et al., 2026), MIRIX(Wang & Chen, 2025), and MMA(Lu et al., 2026). We

additionally report an oracle **clue-only** setting, where human-annotated evidence rounds are directly provided to the model without retrieval, as an upper-bound reference.

Evaluation Protocol. Following MemEye (Guo et al., 2026), we evaluate answer quality using the metric appropriate to each setting. For MemEye-MCQ, we report rotation exact matching (EM), computed by extracting the predicted option under each rotation of the answer choices and averaging exact match across rotations. For open-ended questions, MemEye-Open and Mem-Gallery, we use LLM-as-a-judge and ground scores into $\{0, 0.5, 1\}$. We additionally evaluate retrieval quality based on clue rounds annotated for each question. Let G_q denote the set of clue rounds for question q and R_q denote the set of rounds retrieved by a method. We compute retrieval recall and precision as follow:

$$\text{Recall} = \frac{|R_q \cap G_q|}{|G_q|}, \qquad \text{Precision} = \frac{|R_q \cap G_q|}{|R_q|}, \tag{5}$$

Implementation Details. All methods use Qwen3.5-9B (Qwen Team, 2026) as the generation backbone and Qwen3-VL-Embedding-2B (Li et al., 2026) as the retrieval encoder for both text and image embeddings. Unless otherwise specified, experiments follow the configurations in Appendices A.1 and A.2, with top-K=10 retrieved items per step and a maximum of 5 retrieval steps for adaptive methods. For open-ended evaluation, we use GPT-5.2 at temperature 0 as the judge model, whose alignment with human has been validated in MemEye. The judge follows the rubric shared by both benchmarks and returns a score with a short rationale in JSON format, which is detailed in Appendix A.4.

4.2 Main Results

UnfoldMem consistently outperforms all reported baselines across both benchmarks and three evaluation settings. As shown in Table 1, UnfoldMem achieves overall accuracy of 73.6 and 63.5 under the MemEye-MCQ and MemEye-Open setting, which exceeds strongest baselines Semantic-RAG* and Semantic-RAG by 7.2 and 5.3, respectively. Table 2 shows that UnfoldMem also obtains the highest overall score on Mem-Gallery, reaching 86.4 and outperforming the strongest baseline MiRix by 5.6. UnfoldMem does not lag substantially behind the clue-only reference, where human-annotated evidence rounds are directly provided to the model and memory retrieval is bypassed. As shown in Tables 1 and 2, UnfoldMem even surpasses the clue-only reference on several sub-dimensions and tasks, including MemEye-MCQ's X_1, Mem-Gallery's temporal reasoning (TR), and conflict detection (CD). These results highlight the potential of query-adaptive, on-demand memory exploration to achieve performance comparable to that of a model given direct access to the relevant evidence.

Findings 1: UnfoldMem improves the use of visual evidence at both scene-level and pixel-level granularity. As shown in Table 1, UnfoldMem outperforms all baselines at all four visual-granularity levels in both MCQ and Open settings. The gains are particularly large at two endpoints. At X_1, UnfoldMem improves over the strongest baseline by 13.5 and 9.4 on MCQ and Open. At X_4 the corresponding improvements are 12.9 and 6.4. The same advantage is corroborated by Mem-Gallery results in Table 2. On visually grounded subtasks like visual search (VS) and visual reasoning (VR), UnfoldMem outperforms the strongest baselines by 6.4 and 15.8. These results suggest that by combining compact session-level summaries with iterative memory unfolding design, UnfoldMem can consistently recover visual information from memory across multiple levels of granularity.

Findings 2: UnfoldMem enhances compositional reasoning over distributed and evolving memory. As shown in Table 1, UnfoldMem's advantage becomes most pronounced at Y_2 and Y_3. At Y_2, UnfoldMem outperforms the strongest reported baselines by 12.5 and 4.9 points on MCQ and Open. At Y_3, the corresponding gains are 4.1 and 5.2. This pattern is also reflected in Table 2. UnfoldMem improves its accuracy on temporal reasoning (TR), visual reasoning (VR), and multi-entity reasoning (MR) by 11.0, 15.8, and 6.8 respectively. It further leads by 14.9 on conflict detection (CD), which requires integrating and reconciling information across memory entries. These results suggest that UnfoldMem is particularly effective when reasoning over distributed or conflict evidences.

Table 1: MemEye results on MCQ (exact-matching accuracy) and Open (LLM-judge score), broken down by visual-evidence granularity (X_1–X_4) and reasoning depth (Y_1–Y_3). * denotes the chain-of-thought (CoT) variant. **Bold** and underlined values indicate the best and second-best non-oracle scores.

	X_1		X_2		X_3		X_4		Y_1		Y_2		Y_3		Total	
Method	MC	Op.	MC	Op.	MC	Op.	MC	Op.	MC	Op.	MC	Op.	MC	Op.	MC	Op.
Oracle																
clue-only	79.2	80.2	83.8	66.7	84.3	73.2	63.4	58.7	90.0	79.6	69.2	64.5	70.3	54.2	76.1	68.1
clue-only*	88.5	77.1	85.9	70.8	84.9	71.5	73.3	68.4	87.3	78.3	78.8	66.3	77.1	72.9	81.3	77.1
RAG																
Semantic-RAG	67.2	69.8	76.0	55.2	74.6	65.5	46.8	47.4	83.5	75.8	53.3	52.2	59.4	39.6	63.9	58.2
Semantic-RAG*	71.9	65.6	78.1	56.3	78.5	59.9	47.4	46.2	**87.1**	70.4	56.5	48.0	56.8	47.9	66.4	55.3
UniversalRAG	61.0	64.6	67.7	58.3	70.2	62.3	46.4	48.5	80.8	75.8	52.0	50.2	43.2	39.6	60.2	57.1
Agentic Memory																
A-Mem	71.4	61.5	68.2	58.3	72.0	57.4	43.4	44.4	83.1	72.5	49.6	45.8	55.2	37.5	61.2	53.4
M2A	56.8	30.2	46.4	25.0	50.1	27.5	35.1	31.2	56.7	25.8	40.1	33.2	37.5	17.7	45.1	28.8
MiRix	69.3	52.1	62.0	36.5	56.0	41.6	36.5	27.8	61.5	45.0	46.9	34.2	45.8	31.2	51.5	37.3
MMA	68.8	51.0	57.3	38.5	49.3	45.4	42.3	42.1	55.2	50.4	46.4	42.1	54.7	36.5	50.3	44.1
UnfoldMem (Ours)	**85.4**	**79.2**	**82.3**	**60.4**	**79.1**	**67.3**	**60.3**	**54.9**	85.4	**78.3**	**69.0**	**57.1**	**63.5**	**53.1**	**73.6**	**63.5**

Table 2: Mem-Gallery results measured by the LLM-judge score across subtasks and overall. * denotes the chain-of-thought (CoT) variant. **Bold** and underlined values indicate the best and second-best non-oracle scores.

	Extraction & Adaptation			Reasoning			Knowledge Management			Total
Method	FR	VS	TTL	TR	VR	MR	KR	CD	AR	
Oracle										
clue-only	94.1	98.2	94.4	79.7	90.8	92.0	84.6	67.9	97.8	92.0
clue-only*	84.2	95.6	93.2	69.1	88.8	81.6	63.6	86.4	87.5	86.5
RAG										
Semantic-RAG	76.3	80.6	82.3	68.3	68.7	77.9	61.7	72.8	93.5	78.1
Semantic-RAG*	70.6	80.2	79.4	72.0	69.8	70.4	56.8	71.6	**95.6**	76.1
UniversalRAG	85.8	75.5	81.6	70.3	61.8	81.1	71.6	42.0	92.4	77.0
Agentic Memory										
A-Mem	80.8	77.0	89.5	73.6	60.3	82.5	71.0	61.7	92.4	79.3
M2A	18.2	2.9	50.6	26.8	17.8	10.9	24.1	62.4	86.4	31.3
MiRix	90.4	82.8	88.6	67.9	54.9	84.0	82.7	69.1	85.9	80.8
MMA	60.7	15.7	77.6	29.7	39.9	59.0	36.4	40.7	95.1	53.0
UnfoldMem (Ours)	**93.4**	**89.2**	**90.1**	**84.6**	**85.6**	**90.8**	**83.3**	**87.7**	64.9	**86.4**

4.3 Further Analysis

Accuracy vs. Token Usage. Figure 3 compares task performance with the total token usage in both memory construction and utilization. Across all three settings, UnfoldMem achieves the highest score while operating at an intermediate token scale. It uses more tokens than the lightweight RAG baselines and A-Mem. A-Mem reduces token uasge by representing images primarily through textual descriptions, whereas UnfoldMem keeps the original visual evidence accessible and inspects it on demand. Nevertheless, UnfoldMem's token usage remains substantially below that of MiRIX, MMA, and M2A, which consume more tokens while attaining markedly lower scores.

Adaptive Tool-Use Patterns. We provide a thorough breakdown of UnfoldMem's tool-use patterns in Figure 4 along three dimensions: *whether to invoke tools*, *how extensively to explore*, and *which memory-access tool to use*. Figure 4-(a) shows that the amount of exploration gradually increases with questions requiring finer-grained visual evidence and deeper reasoning, peaking at $X_4 \times Y_3$ with 3.8 calls for MCQ and 3.2 for Open. This trend indicates that UnfoldMem can allocate its exploration effort according to question difficulty rather than following a fixed tool-use

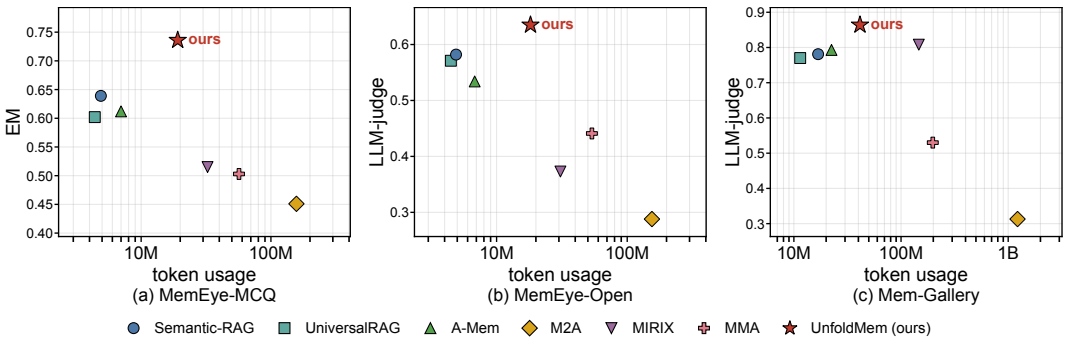

Figure 3: Accuracy–token usage trade-off on MemEye-MCQ (a), MemEye-Open (b), and Mem-Gallery (c). X axis shows total token usage on a logarithmic scale, including memory construction and utilization. Y axis reports rotation exact-match accuracy (EM) for MemEye-MCQ and LLM-judge score for MemEye-Open and Mem-Gallery. Each marker represents one method shown in the legend. The upper-left region indicates better performance with lower token cost.

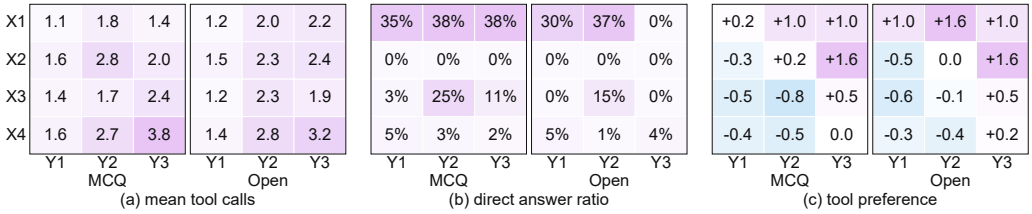

Figure 4: Tool-use patterns of UnfoldMem across MemEye question categories. The top and bottom rows show MCQ and Open, respectively. Each cell corresponds to a visual-evidence level (X_1–X_4) and a reasoning-depth level (Y_1–Y_3). (a) Mean number of tool calls per question. (b) Ratio of questions answered directly without tool use. (c) Tool preference, measured as $\log_2(N_{\text{ask}}/N_{\text{search}})$; positive values (purple) indicate a preference for ask tool, while negative values (blue) indicate a preference for search tools.

budget. In Figure 4-(b), up to 38% of scene-level (X_1) questions are answered directly, as their coarse-grained evidence is often already captured by the session summaries. By contrast, the direct-answer ratio is almost zero in most finer-grained categories, indicating that UnfoldMem does not overconfidently rely on summaries when the question requires evidence beyond their scope. Compared with M2A and MMA's pattern in Appendix B.3, UnfoldMem shows clearer adaptive tool-use patterns. Figure 4-(c) reveals a clear tendency to select different tools for different question types. For questions near the lower-left corner, such as $X_4 \times Y_1$, which typically target a single fine-grained visual detail, the agent tends to use search tools to locate the queried detail directly. In contrast, for questions near the upper-right corner, such as $X_1 \times Y_3$, which require reasoning over multiple pieces of evidence, the agent increasingly uses ask tool, enabling a sub-agent to jointly inspect and reason over the relevant session context.

Retrieval Quality. We compare retrieval quality at the level of original dialogue rounds in Table 3. For UnfoldMem, R_q is the union of rounds exposed to any sub-agent through all tools. For A-Mem, we trace each retrieved memory unit back to its source round. Semantic-RAG and UniversalRAG directly retrieve original rounds. For fair comparison under the same retrieval budget, we set the retrieval top-K as 14 and 16 in MemEye and Mem-Gallery, which are the calculated rounded-average exposed rounds of UnfoldMem. We exclude M2A, MIRIX, and MMA because their heterogeneous rewritten memory units do not permit reliable source-round attribution. Across three settings, UnfoldMem achieves highest recall and precision on average. This demonstrates the benefit of broad and complementary evidence coverage through iterative multi-agent unfolding, which reduces the risk of missing relevant clue rounds. Although Semantic-RAG achieves slightly higher recall on MemEye, its retrieval quality drops behind on Mem-Gallery. This consistency of

Table 3: Retrieval quality on MemEye and Mem-Gallery. **Bold** and underlined values indicate the best and second-best results.

Method	MemEye				Mem-Gallery	
	MCQ		Open		Overall	
	Recall	Precision	Recall	Precision	Recall	Precision
Semantic-RAG	**0.8297**	0.1501	**0.8247**	0.1475	0.7798	0.1242
UniversalRAG	0.7119	0.1261	0.7182	0.1241	0.8085	0.1295
A-Mem	0.6734	0.1052	0.6786	0.1061	0.8244	0.1347
UnfoldMem (Ours)	0.8226	**0.1664**	0.7977	**0.1710**	**0.9501**	**0.1867**

Table 4: Ablation studies of UnfoldMem on MemEye and Mem-Gallery. Each row removes one component from the full system. MemEye-MCQ reports rotation exact-match accuracy, while the other columns report LLM-judge scores.

Variant	MemEye		Mem-Gallery
	MCQ	Open-end	Open-end
UnfoldMem (full)	73.6	63.5	86.4
w/o summary	72.3	60.7	61.7
w/o ask tool	71.2	61.6	61.2
w/o search tools	68.5	59.3	83.1

UnfoldMem comes from the complementary exploration of multiple sub-agents, where each agent focuses on different aspects of the query and contributes focused evidence rather than relying on a single retrieval trajectory.

4.4 Ablation Study

Table 4 evaluates the contribution of the three main components of UnfoldMem by removing session-level summary, the ask tool, or the search tools from the full system. The full version achieves the highest score in all three evaluation settings, indicating that each component provides a distinct contribution. The ask tool and session-level summaries are both essential for exploring long conversational histories. The ask tool provides full access to a selected session, while summaries help the main agent identify which sessions to inspect. Removing either component causes a severe degradation of 25.2 and 24.7 on Mem-Gallery, which contains much longer sessions than MemEye. Although removing search tools leads to smaller decreases in Mem-gallery, the performance drops are more severe on MemEye, with reductions of 5.1 and 4.2. This indicates that search tools are not redundant: even when session summaries and the ask tool remain available, query-driven retrieval helps the main agent efficiently locate relevant evidence across the conversational history.

5 Conclusion

We introduced UnfoldMem, a multi-agent framework for multimodal conversational memory that decouples memory exploration from raw-evidence interpretation. The main agent progressively explores memory by selecting memory access tools and posing focused sub-questions, while temporary sub-agents interpret only the corresponding dialogue rounds and original visual evidence.

Experiments on MemEye and Mem-Gallery show that UnfoldMem achieves the best overall performance among all baselines across all three settings. Further analyses demonstrate that the framework exhibits adaptive tool-use behavior according to visual granularity and reasoning depth, attains substantially higher retrieval quality, and operates in a high-performance, intermediate-token regime. The ablation results further confirm the neccesity of every components with the UnfoldMem. Together, these results suggest that structuring memory access as iterative, context-specific interaction is a promising alternative to exposing all retrieved evidence directly to a single agent.

REPRODUCIBILITY STATEMENT

Section 4.1 specifies the benchmarks, baselines, evaluation protocols, and shared model configuration used in our experiments. Appendix A provides detailed implementation details and baseline-specific hyperparameters. The exact judging procedure and judge prompt are given in Appendix A.4, while Appendix C provides the main-agent and session-agent prompts, tool schemas, and memory-construction prompts. Detailed ablation results and reasoning-budget settings are reported in Appendix B, together providing the experimental and implementation references needed to reproduce the reported results.

AI USE STATEMENT

We used generative AI tools solely to improve the language, grammar, and readability of the manuscript. Generative AI tools were not used to develop the research methodology, formulate hypotheses, design experiments, implement methods, analyze or interpret experimental results, or generate scientific claims. All AI-assisted edits were carefully reviewed and verified by the authors. We take full responsibility for the final content of this work.

REFERENCES

Yuanchen Bei, Tianxin Wei, Xuying Ning, Yanjun Zhao, Zhining Liu, Xiao Lin, Yada Zhu, Hendrik Hamann, Jingrui He, and Hanghang Tong. Mem-Gallery: Benchmarking multimodal long-term conversational memory for MLLM agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 40750–40784, 2026.

Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William Cohen. Murag: Multimodal retrieval-augmented generator for open question answering over images and text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5558–5570, 2022.

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.

Junyu Feng, Binxiao Xu, Jiayi Chen, Mengyu Dai, Cenyang Wu, Haodong Li, Bohan Zeng, Yunliu Xie, Hao Liang, Ming Lu, and Wentao Zhang. M2A: Multimodal memory agent with dual-layer hybrid memory for long-term personalized interactions. *arXiv preprint arXiv:2602.07624*, 2026.

Minghao Guo, Qingyue Jiao, Zeru Shi, Yihao Quan, Boxuan Zhang, Danrui Li, Liwei Che, Wujiang Xu, Shilong Liu, Zirui Liu, Mubbasir Kapadia, Vladimir Pavlovic, Jiang Liu, Mengdi Wang, Yiyu Shi, Dimitris N. Metaxas, and Ruixiang Tang. MemEye: A visual-centric evaluation framework for multimodal agent memory. *arXiv preprint arXiv:2605.15128*, 2026.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*, 2024.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, et al. Memory in the age of ai agents. *arXiv preprint arXiv:2512.13564*, 2025.

Wei-Chieh Huang, Weizhi Zhang, Yueqing Liang, Yuanchen Bei, Yankai Chen, Tao Feng, Zhen Tan, Yu Wang, Tianxin Wei, Shanglin Wu, et al. A survey of agent memory in the second half: Towards self-evolving and long-horizon agents. *Transactions on Machine Learning Research*, 2026.

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.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe

Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, et al. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026.

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

Jiaqi Liu, Zipeng Ling, Shi Qiu, Yanqing Liu, Siwei Han, Peng Xia, Haoqin Tu, Zeyu Zheng, Cihang Xie, Charles Fleming, et al. Omni-simplemem: Autoresearch-guided discovery of lifelong multimodal agent memory. *arXiv preprint arXiv:2604.01007*, 2026a.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. Simplemem: Efficient lifelong memory for llm agents. *arXiv preprint arXiv:2601.02553*, 2026b.

Ziyu Liu, Tao Chu, Yuhang Zang, Xilin Wei, Xiaoyi Dong, Pan Zhang, Zijian Liang, Yuanjun Xiong, Yu Qiao, Dahua Lin, and Jiaqi Wang. MMDU: A multi-turn multi-image dialog understanding benchmark and instruction-tuning dataset for LVLMs. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, Junbo Zhao, and Wei Li. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory. In *International Conference on Learning Representations*, 2026.

Yihao Lu, Wanru Cheng, Zeyu Zhang, and Hao Tang. MMA: Multimodal memory agent. *arXiv preprint arXiv:2602.16493*, 2026.

Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Baolong Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, et al. A survey of context engineering for large language models. *arXiv preprint arXiv:2507.13334*, 2025.

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.

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.

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

Zhoujin Tian, Yao Tian, Hao Zhang, Cheng Chen, Yakun Li, Lei Zhang, and Xiaofang Zhou. Beyond retrieval: Analytic memory for multimodal agents. *arXiv preprint arXiv:2607.29440*, 2026.

Yu Wang and Xi Chen. MIRIX: Multi-agent memory system for LLM-based agents. *arXiv preprint arXiv:2507.07957*, 2025.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, 2025.

Haochen Xue, Feilong Tang, Ming Hu, Yexin Liu, Qidong Huang, Yulong Li, Chengzhi Liu, Zhongxing Xu, Chong Zhang, Chun-Mei Feng, Yutong Xie, Imran Razzak, Zongyuan Ge, Jionglong Su, Junjun He, and Yu Qiao. MMRC: A large-scale benchmark for understanding multimodal large language model in real-world conversation. *arXiv preprint arXiv:2502.11903*, 2025.

Woongyeong Yeo, Kangsan Kim, Soyeong Jeong, Jinheon Baek, and Sung Ju Hwang. Universal-RAG: Retrieval-augmented generation over corpora of diverse modalities and granularities. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3843–3871, 2026.

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. AAAI Press, 2024.

A Experiment Details

A.1 Backbone and Environment

All methods in Table 1 and Table 2 run on the same generation backbone and the same embedding encoder. This section records the shared backbone and embedding encoder's configuration and baseline-specific hyperparameters we used.

Backbone model. Generation uses Qwen3.5-9B greedy decoding with a temperature of 0 and a maximum model context length of $262,144$ tokens. For those who excced the max token length, we conduct context-window truncation that drops the earliest context rounds.

Embedding model. Both text and image embeddings use Qwen3-VL-Embedding-2B with embedding dimension of 2048. Embedding vectors are normalized before cosine similarity is computed. Every method that performs dense retrieval uses this encoder, regardless of whether the input is text-only or multimodal. And we retrieve the top $K = 10$ items at each retrieval step.

Implementation environment. Both Qwen3.5-9B and Qwen3-VL-Embedding-2B are deployed locally on two NVIDIA A100 GPUs with 80 GB of memory each.

A.2 Baseline Details

We compare against two families: RAG systems (*Semantic-RAG* and *UniversalRAG*) and agentic memory systems (*A-Mem*, *M2A*, *MIRIX*, and *MMA*). All of them run with the backbone and embedding model above; the tables below list only the settings that are specific to each method, and all remaining settings follow the shared configuration.

Semantic-RAG. We use the multimodal dense-retrieval version released with MemEye (Guo et al., 2026). Each round is scored by the weighted sum of a text similarity—the cosine similarity between the query and the round's text embedding—and an image similarity—the cosine similarity between the query and the round's image embeddings—and a round that contains no image contributes an image similarity of zero (Table 5). The retrieved rounds are passed to the backbone unchanged.

Table 5: Configuration of Semantic-RAG.

Setting	Value
Neighbor window	0
Text retrieval weight	0.5
Image retrieval weight	0.5

UniversalRAG (Yeo et al., 2026). UniversalRAG adds an explicit query-routing step in front of retrieval, which selects the corpus to search before any evidence is seen (Table 6). We use the LLM routing using the same backbone model (qwen3.5-9b), and an unparsable route falls back to the paragraph category.

A-Mem (Xu et al., 2025). A-Mem builds its memory with caption preprocessing enabled and equal weight of text and image modality. (Table 7).

M2A (Feng et al., 2026). M2A is bounded by iteration budgets on both its memory manager and its query loop (Table 8).

MMA (Lu et al., 2026). MMA fuses source, temporal, and consensus confidence over a dense multimodal index. Its memory search is capped at a retrieval limit of 10 (Table 9).

Table 6: Configuration of UniversalRAG.

Setting	Value
Overall top-K	10
Text top-K	10
Image top-K	10
Neighbor window	0
Text retrieval weight	0.5
Image retrieval weight	0.5
Query routing	Enabled
Routing fallback category	paragraph

Table 7: Configuration of A-Mem.

Setting	Value
Retrieval K	10
Caption preprocessing	Enabled

Table 8: Configuration of M2A.

Setting	Value
Maximum memory-manager iterations	15
Memory-manager context window	5 recent raw messages
Maximum query iterations	5

Table 9: Configuration of MMA.

Setting	Value
Retrieval top-K	10
Source-confidence weight	0.45
Temporal-confidence weight	0.40
Consensus-confidence weight	0.15
Temporal half-life	30 days
Link embedding weight	0.70
Link text weight	0.30
Link top-K	10
Consensus top-K	10

MIRIX (Wang & Chen, 2025). MIRIX is evaluated through its official runtime adapter (Table 10). The adapter is configured with eight memory agents, whose roles are core, resource, semantic, episodic, procedural, knowledge-vault, reflexion, and background, and it writes to memory with read–write scope.

Table 10: Configuration of MIRIX.

Setting	Value
Memory write scope	read_write
Retrieval limit	20
Number of configured memory agents	8

A.3 Query Prompt Templates

QA prompt. All methods are queried through the same prompt format: a fixed benchmark system prompt, then the memory content as messages, then the question as a final user turn. What differs between two methods is therefore only the memory content that they place into this format and, for the chain-of-thought variants, the system prompt. For the retrieval-based methods the content is the retrieved rounds, while for UnfoldMem it is the memory that its own pipeline unfolds. Every round contributes one user message holding the round text and its images and one assistant message holding the round's reply, repeated for each round in the context, in chronological order:

```
[system]
<the benchmark system prompt, one of the four below>

[user]
{round user text}
[img: {image_id_1}]
<image>
[assistant]
{round assistant text} ...  (one user/assistant pair per round placed in the
context, in chronological order; every image follows its [img: ...] marker)

[user] Answer based on the conversation and images above. Be concise and factual.
Question: {question}
[img: question image 1]
<image>
```

A multiple-choice question is appended to the same layout together with its options and a format instruction that names the letters valid for that question:

```
Answer based on the conversation and images above. Be concise and factual.
Question: {question}

A. {option A}
B. {option B}
C. {option C}
D. {option D}

Reply in EXACTLY this XML format, nothing else:
<reason>brief reasoning grounded in the evidence</reason>
<answer>the single option letter (A, B, C, D)</answer>
```

The rows marked * in Table 1 and Table 2 differ only in the system message: rather than answering directly, they answer with a chain-of-thought written in a fixed XML format, where the model first fills a short *reason* field and then the answer in an *answer* field, and only the content of *answer* is scored (Section A.4). We use this format because the native thinking mode of Qwen3.5-9B proved unusable on this workload: on a fraction of the questions the thinking trace grows to several thousand tokens and then degenerates into repeated loops, which inflates latency and leaves no usable final answer within the per-call generation budget.

The three system prompts below are the fixed parts of this format. The open-ended MemEye prompt requires the XML structure and fixes the answer form per question type, so that counting questions

are answered by a bare number, yes/no questions keep their polarity, and descriptive questions stay to one or two sentences.

```
You are an AI assistant being evaluated on MemEye, a vision-centric multimodal
memory benchmark.
Your task is to answer questions about a long-horizon multimodal conversation
using the provided conversation history and images.

Follow these rules:
1. Ground every answer in the conversation history and images. Do not invent
facts.
2. When multiple pieces of information conflict, prefer the most recent visual
evidence unless the question specifically asks about the conflict.
3. Inspect images carefully --- many answers cannot be determined from text
alone.
4. Be precise about spatial positions, colors, textures, and other visual
attributes.
5. If the required information is genuinely absent, say so explicitly rather
than guessing.
6. Output your response in EXACTLY this XML format, nothing else:
<reason>brief reasoning grounded in the evidence</reason>
<answer>the concise answer only</answer>
7. For counting questions, <answer> should contain just the number (e.g. "3").
8. For yes/no questions, <answer> should start with "Yes" or "No".
9. For descriptive questions, <answer> should be a short, direct answer in one
or two sentences at most.
```

The multiple-choice variant is the same prompt with the answer field restricted to a single option letter.

```
You are an AI assistant being evaluated on MemEye, a vision-centric multimodal
memory benchmark.
Your task is to answer multiple-choice questions about a long-horizon multimodal
conversation using the provided conversation history and images.

Follow these rules:
1. Ground every answer in the conversation history and images. Do not invent
facts.
2. When multiple pieces of information conflict, prefer the most recent visual
evidence unless the question specifically asks about the conflict.
3. Inspect images carefully --- many answers cannot be determined from text
alone.
4. Output your response in EXACTLY this XML format, nothing else:
<reason>brief reasoning grounded in the evidence</reason>
<answer>the single option letter (A, B, C, or D)</answer>
```

The format instruction appended to a multiple-choice question is the one below, with {valid} replaced by the letters offered for that question.

```
Reply in EXACTLY this XML format, nothing else:
<reason>brief reasoning grounded in the evidence</reason>
<answer>the single option letter ({valid})</answer>
```

The Mem-Gallery prompt lists the nine evaluation subtasks and fixes the answer convention of each, including image IDs for visual-search questions, a bare affirmative or negative for conflict detection, and an explicit refusal string for answer-refusal questions.

```
You are an AI assistant evaluated on multimodal long-term conversational memory.
For the given question-answering task, your responses must be concise, yet
complete enough to accurately answer the questions.
If multiple pieces of information about the same event appear in the
conversation, always rely on the most recent information.

The question-answering evaluation will contain several multimodal task types:

Factual Retrieval: Retrieve explicit facts mentioned in the conversation for the
answer.
```

```
Multi-entity Reasoning: Combine the retrieved information to reason and infer an
answer.
Temporal Reasoning: Resolve time-dependent questions.
Visual-centric Reasoning: Besides textual information, answer questions using
visual images in the conversation.
Test-time Learning: Learn new visual knowledge from provided images within
historical dialogue and use it in question-answering.
Visual-centric Search: Find the image(s) that match the information in a given
query and return their image ID(s).
Conflict Detection: Detect contradictions between the conversation history and
the information provided in the question.
Knowledge Resolution: Resolve knowledge conflicts or updates by prioritizing the
most recent information.
Answer Refusal: Decline to answer when the information does not exist in the
conversation history.

Follow all instructions strictly. Only answer using information contained within
the multimodal conversation. Do not hallucinate. Always remain consistent and
grounded in the dialogue history.

Output exactly this XML structure and nothing else:
<reason>brief reasoning grounded in the conversation and images</reason>
<answer>the concise answer only</answer>

Put all reasoning inside <reason> and only the final answer inside <answer>.
For counting questions, <answer> must contain only the number.
For yes/no questions, <answer> must start with "Yes" or "No".
For descriptive questions, keep <answer> to one or two concise sentences.
For visual-search questions, preserve the required image_id format from the
question.
For conflict-detection questions, put exactly "Yes." or "No." inside <answer>.
For answer-refusal questions, put "Not mentioned." inside <answer> only when the
information is absent.
```

A.4 LLM-Judge Settings and Prompts

Judging procedure. Each open-ended sample is presented to the judge as three fields—the question, the ground-truth answer, and the assistant answer—and the judge returns a score together with a short rationale as a JSON object that carries a *score* field and a *reasoning* field, for example {"score": 0.75, "reasoning": "short explanation"}. The score could be $\{0, 0.25, 0.5, 0.75, 1\}$. The rationale is retained for inspection and is what we quote in the case study of Appendix D.

The response is parsed by first extracting a JSON object that carries a *score* field and by falling back to a regular expression when the response is not valid JSON. The parsed score $\hat{s}_i$ of sample i is then quantized to a three-point scale before being stored,

$$s_i = \begin{cases} 0, & \hat{s}_i < 0.25, \\ 0.5, & 0.25 \le \hat{s}_i < 0.75, \\ 1, & \hat{s}_i \ge 0.75, \end{cases}$$

so that, although the prompt defines five levels, a stored value is always one of $\{0, 0.5, 1\}$. The reported judge score is the arithmetic mean of the stored values over the samples that were scored successfully,

$$\mathrm{JudgeScore}(S) = \frac{1}{|S|} \sum_{i \in S} s_i,$$

and the same mean is used for the overall result and for every reported task or category group.

Two details of the pipeline bound what is graded. First, only open-ended rows are passed to the judge; the multiple-choice rows of MemEye follow the separate exact-match protocol of Section 4.1. Second, for methods that answer with a reasoning trace, clue-only* and Semantic-RAG* in the experiment, only the final *answer* element of the trace is graded, so the trace itself never affects the score; if that element is missing or empty, the sample is recorded as 0 without an API call. A sample that still fails after all retries raises an error and interrupts the evaluation rather than being silently assigned a score.

Judge prompt. The complete prompt is reproduced below. It is shared by the open-ended portion of both benchmarks.

```
You are an impartial judge evaluating the memory capabilities of an AI assistant
on a question-answering task.
Your task is to compare the Assistant's Answer against the Ground Truth and
assign a score of 0, 0.25, 0.5, 0.75, or 1.

### Important Principles

1. **Semantic equivalence over surface form.** The assistant may use different
wording, synonyms, or sentence structure. As long as the meaning matches the
Ground Truth, it should receive full credit. Do NOT penalize for stylistic
differences.
2. **Numeric / counting questions are binary.** If the Ground Truth is a
specific number (e.g., "3"), the answer is either exactly right (score 1) or
wrong (score 0). There is no partial credit for being off by one.
3. **Negation + correction questions.** Some Ground Truth answers first negate a
premise and then provide the correct information (e.g., "No --- it was May
18-22, not May 15-19"). The assistant must get BOTH the negation AND the
corrected fact right for full credit.
4. **Identity / "Who is" questions.** The Ground Truth may describe a person by
role, actions, or relationships rather than by name. The assistant's answer is
correct if it identifies the same unique person, even if it uses a name instead
of a description or vice versa.
5. **Answer the thing that was asked.** Judge whether the assistant answered the
requested attribute or type of information. If the question asks for a category,
role, relationship, time, count, ordering, or specific item, an answer giving a
different type of information is not fully correct even if it is related.
6. **Wrong entity beats related entity.** If the answer names the wrong person,
object, brand, place, event, or option, do not give high scores just because it
is on-topic. Being from the right semantic domain is not enough.
7. **Multi-item questions require set matching.** If the question asks for two
or more items, compare the full set of required items. Full credit requires all
required items and no incorrect extras. If only some required items are correct,
partial credit may be appropriate. If the set is mostly wrong, score low.
8. **Ordering / chronology questions require the correct item.** If the question
asks which item was first, last, middle, earlier, later, before, or after, the
selected item must match the Ground Truth. A different plausible item is still
wrong.
9. **Do not over-reward fluent explanations.** Extra detail is helpful only if
it stays consistent with the Ground Truth. Confident wording or elaboration
should not increase the score when the core fact is wrong.
10. **Contradicted relations or orderings are usually zero.** If the question
asks about a relationship such as before/after, earlier/later, first/last,
middle, sequence, relative timing, or other ordering, and the assistant gives
the opposite relation, denies the relation, or selects the wrong ordered item,
this is usually a clear contradiction and should normally receive score 0 rather
than 0.25.

### Scoring Rubric

**Score 0 (Incorrect / Miss):**

- The answer contradicts the Ground Truth.
- For Yes/No questions: The answer has the wrong polarity (e.g., says "Yes" when
Ground Truth is "No").
- For numeric questions: The number is wrong.
- For attribute-type mismatches: the answer gives the wrong kind of thing (for
example, a brand when the question asks for a product category, or a related
event when the question asks for the earliest specific item).
- For relation/order questions: if the answer contradicts the required
before/after, earlier/later, first/last, middle, or similar relationship, score
0 in the usual case.
- The answer provides factually wrong information, hallucinations, or fails to
provide any relevant information.
```

```
**Score 0.25 (Poor / Tangential):**

- The answer touches on the topic but misses the **core entity** or key value
required.
- The answer contains a mix of minor correct details and **significant
hallucinations** or wrong associations.
- The answer is excessively vague to the point of being useless (e.g., answering
"a dog" instead of "a golden retriever").
- For multi-item questions: use 0.25 when the answer is clearly about the right
topic but most of the required set is wrong.

**Score 0.5 (Partial / Vague):**

- The answer captures the **main entity or concept** correctly but misses
important supporting details.
- For negation + correction questions: The polarity is correct but the corrected
fact is missing or wrong, or vice versa.
- The answer is too general or misses key qualifiers present in the Ground Truth.
- For multi-item questions: use 0.5 when the answer gets a substantial part of
the required set right (for example, one of two required items is correct and
the rest is wrong or missing).

**Score 0.75 (Good / Minor Imperfection):**

- The answer is largely accurate and captures the core information confidently.
- It misses only **minor details** (e.g., specific adjectives or secondary
details) that do not alter the main truth.
- The answer contains all the correct information but includes unnecessary
filler that reduces precision.

**Score 1 (Correct / Exact):**

- The answer is accurate, precise, and confident.
- For Yes/No questions: The polarity matches the Ground Truth perfectly.
- For numeric questions: The number matches exactly.
- For descriptive questions: The answer contains **all** the core information
and necessary details required by the Ground Truth without hallucinations.
- Semantic equivalence is sufficient --- exact wording is NOT required.
- For multi-item questions: all required items are present and there are no
incorrect extra items.

### Input Data

Question: {{question}} Ground Truth: {{ground_truth}} Assistant Answer:
{{model_output}}

### Output Format

Output strictly in the following JSON format:
{"score": <0, 0.25, 0.5, 0.75, or 1>, "reasoning": "<short explanation>"}
```

B Additional Result

B.1 Detail Ablation Results

This section reports the per-stratum numbers behind the aggregate results of Section 4.4 and Table 4, in the same format as the main results tables. The variants share the backbone, the embedding encoder, the prompts, and the prebuilt memory of the full system; only the interface that the main agent sees changes. Table 11 mirrors Table 1 and Table 12 mirrors Table 2, with the ablation variants in place of the baselines, and Table 13 varies the reasoning budget of the full system. Each ablated variant is a separate run of the same configuration with a single component disabled. Which component is absent can be read off the run itself, since the *w/o ask* runs issue no *ask* call at all (two stray calls out of hundreds of thousands of tokens on MemEye), the *w/o search* runs issue nothing but *ask*, and the *w/o summary* run keeps both tools but is given a session index without any content information instead of the summaries.

MemEye. Table 11 gives the two MemEye scores at every level of the visual-evidence axis (X_1–X_4) and the reasoning-depth axis (Y_1–Y_3), together with the total. The removals are not spread evenly. On MemEye-MCQ the largest single gaps are on the visual axis at X_3 (0.790 for the full system against 0.715 for *w/o search*) and on the reasoning axis at Y_1 (0.854 against 0.775); on the open-ended questions they are at X_1 (0.792 against 0.615) and X_4 (0.549 against 0.470 for *w/o ask*). The ordering is not uniform across levels, however: removing the summaries leaves X_3 above the full system on MCQ (0.805 against 0.790), and removing the *ask* tool leaves X_3 and Y_1 above it on the open-ended questions (0.725 against 0.673 and 0.796 against 0.783). The two tools therefore substitute for each other in some cells even though the totals separate cleanly.

Table 11: MemEye ablation detail: exact-match accuracy on MCQ and LLM-judge score on Open, broken down by visual-evidence granularity (X_1–X_4), reasoning depth (Y_1–Y_3), and overall. The layout follows Table 1, with the ablation variants in place of the baselines.

	X_1		X_2		X_3		X_4		Y_1		Y_2		Y_3		Total	
Variant	MCQ	Open	MCQ	Open	MCQ	Open	MCQ	Open	MCQ	Open	MCQ	Open	MCQ	Open	MCQ	Open
UnfoldMem (full)	0.854	0.792	0.823	0.604	0.790	0.673	0.603	0.549	0.854	0.783	0.690	0.571	0.635	0.531	0.736	0.635
w/o summary	0.755	0.625	0.781	0.656	0.805	0.662	0.603	0.523	0.863	0.771	0.658	0.527	0.651	0.531	0.723	0.606
w/o ask tool	0.823	0.771	0.781	0.542	0.776	0.725	0.579	0.470	0.840	0.796	0.650	0.520	0.656	0.573	0.712	0.616
w/o search tools	0.818	0.615	0.760	0.635	0.715	0.616	0.577	0.545	0.775	0.733	0.647	0.515	0.620	0.573	0.685	0.593

Mem-Gallery. Table 12 reports the nine Mem-Gallery subtasks, and the two removals do not damage the same ones. Removing the summaries costs most on factual retrieval ($0.934 \rightarrow 0.466$), multi-entity reasoning ($0.908 \rightarrow 0.454$), temporal reasoning ($0.846 \rightarrow 0.492$), and knowledge resolution ($0.833 \rightarrow 0.543$)—the subtasks whose evidence the index is supposed to locate. Removing the *ask* tool costs most on visual-centric search ($0.892 \rightarrow 0.330$) and visual-centric reasoning ($0.856 \rightarrow 0.353$), and it is the removal with the largest overall drop on this benchmark ($0.864 \rightarrow 0.612$). Removing the search tools is the mildest of the three here ($0.864 \rightarrow 0.831$), and no subtask loses more than 0.09 to it: when the sessions are long, a session can still be unfolded by name, so the agent loses the ability to *find* a session by query but not the ability to *read* one. Test-time learning is insensitive to all three removals (0.901 against 0.840, 0.865, and 0.868), and on answer refusal every ablation scores above the full system (0.793, 0.859, and 0.793 against 0.649), so the full system's known deficit on that subtask is not caused by any of the three components.

Table 12: Mem-Gallery ablation detail: LLM-judge score for each of the nine subtasks and overall. The layout follows Table 2, with the ablation variants in place of the baselines: FR, VS, and TTL belong to extraction and adaptation, TR, VR, and MR to reasoning, and KR, CD, and AR to knowledge management.

	Extraction & Adaptation			Reasoning			Knowledge Management			Total
Variant	FR	VS	TTL	TR	VR	MR	KR	CD	AR	
UnfoldMem (full)	0.934	0.892	0.901	0.846	0.856	0.908	0.833	0.877	0.649	0.864
w/o summary	0.466	0.562	0.840	0.492	0.595	0.454	0.543	0.630	0.793	0.617
w/o ask tool	0.642	0.330	0.865	0.748	0.353	0.464	0.605	0.716	0.859	0.612
w/o search tools	0.854	0.802	0.868	0.817	0.842	0.867	0.765	0.790	0.793	0.831

B.2 Reasoning budget ablation

The full system runs its ReAct loop with a maximum of 6 iterations (5 retrieval steps), and Table 13 varies that budget on MemEye-Open while holding everything else fixed. The score rises from 0.542 at 2 iterations to 0.597 at 4 and 0.635 at the default 6, then flattens and falls back to 0.617 at 8 iterations, so the additional turns are not converted into evidence on this benchmark. The breakdown shows where the earlier turns are spent: the scene level X_1 is already close to its final value at 2 iterations (0.771 against 0.792), and the run gains most between 2 and 4 iterations at the pixel level X_4 ($0.383 \rightarrow 0.496$), which is the level that needs an extra unfolding step to reach. The reasoning axis gains most at Y_3 ($0.354 \rightarrow 0.531$ between 2 and 6 iterations) and Y_2 ($0.488 \rightarrow 0.584$), while Y_1 already reaches 0.708 at 2 iterations and moves by at most 0.075 across the whole range.

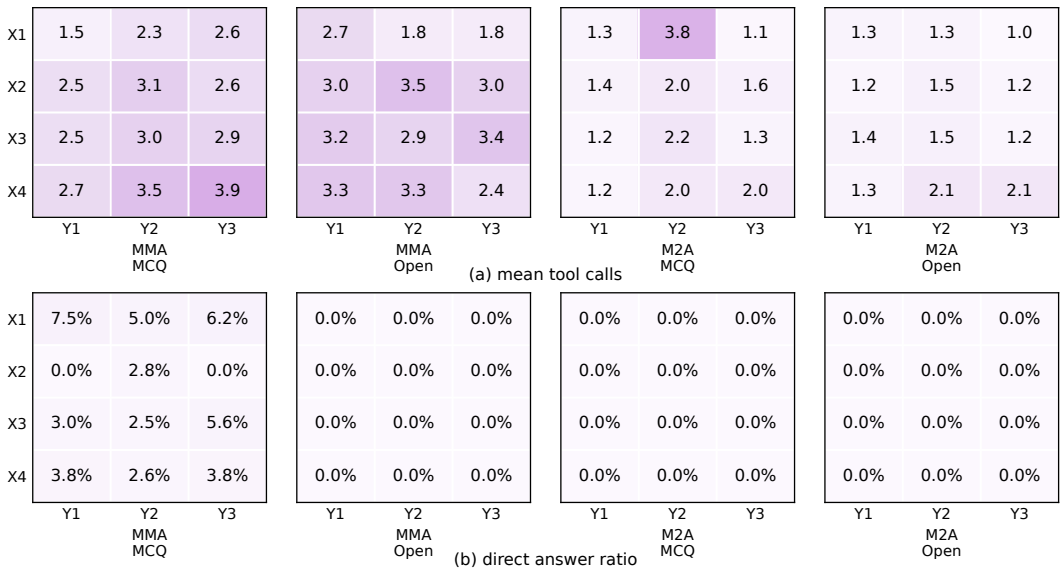

Figure 5: Tool-use patterns of M2A and MMA across MemEye question categories. The top and bottom rows show MCQ and Open, respectively. Each cell corresponds to a visual-evidence level (X_1–X_4) and a reasoning-depth level (Y_1–Y_3). (a) Mean number of tool calls per question. (b) Ratio of questions answered directly without tool use.

B.3 Tool-use patterns of adaptive baselines

Figure 5 summarizes how the adaptive baselines MMA and M2A use tools across different MemEye question categories. We report both the average number of tool calls per question and the fraction of questions answered without invoking any tool, as a comparison with UnfoldMem's pattern discussed in Sec 4.3.

Table 13: Reasoning-budget ablation on MemEye-Open: the maximum number of iterations of the agent loop is varied while everything else is held fixed. The row for 6 iterations is the default budget used by every other experiment in the paper.

Max. iterations	X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Total
2	0.771	0.510	0.623	0.383	0.708	0.488	0.354	0.542
4	0.771	0.604	0.630	0.496	0.729	0.534	0.531	0.597
6 (default)	0.792	0.604	0.673	0.549	0.783	0.571	0.531	0.635
8	0.823	0.552	0.634	0.549	0.721	0.584	0.500	0.617

C UnfoldMem Prompt Templates and Tool Schemas

This section lists the prompt templates and tool schemas used in UnfoldMem. Braces mark content that is inserted at run time.

Main-agent system prompt. The main agent is the top-level orchestrator of the two-layer ReAct loop. This prompt gives it the summarized flat session index, the meaning of its three tools, and the rules that decide when the index alone may be answered from, when a tool must be called, and how queries should be split or aggregated.

```
You are the long-term memory keeper for a multimodal chat agent. You hold an
index of past conversation sessions, each summarized in a few sentences.
A new question about those past conversations is given.
```

```
You are the top-level orchestrator in a two-layer ReAct system. You have access
to tools (ask / search_by_text / search_by_img) that progressively unfold the
memory; see each tool's description in the schema for full details.
Each sub-agent spawned by ask / search_by_text / search_by_img sees ALL relevant
images directly.
Be concise and factual. Answer the question directly when you have enough
information.
```

Session-agent system prompt. Every *ask*, *search_by_text*, and *search_by_img* call delegates the selected memory context to a session-scoped sub-agent that sees the context and its images directly. This prompt confines the sub-agent to that context, tells it how to cite images and sessions, and requires it to report insufficient evidence instead of guessing.

```
You are a session-scoped memory agent spawned by the top-level memory keeper.
You answer a focused question using a bounded context of past conversation
messages and their actual images.

All the images from this context are shown to you directly --- you can see them
as part of this message. Use both the conversation text and the images to answer
the question.

Rules:
- Base your answer strictly on the provided context.
- Cite images using ###<img src="<path>">### when you are unsure or uncertain
about your answer, when a specific image is directly relevant to your analysis,
or whenever you need to refer to a particular image --- copy the path EXACTLY
from the context's ###<img src="...">### markers.
  The caller can see the image directly through this reference and visually
verify your judgment.
- Do NOT invent image paths and do NOT add or change any directory prefix.
- NEVER fabricate, guess, or invent information (dates, names, numbers, facts)
that is not present in the context. If you do not have the information, say "the
context does not contain this information" and then provide any directly related
details that ARE present in the context, citing relevant images.
- Explain your reasoning before giving the answer.

Example:
Q: What does Tom look like?
A: This is Tom's photo ###<img src="Vacation_Album/tom_photo.jpg">###
```

Tool Schema. The three tool schemas below are exposed to the main agent as native function calls.

The schema of the *ask* tool:

```
{
  "type": "function",
  "function": {
    "name": "ask",
    "description": "Ask a memory agent a focused question about a bounded
context. Pass session_name (one) or session_list (up to 10) using the values
shown in the [session_name:...] tags of the current index. An ask drills into
that session's content --- the sub-agent sees that session's rounds. When
multiple entries are given (session_list), their full contexts are concatenated
into ONE sub-agent context (with markers), allowing the sub-agent to count,
compare, or aggregate across entries directly. This is like a search but with
hand-picked entries instead of semantic retrieval --- use this when you know
which entry(ies) contain the answer.",
    "parameters": {
      "type": "object",
      "properties": {
        "session_name": {
          "type": "string",
          "description": "The session_name value as shown in the
[session_name:...] tag in the current session index. Use this for a single-entry
query."
        },
```

```
        "session_list": {
          "type": "array",
          "description": "List of up to 10 session_name values (as shown in the
[session_name:...] tags). All rounds from these entries are concatenated into
ONE sub-agent context. Use when you need cross-entry counting, comparison, or
aggregation.",
          "items": {"type": "string"}
        },
        "question": {
          "type": "string",
          "description": "The FOCUSED sub-question to ask about the entry(ies)
--- authored by you (the caller). When a list with multiple entries is used, the
sub-agent sees ALL rounds from all listed entries concatenated and CAN count,
compare, and aggregate across them directly. For visual detail questions, phrase
the question so the sub-agent knows to inspect the relevant images. REQUIRED:
you already see the index/summaries, so always ask a focused question targeting
specific details."
        },
        "question_img_name": {
          "type": "string",
          "description": "Optional: the bare FILENAME of a question image (from
the separate question-image directory) to pass along to the sub-agent. Not an
<img src=\"...\"> marker."
        }
      },
      "required": ["question"],
      "additionalProperties": False
    }
  }
}
```

The schema of the *search_by_text* tool:

```
{
  "type": "function",
  "function": {
    "name": "search_by_text",
    "description": "Retrieve the most relevant conversation rounds across all
sessions by vector similarity: rounds are ranked by fused text+image similarity
(text cosine + image cosine averaged per round), the top-K are returned in
chronological order, then a sub-agent answers `question` from that gathered
context. This is a semantic search --- it retrieves rounds whose content is
similar to your query text. Use when the target session is unclear or the
question spans many sessions. Returns the sub-agent's textual answer.",
    "parameters": {
      "type": "object",
      "properties": {
        "query_text": {
          "type": "string",
          "description": "The text query to search for in past conversations.
Center it on ONE coherent fact; for a multi-fact question whose facts are not
discussed together, issue separate searches per fact instead of concatenating
them into one query."
        },
        "question": {
          "type": "string",
          "description": "The FOCUSED sub-question to answer from the retrieved
context --- authored by you (the main agent), not necessarily the original
question. You MAY copy the original question verbatim if it is already focused
and appropriate. If omitted, the sub-agent is asked \"what is {query_text}\"."
        }
      },
      "required": ["query_text"],
      "additionalProperties": False
    }
  }
}
```

The schema of the *search_by_img* tool:

```
{
  "type": "function",
  "function": {
    "name": "search_by_img",
    "description": "Retrieve rounds whose images+text are most similar to a
query image, using fused image+text similarity, then hand them (in chronological
order) to a sub-agent that answers `question` from that gathered context. Use
when the question is anchored on a visual cue.
      The query image is specified EITHER by `query_img_name` (an attached
question image filename) OR by `img_rel` (an image path surfaced by a prior
sub-agent's answer, e.g. from an <img src=\"<path>\"> marker).
      At least one must be provided. Returns the sub-agent's textual answer.",
    "parameters": {
      "type": "object",
      "properties": {
        "query_img_name": {
          "type": "string",
          "description": "The bare FILENAME of an attached question image (from
the separate question-image directory) to match against past conversation
images. Not an <img src=\"...\"> marker.
             Provide this OR img_rel (at least one required)."
        },
        "img_rel": {
          "type": "string",
          "description": "An in-conversation image path surfaced by a prior
sub-agent's answer (the <path> inside an <img src=\"<path>\"> marker), to use as
the query image for reverse similarity search. Provide this OR query_img_name
(at least one required)."
        },
        "question": {
          "type": "string",
          "description": "The FOCUSED sub-question to answer from the retrieved
context --- authored by you (the main agent), not necessarily the original
question. You MAY copy the original question verbatim if it is already focused
and appropriate. If omitted, the sub-agent is asked \"what is this img <image>\"
with the query image attached."
        }
      },
      "required": [],
      "additionalProperties": False
    }
  }
}
```

Memory-construction prompts. These two prompts are used only when summary is built from the raw sessions. The image-description prompt converts each image into a short caption that is indexed together with its round.

```
You are an image description tool. Your ONLY purpose is to output a short
description of the image.
RESPOND WITH ONLY THE DESCRIPTION TEXT AND NOTHING ELSE. No analysis, no
thinking, no explanation.

Format your description as 1-3 short sentences:
- If the image contains text: summarize its MAIN MESSAGE in one sentence.
  DO NOT reproduce the text verbatim --- condense it. For long passages, write a
single sentence capturing the core idea, never copy word-for-word.
- If the image has no text: describe only the key visual content.
- Do not speculate beyond what is visible.
```

The session-summary prompt condenses a full session transcript, together with its images, into the short summary that appears in the flat session index.

```
You are a conversation summarizer tool. Your ONLY purpose is to output a short
summary of a conversation session.
RESPOND WITH ONLY THE SUMMARY TEXT AND NOTHING ELSE. No analysis, no thinking,
no explanation.

Cover these points in 3-5 short sentences:
- main topics discussed - key events, decisions, or information exchanged -
which images were involved (describe them by content, do NOT include file paths)

It may help to base the summary on what the images actually show (objects,
characters, scenes, actions). Do not narrate every turn. Do not add information
not in the dialogue.
```

D CASE STUDY

D.1 SUCCESS CASES

We select questions on which UnfoldMem is judged correct. Every trace is reproduced from the recorded run of the corresponding method. Each case states the question, the method's answer, and our analysis as text, and then reproduces the recorded run as a trace: the trace opens with the raw benchmark question, and each agent turn is then set as a box that opens with the iteration label, followed by the tool call as the agent issued it and the report it received, quoted verbatim. Images are reproduced once, in the figures described in the text.

D.1.1 CASE 1: DIRECT ANSWER

Question. "Approximately how many distinct people have you actually been chatting with across all your conversations, excluding any names mentioned secondhand or in correction chats?" *(MemEye social_chat_memory_test, idx 26).* Ground truth: *Around 10.* The archive holds 39 sessions and 53 rounds, 43 of which carry a screenshot. Exactly ten people appear in it—Daniel, Elena, Helen, Jordan, Marcus, Mia, Priya, Ryan, Sara, Tomas—introduced by seven "first conversation" rounds, which form the clue set. The question is a set union and a count over the entire archive.

Answer. UnfoldMem answers 11 and the judge scores it 1.0.

UnfoldMem operation trace.

Question

Approximately how many distinct people have you actually been chatting with across all your conversations, excluding any names mentioned secondhand or in correction chats?

No tool call

(none).

Final answer (judge 1.0)

11

Analysis. No retrieval is needed because the summary already records which people appeared in each session, so the question reduces to reading and counting rather than checking the raw images and de-duplicating across all separate screenshots. The answer 11 against the ground-truth "around 10" is an aggregation error the judge tolerates by design.

Baseline failures. Table 14 lists the ten rounds each baseline actually retrieved. Semantic-RAG's window covers all ten people, each of whom speaks in at least one of its rounds, so the evidence it needs is present; what it cannot do is aggregate, because the count has to be de-duplicated across ten separate screenshots, and it answers 7 (judge 0.0). A-Mem's nine other rounds name nobody at all, and only CHAT_S3:R2 carries people (Elena, Ryan), so the union it can form is two—exactly its answer (judge 0.0).

Table 14: Case 1: the ten rounds each baseline retrieved, and the people appearing in them.

Semantic-RAG (answers 7)		A-Mem (answers 2)	
Round	People	Round	People
CHAT_S2:R2	Marcus, Priya	CHAT_S10:R1	*none*
CHAT_S3:R1	Elena, Ryan	CHAT_S20:R1	*none*
CHAT_S3:R2	Elena, Ryan	CHAT_S16:R1	*none*
CHAT_S4:R2	Daniel, Marcus, Priya	CHAT_S33:R1	*none*
CHAT_S8:R2	Elena, Ryan	CHAT_S29:R1	*none*
CHAT_S15:R2	Marcus, Priya, Sara	CHAT_S39:R1	*none*
CHAT_S18:R2	Helen, Mia, Tomas	CHAT_S14:R1	*none*
CHAT_S19:R2	Elena, Jordan	CHAT_S37:R1	*none*
CHAT_S26:R2	Elena, Ryan	CHAT_S3:R2	Elena, Ryan
CHAT_S35:R2	Priya, Sara	CHAT_S5:R1	*none*

D.1.2 Case 2: Single-Turn Search

Question. "What identification tag number is currently displayed in the fossil room case?" *(MemEye multi_scene_visual_case_archive_assistant, idx 47).* Ground truth *A-209*. The memory photographs the same display case twice. Reading the images directly confirms the following six rounds, which are all the images of the two visits.

Answer. UnfoldMem answers A-209 and the judge scores it 1.0.

Table 15: Case 2: the six image rounds of the two visits.

Round	Date	Image	Tag on the case
SCENE_S7:R2	2025-06-20	S7-IMG1	**C-1127**
SCENE_S7:R4	2025-06-20	S7-IMG2	torn blank label
SCENE_S7:R5	2025-06-20	S7-IMG3	**C-1127**
SCENE_S7:R7	2025-06-20	S7-IMG4	**C-1127**
SCENE_S9:R1	2025-06-25	S9-IMG1	**A-209**
SCENE_S9:R7	2025-06-25	S9-IMG4	**A-209**



Figure 6: Case 2: the same display case on the two visits. Left: 2025-06-20, tag C-1127 (SCENE_S7:R2). Right: 2025-06-25, tag A-209 (SCENE_S9:R1). The question turns on which of the two is displayed *currently*, a cue that no caption and no similarity score represents.

UnfoldMem operation trace.

Question

What identification tag number is currently displayed in the fossil room case?

Iteration 0

```
search_by_text(
  query_text="fossil room case identification tag label",
  question="What identification tag number is currently displayed in the
```

```
fossil room case?"
)
```
[Author note: the search brings 2 of the 4 clue rounds into the context (SCENE_S7:R7 from the first visit and SCENE_S9:R1 from the second); the answer is read from SCENE_S9:R1.]

> The identification tag number **A-209 is currently displayed** in the fossil room case, attached to the upper left of the glass frame in **the most recent image** (session SCENE_S9). Reference to the specific image: `###<img src="32.png">###`, which is the right panel of Figure 6.

Final answer (judge 1.0)

A-209.

Analysis. The main agent calls the search_by_text tool to find the relevant rounds for the sub-agent, and the sub-agent returns the correct answer.

D.1.3 CASE 3: FURTHER CONFIRMATION

Question. "Was the crown given willingly by the king or taken by force?" *(MemEye cartoon_entertainment_companion, idx 46)*. Ground truth: *Given willingly by the king.* The clue round is CARTOON_S40:R1, page 27 of the comic (session date 2025-03-15), whose image caption reads "Comic strip titled 'King for a Day' shows a caveman king trying to convince another caveman to take his place because ruling is hard work." The trap is that the surrounding storyline contains violence and a crown hand-back, so the correct answer is reached only if the context contains this page and not merely the pages around it.

Answer. UnfoldMem answers that the crown was **offered willingly by the King** and not taken by force, and the judge scores it 1.0.

UnfoldMem operation trace.

Question

Was the crown given willingly by the king or taken by force?

Iteration 0

```
ask(
  session_name="CARTOON_S40",
  question="Was the crown offered willingly by the king or was it taken
by force?  What was the exact sequence of events on page 27?"
)
```
[Author note: this call opens CARTOON_S40, i.e. the clue round CARTOON_S40:R1 (page 27), and the answer is read from it. The four later calls probe other sessions and touch no clue round.]

> Based on the comic page provided (64.png), the crown was **offered willingly by the king**; it was **not taken by force**. The king explains that he must leave town and asks Oop to fill in for him temporarily.
>
> ...
>
> *The report cites the page as plain text only and inlines no image; page 27 is reproduced for reference in Figure 7(a).*

Iteration 1

```
ask(
  session_name="CARTOON_S41",
  question="What happens in the punishment/crowd scene?  Was this after
Oop was made king or before?  What specifically does the crowd do to
the ruler?"
)
```

> Based on the text in the panels on page 28 `###<img src="65.png">###` (Figure 7(b)), here are the details:
>
> **Timeline:** This scene happens **after Oop was made king**. ...

Iteration 2

```
ask(
  session_name="CARTOON_S59",
  question="What does page 32 show?  When does Oop hit the king with a
club?  Is this after being made king temporarily?"
)
```

> Based on the visual content of page 32: ... Oop pops up from behind the seated character shouting "**PEEK-A-BOO! GUESS WHO!**" (###<img src="77.png">###, Figure 7(c)). In the final panel, **Oop strikes the character with a weapon** (likely a club or bone). ...
>
> **Is this after being made king temporarily?** The provided text on this page **does not explicitly state whether Oop has been made king temporarily**. ...

Iteration 3

```
ask(
  session_name="CARTOON_S61",
  question="What happens on the crown return page?  Does Oop voluntarily
give back the crown or was he removed from power?  What context is
given about whether he had earned or received the crown?"
)
```

> The user is asking for specific details about the events on page 33 (the image provided). I need to determine: 1. Whether Oop voluntarily gave back the crown or was removed. 2. The context given regarding how he earned/received the crown. ...

[Author note: the report then runs for 15,049 characters of panel-by-panel self-correction, never reaching a conclusion, and carries no marker, so no image is inlined for this turn.]

Iteration 4

```
ask(
  session_name="CARTOON_S54",
  question="What exactly happens on page 30 with the clam dispute?
Is this related to Oop's kingship?  How does this connect to the
crown-taking/coronation sequence?"
)
```

> On page 30 (image ###<img src="73.png">###, Figure 7(d)), a dispute breaks out where two subjects are fighting over **50 clams**. ...
>
> **Connection to Oop's Kingship:** This scene confirms Oop's status as the ruler. He is depicted wearing a crown and is addressed by his subjects as "**Your Majesty**." ...

Final answer (judge 1.0)

The crown was initially **offered willingly by the King**, who asked Alley Oop to take his place temporarily because he had to leave town ("Need someone to take my place!"). While Oop fits the physical requirements for rulership (being measured at exactly six feet tall), the transfer of power in the primary coronation scene was initiated by the King's request, **not a forcible seizure**.

Analysis. The main agent finds the clue session that includes the major comic-page storyline from the summaries, and calls the ask tool to get the details. Then it calls the ask tool four times to check the later story and confirm the answer.

Baseline failures. Semantic-RAG recalls the clue: CARTOON_S40:R1 enters its ten-round window at rank 2, but the window carries nine other rounds and its top hit, CARTOON_S61:R2, is the club-strike page, so the recall is complete and the answer is still "The crown was taken by force" (judge 0.0). A-Mem recalls no clue round at all—it returns CARTOON_S40:R2 and CARTOON_S40:R3 from the clue session but never CARTOON_S40:R1—and answers "The crown was taken by force when Oop struck the crowned figure" (judge 0.0).

D.2 Failure Cases

We also report two questions on which UnfoldMem is judged wrong, one at each end of its tool-use range. In the first the agent spends five calls and still never retrieves the page that carries the answer; in the second it makes no call at all, answers from the session index, and returns a brand that the index alone cannot settle. Both are reported as failures of the method itself.

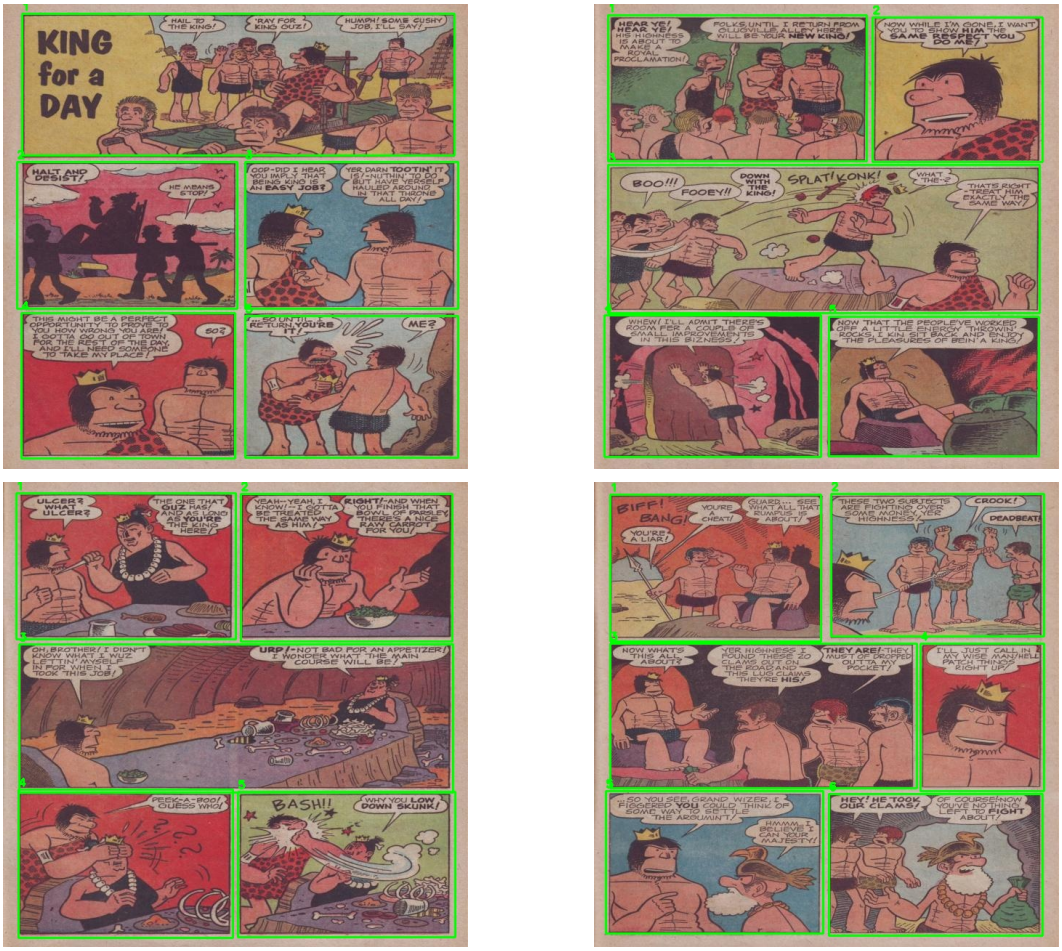

Figure 7: Case 3: the four comic pages the trace turns on. **(a)** (top left) page 27, the clue round `CARTOON_S40:R1`, cited by the report of iteration 0 as plain text but not inlined. **(b)** (top right) page 28, inlined at iteration 1. **(c)** (bottom left) page 32, inlined at iteration 2. **(d)** (bottom right) page 30, inlined at iteration 4.

Table 16: Case 3: the ten rounds each baseline retrieved, and whether a clue round is among them. RAG's window holds the first endpoint of the subtraction but not the second; A-Mem's holds neither.

Semantic-RAG (answers `2026-01-09`)		A-Mem (answers `10 days`)	
Round	Clue	Round	Clue
`CHAT_S1:R1`	**yes** (2026-01-09)	`CHAT_S11:R2`	no
`CHAT_S7:R1`	no	`CHAT_S20:R1`	no
`CHAT_S8:R1`	no	`CHAT_S10:R1`	no
`CHAT_S8:R2`	no	`CHAT_S39:R1`	no
`CHAT_S11:R1`	no	`CHAT_S18:R2`	no
`CHAT_S11:R2`	no	`CHAT_S18:R1`	no
`CHAT_S12:R1`	no	`CHAT_S14:R1`	no
`CHAT_S18:R2`	no	`CHAT_S33:R1`	no
`CHAT_S25:R1`	no	`CHAT_S7:R1`	no
`CHAT_S31:R1`	no	`CHAT_S16:R1`	no

D.2.1 CASE 4: THE DECISIVE PAGE IS NEVER FOUND

Question. "After the earlier crowd and chore scenes, what problem appears on the last palace page?" *(MemEye cartoon_entertainment_companion, idx 49)*. Ground truth: *Noisy guards interrupt him at the palace*. The benchmark gives three clue rounds, and labels the first two stale: CARTOON_S41:R1 (page 28, the crowd pelting the temporary king with rocks), CARTOON_S47:R1 (page 29, the chore sequence) and the decisive CARTOON_S56:R1 (page 31). Page 31 opens with the king shouting "GUARD! DO SEE WHAT ALL THAT RACKET IS OUTSIDE!" over WHAM! CRASH! BAM! effects, which is the answer; the rest of the page is dinner and the advisor.

Answer. UnfoldMem answers that the time machine malfunctions on page 35, and the judge scores it 0.0.



Figure 8: Case 4: the three clue pages. Left to right: page 28 (the crowd scene, `CARTOON_S41:R1`), page 29 (the chore scene, `CARTOON_S47:R1`), and page 31 (`CARTOON_S56:R1`), whose first panel shows the noisy guards. **None of the three entered the context of the run analysed below.**

The agent issued five calls over five iterations—two searches and three single-session asks—and none of them reached pages 28, 29 or 31 (Figure 8).

UnfoldMem operation trace.

Question

After the earlier crowd and chore scenes, what problem appears on the last palace page?

Iteration 0

```
search_by_text(
  query_text="last palace page problem issue after crowd chore scenes",
  question="What problem appears on the last palace page after the crowd
and chore scenes?"
)
```

> "The search context **does not provide explicit details** about what specific problem appears on 'the last palace page after the crowd and chore scenes.' The available palace-aftermath pages mentioned include those involving a **stolen-clam** . . . "

[Author note: no clue round is retrieved.]

Iteration 1

```
search_by_text(
  query_text="palace aftermath crowed chore Time-and-a-Hap time
machine",
  question="What problem appears on the final palace page in the alley
oop sequence?"
)
```

> "The context **does not contain information** about an 'alley oop sequence' or what specific problem appears on the final palace page in that sequence. The retrieved memory contains many discussions about palace pages, crown sequences, throne offers, crowd scenes . . . "

[Author note: this is the query in which the model first names the time-machine storyline, and the answer it eventually gives. No clue round is retrieved either: the two searches together contribute CAR-

TOON_S63:R2, CARTOON_S13:R3, CARTOON_S65:R2, CARTOON_S37:R1, CARTOON_S59:R2, CARTOON_S38:R2, CARTOON_S53:R1, CARTOON_S79:R1 and CARTOON_S40:R2 to the context, none of them a clue round.]

Iteration 2

```
ask(
  session_name="CARTOON_S45",
  question="What was described as the final palace aftermath page after
the crowd and chore scenes?  What specific problem appears on it?"
)
```

> "Based on the provided context, I **cannot find specific details** about the final palace aftermath page after the crowd and chore scenes or what specific problem appeared on it. The conversation mentions palace pages involving power and . . . "

Iteration 3

```
ask(
  session_name="CARTOON_S57",
  question="What is described about the final palace aftermath pages?
What specific problem or scene appears on the last page of the palace
sequence?"
)
```

> "The final palace pages appear **deliberately ambiguous** in their organization . . . "

[Author note: no page is identified.]

Iteration 4

```
ask(
  session_name="CARTOON_S61",
  question="What specific problem or scene is shown in page 33 after Oop
returns his crown?  Does he carry anything that distinguishes it as the
final palace page?"
)
```
[Author note: the only session the agent chose that lies on the palace timeline, and the only one whose report is conclusive—about a page that comes after the answer.]

> "Based on the visual content of **page 33** . . . Following the apology sequence where Oop gives up the crown . . . "

[Author note: the report inlines page 33 here (79.png); the page is reproduced in Figure 9.]

Final answer (judge 0.0)

"**On page 35**, the **time machine** owned by Time-and-a-Hap malfunctions during its first test, accidentally materializing a muscular warrior instead of successfully summoning Oop from the past, causing significant panic."



Figure 9: Case 4: page 33, the page inlined by the report of iteration 4 (`79.png`). It follows the apology sequence in which Oop gives up the crown, so it comes *after* the last palace page the question asks about.

Analysis. It fails first because it cannot localise the decisive page from the summaries, and when it turns to the search tool, neither of the two searches retrieves the needed clue; the second search already injects the storyline that the answer later follows (the page 35 time machine), and the session-by-session walk ends on page 33—one page past the answer.

D.2.2 CASE 5: ANSWERING FROM THE INDEX ALONE

Question. "What brand appeared in the very last advertisement mentioned in the entire conversation?" *(MemEye brand_memory_test, idx 3)*. Ground truth: *Burger King*. The clue round is BRAND_S41:R1 (2024-05-26), whose user turn reads "This is the final BK piece, with surreal creative." and whose image caption reads "A whimsical tower of clay cows balances beside the **Burger King** Big Stacker logo."

Answer. UnfoldMem answers Coca-Cola and the judge scores it 0.0.

Table 17: Case 5: the last four sessions of the memory, in date order.

Session	Date	What the session is	Brand it names
BRAND_S39	2024-05-18	"the last Pepsi piece", with image	Pepsi
BRAND_S40	2024-05-22	"the final Starbucks piece", with image	Starbucks
BRAND_S41	2024-05-26	"the final BK piece", with image—**the clue round**	**Burger King**
BRAND_S42	2024-05-30	folder clean-up, no image: the user asks to verify two names on a "Share a Coke with a friend" picture	Coca-Cola

The agent made no tool call at all. The recorded trace has no entries and *context_round_ids* is empty, so no round—including the clue round—was read; the answer was composed from the session index alone.

UnfoldMem operation trace.

Question

What brand appeared in the very last advertisement mentioned in the entire conversation?

No tool call

(none). [Author note: the agent answers from the flat session index without retrieving anything.]

Final answer (judge 0.0)

Coca-Cola

Analysis. The failure is not that the memory lacks the fact but that the agent never issued an *ask*: with an empty *tool_trace* and no round ever opened, nothing was confirmed against the raw information, and the index alone cannot separate the last *advertisement* from the last session that names a brand.