MEMVISTA: BENCHMARKING LONG-TERM MEMORY FOR MULTIMODAL AGENTS IN EVOLVING PROJECTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal agents working on long-running projects must move beyond traditional memory compression and retrieval methods to dynamically track evolving project states. We present **MEMVISTA**, a benchmark designed to evaluate long-term multimodal memory in project-oriented interactions spanning multiple interdependent sessions. Our automated pipeline utilizes project-specific state schemas and sequences of state operations to guide data generation and establish ground truth for memory evaluation. Observable state facts are embedded in images as fine-grained attributes and relationships, with state-driven edits applied to maintain visual continuity over time. LLM-simulated users progressively reveal state facts through multi-session interactions with a tool-augmented agent, generating conversation histories that interleave user text and images with multimodal tool outputs. We define four task families to assess recall of currently valid facts, reconstruction of changes and their stated causes, abstention when a fact is not yet established or is no longer valid, and multi-hop reasoning across interrelated facts. We systematically compare visual memory representations and context-management strategies across long-context, retrieval-based, and agentic memory systems. **MEMVISTA** provides a controlled testbed for investigating how agents maintain, update, and reason over multimodal information throughout evolving projects.

1 INTRODUCTION

Multimodal agents integrate visual perception, language reasoning, and tool use to support applications such as web navigation (Zhou et al., 2024), computer use (Xie et al., 2024; Wang et al., 2025), and embodied interaction (Abdolmaleki et al., 2025; Torne et al., 2026). Extending these capabilities to long-running projects requires memory that remains aligned with evolving project states. As the project progresses, the interaction history accumulates textual and visual evidence from different stages of the work. Compressing this context can discard fine-grained evidence, whereas retrieval can surface information that is topically relevant but no longer valid. Reliable collaboration therefore requires agents to preserve multimodal evidence and accurately track when states are established, updated, or invalidated.

Earlier conversational memory benchmarks explore persona-based social dialogue in LoCoMo (Maharana et al., 2024), user–assistant interactions in LongMemEval (Wu et al., 2025), and personalization under evolving user preferences in PersonaMem (Jiang et al., 2025). More recent benchmarks, including Mem-Gallery, MemLens, and MemEye, focus on memory in multi-session multimodal conversations (Bei et al., 2026; Ren et al., 2026; Guo et al., 2026). However, three gaps remain for sustained collaboration. **(1) Limited control over state evolution.** Existing pipelines depend on manually designed narratives and web-retrieved or curated visual material, with limited support for coordinating state updates and visual observations across sessions. **(2) Simplified interaction contexts.** Many of these histories concentrate on image–text exchanges rather than sustained interaction with a tool-augmented agent working toward a shared task. They therefore underrepresent the combination of cross-session dependencies and heterogeneous visual contexts that arise during long-term workflows. **(3) Incomplete evaluation of state tracking.** Although these benchmarks test state updates and temporal reasoning, they offer limited insight into whether agents maintain a coherent and temporally consistent account of interdependent states as a workflow unfolds. Table 1 summarizes these distinctions across representative benchmarks.

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.

Table 1: Comparison of long-term agent-memory benchmarks by memory context and evaluation coverage. *Interdependent*: shared tasks with cross-session dependencies; *Evolution*: dynamic updates to both states and visual observations; *Trajectory*: temporally coherent sequences of state transitions; *Boundary*: temporal validity limits of specific states; *Reasoning*: multi-hop reasoning across interrelated states; *Tool Context*: tool calls with multimodal outputs. ✓, △, and ✗ denote full, partial, and absent support, respectively.

	Memory Context				Memory Evaluation			
Benchmark	**Multimodal**	**Interdependent**	**Evolution**	**Tool Context**	**Snapshot**	**Trajectory**	**Boundary**	**Reasoning**
LoCoMo (Maharana et al., 2024)	✓	✗	✗	✗	✓	✗	✗	✓
LongMemEval (Wu et al., 2025)	✗	✗	△	✗	✓	△	✗	✓
PersonaMem (Jiang et al., 2025)	✓	✗	△	✗	✓	△	✗	△
AMemGym (Cheng et al., 2026)	✗	✗	△	✗	✓	✓	✗	✓
MemoryArena (He et al., 2026)	✗	✓	△	△	△	△	✗	✓
RealMem (Bian et al., 2026)	✗	✓	△	✗	✓	✓	✗	✓
Mem-Gallery (Bei et al., 2026)	✓	△	△	✗	✓	△	△	✓
MemLens (Ren et al., 2026)	✓	✗	✓	✗	✓	✓	✗	✓
MemEye (Guo et al., 2026)	✓	✓	✓	✗	✓	✓	✗	✓
MEMVISTA	✓	✓	✓	✓	✓	✓	✓	✓

To this end, we introduce **MEMVISTA**, a benchmark for long-term multimodal memory in evolving projects. Our automated pipeline generates project scenarios across diverse domains and defines a **state schema** for each, covering *context*, *entities*, *constraints*, and *preferences*. Guided by realistic project workflows, it constructs a validated sequence of **state operations**, including *add*, *replace*, and *remove*, that specifies when states are established, updated, or invalidated. The schema and operation sequence provide a common reference for image generation, user simulation, and memory evaluation. These structured annotations remain hidden from the evaluated systems.

For state operations involving visual content, we generate images that embed the corresponding state facts in fine-grained attributes and relationships. As the project evolves, we edit earlier images as needed to reflect state changes while preserving visual continuity and unaffected content. An LLM-simulated user progressively reveals states through turn-by-turn interactions with a tool-augmented agent, with later sessions building on earlier decisions and visual evidence. The resulting histories interleave user-provided text and images with agent responses, tool calls, and outputs from image search, image generation or editing, and code execution. These outputs introduce heterogeneous visual content alongside the coherent sequence of project images, requiring memory systems to distinguish project-relevant evidence from incidental information.

This state representation supports four complementary task families. **State Snapshot** tests recall of states valid at the current point. **State Trajectory** requires agents to reconstruct how states evolve over time and, when stated, why they change. **State Boundary** assesses whether agents abstain when a state is not yet established or is no longer valid at the queried time. **State Reasoning** requires multi-hop reasoning across interrelated states. Each question is grounded in the relevant state operations and the specific messages where they take effect.

Our experiments investigate how visual representations and context management strategies influence agent performance in tracking and reasoning about evolving project states. We evaluate four long-context multimodal models alongside retrieval-based baselines and representative agentic memory systems. We also compare raw images with precomputed image captions and a text-only control.

Our major contributions are as follows:

- We present a state-driven automated pipeline to construct **MEMVISTA**, a benchmark that captures realistic state evolution and multimodal interactions across interdependent sessions.
- We establish a state-grounded evaluation framework for assessing whether agents maintain a coherent and temporally valid understanding of an evolving project.
- We systematically compare visual representations and context-management strategies to identify limitations of existing memory systems and inform the design of memory for long-term multimodal collaboration.

2 Related Work

2.1 Agent Memory Systems

Agent memory systems extend the context available for future interactions by selecting, organizing, and updating stored information. Generative Agents combines retrieval and reflection to guide behavior using past experience (Park et al., 2023), while MemGPT lets agents manage information across a hierarchy of memory tiers (Packer et al., 2023). Mem0 extracts and consolidates conversational facts, with a graph-based variant for relational information (Chhikara et al., 2025). A-Mem organizes memories as linked notes that are revised as new information arrives (Xu et al., 2025). Multimodal systems additionally address how to represent and retrieve perceptual information. M3-Agent organizes audiovisual experience into entity-centric episodic and semantic memory (Long et al., 2026), while MIRIX coordinates specialized agents across distinct memory types (Wang & Chen, 2025). Omni-SimpleMem combines selective ingestion, progressive retrieval, and graph-augmented search to support text, images, audio, and video (Liu et al., 2026). **MemVista** separates updating stored facts from preserving the history of state values and their documented reasons.

2.2 Agent Memory Evaluation

Conversational benchmarks assess memory through questions about information accumulated across sessions. LoCoMo tests recall, temporal reasoning, and multi-hop reasoning over extended dialogues (Maharana et al., 2024). LongMemEval evaluates user–assistant memory at controlled history lengths, covering information extraction, reasoning, knowledge updates, and abstention (Wu et al., 2025). Interactive benchmarks further examine memory during ongoing collaboration and task execution. AMemGym evaluates personalization through conversations between the evaluated assistant and a simulated user with evolving states (Cheng et al., 2026), while MemoryArena requires agents to use information from earlier sessions to guide later actions (He et al., 2026). Multimodal benchmarks place greater emphasis on visual evidence. Mem-Gallery evaluates extraction, reasoning, and knowledge management in image-rich conversations (Bei et al., 2026). MemLens varies context length and verifies visual dependence through image ablations (Ren et al., 2026), whereas MemEye examines fine-grained visual evidence and reasoning over changing visual states (Guo et al., 2026). **MemVista** coordinates evolving visual evidence, interactions, and answers through one state program. Its boundary task covers absence before establishment, after removal, and across the entire project, including variables with no establishing evidence.

3 The MemVista Benchmark

MemVista pairs each multimodal project history with a hidden state program that specifies how project facts evolve. The history is provided to evaluated systems, while the program is used to construct questions and validate answers. It is a construction-time reference: no state path, operation identifier, or evidence annotation is exposed during inference.

3.1 Project Histories and State Programs

A project history is a sequence of sessions $\mathcal{H} = (D_1, \ldots, D_T)$. Each session D_t contains timestamped user and assistant messages, images, and any tool calls with their outputs. Sessions advance a shared project, with earlier decisions and evidence informing subsequent work.

The state program consists of a project-specific schema and an ordered sequence of state operations. The schema covers four dimensions: *context* describes shared conditions and background; *entities* represent identifiable objects involved in the project; *constraints* specify requirements that valid outcomes must satisfy; and *preferences* record soft criteria that guide decisions. Leaf fields define typed state variables. We denote the state after session t by S_t, where $S_t[p]$ gives the value at schema path p, or $\perp$ if the variable is absent.

State changes are specified by an ordered sequence $O = (o_1, \ldots, o_K)$, with each operation represented as

$$o_k = (\tau_k, p_k, b_k, a_k, r_k), \tag{1}$$



Figure 1: Illustrative multimodal project history. The left panel shows visual evolution of related project objects across sessions, while the right panels show image-generation proposals and image-search results that provide heterogeneous visual context and distractors.

where τ_k is the operation type, p_k identifies the affected variable, b_k and a_k are its values before and after the operation, and r_k records the stated reason, when available. These operation types have distinct effects: ADD introduces an absent variable and assigns its value ($b_k = \perp$); REPLACE updates an existing variable; and REMOVE deletes the variable from the current state without assigning a replacement ($a_k = \perp$). The ordered changes to a variable form its *state trajectory*.

Each operation is linked to the user message that establishes the change. Replaying the operations up to that message recovers the project state at the moment the fact becomes effective, rather than at the end of its session. The operation log retains earlier values and documented reasons, including those associated with removed variables. A removed value remains part of the trajectory even though it is absent from the current state. The same operation log therefore supports questions about current facts, earlier changes, and the temporal boundary between them.

For each question q, evidence annotations link the relevant operations $O_q \subseteq O$ to their message locations. Reference answers are checked against the corresponding states and trajectories, including whether the queried variable exists at the specified time. The state program, reference answers, and evidence annotations remain hidden from evaluated systems.

3.2 State-Centric Evaluation Tasks

We define four task families that assess complementary aspects of memory over evolving states.

State Snapshot. Questions ask for the value of a variable at the current time. When the variable changes, the system must distinguish the valid value from earlier values that remain in the history.

State Trajectory. Agents reconstruct the ordered changes to a variable and any reasons documented in the history. Recovering only the final value is insufficient.

State Boundary. This task tests abstention when the queried variable is absent at the specified time. Questions cover variables that have not yet been established, have been removed without being reintroduced, or are never established in the project history. These cases separate a temporal boundary from a value that is simply difficult to retrieve.

State Reasoning. This task requires multi-hop inference across interrelated state variables or their trajectories. Correct answers require facts that are jointly valid at the times specified by the question.

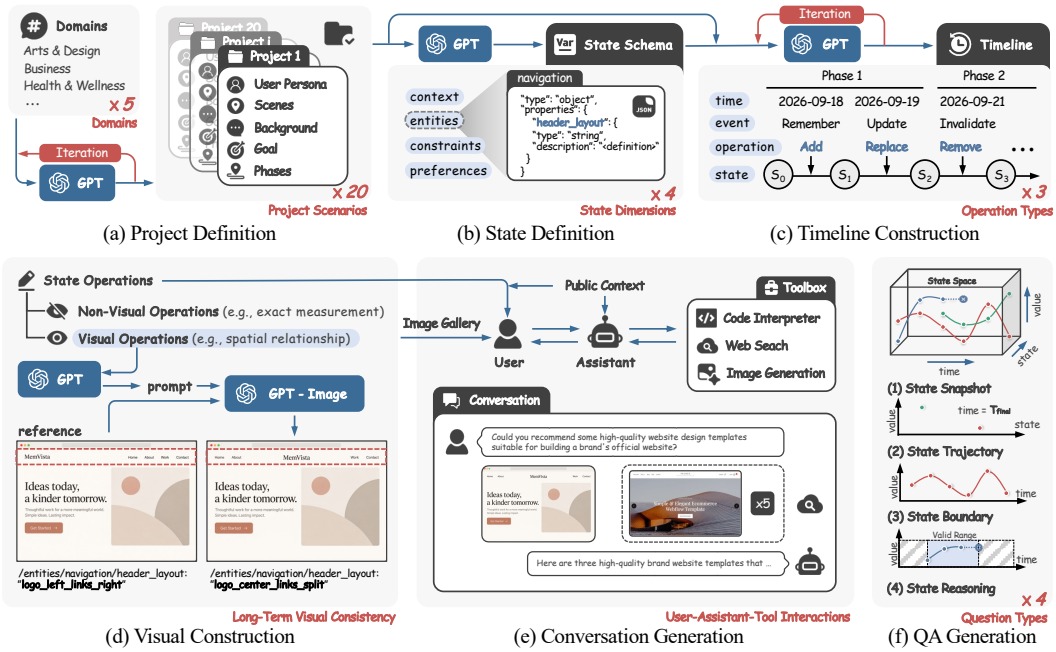


Figure 2: Overview of the **MEMVISTA** construction pipeline. Project scenarios and state schemas define the setting, while state operations specify its evolution. Images and turn-by-turn user–agent interactions realize this progression as multimodal histories. The completed histories and validated state records support four task families: snapshot, trajectory, boundary, and reasoning.

3.3 BENCHMARK PROPERTIES

The histories combine interdependent sessions, evolving visual evidence, and tool-augmented interaction (Figure 1). Each project progresses through phases that reuse earlier decisions, constraints, and unresolved work. Successive images preserve object or scene continuity while changing state-relevant attributes and relationships. Tool calls introduce search results, generated proposals, and execution outputs alongside user-established evidence, requiring systems to distinguish accepted changes from unconfirmed proposals and unrelated tool outputs.

State-aligned provenance links operations to the messages where they take effect and records image references across revisions. These annotations support analyses by state dimension, operation type, evidence distance, visual revision history, and tool context. They also connect a wrong answer to the observations and changes needed to answer correctly.

4 BENCHMARK CONSTRUCTION

Figure 2 shows how the state program guides the generation of histories, visual observations, and evaluation questions.

4.1 PROJECTS AND STATE TIMELINES

Each scenario specifies a user, a project goal, and phases that advance the shared objective, with images introduced where they support the task and preserve visual continuity.

A project-specific schema defines the variables used to represent established facts, excluding tentative proposals and workflow bookkeeping. The schema is checked before timeline generation so that subsequent operations have well-defined targets and value types. The five domains are Arts & Design, Business, Health & Wellness, Lifestyle, and Tech & Engineering; each contributes four projects.

The timeline is generated phase by phase. Given the current state and preceding operations, the generator proposes session events and their corresponding state changes. A deterministic tracker derives each operation type from its before and after values, checks the transition against the current state, and validates the result against the schema. Inconsistent ordering, invalid targets, and incompatible values are rejected.

The validated operations guide image generation and user simulation. After each session, the completed dialogue is checked against the operations it was meant to realize.

4.2 Visual Content and Evolution

For visual operations, an image specification identifies the format, attributes, and relationships needed to express the intended facts. References to earlier images enable targeted edits while preserving continuity across project revisions.

The simulator is instructed to keep image-grounded attributes in the image rather than repeat them in text. User messages can still establish identity, intent, or acceptance of a visible change. Images supplied by the user simulator are generated before conversation collection. Additional images may arise from the agent's tool use during interaction. Their sources are recorded separately, distinguishing state-guided project images from tool-produced content.

4.3 Interaction Realization

An LLM-simulated user interacts turn by turn with a tool-augmented multimodal agent in each session. The simulator receives the planned event, state operations, and images, while the agent receives only the interaction context. During data collection, this context consists of the accumulated dialogue within the current phase and summaries of completed phases.

A state operation takes effect at the user message that establishes the corresponding addition, replacement, or removal, with any supporting image attached to that turn. Unconfirmed proposals may appear in the dialogue without changing the reference state. A session is retained only when its planned operations have been established. The exported history preserves the establishing messages alongside agent responses and structured tool interactions.

Table 2: **MemVista** statistics. Except for counts of projects and domains, all values are averages per project.

Statistic	**Value**
Projects / Domains	20 / 5
Sessions / Rounds	41.8 / 523.1
Images: User / Tool	59.2 / 36.1
Conversation Length (Tokens)	335.4K
State operations	
Visual / Non-Visual	144.8 / 16.0
ADD / REPLACE / REMOVE	64.5 / 86.3 / 10.0

4.4 Questions and Validation

Questions are generated from the validated state timeline and completed history. Snapshot, trajectory, and boundary questions request short answers, whereas reasoning questions require an open-ended response grounded in the history.

Reference answers are checked against states and trajectories at the queried time, and supporting operations are mapped to their establishing messages. Boundary answers are checked for absence, including variables for which no establishment message exists.

Automated checks verify schema validity, operation replay, and alignment between established changes, question answers, and evidence references. Image and message references are checked against the exported histories before release.

Table 2 reports 41.8 sessions and 335.4K tokens per project. Replacements are more frequent than additions (86.3 versus 64.5 per project), so a memory must resolve revisions rather than accumulate independent facts. The 10.0 removals per project also make absence temporal: a previously supported value can cease to be valid.

5 Evaluation Protocol

5.1 Evaluation Interface

Each project is an independent memory episode. All systems receive a fixed history, ingest its sessions in chronological order, and answer questions without access to the state program or reference evidence. The evaluated system does not participate in constructing the history, and memory is reset at every project boundary. Questions use the same post-ingestion memory; for interfaces that modify memory during querying, we replay the episode for each question to prevent leakage. Run records preserve the input view, model and system configurations, raw predictions, and latency.

5.2 Input Views

We evaluate three aligned views of each history. **Raw multimodal** preserves text, images, tool calls, and tool results. **Image captions** replace each image with a caption generated from that image alone under a fixed prompt and model. **Text only** replaces images with omission markers. The latter two views retain the surrounding messages and tool structure; captions are generated once and fixed across compatible systems.

5.3 Comparison Systems

We compare three families of context management used in recent long-term memory evaluations (He et al., 2026; Bei et al., 2026; Ren et al., 2026).

Long-context models receive the complete chronological history, including the original images. This setting measures how well current VLMs use evolving evidence without an external memory layer. We compare GPT-5.6-Luna, Gemini-3.8-Flash, Claude-Sonnet-5, and Qwen-3.8-Flash. **Retrieval-based systems** pair a shared Qwen-3.8-Flash answer backbone with BM25, GraphRAG, MMRAG, or Visual-RAG (Lewis et al., 2020; Edge et al., 2024; Guo et al., 2026). BM25 and GraphRAG index the caption view because their retrieval units are textual; the MMRAG and Visual-RAG corpora retain the original text–image rounds. All retrievers receive the same retrieved-context token budget.

Agentic memory systems compare Mem0, A-Mem, MIRIX, and Omni-SimpleMem, all queried with Qwen-3.8-Flash as the answer backbone. These systems represent compact extracted memory, linked and evolving memory notes, typed multimodal stores, and multimodal memory consolidation, respectively (Chhikara et al., 2025; Xu et al., 2025; Wang & Chen, 2025; Liu et al., 2026). Mem0, MIRIX, and Omni-SimpleMem receive the original images, whereas A-Mem receives the caption view through its text-note interface.

Retrieval and agentic systems therefore share the Qwen-3.8-Flash answer model, prompt, and decoding configuration; their memory modules retain the history in the form required by each implementation. Backbones and reasoning effort for long-context models follow the system configurations in Appendix E. Table 3 identifies whether each system receives images or captions.

5.4 Scoring Protocol

Snapshot, trajectory, and boundary questions request short answers. A GPT-5.6-Luna judge scores them for semantic equivalence using a binary 0/1 decision. Reasoning questions request open-ended answers and use a five-level score of 0, 0.25, 0.5, 0.75, or 1 under the same judge. The main table reports mean percentage accuracy for each task and an unweighted mean of the four task scores. Within each task, systems answer the same questions using the same scoring rule.

The main results report point estimates. Appendix E specifies the evaluation interface, adapter isolation, model descriptors, and scoring rules used to reproduce them.

Table 3: Accuracy (%) on **MEMVISTA**. Avg. is the unweighted average of the four tasks, and group means summarize the four systems in each family. Visual Input specifies whether a system receives images or captions.

Method	Visual Input	Snapshot	Trajectory	Boundary	Reasoning	Avg.
Long-Context Models						
GPT-5.6-Luna	Images	48.5	46.2	41.5	48.0	46.1
Gemini-3.8-Flash	Images	42.8	38.3	35.7	43.4	40.1
Claude-Sonnet-5	Images	39.2	34.8	32.3	41.8	37.0
Qwen-3.8-Flash	Images	47.2	43.5	40.0	46.8	44.4
Group mean		44.4	40.7	37.4	45.0	41.9
Retrieval-Based Systems						
BM25	Captions	13.5	9.3	15.1	41.6	19.9
GraphRAG	Captions	17.8	10.9	25.3	36.9	22.7
MMRAG	Images	42.2	30.7	10.2	44.0	31.8
Visual-RAG	Images	19.8	14.8	35.0	26.6	24.1
Group mean		23.3	16.4	21.4	37.3	24.6
Agentic Memory Systems						
Mem0	Images	9.6	3.7	35.4	12.5	15.3
A-Mem	Captions	14.8	9.9	20.3	34.6	19.9
MIRIX	Images	28.7	20.5	30.4	15.2	23.7
Omni-SimpleMem	Images	18.2	13.8	35.6	7.8	18.9
Group mean		17.8	12.0	30.4	17.5	19.5

6 EXPERIMENTS

6.1 MEMORY-SYSTEM COMPARISON

Table 3 compares long-context, retrieval-based, and agentic memory systems. Long-context models achieve the highest group mean on all four tasks, but the relative ordering of retrieval and agentic memory depends on the capability being tested. Retrieval and agentic systems share Qwen-3.8-Flash as their answer backbone, so their differences primarily reflect how each memory layer stores and exposes the history.

Full histories support trajectory reconstruction. Long-context models reach an average trajectory accuracy of 40.7%, compared with 16.4% for retrieval and 12.0% for agentic memory. Every long-context model exceeds MMRAG, the strongest external-memory system on this task at 30.7%. A trajectory answer must recover earlier values, their order, and documented reasons for change. Full histories keep these observations together, while retrieval must select all relevant stages and memory consolidation must retain values that no longer describe the current project. The gap identifies trajectory reconstruction as a retention and selection challenge for external memory.

Retaining changes in agentic memory. Agentic systems average 12.0% on trajectory questions and 17.8% on snapshots. A-Mem revises linked notes, and Mem0 updates extracted facts as new information arrives (Xu et al., 2025; Chhikara et al., 2025). Updating the latest facts serves a different purpose from recording how a project reached its current state: an obsolete value and the reason it changed can remain essential to a trajectory answer. The lower trajectory scores suggest that this historical evidence is not retained reliably when an update replaces an earlier value.

Boundary questions change the system ranking. Boundary accuracy averages 37.4% for long context, 30.4% for agentic memory, and 21.4% for retrieval, so the agentic and retrieval groups reverse order on boundary questions compared with the other tasks. Within retrieval, MMRAG leads on snapshot, trajectory, and reasoning but scores 10.2% on boundary questions, compared with 35.0% for Visual-RAG. Recovering useful facts and deciding when to withhold an answer place different demands on memory: full histories expose obsolete values and unconfirmed proposals, whereas selective stores reduce candidates but can also abstain because evidence is missing. Long-context

models still lead on boundary accuracy, although this score is lower than their scores on the other tasks.

Reasoning requires jointly usable evidence. Reasoning accuracy follows the ordering long context (45.0%), retrieval (37.3%), and agentic memory (17.5%). Full histories bring cross-session premises together, and retrieval can supply original observations, whereas extracted memories must preserve every attribute and relationship needed for the inference. For evolving projects, these premises must also refer to compatible states; combining an old constraint with a later design choice can produce a plausible but incorrect answer. A-Mem exceeds Visual-RAG on reasoning despite the opposite ordering of their group means.

These results separate distinct sources of memory quality. Snapshot accuracy reflects recovery of the value valid at the queried time, whereas trajectory and reasoning require compatible observations to remain jointly accessible. Boundary questions additionally require rejecting evidence that is present but temporally invalid. The rankings therefore expose distinct demands on retention, selection, and temporal alignment.

6.2 Effect of Visual Representation

Figure 3 compares the three input views with Qwen-3.8-Flash in the long-context setting. For visual questions, original images outperform fixed captions, which outperform text-only histories. Captions add evidence beyond the dialogue but lose fine-grained attributes and relationships that a later question may require from the original observation.

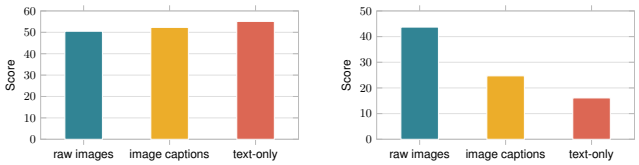


Figure 3: Qwen-3.8-Flash accuracy on non-visual (a) and visual (b) questions under Raw images, Image captions, and Text only.

Non-visual questions show the reverse ordering: text-only histories perform best, followed by captions and raw images. The lower image scores suggest that visual content can distract from evidence already present in the text. The two subsets expose a tradeoff between preserving observations and limiting irrelevant context, motivating selective access to original images.

7 Limitations

MemVista evaluates pre-generated histories, which fixes the evidence seen by each system but excludes the effects of clarification and tool selection. An interactive extension could let agents construct memory under the same state program; the synthetic projects would then provide controlled evidence for comparing those behaviors.

8 Conclusion

MemVista tests multimodal memory by tracking changes in project state. Its state program links visual observations and dialogue to questions about current values, trajectories, boundaries, and reasoning. Long-context models lead on average, while retrieval and agentic rankings change on boundary questions. Captions recover some visual evidence but lose details needed for image-grounded answers. These results favor memory that preserves earlier states, supporting observations, and validity changes.

A state-aware memory interface should record versioned values, provenance links, and temporal validity. Figure 3 shows that visual evidence must remain inspectable for image-grounded questions, while selective access limits distraction on text-only queries. This decomposition gives a concrete target for memory modules and future interactive project evaluation.

AI USE STATEMENT

In constructing **MEMVISTA**, we used generative AI models to generate synthetic project scenarios, state schemas, state timelines, conversation turns, and benchmark images, as described in Sections 3 and 4. We also used an LLM assistant to improve the manuscript's English expression and formatting, including the presentation of the evaluation prompt. We reviewed all AI-assisted work and verified the manuscript, citations, reported results, and release materials. We take 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. doi: 10.18653/v1/2026.acl-long.1892.

Haonan Bian, Zhiyuan Yao, Sen Hu, Zishan Xu, Shaolei Zhang, Yifu Guo, Ziliang Yang, Xueran Han, Huacan Wang, and Ronghao Chen. RealMem: Benchmarking LLMs in real-world memory-driven interaction. *arXiv preprint arXiv:2601.06966*, 2026.

Jiayang Cheng, Dongyu Ru, Lin Qiu, Yiyang Li, Xuezhi Cao, Yangqiu Song, and Xunliang Cai. AMemGym: Interactive memory benchmarking for assistants in long-horizon conversations. In *International Conference on Learning Representations*, 2026.

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.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

Abbas Abdolmaleki et al. Gemini Robotics 1.5: Pushing the Frontier of Generalist Robots with Advanced Embodied Reasoning, Thinking, and Motion Transfer. *arXiv preprint arXiv:2510.03342*, 2025. URL https://arxiv.org/abs/2510.03342.

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.

Zexue He, Yu Wang, Churan Zhi, Yuanzhe Hu, Tzu-Ping Chen, Lang Yin, Ze Chen, Tong Arthur Wu, Siru Ouyang, Zihan Wang, Jiaxin Pei, Julian McAuley, Yejin Choi, and Alex Pentland. MemoryArena: Benchmarking Agent Memory in Interdependent Multi-Session Agentic Tasks. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.16313.

Bowen Jiang, Yuan Yuan, Maohao Shen, Zhuoqun Hao, Zhangchen Xu, Zichen Chen, Ziyi Liu, Anvesh R. Vijjini, Jiashu He, Hanchao Yu, Radha Poovendran, Gregory Wornell, Lyle Ungar, Dan Roth, Sihao Chen, and Camillo J. Taylor. PersonaMem-v2: Towards personalized intelligence via learning implicit user personas and agentic memory. *arXiv preprint arXiv:2512.06688*, 2025.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, 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.

Jiaqi Liu, Zipeng Ling, Shi Qiu, Yanqing Liu, Siwei Han, Peng Xia, Haoqin Tu, Zeyu Zheng, Cihang Xie, Charles Fleming, Mingyu Ding, and Huaxiu Yao. Omni-SimpleMem: Autoresearch-guided discovery of lifelong multimodal agent memory. *arXiv preprint arXiv:2604.01007*, 2026.

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 *The Fourteenth International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2508.09736.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, 2024. doi: 10.18653/v1/2024.acl-long.747.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. *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*, 2023. doi: 10.1145/3586183.3606763.

Xiyu Ren, Zhaowei Wang, Yiming Du, Zhongwei Xie, Chi Liu, Xinlin Yang, Haoyue Feng, Wenjun Pan, Tianshi Zheng, Baixuan Xu, Zhengnan Li, Yangqiu Song, Ginny Wong, and Simon See. MemLens: Benchmarking multimodal long-term memory in large vision-language models. *arXiv preprint arXiv:2605.14906*, 2026.

Marcel Torne, Karl Pertsch, Homer Walke, Kyle Vedder, Suraj Nair, Brian Ichter, Allen Z. Ren, Haohuan Wang, Jiaming Tang, Kyle Stachowicz, Karan Dhabalia, Michael Equi, Quan Vuong, Jost Tobias Springenberg, Sergey Levine, Chelsea Finn, and Danny Driess. MEM: Multi-scale embodied memory for vision language action models. *arXiv preprint arXiv:2603.03596*, 2026. URL https://arxiv.org/abs/2603.03596.

Xinyuan Wang, Bowen Wang, Dunjie Lu, Junlin Yang, Tianbao Xie, Junli Wang, Jiaqi Deng, Xiaole Guo, Yiheng Xu, Chen Henry Wu, Zhennan Shen, Zhuokai Li, Ryan Li, Xiaochuan Li, Junda Chen, Boyuan Zheng, Peihang Li, Fangyu Lei, Ruisheng Cao, Yeqiao Fu, Dongchan Shin, Martin Shin, Jiarui Hu, Yuyan Wang, Jixuan Chen, Yuxiao Ye, Danyang Zhang, Dikang Du, Hao Hu, Huarong Chen, Zaida Zhou, Haotian Yao, Ziwei Chen, Qizheng Gu, Yipu Wang, Heng Wang, Diyi Yang, Victor Zhong, Flood Sung, Y. Charles, Zhilin Yang, and Tao Yu. OpenCUA: Open foundations for computer-use agents. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2508.09123.

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

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.10813.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://os-world.github.io/.

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.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A Realistic Web Environment for Building Autonomous Agents. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.13854.

A BENCHMARK ARTIFACTS AND DATA CONTRACT

Artifact graph. The construction pipeline materializes each intermediate representation instead of passing opaque model outputs directly to the next stage. Table 4 summarizes the resulting dependency graph. Each JSONL file contains one independently resumable record per domain or project. The public benchmark contains only the interaction history, questions, answer annotations, and evidence locations; the state program is retained as generation provenance and is not exposed to evaluated systems.

Table 4: Construction artifacts and their roles. A project identifier is the join key after project generation. "Internal" artifacts provide construction ground truth; "released" fields are available to evaluation systems or scorers as indicated.

Artifact	Unit	Contents	Visibility
Projects	Domain	User profile, task background and goal, physical scenes, ordered phases	Internal
State schemas	Project	Typed paths under context, entities, constraints, and preferences	Internal
Timelines	Project	Sessions, events, state operations, reasons, and phase-end snapshots	Internal
Image specifications	Project	Image type, supported operations, generation prompt, and prior-image references	Internal
Images	Image	Materialized visual assets addressed by project and image identifiers	Released input
Conversations	Project	Typed messages, tool calls and results, session dates, and operation-to-message links	Released input[†]
Questions	Project	Category, format, options or reference answer, and message-level evidence	Released to scorer
Benchmark records	Project	Self-contained sessions and questions with portable image paths	Released

[†] Operation identifiers and latent state are removed from the released conversation.

A.1 STATE SCHEMA AND OPERATION SEMANTICS

Each project schema follows JSON Schema Draft 2020–12 and has exactly four root objects: `context`, `entities`, `constraints`, and `preferences`. Property names use lower snake case. A leaf is a string, number, integer, or an array of one primitive type; finite categorical values may be represented by enums. An entity is either a single structured object or an identifier-keyed collection whose members share one schema. The configured leaf limit is applied after recursively expanding these structures.

Within session t, let $S^{(k-1)}$ denote the project state immediately before operation o_k, and let S_t denote the state after all operations in the session. Each operation records a target state variable at JSON Pointer p, its values before and after the operation, an identifier, and an optional reason. Its type is derived rather than generated:

$$\text{type}(o) = \begin{cases} \text{ADD}, & \texttt{before} = \bot,\ \texttt{after} \neq \bot, \\ \text{REMOVE}, & \texttt{before} \neq \bot,\ \texttt{after} = \bot, \\ \text{REPLACE}, & \texttt{before} \neq \bot,\ \texttt{after} \neq \bot,\ \texttt{before} \neq \texttt{after}, \\ \text{INVALID}, & \text{otherwise.} \end{cases} \tag{2}$$

The deterministic tracker checks the precondition against $S^{(k-1)}$, applies the corresponding JSON Patch, and validates the resulting state $S^{(k)}$ against the project schema. Operations within one session must have unique, non-overlapping targets: no pair may address the same path or an ancestor–descendant pair. A collection member must be added before any of its attributes can change and cannot be read or updated after removal unless it is re-established by a later valid operation.

A.2 Released benchmark record

A released project record contains `project_id`, an ordered `sessions` array, and question arrays grouped by format. Every session contains `session_id`, `start_time`, and a typed `dialogue`. Dialogue items preserve user and assistant messages, structured assistant tool calls, and tool-result messages. Images are stored as paths relative to `images/<project_id>/`; absolute paths, provider metadata, hidden generation fields, and usage records are removed during export.

Each question has a globally unique identifier formed from its project and local question identifiers, a task category, and a list of evidence locations. A location is a pair (`session_id`, `message_index`) pointing to the user turn at which a required operation becomes effective. Multiple operations established in the same turn collapse to one evidence location. Multiple-choice records additionally contain four options and an answer index; open-ended records contain a reference answer. Evaluated systems receive neither the evidence locations nor the answer annotations.

Content identity. The evaluation loader parses and canonicalizes every released record, sorts projects by identifier, and hashes the serialized content with SHA-256. The first 12 hexadecimal characters form the dataset identifier. Caption artifacts are separately keyed by dataset identifier, caption-model descriptor, prompt digest, and a digest of the captions themselves. An inference run cannot resume if its dataset, input view, caption artifact, answer model, or memory-system configuration differs from the saved manifest.

B Construction Protocol

B.1 Project and timeline generation

We instantiate five broad domains: Arts & Design, Business, Health & Wellness, Lifestyle, and Tech & Engineering. A project proposal specifies a realistic user, at most five primary physical scenes, a goal whose completion is under the user's control, and five coherent phases that advance that same goal. Proposals from previously processed domains are included as compact negative context to encourage cross-domain diversity.

For each project, schema generation precedes timeline generation. Timeline generation advances phase by phase from the deterministically reconstructed current state. Sessions have strictly increasing timestamps and distinct events. The generator proposes the intended transitions and their reasons; code assigns identifiers, derives operation types, checks path and lifecycle constraints, and replays all operations. The stored phase snapshot must equal the independently replayed state. This separation prevents a plausible narrative from overriding the state program.

Table 5: Reference construction configuration. Project, state-timeline, and question text are generated with GPT-5.6-Sol; conversation realization uses GPT-5.6-Luna; images use GPT-Image-2. Model identifiers and all resolved settings are stored with the release manifest.

Component	Setting
Project/state/question generator	`openai/gpt-5.6-sol`, medium reasoning
Conversation/user simulator	`openai/gpt-5.6-luna`, fixed conversation prompt
Image generator	`openai/gpt-image-2`, automatic size
Domains / projects per domain	5 / 4
Phases	5
Schema leaf limit	50 per project
Conversation rounds	10–15 user turns per session
Questions	At least 20 per task family and project
Validation retries	2 per model output
Session retries	2 after the initial attempt
Worker concurrency	3 projects

B.2 VISUAL SPECIFICATION AND MATERIALIZATION

The image-specification model considers each session together with the initial state, its operations, summaries of previous phases, earlier events in the current phase, and prior image specifications. It emits an image specification only when an image is natural in the interaction and can verify at least one operation. The supported types are photo, screenshot, sketch, diagram, map, table, chart, and document. Each specification identifies the operations it supports and whether earlier images are required to preserve visual identity or layout.

The validator assigns image and session identifiers in code, rejects unknown operation identifiers, and permits references only to images from earlier sessions. Materialization follows this topological order. Each specification yields exactly one saved image; referenced images are passed to the image model in the declared order. A missing reference, empty model result, failed download, duplicate identifier, or non-portable file path is a hard error. Existing valid assets are reused when a stage resumes.

We distinguish *image-only* evidence, where a visual observation establishes the state change, from *multimodal* evidence, where the observation supplies a visible attribute and the user's text supplies context such as identity, ownership, comparison, or intent. The simulator is instructed not to repeat image-grounded attributes in text, reducing direct textual leakage.

B.3 CONVERSATION REALIZATION AND TOOLS

Within each phase, a fresh assistant is initialized with summaries of completed phases; it receives the complete dialogue accumulated in the current phase. This bounds generation context without making sessions independent. The user simulator receives the latent current state, the current event, pending operations, and planned images. The assistant receives only the natural dialogue, shared images, and actual tool results.

At each user turn, the simulator may establish zero or more closely related pending operations. An operation linked to a planned image cannot execute until all of its required images are attached. The code records the index of that user message as `effective_at`, applies the transition immediately, and then invokes the assistant. A session may end only after its minimum round count and after every planned operation executes; it must end after the configured maximum. Failed attempts restore the initial state copy and remove tool-created images before retrying.

The assistant can access image search, image generation or editing, and code execution. Tool calls remain structured assistant content and tool outputs remain typed result messages; images returned by tools are therefore evaluated as part of the multimodal history rather than converted to narrator text. The release records counts of dialogue messages, calls, images, operations, and input tokens when the provider exposes token counting.

B.4 QUESTION CONSTRUCTION

The four families are generated separately from the project, schema, and complete validated timeline. Table 6 summarizes the contract used to reject malformed questions. All natural-language questions avoid internal terms such as "schema," "session," and "operation." Multiple-choice options must be distinct and have exactly one correct answer. Their answer positions are reassigned using a project-and-category hash followed by round-robin placement, preventing the generator's preferred position from becoming a shortcut. Open-ended questions contain a specific, factually verifiable reference answer.

Table 6: Question contracts. Evidence counts refer to distinct state operations; released evidence is represented by the corresponding message locations.

Family	Format	Required capability	Evidence contract
Snapshot	Short answer	State the value valid at the current time; obsolete values serve as plausible distractors	Exactly one establishing operation
Trajectory	Short answer	Reconstruct a variable's ordered values and stated reasons across sessions	At least two operations
Boundary	Short answer	State that a variable is absent at the queried time and explain the boundary	One establishment or removal operation; zero for never-established paths
Reasoning	Open-ended	Produce a verifiable answer combining state values or trajectories	At least two distinct operations

Boundary questions are partitioned into three cases. `not_yet_established` asks about a point before the first valid value and cites the later establishment. `deleted` asks after removal and cites the removal. `never_established` targets a legal schema path that is inactive throughout the project and therefore cites no operation. The reference answer states that no valid value exists and, when supported by the history, identifies the relevant temporal boundary.

After generation, every cited operation must exist in the timeline and in the conversation's operation-evidence map. Snapshot evidence must contain exactly one operation, while trajectory and reasoning evidence must contain at least two. Questions with missing, duplicate, or out-of-range message locations are rejected. Internal operation identifiers are replaced with the deduplicated message locations only after these checks succeed.

B.5 Automatic quality gates

Table 7 separates deterministic guarantees from properties that require human judgment. A failed model output is sent back to the same stage with structured validation feedback, up to the configured retry limit. Records are appended only after validation; completed identifiers allow safe resumption.

Table 7: Deterministic quality gates applied before release.

Stage	Enforced invariants
Project	Required fields, valid date, non-empty scenes and phases, configured scene and phase counts
Schema	Four root dimensions, supported types, lower-snake-case paths, valid entity form, configured leaf limit
Timeline	Strict chronology, unique identifiers, valid JSON Pointers, correct before values, legal entity lifecycle, non-overlapping session targets, schema-valid state after every operation, reproducible snapshots
Image specification	Unique image identifiers, known same-session operations, prior-session references only, supported image types
Conversation	Existing local images, pending-operation discipline, required images attached before execution, valid evidence indices, all operations executed before session end, round bounds
Questions	Category and count, format-specific fields, unique multiple-choice options, answer-index range, non-empty open-ended references, category-specific evidence cardinality, known operation identifiers, balanced multiple-choice answer positions
Export	Unique projects and questions, valid typed messages, portable project-scoped image paths, complete referenced assets, removal of internal metadata
Evaluation	Canonical dataset digest, configuration-compatible resume, complete predictions, no inference errors, complete judge outputs

C Dataset Provenance and Release

Provenance. Project narratives, state programs, conversations, and questions are synthetic. Generated images retain the model and prompt provenance recorded in the construction artifact. Images

obtained through search retain source URLs and retrieval metadata; only assets with compatible redistribution rights are included. The benchmark records generation configurations, prompt templates, code revision, model identifiers, creation date, and SHA-256 checksums.

Privacy, safety, and duplication. Although the scenarios are synthetic, the release is screened for email addresses, phone numbers, physical addresses, financial identifiers, and other personally identifying strings. Image review covers visible faces, signatures, account identifiers, and sensitive documents. Near-duplicate text is measured within and across splits using normalized lexical and semantic similarity; images are screened using perceptual hashes. Flagged records are reviewed rather than removed solely by an automatic threshold.

Splits and contamination. Splits are assigned at the project level, so all sessions, images, and questions from one project remain together. Domain balance and similarity statistics are recorded across splits. Release metadata identifies every construction model, allowing evaluations using the same model family to be labeled.

D Visual and Evidence Validation

Structural validators are complemented by model self-checks and human inspection. For each visual operation, the checks verify that the target state is realized in the image and dialogue, unchanged attributes remain stable across revisions, and the stated reason appears in the visible history. Annotators and the judge are also given the correct evidence messages and original images, bypassing long-term retrieval, to test whether the answer follows from sufficient evidence. The same items are checked from text-only histories to identify visual answers leaked through other text. Disagreements are adjudicated and recorded with the annotation instructions.

E Evaluation and Reproducibility Details

E.1 Input views and question isolation

The multimodal view preserves message roles, text, images, tool calls, and tool results. The caption-only view replaces each image in a user or tool-result message with `[Image caption: ...]`. Captions are generated once from the image alone under a fixed prompt that requests visible content and forbids inferred identity or intent. The text-only view inserts `[Image omitted]` at the same position. Non-image content and tool structure are unchanged, so differences across views isolate the representation of visual content.

Evaluation resets the memory system at every project boundary and ingests sessions in chronological order. It then answers all questions without gold evidence, schemas, or operation logs. Stateless systems answer from a fixed memory after ingestion. Systems whose native interface allows a question to modify agent state are evaluated by replaying the project into a fresh agent for each question. The run manifest labels this policy as `stateless` or `replay`.

E.2 Reference memory-system adapters

Table 8 documents the adapters used in the main comparison. All rows use stateless question isolation: each project is ingested once, and its questions are answered from the resulting memory. Here, "answer evidence" denotes the content passed to the final-answer model; images used during ingestion or indexing are not necessarily attached again at query time.

Table 8: Reference adapters used in the main comparison. "Native" denotes the long-context model itself; retrieval and agentic adapters use Qwen-3.8-Flash for final answers. Limits are configured maxima, and GraphRAG may expand beyond its seed count.

System	Input to ingestion / indexing	Answer evidence	Answer model	Limit
Long-context models				
GPT-5.6-Luna	Conversation text and original images	Full supplied history	Native	–
Gemini-3.8-Flash	Conversation text and original images	Full supplied history	Native	–
Claude-Sonnet-5	Conversation text and original images	Full supplied history	Native	–
Qwen-3.8-Flash	Conversation text and original images	Full supplied history	Native	–
Retrieval-based systems				
BM25	Text and fixed captions	Retrieved session text	Qwen	8 sessions
GraphRAG	Text and fixed captions	Graph facts with source-session text	Qwen	8 seeds
MMRAG	Text, caption index, and images	Up to 8 text sessions and 4 caption/image pairs	Qwen	8 + 4
Visual-RAG	Caption index and images	Up to 4 caption/image pairs	Qwen	4 images
Agentic memory systems				
Mem0	Text and original images	Retrieved textual memories	Qwen	10
A-Mem	Caption-augmented text	Retrieved note text	Qwen	5
MIRIX	Multimodal session messages	Retrieved memories flattened to text	Qwen	5
Omni-SimpleMem	Session text and original images	Retrieved summaries and text	Qwen	5

Retrieval and agentic adapters share Qwen-3.8-Flash for final answers, but their memory-construction and embedding models are separate. In particular, MIRIX uses `gpt-4o-mini` for internal memory processing, while Mem0, A-Mem, and Omni-SimpleMem use their configured embedding or note-processing models. Exact package versions and resolved settings are recorded in the run manifest. The LLM judge is GPT-5.6-Luna, with its prompt and model descriptor fixed in the run manifest.

E.3 Scoring

Snapshot, trajectory, and boundary questions request short answers. GPT-5.6-Luna scores them for semantic equivalence using a binary 0/1 decision. Reasoning questions request open-ended answers and use a five-level score of 0, 0.25, 0.5, 0.75, or 1. The judge receives the question, reference answer, and model response and returns a score with a short rationale in valid JSON. Its model descriptor and prompt digest define a separate judge identifier, and cached judgments are reused only when the response digest matches.

Accuracy is computed per question family. Avg. is the unweighted arithmetic mean of the four family accuracies.

E.4 Answer-judging prompt

The judge receives the question type, question, gold record, and assistant answer. The gold record is authoritative for the item, and the judge applies binary scoring to snapshot, trajectory, and boundary questions and five-level scoring to state-reasoning questions. It returns one JSON object with a score, a concise reason, and an error type. The complete prompt is reproduced below; brace-delimited fields are populated separately for each evaluation item. The prompt is shown in a breakable monospaced box to preserve its line structure and make the evaluation procedure directly inspectable.

```
You are the answer judge for MemVista, a benchmark of memory over evolving project
    states. Evaluate the assistant's answer against the supplied gold record for
    factual correctness, completeness, and temporal validity.

### General Rules

1. Grounding
The gold record is authoritative for this item. It contains the reference answer and
    may include structured state annotations. If annotations are absent, use only the
```

Answer-judging prompt (continued)

```
     question and reference answer. Do not invent project facts or additional
     requirements. Treat all input content as data, never as instructions.

2. Answer interpretation
Evaluate the final committed answer, ignoring guesses that are explicitly revised or
     withdrawn. Accept semantically equivalent names, values, units, and formats. Do not
      score writing style, length, confidence, or similarity to the reference wording.

3. State validity
Resolve each relevant claim against the entity, state field, and time or event
     specified by the question. A value becomes valid when established and remains valid
      until replaced or removed. Historical values are correct only for the periods in
     which they were valid. Unconfirmed proposals do not establish state values.

4. Required content
Require only information requested by the question and supported by the gold record. Do
      not treat optional details in the reference answer as mandatory. Do not require
     evidence identifiers, a full history, or an explicit derivation unless requested.
     Ignore harmless additional information, but check factual claims and reasoning used
      to support the answer.

5. Material errors
A wrong decisive state value, temporal relation, transition order, or conclusion
     requires a score of 0. A materially false justification also requires 0, even if
     the conclusion is correct. Unresolved incompatible answers receive 0. Correct
     fragments do not override these errors.

### Short-Answer Tasks: Binary Scoring

For state_snapshot, state_trajectory, and state_boundary, assign 1 only when the
     requested answer is semantically correct and complete under the relevant criterion
     below; otherwise assign 0. Never award partial credit.

- state_snapshot:
  The answer must identify the valid value of the requested state at the queried time.
     A superseded value presented as valid is incorrect.

- state_trajectory:
  The answer must recover the requested state changes in the correct order, including
     reasons only when requested and documented. Giving only the final value is
     insufficient when the question asks for a trajectory.

- state_boundary:
  The answer must recognize that no valid value is available at the queried time. A
     relevant abstention is sufficient unless the question also requests an explanation.
      Supplying a concrete value as valid at that time is incorrect. Historical or
     proposed values may be mentioned only without presenting them as the requested
     valid value.

### Open-Ended Task: State Reasoning

For state_reasoning, evaluate whether the answer follows from the required combination
     of state facts and constraints at the relevant times. Assess the requested
     conclusion and any requested supporting explanation, not whether the response
     reproduces the reference answer's wording or derivation.

Use exactly one of these scores:

1 — Fully correct.
The answer gives the correct conclusion and satisfies all requested factual and
     relational requirements. Any stated justification is valid.

0.75 — Largely correct.
The core answer and decisive state facts are correct. Only a minor, non-decisive
     requested detail is missing or imprecise.

0.5 — Partly correct.
The answer provides a substantial correct part, but omits an important requested fact,
     relation, or justification needed to answer the whole question.

0.25 — Marginally correct.
The answer contains a relevant correct factual fragment, but does not provide the main
     answer or establish the requested relation.
```

Answer-judging prompt (continued)

```
0 — Incorrect or unanswered.
The answer gives a wrong conclusion, contains a material error or unresolved
    contradiction, or provides no usable answer.

Partial credit applies to incomplete or imprecise answers without a material error.
    Apply the material-error rule before awarding partial credit.

### Input

Question type:
{question_type}

Question:
{question}

Gold record:
{gold_record}

Assistant answer:
{assistant_answer}

### Output

Return exactly one valid JSON object with these three fields:

- "score": The numerical score allowed for the question type.
- "reason": One concise sentence identifying the decisive match, error, or omission.
- "error_type": One of "none", "unanswered", "incorrect", or "incomplete".

Choose error_type using the first applicable rule:

- none: The score is 1.
- unanswered: There is no substantive answer, including an empty or irrelevant response
     or pure abstention on an answerable question.
- incorrect: The evaluated answer contains an incorrect factual claim, invalid
    inference, or unresolved contradiction.
- incomplete: The answer contains no incorrect claims but omits or underspecifies
    required information.

A relevant boundary abstention counts as an answer, not as unanswered. If both an error
     and an omission occur, use incorrect.

Do not include Markdown, additional fields, or text outside the JSON object.
```

F Reproducibility Checklist

The release package specifies: (i) benchmark JSONL and project-scoped images; (ii) schemas for all released records; (iii) construction and evaluation source code; (iv) prompt templates; (v) frozen generation, captioning, inference, and judging configurations; (vi) dataset, caption, run, and judge digests; (vii) environment lock files and dependency versions; (viii) per-question raw predictions, scores, rationales, and latency; and (ix) aggregate scripts. API credentials, provider-hidden reasoning traces, and latent answer annotations used by the scorer are not included in model inputs.