OPCTX: EVOLVING EVENT VIEWS FOR BUDGET-CONSTRAINED AGENT MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term memory enables language agents to accumulate knowledge across interactions. However, as histories grow, answering new questions requires organizing observations of events that evolve across interactions and recovering useful evidence within limited reading budgets. These challenges are amplified in multimodal settings, where images contain overlapping details about objects, attributes, and relations. Existing memory pipelines still face two difficulties in addressing these challenges. First, they leave relationships among observations of the same event implicit, requiring the answering model to reconstruct these relationships from scattered evidence. Second, they retrieve multiple facts conveying the same information when selection relies primarily on similarity, leaving less of the retrieval budget for complementary evidence. Retrieving more candidates and applying model-based reranking can improve selection, but reranking costs grow with candidate text. To address these difficulties, we propose OpCtx, a memory architecture connecting online event organization with question-conditioned evidence reading. OpCtx organizes source-linked observations into cross-interaction event views and, using a training-free Reader, aggregates selected facts and supported event information into structured evidence. This helps the answering model interpret relationships and changes across interactions. To improve evidence coverage within the reading budget, a coverage-preserving selector balances lexical and semantic signals under explicit fact-count limits and an optional character budget. Experiments on MemEye, LoCoMo, and LongMemEval demonstrate the effectiveness of OpCtx across multimodal and text-only memory tasks.

1 INTRODUCTION

Long-term memory enables agents to accumulate and update knowledge about users and their environments through continued interaction across sessions. As histories grow, systems must not only retain past information but also recover the evidence needed to answer the current question within a limited reading budget (Maharana et al., 2024; Wu et al., 2025). Multimodal settings further complicate this task. A single image may contain multiple objects, attributes, and spatial relations, while successive images and conversations may record different stages of the same event (Guo et al., 2026). These observations are both rich and overlapping, and their storage boundaries need not match the evidence boundaries of later questions.

Existing memory systems organize histories through facts, summaries, and relational structures, then provide retrieved content to a language model for answering. Segment-based and hierarchical approaches preserve coherent units and summaries (Pan et al., 2025; Sarthi et al., 2024). Fact-based systems extract compact records and reconcile new information with stored memory (Chhikara et al., 2025; Liu et al., 2026). Dynamic architectures update contextual links, represent temporal relations, or consolidate memories asynchronously (Xu et al., 2025; Rasmussen et al., 2025; Fang et al., 2026). At reading time, systems may further adjust retrieval granularity, filter candidates, or organize them into question-conditioned evidence (Xu et al., 2026b; Ma et al., 2026). These approaches improve access to historical information through complementary mechanisms.

However, two difficulties remain. First, existing methods that write memory within individual sessions or dialogue segments can separate observations of the same evolving event. For example, a household assistant may observe a blue cabinet in an image and later learn that a white one has re-

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.

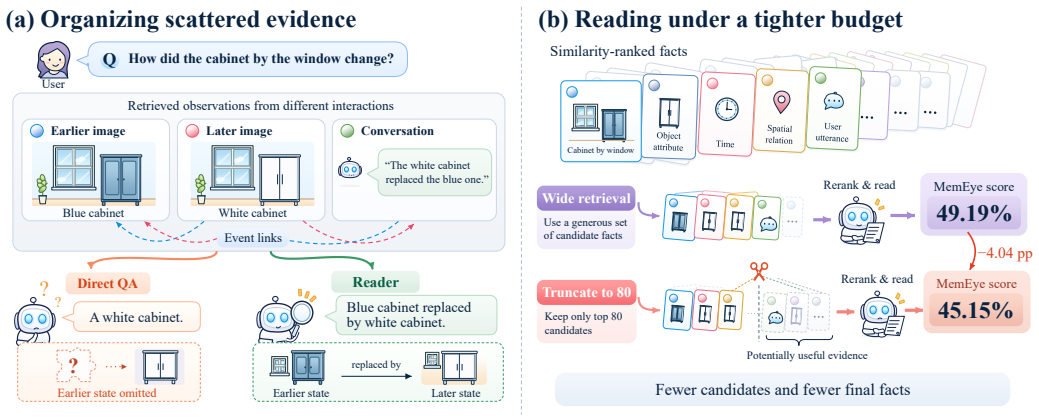


Figure 1: Two challenges in multimodal memory. (a) An illustrative example shows how scattered observations must be organized to recover an object's earlier state, transition, and later state. (b) On MemEye, tightening the candidate and final-fact budgets reduced the mapped Judge score from 49.19% to 45.15%, motivating evidence selection before costly model-based reranking.

placed it (Figure 1a). Answering questions about this change requires reconstructing object identity and temporal relationships from scattered evidence. Simply adding complete event descriptions may introduce repetition and obscure relevant details.

Figure 1b illustrates the second difficulty, a budget trade-off on MemEye. Similarity-based ranking can select multiple records conveying the same information, leaving less budget for complementary evidence. Expanding the candidate pool before model-based reranking can improve selection, but processing more candidate text increases reranking cost. Conversely, early truncation risks discarding useful evidence before reranking.

To address these difficulties, we propose OpCtx, which connects ongoing event organization with question-conditioned evidence reading. OpCtx preserves source-linked facts and organizes related observations into reusable cross-interaction event views. The Reader aggregates selected facts with session summaries and supported event claims into question-relevant evidence for QA.

To improve evidence selection within the budget, OpCtx combines object-anchored attribute queries with coverage-preserving selection. The queries retain the object and relation constraints associated with each requested attribute. The selector prioritizes query-term coverage, then balances lexical and semantic signals to select complementary evidence under the configured budget. This avoids manually specifying fusion weights and reduces reliance on model-based reranking.

In summary, our contributions are threefold.

- **Event-conditioned evidence organization.** We connect ongoing event organization with question-conditioned reading. Source-linked facts preserve observations, online event views maintain reusable cross-interaction associations, and the Reader aggregates question-relevant evidence for answering.

- **Coverage-preserving budgeted retrieval.** We formulate evidence selection as coverage-constrained utility retention, combining object-anchored attribute queries with a normalized max–min objective to balance lexical and semantic signals under reading budgets.

- **Evaluation across three memory benchmarks.** We conduct experiments on MemEye, LoCoMo, and LongMemEval, with component ablations and evidence analyses on MemEye and LoCoMo to examine the contributions of event organization, question-conditioned reading, and budgeted selection.

2 Related Work

Memory construction and maintenance. Agent memory systems use facts, summaries, and relational structures to retain interaction history (Chhikara et al., 2025; Pan et al., 2025; Liu et al., 2026; Sarthi et al., 2024; Jiménez Gutiérrez et al., 2024). Dynamic systems update contextual links, temporal relations, or consolidated memories (Xu et al., 2025; Rasmussen et al., 2025; Hu et al., 2026; Fang et al., 2026; Liao et al., 2026). StructMem preserves event bindings with hierarchical connections and periodic consolidation (Xu et al., 2026a); SpeakerMem-R1 combines speaker-labeled messages with derived person and group states (Zheng et al., 2026). These systems already motivate structured access beyond isolated records. OpCtx revises event membership and supported claims separately from its source facts, then couples this evolving representation with question-conditioned evidence selection and subsequent aggregation.

Budget-constrained evidence retrieval. Lexical and dense retrieval provide complementary signals (Robertson & Zaragoza, 2009; Karpukhin et al., 2020; Izacard et al., 2021; Khattab & Zaharia, 2020; Formal et al., 2021); diversification methods address coverage and redundancy (Carbonell & Goldstein, 1998; Santos et al., 2010; Lin & Bilmes, 2011). Recent set-selection methods use requirement matching or coverage and interaction objectives (Lee et al., 2025; Zhang et al., 2026a; Singh & Vadlamani, 2026). MemGAS combines granularity routing, graph propagation, and model filtering (Xu et al., 2026b). The SLM-based LightMem of Zhang et al. (2026b), distinct from Fang et al. (2026), uses vector retrieval followed by reranking. ContextBudget learns sequential history-compression decisions (Wu et al., 2026). OpCtx instead selects a set for each question, maximizing literal coverage before balancing normalized channel retention under shared constraints. Its guarantee concerns retrieval utilities, while QA performance is evaluated empirically.

Question-conditioned reading and multimodal memory. Visual memory must preserve details and relationships across observations (Song et al., 2024; Guo et al., 2026). MemLens evaluates multimodal memory across session histories and context lengths (Ren et al., 2026). MemVerse maintains hierarchical multimodal memory (Liu et al., 2025), while PMMC prepares question-conditioned access programs (Sun et al., 2026). MemChain learns evidence composition through supervised fine-tuning and reinforcement learning (Ma et al., 2026). OpCtx uses prompting and operation-conditioned context rules without additional policy training. Its Reader aggregates retrieved facts with session summaries and supported event claims for answering.

3 Method

OpCtx (Figure 2) connects online event organization, budgeted selection, and evidence aggregation over a source-linked fact store. Online views connect observations across writing batches. Selector allocates a fact budget across complementary retrieval signals, and Reader aggregates the resulting evidence for QA. Source facts retain their identities throughout.

3.1 Preliminaries

Let $B_1, \ldots, B_t$ be observed writing batches containing dialogue and available visual material; t is the latest visible frontier. A local writer produces a summary and source-linked facts, $x_t = (s_t, \mathcal{F}_t)$. For question q, Selector chooses $S \subseteq \mathcal{P}$ from a candidate fact pool. Constraints include $|S| \leq K$ and an optional rendered-character ceiling $c(S) \leq b(q)$; $b = \infty$ denotes no character ceiling. Appendix E distinguishes Selector and final retrieval limits. Fixed query subscripts are omitted.

3.2 Evidence-Preserving Event Evolution

Preserve observations while revising their relationships. A useful writing boundary need not delimit an event. We therefore keep the writer's facts unchanged and maintain event membership separately. Each fact has a persistent identifier, text, source references, and available temporal metadata. The evidence store appends $\mathcal{F}_t$ to its observed prefix without merging similar statements. Local summaries, embeddings, and a lexical index provide access to this store.

An event view contains member identifiers J_v, a relation type, evidence-linked claims, and a revision. A shared event links one occurrence; a continuing process links stages of an activity; an update

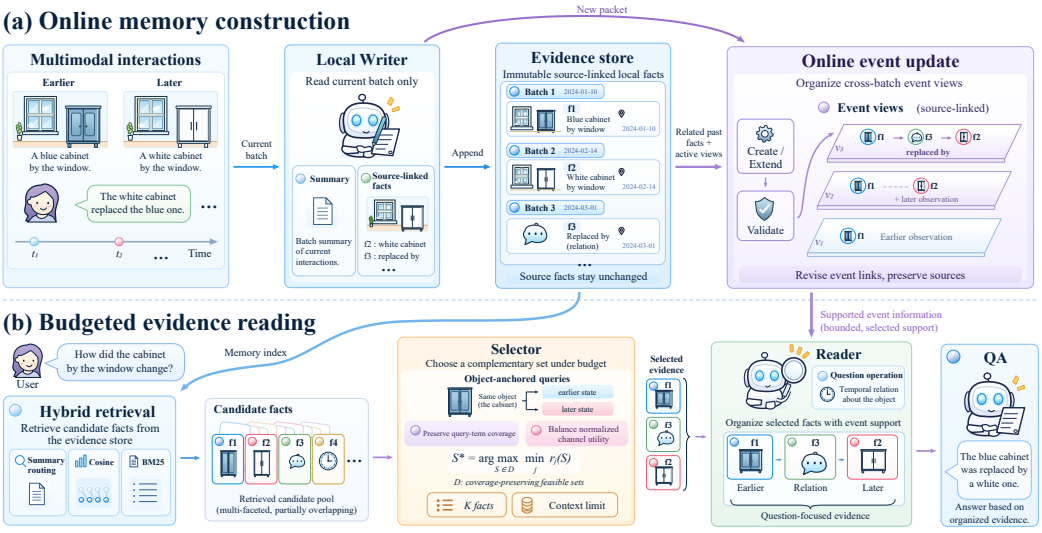


Figure 2: Overview of OpCtx. (a) The local writer appends source-linked facts, while online updates organize related observations into evolving event views without overwriting their sources. (b) Hybrid retrieval supplies candidate facts to the budgeted selector. The operation-conditioned Reader organizes selected facts with supported event information for QA. The cabinet example illustrates how evidence from different interactions forms an update chain.

chain links changes in an object or relation. Shared topic or person alone does not establish event identity. Claims cite nonempty support within J_v and collectively cover its membership. Facts can support several views without changing their identities.

Neighborhood-based online updates. An event may continue across batches. A prefix-only retriever combines summary routing, dense matching, and BM25 to expose a neighborhood $\mathcal{N}_t$ of historical facts and active views. It uses the current packet x_t to search only the observed prefix. The update model U then proposes operations Δ_t, and a validation-and-application function A commits admissible changes:

$$\Delta_t = U(x_t, \mathcal{N}_t), \qquad \mathcal{V}_t = A(\mathcal{V}_{t-1}, \Delta_t). \tag{1}$$

Here, $\mathcal{V}_{t-1}$ and $\mathcal{V}_t$ are the event layers before and after updating. A *create* operation connects current and historical evidence; an *extend* operation retains an exposed view's members and adds current evidence. Function A checks reference visibility, claim-support membership, and operation validity before revising affected views and their embeddings. Subsequent updates and retrieval access the revised layer; the writer remains local to its batch.

3.3 Coverage-Constrained Minimax Evidence Selection

Rich visual descriptions can repeatedly match one attribute while missing another. We preserve question-term coverage, then balance signals by the fraction of attainable utility each retains.

Object-anchored query channels. Attribute queries retain the queried instance. A query-only model extracts verbatim entity, relation, and scope anchors with requested attributes. Each projection joins the anchors with one attribute in original textual order, specializing aspect-based retrieval (Santos et al., 2010). We retain the original full question $q_0 = q$, remove duplicate projections, and fall back to q_0 if structured extraction is invalid.

Each query supplies cosine and BM25 channels (Robertson & Zaragoza, 2009). Channel j is a query–scorer pair. Its nonnegative utility $u_j(f)$ derives from within-pool ranks, avoiding addition of raw scores with different scales. Appendix E specifies ranks, ties, and inactive channels.

Coverage under a shared budget. Each question term earns coverage once. Let $\mathcal{T}(q)$ be non-stopword question tokens and $\mathcal{T}(f)$ fact tokens. Feasible sets $\mathcal{B}$ satisfy $|S| \leq K$, $c(S) \leq b$, and provenance-duplicate exclusions. Coverage $h(S)$ counts distinct matches; $\mathcal{D}$ contains its feasible maximizers:

$$h(S) = \left|\mathcal{T}(q) \cap \bigcup_{f \in S} \mathcal{T}(f)\right|, \qquad \mathcal{D} = \arg\max_{S \in \mathcal{B}} h(S). \tag{2}$$

Cost $c(S)$ counts characters in facts, source-session summaries, and eligible event claims, with separators included and each shared header charged exactly once. Subsequent selection decisions stay in $\mathcal{D}$, preserving maximal term coverage.

Same-budget utility retention and selection. We measure each channel's retained fraction against its independent optimum, then maximize the smallest fraction to protect the least-served signal:

$$U_j(S) = \sum_{f \in S} u_j(f), \qquad I_j = \max_{T \in \mathcal{D}} U_j(T), \qquad r_j(S) = \frac{U_j(S)}{I_j}. \tag{3}$$

$$S^* \in \arg\max_{S \in \mathcal{D}} \min_{j \in \mathcal{J}} r_j(S). \tag{4}$$

Here, $U_j(S)$ is selected utility, I_j the channel's attainable best value, and $r_j(S) \in [0, 1]$ its retained fraction. Each channel has its own reference set under identical count, cost, and coverage constraints. Distinct channels with $I_j > 0$ form $\mathcal{J}$.

This normalized max–min rule (Steuer & Choo, 1983; Krause et al., 2008) couples object-anchored channels with coverage-preserving, same-budget references. Ties favor greater total retention $\sum_j r_j(S)$, then lower cost $c(S)$. The Selector uses cached scores.

A uniform retention guarantee. The minimum retention $r(S) = \min_{j \in \mathcal{J}} r_j(S)$ also protects every nonnegative channel mixture. Let $w_j \geq 0$ be a channel weight with $\sum_j w_j = 1$, and $\mathcal{W}$ contain all such weight vectors. For mixture utility $v_w(S) = \sum_j w_j U_j(S)$ and its best attainable value $v_w^* = \max_{T \in \mathcal{D}} v_w(T)$,

$$r(S) = \min_{w \in \mathcal{W}} \frac{v_w(S)}{v_w^*}. \tag{5}$$

With exact channel ideals, $r(S)$ bounds retained rank-derived utility for every nonnegative channel mixture on the same domain. It does not guarantee gold-evidence recall or QA accuracy. The property applies to Selector output before downstream reranking; Appendix D proves it and specifies exact-solution requirements.

3.4 Operation-Conditioned Reader

Directly concatenating memory records can obscure changes and relationships across observations. Given a question q and retrieved facts $\mathcal{R}$, the Reader identifies the answering operation. It combines the facts with their session summaries and supported event claims, then organizes and aggregates them into direct evidence, counts, timelines, or comparisons. QA receives only the evidence packet. Appendix E.4 specifies evidence construction, claim filtering, and ablation controls.

4 Experiments

4.1 Experimental Settings

Datasets and metrics. We evaluate on MemEye (Guo et al., 2026), LoCoMo (Maharana et al., 2024), and LongMemEval-S (Wu et al., 2025). MemEye contributes 371 visually grounded questions across eight scenarios; LoCoMo contributes 1,540 non-adversarial questions from ten conversations. LongMemEval-S contains 500 questions with separate timestamped histories. Appendix A describes their categories and source annotations. The primary metric is question-averaged Judge score: binary accuracy for LoCoMo and LongMemEval, and a fixed partial-credit mapping for MemEye. We additionally report BLEU-1 for MemEye's open-ended answers, alongside category scores and rendered input usage. Metric definitions appear in Appendix B.3.

Table 1: MemEye category results (%). The X and Y columns report Judge scores, matching the component study. Overall, EM, and BLEU-1 are question-weighted; EM evaluates the mirrored multiple-choice questions.

Method	X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Overall	EM	BLEU-1
Questions	48	48	142	133	120	203	48	371	371	371
Text-memory baselines										
FC(T)	53.13	30.21	29.58	22.56	26.67	32.76	28.13	30.19	41.58	17.81
SRAG(T)	46.88	26.04	21.13	16.54	22.92	22.17	30.21	23.45	36.39	13.66
Reflexion	47.92	22.92	24.65	9.40	22.08	20.94	26.04	21.97	35.98	11.41
Generative Agents	35.42	20.83	9.86	7.14	5.42	17.73	16.67	13.61	26.82	8.38
MemoryOS	16.67	9.38	13.73	13.16	6.67	18.23	9.38	13.34	28.91	9.64
A-MEM	38.54	15.63	23.24	10.53	17.92	20.69	19.79	19.68	33.69	11.66
SM(T)	36.46	17.71	17.96	9.02	14.58	17.98	19.79	17.12	32.01	9.69
Visual-memory baselines										
FC(V)	47.92	33.33	42.61	35.71	46.67	37.19	32.29	39.62	46.83	19.75
SRAG(V)	52.08	33.33	39.79	26.69	46.67	30.79	30.21	35.85	46.56	18.36
MMA	45.83	34.38	41.90	31.95	52.08	31.03	31.25	37.87	45.42	20.78
M2A	17.71	9.38	17.61	10.90	16.25	15.02	5.21	14.15	32.41	10.85
SM(V)	38.54	19.79	21.13	7.89	20.00	17.24	19.79	18.46	34.70	10.35
OpCtx										
OpCtx	61.46	48.96	52.82	36.09	50.00	46.55	44.79	47.44	50.13	26.47

Baselines. We include two direct-access references, Full-context and Naive-RAG; compact memory systems, Mem0 (Chhikara et al., 2025), SimpleMem (Liu et al., 2026), and LightMem (Fang et al., 2026); and structured systems, HyperMem (Yue et al., 2026), A-MEM (Xu et al., 2025), and MemoryOS (Kang et al., 2025). LoCoMo additionally includes RAPTOR (Sarthi et al., 2024), HippoRAG 2 (Jiménez Gutiérrez et al., 2025), RecurSum (Wang et al., 2023), SeCom (Pan et al., 2025), and MemGAS (Xu et al., 2026b). MemEye compares the benchmark's text and visual FC, semantic RAG, and SimpleMem variants, together with Reflexion, Generative Agents, A-MEM, MemoryOS, MMA, and M2A. The w/o Online control retains the same local Writer and removes cross-session event updates. Dense Top-K provides a selector control within a fixed memory store. Appendix B describes the systems and reproduction settings.

Implementation details. Tables 2 and 3 report two backbones. Models explicitly prescribed by each method are retained; other generative stages share the reported backbone (Appendix B). LoCoMo ablations use Gemini 2.5 Flash. LoCoMo and LongMemEval-S use Gemini 3 Flash as Judge. MemEye uses Gemini 2.5 Flash-Lite for generation and Gemini 3.5 Flash-Lite for judging. Default embeddings and reranking use gemini-embedding-001 and Qwen3-Reranker-4B, respectively. OpCtx pools facts from summary/view routing, BM25, and dense retrieval before Selector; dataset-specific retrieval budgets appear in Appendix E.2. Each answer receives one valid Judge verdict, and ablation variants share frozen local Writer facts.

4.2 Overall Results

OpCtx achieves the highest overall QA scores on all three benchmarks (Tables 1–3). Improvements are consistent across both backbones on LoCoMo and LongMemEval-S, and extend to EM and BLEU-1 on MemEye. The consistency across backbones suggests that the gains are not confined to a single generative model.

Performance across query types. OpCtx leads on LoCoMo multi-hop and temporal questions, and LongMemEval-S knowledge updates and multi-session questions. MemEye likewise shows larger gains in relational association and evolutionary synthesis than in atomic retrieval. These categories require combining observations or resolving changes over time, making them useful

Table 2: LoCoMo category accuracy (%) across two backbones on all 1,540 questions. Higher scores are better.

Method	Single-hop (841)	Multi-hop (282)	Temporal (321)	Open-domain (96)	Overall (1540)
Gemini 2.5 Flash					
Full Context	89.77	64.89	56.39	59.38	76.36
RAPTOR	85.49	54.26	63.24	55.21	73.25
HippoRAG 2	76.46	37.23	50.47	52.08	62.34
HyperMem	84.78	63.83	70.40	57.29	76.23
Mem0	80.14	69.50	78.82	51.04	76.10
A-MEM	82.16	60.64	37.07	53.12	67.01
RecurSum	51.01	42.20	19.94	48.96	42.79
MemGAS	57.43	26.95	33.64	41.67	45.91
Naive RAG	65.40	25.89	40.50	41.67	51.49
SeCom	86.56	57.80	13.08	51.04	63.77
MemoryOS	78.83	42.20	33.02	50.00	60.78
LightMem	77.88	42.91	74.14	39.58	68.31
SimpleMem	86.33	52.84	72.90	51.04	75.19
OpCtx	86.68	73.76	83.18	57.29	81.75
DeepSeek V4 Flash					
Full Context	90.37	68.09	52.02	55.21	76.10
RAPTOR	84.78	51.42	48.29	52.08	69.03
HippoRAG 2	63.14	34.75	37.07	45.83	51.43
HyperMem	79.55	63.48	52.34	50.00	69.09
Mem0	75.86	63.48	70.09	50.00	70.78
A-MEM	80.74	58.51	36.45	52.08	65.65
RecurSum	48.16	35.11	17.76	51.04	39.61
MemGAS	56.60	29.08	32.40	39.58	45.45
Naive RAG	83.95	50.35	50.78	44.79	68.44
SeCom	83.59	53.19	11.84	55.21	61.30
MemoryOS	55.29	26.60	32.40	39.58	44.29
LightMem	78.83	44.68	57.94	42.71	65.97
SimpleMem	86.09	56.38	65.42	51.04	74.16
OpCtx	86.21	69.86	76.64	54.17	79.22

tests of whether memory supplies related evidence together. Full Context remains stronger on LoCoMo single-hop and LongMemEval-S single-session assistant questions. These exceptions identify direct-access tasks on which memory construction and evidence selection still need improvement.

4.3 Hyperparameter Analysis

Increasing retrieved facts k generally improves overall QA, with uneven gains across categories (Figure 3). LoCoMo continues to improve over the tested range, whereas MemEye gains little beyond 40 facts despite increasing QA input. The category curves are not monotonic, so increasing the budget does not uniformly benefit every question type. These trends suggest that useful context budgets depend on the evidence requirements of each dataset. The marked settings represent operating points on a quality–input trade-off; they do not maximize accuracy across the tested budgets.

4.4 Ablation Study

Effect of Online. Removing Online produces the largest overall drop on both datasets (Figure 4a,b). LoCoMo multi-hop accuracy is most affected, supporting the role of cross-session organization in connecting separately written observations. The same-Writer analysis in Appendix C.5 provides a more direct mechanism check: Online improves access to annotated evidence sources, while shuffling event links reduces candidate-source recall. The benefits remain task-dependent:

Table 3: LongMemEval-S category accuracy (%) across two backbones on all 500 questions. Overall is question-weighted.

Method	KU (78)	MS (133)	SSA (56)	SSP (30)	SSU (70)	TR (133)	Overall (500)
Gemini 2.5 Flash							
Full Context	87.18	63.91	98.21	46.67	90.00	56.39	72.00
SimpleMem	79.49	69.17	80.36	53.33	90.00	61.65	72.00
Mem0	61.54	36.84	46.43	46.67	75.71	46.62	50.40
Naive RAG	10.26	10.53	26.79	16.67	32.86	12.78	16.40
OpCtx	91.03	75.19	75.00	73.33	94.29	78.20	81.00
DeepSeek V4 Flash							
Full Context	75.64	53.38	100.00	20.00	94.29	62.41	68.20
SimpleMem	75.64	61.65	76.79	23.33	91.43	69.92	69.60
Mem0	76.92	60.90	60.71	30.00	82.86	76.69	68.80
Naive RAG	26.92	10.53	75.00	6.67	64.29	18.05	29.60
OpCtx	88.46	72.93	73.21	73.33	92.86	75.94	79.00

KU: knowledge update; MS: multi-session; SSA/SSP/SSU: single-session assistant/preference/user; TR: temporal reasoning.

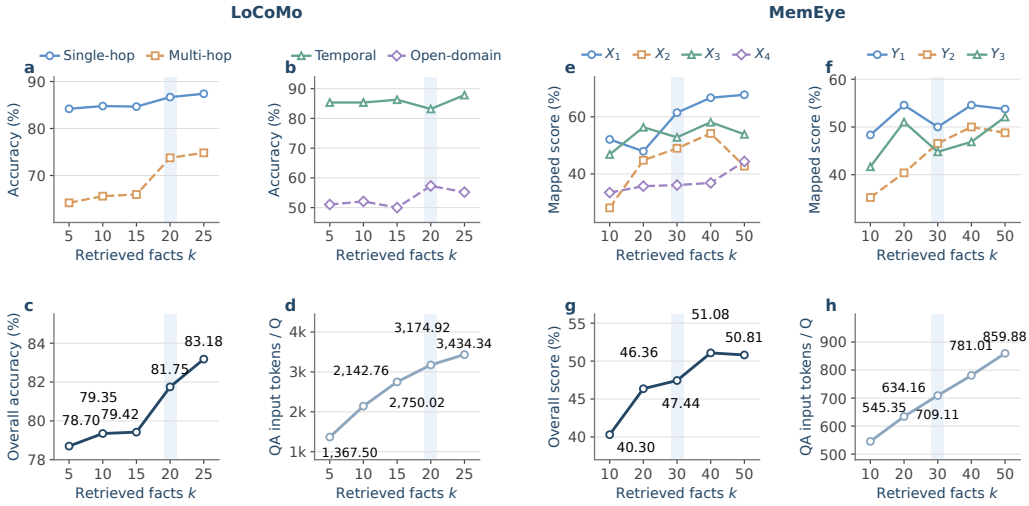


Figure 3: Retrieved facts k sensitivity on LoCoMo and MemEye: category and overall scores, and mean QA input tokens per question. Shaded bands mark $k = 20$ and $k = 30$, respectively.

MemEye improves overall with Online, but its evolutionary-synthesis category scores higher without it.

Effect of Reader. Bypassing Reader reduces overall performance on both datasets despite retaining the same selected facts. This supports the combined use of event claims and evidence aggregation. Replacing Reader with a generic reasoning prompt also lowers performance when facts, summaries, and event claims are held fixed (Appendix C.5), supporting the value of question-conditioned evidence organization. Appendix F.1 illustrates counting and comparison behavior. The lexical and semantic metrics on MemEye follow the same pattern (Figure 4d), supporting improvements beyond Judge scoring.

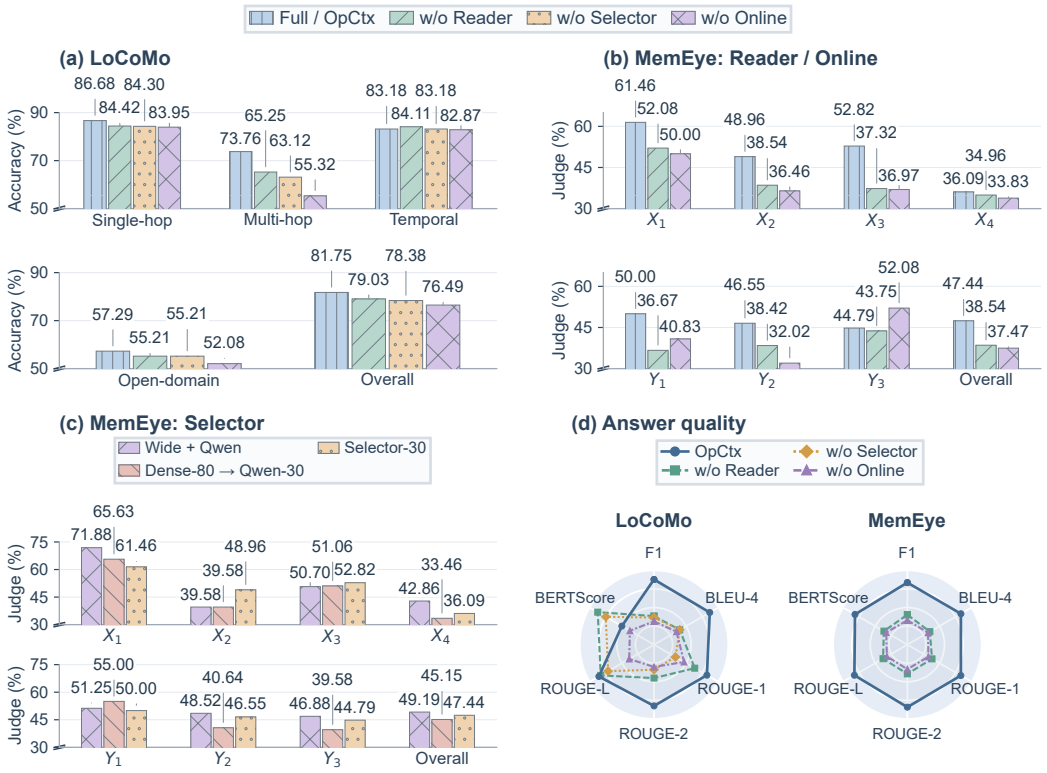


Figure 4: Ablations on (a) LoCoMo at 20 facts and (b) MemEye at 30 facts; (c) MemEye Selector versus Qwen reranking; (d) answer-quality metrics. Bars use truncated axes. Radar axes are scaled independently; exact scores appear in Appendix Table 9.

Effect of Selector. On LoCoMo, replacing Selector with dense retrieval hurts multi-hop more than single-hop accuracy, consistent with the need to preserve complementary facts. MemEye controls using the same candidate pool and budgets also favor OpCtx over dense retrieval, BM25, MMR, and equal-weight fusion (Appendix C.5). This supports the selection policy beyond simply combining retrieval channels. On MemEye, the Selector-30 pipeline exceeds Dense-80 → Qwen-30 by 2.29 Judge points without reranking (Figure 4c). This is a pipeline comparison; wider Qwen retrieval still scores higher.

5 Conclusion

OpCtx connects evolving event views, coverage-constrained selection, and question-conditioned evidence aggregation. Across three memory benchmarks, it improves overall QA, while component analyses link these gains to cross-session evidence access and the organization of selected content. The results support treating retrieval and reading as complementary parts of memory design, with their benefits depending on the question type and evidence budget. Our cost measurements cover QA input usage and Selector latency. Because baselines retain their prescribed auxiliary models, complete cost comparisons require profiling all model-specific stages. Future work will examine end-to-end costs and smaller models for intermediate OpCtx operations.

Ethics Statement

This work evaluates agent memory on existing benchmarks. Persistent conversational and visual memories may contain sensitive information or preserve incorrect inferences. Deployment should include informed consent, access controls, and mechanisms to inspect, correct, and delete stored information. Source links improve traceability but do not guarantee the accuracy of generated content.

Reproducibility Statement

Dataset and evaluation details are provided in Appendices A and B. Appendices D and E document the theoretical assumptions, proofs, and implementation settings. Prompt templates and Selector runtime diagnostics appear in Appendices G and C.6.

AI assistance disclosure

Generative AI tools assisted with language polishing, coding, and debugging, and also supported translation, methodological discussion, mathematical exposition, and interpretation of experimental results. The authors are responsible for the final content.

References

Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 335–336, 1998. doi: 10.1145/290941.291025. URL https://doi.org/10.1145/290941.291025.

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. URL https://arxiv.org/abs/2504.19413.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. LightMem: Lightweight and efficient memory-augmented generation. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.18866.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. SPLADE: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021. URL https://arxiv.org/abs/2107.05720.

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. URL https://arxiv.org/abs/2605.15128.

Yuyang Hu, Jiongnan Liu, Jiejun Tan, Yutao Zhu, and Zhicheng Dou. Memory matters more: Event-centric memory as a logic map for agent searching and reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 22389–22407, 2026. doi: 10.18653/v1/2026.findings-acl.1123. URL https://aclanthology.org/2026.findings-acl.1123/.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. arXiv preprint arXiv:2112.09118, 2021. URL https://arxiv.org/abs/2112.09118.

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*, volume 37, 2024. URL https://arxiv.org/abs/2405.14831.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21497–21515. PMLR, 2025. URL https://proceedings.mlr.press/v267/gutierrez25a.html.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI agent. arXiv preprint arXiv:2506.06326, 2025. URL https://arxiv.org/abs/2506.06326.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, 2020. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

Omar Khattab and Matei Zaharia. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020. URL https://arxiv.org/abs/2004.12832.

Andreas Krause, H. Brendan McMahan, Carlos Guestrin, and Anupam Gupta. Robust submodular observation selection. *Journal of Machine Learning Research*, 9(93):2761–2801, 2008. URL https://www.jmlr.org/papers/v9/krause08b.html.

Dahyun Lee, Yongrae Jo, Haeju Park, and Moontae Lee. Shifting from ranking to set selection for retrieval augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17606–17619, 2025. doi: 10.18653/v1/2025.acl-long.861. URL https://aclanthology.org/2025.acl-long.861/.

Yuxin Liao, Le Wu, Min Hou, Hao Liu, Han Wu, and Zishu Wang. LeanMem: Simple and efficient long-term memory for LLM agents. arXiv preprint arXiv:2608.03463, 2026. URL https://arxiv.org/abs/2608.03463.

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 510–520, 2011. URL https://aclanthology.org/P11-1052/.

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, 2026. URL https://arxiv.org/abs/2601.02553v3. Version 3.

Junming Liu, Yifei Sun, Weihua Cheng, Haodong Lei, Yirong Chen, Licheng Wen, Xuemeng Yang, Daocheng Fu, Pinlong Cai, Nianchen Deng, Yi Yu, Shuyue Hu, Botian Shi, and Ding Wang. MemVerse: Multimodal memory for lifelong learning agents. arXiv preprint arXiv:2512.03627, 2025. URL https://arxiv.org/abs/2512.03627v2. Version 2, revised June 2026.

Yiwen Ma, Songjun Tu, Qichao Zhang, Dong Li, Linjing Li, and Dongbin Zhao. MemChain: Learning interpretable memory traces for memory-augmented LLM agents. arXiv preprint arXiv:2607.24097, 2026. URL https://arxiv.org/abs/2607.24097.

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)*, 2024. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Jianfeng Gao. SeCom: On memory construction and retrieval for personalized conversational agents. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/e56f394bbd4f0ec81393d767caa5a31b-Abstract-Conference.html.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory. arXiv preprint arXiv:2501.13956, 2025. URL https://arxiv.org/abs/2501.13956.

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. URL https://arxiv.org/abs/2605.14906.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/1500000019. URL https://doi.org/10.1561/1500000019.

Rodrygo L. T. Santos, Jie Peng, Craig Macdonald, and Iadh Ounis. Explicit search result diversification through sub-queries. In *Advances in Information Retrieval*, pp. 87–99, 2010. doi: 10.1007/978-3-642-12275-0_11. URL https://terrierteam.dcs.gla.ac.uk/publications/ecir2010_rodrygo_div.pdf.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2401.18059.

Rahul Singh and Madhav Vadlamani. QUBO-optimized evidence selection for retrieval-augmented question answering with unconventional solvers. arXiv preprint arXiv:2607.12334, 2026. URL https://arxiv.org/abs/2607.12334.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang. MovieChat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2307.16449.

Ralph E. Steuer and Eng-Ung Choo. An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26(3):326–344, 1983. doi: 10.1007/BF02591870. URL https://link.springer.com/article/10.1007/BF02591870.

Jingyu Sun, Yan Lin, Yuyang Xue, Yifan Wang, Zhengtao Yao, Rui Qian, Zefeng Xu, Jiachen Li, Xianyang Liu, Jiancheng Pan, Jingyuan Sun, Syed Murtuza Baker, and Hongpeng Zhou. PMMC: Prospective multimodal memory compilation for long-term LVLM agents. arXiv preprint arXiv:2608.00962, 2026. URL https://arxiv.org/abs/2608.00962.

Qingyue Wang, Liang Ding, Yanan Cao, Zhiliang Tian, Shi Wang, Dacheng Tao, and Li Guo. Recursively summarizing enables long-term dialogue memory in large language models. arXiv preprint arXiv:2308.15022v1, 2023. URL https://arxiv.org/abs/2308.15022v1. Version 1.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.10813.

Yong Wu, YanZhao Zheng, TianZe Xu, ZhenTao Zhang, YuanQiang Yu, JiHuai Zhu, Chao Ma, BinBin Lin, BaoHua Dong, HangCheng Zhu, RuoHui Huang, and Gang Yu. ContextBudget: Budget-aware context management for long-horizon search agents. *arXiv preprint arXiv:2604.01664*, 2026. URL https://arxiv.org/abs/2604.01664.

Buqiang Xu, Yijun Chen, Jizhan Fang, Ruobin Zhong, Yunzhi Yao, Yuqi Zhu, Lun Du, and Shumin Deng. StructMem: Structured memory for long-horizon behavior in LLMs. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 122–146. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.acl-short.12. URL https://aclanthology.org/2026.acl-short.12/.

Derong Xu, Yi Wen, Pengyue Jia, Yingyi Zhang, Wenlin Zhang, Yichao Wang, Huifeng Guo, Ruiming Tang, Xiangyu Zhao, Enhong Chen, and Tong Xu. From single to multi-granularity: Toward long-term memory association and selection of conversational agents. In *International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=i2yIvZARnG.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2502.12110.

Juwei Yue, Chuanrui Hu, Jiawei Sheng, Zuyi Zhou, Wenyuan Zhang, Tingwen Liu, Li Guo, and Yafeng Deng. HyperMem: Hypergraph memory for long-term conversations. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 35237–35254, 2026. doi: 10.18653/v1/2026.acl-long.1627. URL https://aclanthology.org/2026.acl-long.1627/.

Bingxue Zhang, Jianying Jia, and Feida Zhu. Covering the unseen: Information demand coverage optimization for retrieval-augmented generation. arXiv preprint arXiv:2606.29328, 2026a. URL https://arxiv.org/abs/2606.29328.

Jiaquan Zhang, Chaoning Zhang, Shuxu Chen, Zhenzhen Huang, Pengcheng Zheng, Zhicheng Wang, Ping Guo, Fan Mo, Sung-Ho Bae, Jie Zou, Jiwei Wei, and Yang Yang. Lightweight LLM agent memory with small language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12914–12929. Association for Computational Linguistics, 2026b. doi: 10.18653/v1/2026.acl-long.588. URL https://aclanthology.org/2026.acl-long.588/.

Haobo Zheng, Tan Tang, Yan Chen, Weijie Wang, and Yingcai Wu. SpeakerMem-R1: Speaker-centered dual-track memory for multi-party dialogue. *arXiv preprint arXiv:2609.26780*, 2026. URL https://arxiv.org/abs/2609.26780.

Technical Appendix

TABLE OF CONTENTS

A Dataset Details **15**

B Baseline and Evaluation Details **16**

B.1 External Memory Systems 16

B.2 Internal Controls and Reproduction Scope 16

B.3 Metrics and Resource Accounting 17

C Category Results and Component Controls **18**

C.1 MemEye Baseline Results by Joint Category 18

C.2 MemEye Ablations at 30 Facts 20

C.3 Lexical and Semantic Answer Quality 20

C.4 Sensitivity to the Reading Budget 21

C.5 Supplementary Component Analyses 21

C.6 Selector Optimization and Runtime 22

C.7 QA Input Usage 22

D Properties of the Evidence Objective **23**

E Implementation Details **24**

E.1 Online Updates 24

E.2 Retrieval and Budgets 24

E.3 Selector Execution 24

E.4 Reader Evidence Aggregation 25

E.5 Component Controls 25

F Case Study **26**

F.1 Reader: Using Retrieved Evidence 26

F.2 Online: Preserving Answer Evidence 27

F.3 Baseline Pipeline Comparisons 28

G Prompt Templates **29**

A Dataset Details

MemEye. MemEye (Guo et al., 2026) evaluates visual memory across 371 questions from eight everyday scenarios. We evaluate open-ended QA and report EM on the paired multiple-choice questions. The X axis specifies visual granularity: scene (X_1), region (X_2), instance (X_3), and pixel-level detail (X_4). The Y axis specifies the reasoning operation: atomic retrieval (Y_1), relational association (Y_2), and evolutionary synthesis (Y_3).

Table 4 reports the number of questions in each category.

Table 4: Question distribution by MemEye category. Rows denote visual granularity; columns denote reasoning operation.

Visual granularity	Y_1: atomic	Y_2: relational	Y_3: evolutionary	Total
X_1: scene	10	30	8	48
X_2: region	16	27	5	48
X_3: instance	74	59	9	142
X_4: pixel detail	20	87	26	133
Total	120	203	48	371

Joint visual–reasoning categories. Each joint label X_iY_j combines visual granularity X_i with reasoning operation Y_j. For example, X_1Y_1 denotes scene-level atomic retrieval, and X_2Y_2 denotes region-level relational association. Table 5 defines the twelve categories covering all 371 questions. Appendix Table 8 reports their 30-fact ablation results.

Table 5: Joint categories in MemEye. Each row combines one visual granularity with one reasoning operation.

Category	Visual granularity	Reasoning operation	n
X_1Y_1	Scene	Atomic retrieval	10
X_1Y_2	Scene	Relational association	30
X_1Y_3	Scene	Evolutionary synthesis	8
X_2Y_1	Region	Atomic retrieval	16
X_2Y_2	Region	Relational association	27
X_2Y_3	Region	Evolutionary synthesis	5
X_3Y_1	Instance	Atomic retrieval	74
X_3Y_2	Instance	Relational association	59
X_3Y_3	Instance	Evolutionary synthesis	9
X_4Y_1	Pixel detail	Atomic retrieval	20
X_4Y_2	Pixel detail	Relational association	87
X_4Y_3	Pixel detail	Evolutionary synthesis	26
Total			371

LoCoMo. LoCoMo (Maharana et al., 2024) contains timestamped, multi-session conversations with question-level evidence references. We evaluate ten conversations and all 1,540 non-adversarial questions: single-hop (841), multi-hop (282), temporal (321), and open-domain (96). The 446 adversarial questions are excluded. Each conversation has an independent memory built from the fixed text-only dialogue rendering.

LongMemEval-S. LongMemEval (Wu et al., 2025) provides an independent timestamped conversation history for each question. We evaluate all 500 questions in LongMemEval-S using their complete histories. The six types are knowledge update (78), multi-session (133), single-session assistant (56), single-session preference (30), single-session user (70), and temporal reasoning (133). These tasks test information updates, cross-session integration, speaker-specific recall, preferences,

and temporal relations. Abstention cases occur within these categories. Each question uses an isolated memory namespace; source-session and answer-bearing-turn annotations are reserved for evaluation.

Visibility and population accounting. Memory construction uses the observed history, with reference answers, evidence annotations, and category labels reserved for evaluation. Overall QA scores are weighted by the number of questions.

B Baseline and Evaluation Details

B.1 External Memory Systems

Full-context and Naive-RAG. Full-context passes the conversation history directly to QA, whereas Naive-RAG retrieves relevant text units before answering. These baselines compare full-history access with selective retrieval.

Mem0. Mem0 (Chhikara et al., 2025) extracts salient information, reconciles new records with existing memory, and retrieves relevant records for subsequent interactions.

SimpleMem. SimpleMem (Liu et al., 2026) compresses interactions into structured memory units, synthesizes related information within a session, and plans retrieval according to query intent.

LightMem. LightMem (Fang et al., 2026) uses sensory, short-term, and long-term memory stages. It filters incoming content, groups related topics, and consolidates memory through updates outside the immediate interaction path.

HyperMem. HyperMem (Yue et al., 2026) organizes topics, episodes, and facts in a hierarchical hypergraph. Hyperedges connect memory elements, and lexical–semantic indexing supports coarse-to-fine retrieval.

A-MEM and MemoryOS. A-MEM (Xu et al., 2025) creates contextualized memory notes and links them to related records as memory evolves. MemoryOS (Kang et al., 2025) manages short-term, mid-term, and long-term memory through storage, updating, retrieval, and generation modules.

RAPTOR, HippoRAG 2, SeCom, and RecurSum. RAPTOR (Sarthi et al., 2024) recursively clusters and summarizes text for retrieval at multiple abstraction levels. HippoRAG 2 (Jiménez Gutiérrez et al., 2025) combines graph-based associative retrieval, passage representations, and LLM-assisted access. SeCom (Pan et al., 2025) organizes dialogue into semantically coherent segments. RecurSum (Wang et al., 2023) recursively integrates earlier summaries with newly available dialogue.

MemGAS. MemGAS (Xu et al., 2026b) constructs associations across memory granularities, selects the retrieval granularity, and filters retrieved evidence. We reproduce its LoCoMo results under the shared evaluation protocol.

B.2 Internal Controls and Reproduction Scope

Model protocol. We retain auxiliary models explicitly specified by the original method designs. All other generative stages use the designated experimental backbone, with the same QA and Judge protocols across methods. The default embedding and reranking models are gemini-embedding-001 and Qwen3-Reranker-4B.

MemEye baseline reproduction. Text-based memory architectures use the image descriptions provided by MemEye. OpCtx and the other visual-memory architectures process images with Gemini 2.5 Flash-Lite. All methods share the same QA and Judge protocols.

Online ablations. On LoCoMo and MemEye, w/o Online retains the same Writer facts, session summaries, and source links as Full while omitting cross-session event updates.

Selector ablations. On LoCoMo, w/o Selector uses cosine similarity to select 30 facts from the same candidate pool, followed by Qwen3 reranking to 20. On MemEye, three pipelines share the same initial pool. Wide + Qwen reranks at most 200 candidate facts and retains at most 80; Dense-80 $\rightarrow$ Qwen-30 selects 80 facts by cosine similarity and reranks to 30; Selector-30 selects 30 facts without reranking. Appendix E specifies the retrieval and character budgets.

B.3 Metrics and Resource Accounting

Answer scores. Accuracy on LoCoMo and LongMemEval-S is the percentage of questions receiving a correct verdict. On MemEye, Judge scores in $\{0, 0.25, 0.5, 0.75, 1\}$ map to 0 for raw 0, 0.5 for raw 0.25 or 0.5, and 1 for raw 0.75 or 1. We report the mean mapped score multiplied by 100. BLEU-1 uses benchmark answer normalization and sentence-level unigram precision with a brevity penalty, averaged over questions and multiplied by 100. EM uses the paired multiple-choice questions with the same retrieved evidence. Correctness is averaged over four option orderings per question and then across questions; options are supplied only during answering.

Input accounting. Table 13 and Figure 3 report mean QA input tokens per question, counting QA calls only. For Full, the evidence supplied to QA is the packet produced by Reader aggregation. Appendix C.6 separately reports Selector latency; neither measure represents total pipeline cost.

C Category Results and Component Controls

This appendix examines joint-category performance, component contributions, answer quality, and sensitivity to the retrieved-fact budget.

C.1 MemEye Baseline Results by Joint Category

Tables 6 and 7 expand Table 1 into the twelve X_iY_j categories defined in Appendix A. This breakdown separates visual granularity from reasoning requirements when comparing text-based and visual memory methods.

Table 6: MemEye text-memory baselines and OpCtx by joint category (%). Refl.: Reflexion; GA: Generative Agents. Bold marks the best score among the methods shown in each row.

Category	n	FC(T)	SRAG(T)	Refl.	GA	MemOS	A-MEM	SM(T)	**OpCtx**
LLM-as-a-Judge									
X_1Y_1	10	**60.00**	45.00	45.00	10.00	10.00	35.00	20.00	50.00
X_1Y_2	30	45.00	43.33	41.67	45.00	20.00	38.33	38.33	**63.33**
X_1Y_3	8	**75.00**	62.50	**75.00**	31.25	12.50	43.75	50.00	68.75
X_2Y_1	16	12.50	28.13	18.75	15.63	12.50	15.63	15.63	**56.25**
X_2Y_2	27	37.04	22.22	20.37	22.22	9.26	11.11	14.81	**38.89**
X_2Y_3	5	50.00	40.00	50.00	30.00	0.00	40.00	40.00	**80.00**
X_3Y_1	74	23.65	19.59	21.62	4.05	5.41	17.57	12.84	**48.65**
X_3Y_2	59	34.75	19.49	25.42	11.86	25.42	27.12	21.19	**58.47**
X_3Y_3	9	44.44	44.44	44.44	44.44	5.56	44.44	38.89	**50.00**
X_4Y_1	20	32.50	20.00	15.00	0.00	5.00	12.50	17.50	**50.00**
X_4Y_2	87	25.86	16.67	10.92	10.92	15.52	13.22	9.77	**35.06**
X_4Y_3	26	3.85	13.46	0.00	0.00	11.54	0.00	0.00	**28.85**
Overall	371	30.19	23.45	21.97	13.61	13.34	19.68	17.12	**47.44**
BLEU-1									
X_1Y_1	10	**27.75**	7.79	19.47	9.40	9.25	13.39	9.46	21.17
X_1Y_2	30	40.03	40.64	33.28	38.45	19.80	32.95	29.67	**50.41**
X_1Y_3	8	27.11	19.03	22.74	12.80	6.27	15.08	19.10	**31.06**
X_2Y_1	16	4.03	6.65	5.74	2.15	8.87	7.75	4.89	**11.03**
X_2Y_2	27	14.45	13.19	7.14	8.44	5.63	9.50	12.65	**14.67**
X_2Y_3	5	20.71	16.10	13.03	7.71	6.67	19.89	14.58	**44.33**
X_3Y_1	74	15.22	9.43	12.89	4.05	4.89	8.27	6.45	**27.42**
X_3Y_2	59	14.08	10.79	9.56	4.84	9.80	11.65	9.65	**22.37**
X_3Y_3	9	10.65	12.58	10.03	12.37	9.36	9.71	10.62	**17.38**
X_4Y_1	20	16.01	12.67	9.07	4.60	3.27	13.77	15.00	**29.85**
X_4Y_2	87	21.42	13.00	8.01	7.37	12.89	10.67	5.87	**28.41**
X_4Y_3	26	4.62	9.38	0.76	1.00	11.63	0.38	0.52	**18.44**
Overall	371	17.81	13.66	11.41	8.38	9.64	11.66	9.69	**26.47**

Table 7: MemEye visual-memory baselines and OpCtx by joint category (%). Bold marks the best score among the methods shown in each row.

Category	n	FC(V)	SRAG(V)	MMA	M2A	SM(V)	**OpCtx**
LLM-as-a-Judge							
X_1Y_1	10	60.00	**80.00**	60.00	15.00	25.00	50.00
X_1Y_2	30	43.33	46.67	45.00	20.00	41.67	**63.33**
X_1Y_3	8	50.00	37.50	31.25	12.50	43.75	**68.75**
X_2Y_1	16	46.88	43.75	43.75	15.63	28.13	**56.25**
X_2Y_2	27	20.37	24.07	24.07	7.41	9.26	**38.89**
X_2Y_3	5	60.00	50.00	60.00	0.00	50.00	**80.00**
X_3Y_1	74	43.24	41.22	**54.05**	16.89	18.92	48.65
X_3Y_2	59	41.53	38.98	31.36	18.64	21.19	**58.47**
X_3Y_3	9	44.44	33.33	11.11	16.67	38.89	**50.00**
X_4Y_1	20	**52.50**	**52.50**	47.50	15.00	15.00	50.00
X_4Y_2	87	**37.36**	21.84	28.16	13.22	8.62	35.06
X_4Y_3	26	17.31	23.08	**32.69**	0.00	0.00	28.85
Overall	371	39.62	35.85	37.87	14.15	18.46	**47.44**
BLEU-1							
X_1Y_1	10	23.52	**33.80**	30.55	12.79	9.58	21.17
X_1Y_2	30	29.36	36.66	36.17	17.72	30.01	**50.41**
X_1Y_3	8	18.69	22.73	17.32	5.52	16.83	**31.06**
X_2Y_1	16	13.95	19.07	**19.91**	11.09	6.69	11.03
X_2Y_2	27	7.33	11.72	**16.51**	6.04	7.82	14.67
X_2Y_3	5	28.81	34.40	18.79	10.20	11.77	**44.33**
X_3Y_1	74	21.43	21.45	22.62	14.53	10.08	**27.42**
X_3Y_2	59	16.53	13.17	15.68	10.01	10.98	**22.37**
X_3Y_3	9	12.75	11.79	8.32	12.78	10.12	**17.38**
X_4Y_1	20	20.49	20.89	14.97	6.84	17.88	**29.85**
X_4Y_2	87	23.80	11.87	19.70	11.06	5.48	**28.41**
X_4Y_3	26	13.05	18.42	**24.46**	1.98	0.48	18.44
Overall	371	19.75	18.36	20.78	10.85	10.35	**26.47**

C.2 MemEye Ablations at 30 Facts

Table 8 complements Figure 4b with the joint categories defined in Appendix A. All three configurations use the same generation and judging models across 371 questions, with at most 30 retrieved facts and no model reranking.

Full achieves 47.44%, compared with 38.54% for w/o Reader and 37.47% for w/o Online. Both components improve instance-level relational association, consistent with the need to connect evidence across observations. Pixel-level evolutionary synthesis shows a different pattern, with both ablations scoring higher than Full.

Table 8: MemEye ablations by joint category (mapped Judge score, %; 371 questions). All columns use at most 30 facts without model reranking.

Category	n	Full	w/o Reader	w/o Online
X_1Y_1	10	50.00	50.00	60.00
X_1Y_2	30	63.33	53.33	46.67
X_1Y_3	8	68.75	50.00	50.00
X_2Y_1	16	56.25	25.00	28.12
X_2Y_2	27	38.89	40.74	33.33
X_2Y_3	5	80.00	70.00	80.00
X_3Y_1	74	48.65	35.14	39.19
X_3Y_2	59	58.47	39.83	33.05
X_3Y_3	9	50.00	38.89	44.44
X_4Y_1	20	50.00	45.00	47.50
X_4Y_2	87	35.06	31.61	25.86
X_4Y_3	26	28.85	38.46	50.00
Overall	371	47.44	38.54	37.47

Reader evidence aggregation. Reader constructs evidence using the common rules in Appendix E.4 and aggregates it for QA. The comparison therefore examines supporting-context inclusion and evidence aggregation together.

C.3 Lexical and Semantic Answer Quality

Figure 4d and Table 9 compare six lexical and semantic answer-quality metrics. OpCtx leads on five LoCoMo metrics, while w/o Reader has the highest BERTScore. On MemEye, OpCtx leads on all six metrics among the three configurations, consistent with the Judge-based comparisons.

Table 9: Lexical and semantic answer-quality scores. Higher is better for each metric; bold marks the highest score within each dataset.

Configuration	F1	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore
LoCoMo						
OpCtx	**39.45**	**5.72**	**25.94**	**12.16**	**24.70**	88.32
w/o Reader	37.23	5.44	25.77	11.86	24.68	**88.35**
w/o Selector	37.14	5.44	25.50	11.77	24.60	88.34
w/o Online	36.91	5.41	25.62	11.74	24.37	88.31
MemEye						
OpCtx	**31.04**	**8.66**	**30.64**	**11.34**	**29.65**	**89.78**
w/o Reader	23.26	4.80	21.67	6.63	21.19	88.52
w/o Online	21.90	4.58	20.53	5.97	20.04	88.42

For metric j, the radar radius is $r_j = (s_j - L_j)/(U_j - L_j)$ with display bounds $[L_j, U_j]$. Bounds are shared across methods within each dataset and metric, allowing small score differences to remain visible. Markers and line styles distinguish nearby or identical scores.

C.4 Sensitivity to the Reading Budget

We vary the final retrieved-fact limit under the configurations in Figure 4a,b. LoCoMo uses a fixed candidate pool and Selector limit of 30, followed by Qwen3-Reranker-4B at $k \in \{5, 10, 15, 20, 25\}$. MemEye varies the Selector limit at $k \in \{10, 20, 30, 40, 50\}$ without model reranking (Figure 4b). Memory, Reader rules, models, and prompts remain fixed. The reference budgets are $k = 20$ and $k = 30$, respectively.

Figure 3 shows different responses to increasing the fact budget. LoCoMo improves throughout the tested range, indicating that additional retrieved evidence remains useful. MemEye improves initially but changes little beyond forty facts, while QA input continues to grow. The curves indicate dataset-dependent gains from additional context rather than a uniform benefit from larger budgets.

C.5 Supplementary Component Analyses

Online evidence access. We compare Online with w/o Online on LoCoMo multi-hop questions using the same Writer fact store and a final limit of 20 facts. Online improves source recall both in the candidate pool and after final selection (Table 10), supporting a retrieval-side benefit before Reader aggregation. Randomizing event links over 20 fixed seeds reduces mean candidate-source recall to 86.84%, 1.99 percentage points below Online. These measurements track recovery of annotated dialogue sources, rather than the semantic completeness of retrieved evidence.

Table 10: LoCoMo multi-hop source recall and QA accuracy at 20 facts (%).

Metric	w/o Online	Online
Candidate-source recall	67.30	**88.83**
Final-20 source recall	61.97	**79.48**
QA accuracy	55.32	**73.76**

Matched selection controls. The MemEye selection controls share the candidate pool, 30-fact limit, character ceiling $b(q)$, Reader, and QA settings, without model reranking. OpCtx exceeds dense retrieval, BM25, MMR, and equal-weight multichannel fusion (Table 11). Equal-weight fusion is the strongest alternative at 45.55%, compared with 47.44% for OpCtx. This comparison supports the combined selection objective beyond using multiple retrieval channels, complementing the pipeline comparison in Figure 4c.

Table 11: MemEye selection and Reader controls (mapped score, %; Gemini 3.5 Flash-Lite Judge).

Configuration	Score
Selection controls with OpCtx Reader	
OpCtx Selector-30	**47.44**
Dense Top30	43.66
BM25 Top30	45.28
MMR Top30	40.97
Equal-weight multichannel fusion Top30	45.55
Reader controls with Selector-30	
OpCtx Reader	**47.44**
Generic reasoning Reader	41.64
Direct QA (w/o Reader)	38.54

Reader controls and Judge robustness. Replacing only the Reader prompt with generic reasoning yields 41.64%, while retaining the same selected facts, session summaries, and event claims. OpCtx improves this score by 5.80 percentage points, supporting its question-conditioned evidence organization beyond generic reasoning. With the selected facts fixed, Reader also improves over direct QA by 8.89 percentage points under Gemini 3.5 Flash-Lite and 8.49 under Gemini 3.5 Flash. The observed mean gain remains positive under both Judges.

C.6 Selector Optimization and Runtime

We examine Selector on all 371 MemEye questions with the 30-fact limit and character budget from Appendix E.2. Candidate pools average 395.35 facts, with both dense and lexical channels active for every question. Table 12 reports problem size, numerical certification, and latency.

Table 12: Selector diagnostics on MemEye. Certification uses post-solve numerical checks; times are in seconds.

Metric	Result
Candidate facts, mean (range)	395.35 (122–889)
Active utility channels, mean	3.64
Coverage certified	371/371
All channel ideals certified	371/371
Coverage, ideals, and minimax certified	367/371 (98.92%)
All stages certified, including tie-breaks	315/371 (84.91%)
Hard-constraint violations in returned sets	0
Minimum channel retention, mean	0.9557

Timing scope	Median	P95	Mean
Executed solver stages, cumulative	0.165	1.138	0.338
Complete Selector function	0.187	1.168	0.359

Numerical certification. Certification requires post-solve optimality and reconstruction checks at approximately 10^{-7} tolerance. Coverage and all channel ideals pass for every question, and 367 questions also pass minimax checks. Of the 56 questions without all-stage certification, one reaches the time limit during the total-utility tie-break; 38 fail objective-gap checks (four at minimax and 34 at the utility tie-break); and 17 fail integrality or reconstruction checks at the cost tie-break. The latter 55 return an optimal solver status but fail the stricter checks. These are floating-point certificates, distinct from the exact-solution assumptions in Appendix D.

Runtime and stability. Complete-function latency includes initial model construction and all executed solving stages, but excludes upstream embeddings, query-projection model calls, Reader, and QA. Cumulative stage timing includes solution reconstruction and numerical checks but excludes initial modeling outside those timers. These measurements come from the original run logs. Extending the per-stage limit from 5 to 30 seconds for the 56 uncertified cases preserves every selected set and its order. Only the previously timed-out case gains all-stage certification; the other 55 remain numerically uncertified. Thus the returned evidence is stable under this extension, without establishing optimality for every case.

C.7 QA Input Usage

Table 13 reports mean QA input usage on MemEye with Gemini 2.5 Flash-Lite and on LoCoMo with Gemini 2.5 Flash. These measurements cover input to the final QA model; memory construction, Reader, reranking, and Judge calls are excluded.

Table 13: Mean QA input tokens per question (Input/Q) on MemEye and LoCoMo.

MemEye method	Input/Q	LoCoMo method	Input/Q
FC(T)	8,607	Full Context	32,176
SRAG(T)	1,063	RAPTOR	2,106
Reflexion	8,092	HippoRAG 2	2,237
Generative Agents	1,022	HyperMem	2,151
MemoryOS	2,655	Mem0	3,489
A-MEM	15,058	A-MEM	2,041
SM(T)	1,937	RecurSum	1,843
FC(V)	27,786	MemGAS	6,708
SRAG(V)	3,560	Naive RAG	2,673
MMA	2,192	SeCom	2,034
M2A	263	MemoryOS	3,424
SM(V)	1,942	LightMem	982
		SimpleMem	1,568
OpCtx ($k = 30$)	709.11	**OpCtx** ($k = 20$)	3,174.92

D Properties of the Evidence Objective

The following properties use a fixed candidate pool, duplicate policy, renderer, budget, and set of active channels. The feasible domain is nonempty, and channel ideals are computed exactly on that domain.

Proposition: exact worst-case utility retention. Assume $\mathcal{J} \neq \varnothing$, with coverage-maximizing domain $\mathcal{D}$ from Equation 2, channel ideals $I_j > 0$ from Equation 3, and normalized utilities $r_j(S) = U_j(S)/I_j$. For each $S \in \mathcal{D}$, Equation 5 states that the worst-case fraction of attainable mixture utility equals $r(S) = \min_j r_j(S)$. Equation 4 therefore maximizes this uniform retention guarantee.

Proof. Fix S and write $a = r(S)$. Nonnegative utilities give $a \geq 0$ and $U_j(S) \geq aI_j$ for every active channel. Since $U_j(T) \leq I_j$ for every $T \in \mathcal{D}$, any $w \in \mathcal{W}$ satisfies

$$v_w(S) = \sum_j w_j U_j(S) \geq a \sum_j w_j I_j \geq a v_w^*. \tag{D.1}$$

The denominator v_w^* is positive because some $w_j > 0$ and a feasible set attains $I_j > 0$. Equation D.1 gives $v_w(S)/v_w^* \geq a$. Choose $k \in \arg\min_j r_j(S)$ and set $w_k = 1$, with all other weights zero. The ratio is then $U_k(S)/I_k = a$, attaining the bound and proving Equation 5. An exact maximizer S^* therefore achieves the largest uniform retention factor on the shared candidate, coverage, and cost domain.

The weight domain is the full simplex, including the single-channel weights used to attain the bound.

Minimax equivalence and solver form. The implementation solves Equation 4 through normalized channel constraints. Equivalently, introduce the worst relative shortfall $\rho = 1 - r(S)$ and solve

$$\begin{aligned} \min_{S,\rho} \quad & \rho \\ \text{subject to} \quad & S \in \mathcal{D}, \quad 0 \leq \rho \leq 1, \\ & U_j(S) \geq (1-\rho) I_j, \quad j \in \mathcal{J}. \end{aligned} \tag{D.2}$$

For fixed S, the smallest feasible ρ in Equation D.2 is

$$\rho(S) = \max_{j \in \mathcal{J}} \big(1 - r_j(S)\big) = 1 - \min_{j \in \mathcal{J}} r_j(S) = 1 - r(S).$$

The two formulations therefore have the same optimal sets and retain the same total-retention and cost tie-breaks.

Property 1: scale invariance. Multiplying each channel utility by $a_j > 0$ preserves selection on the fixed domain and active channel set. Since $\widetilde{U}_j(S) = a_j U_j(S)$ and $\widetilde{I}_j = a_j I_j$, we have $\widetilde{r}_j(S) = r_j(S)$. The max–min objective and both tie-breaks are unchanged.

Property 2: Pareto efficiency within the covering domain. Let S^* maximize $r(S)$ and, among those maximizers, maximize $\sum_j r_j(S)$. No $S' \in \mathcal{D}$ can satisfy $U_j(S') \geq U_j(S^*)$ for every active channel and a strict inequality for at least one. Otherwise $r(S') \geq r(S^*)$. A strict increase contradicts max–min optimality; equality makes the total normalized utility strictly larger, contradicting the tie-break. Minimum cost only resolves remaining ties and preserves this property.

Computational scope. The properties apply to Selector output before reranking. Certified maximum coverage defines $\mathcal{D}$; a partially solved coverage stage instead locks $\widehat{h}$ and uses $\widehat{\mathcal{D}} = \{S \in \mathcal{B} : h(S) = \widehat{h}\}$. Equation 5 requires exact ideals on the chosen domain. Optimal selection and Pareto efficiency additionally require exact selection and tie-break stages. Uncertified stages are labeled partial. If $\mathcal{J}$ is empty, cost resolves coverage ties.

E Implementation Details

E.1 Online Updates

The local Writer produces $(s_t, \mathcal{F}_t) = W(B_t)$ and appends the source facts unchanged. Online retrieves historical facts and active views from the observed prefix, then proposes and commits valid create/extend operations. Checks cover source-reference visibility, claim-support membership, and operation validity. Writing batches are processed in causal order. Each update returns at most eight operations, with at most 24 member facts per operation.

E.2 Retrieval and Budgets

The candidate pool combines facts expanded from retrieved summaries and event views with dense and BM25 facts. Summaries and event views are ranked jointly. On LoCoMo and LongMemEval-S, we expand the top five and add the top ten facts from each fact-level retriever. On MemEye, we expand the top ten and add the top twenty BM25 facts and top twenty dense facts to accommodate richer multimodal content. Selector chooses from the merged pool. Its limit K controls preselection, while k limits the final facts supplied to Reader:

Dataset	K	After Selector	k	Character ceiling
LoCoMo	30	Qwen rerank	20	None
MemEye	30	No model reranking	30	$c(D_{50}(q))$
LongMemEval-S	60	Qwen rerank	20	None

LoCoMo and LongMemEval-S use Qwen3-Reranker-4B after Selector; MemEye passes the selected facts directly to Reader. MemEye uses $b(q) = c(D_{50}(q))$, where $D_{50}(q)$ contains the top 50 facts by dense cosine similarity from the same merged candidate pool. Both S and $D_{50}(q)$ use the same rendering of facts, source-session summaries, and eligible event claims, including separators and counting each shared header once. LoCoMo and LongMemEval-S use $b = \infty$, with rendered cost retained as a tie-break.

E.3 Selector Execution

Channels and solvers. Let $\mathcal{A}$ be the exact entity, relation, and scope spans extracted from q, and $a_1, \ldots, a_m$ its requested attribute spans. The projected queries are

$$q_0 = q, \qquad q_\ell = g(\mathcal{A} \cup \{a_\ell\}), \quad \ell = 1, \ldots, m, \tag{E.1}$$

where g deterministically merges overlapping spans and joins them in their original question order.

For each original or projected query q_ℓ, the dense and lexical scores are

$$\begin{aligned} s_{\ell,d}(f) &= \frac{e(q_\ell)^\top e(f)}{\|e(q_\ell)\|_2 \, \|e(f)\|_2}, \\ s_{\ell,b}(f) &= \mathrm{BM25}(q_\ell, f), \end{aligned} \tag{E.2}$$

where e is the frozen gemini-embedding-001 embedding function. For channel $j = (\ell, d)$ or (ℓ, b) and $N = |\mathcal{P}| > 1$, its initial utility is

$$u_j(f) = \frac{\mathrm{MidRank}_{\mathcal{P}}(s_j(f)) - 1}{N - 1}. \tag{E.3}$$

The ascending midrank is one plus the number of lower scores plus half the number of other tied scores. Ranks include all N candidates; nonpositive BM25 utilities are then set to zero. For $N = 1$, dense utility is zero, and BM25 utility is one only for a positive score. Empty pools return no facts. Constant positive BM25 channels remain active, constant dense channels are disabled, and identical utility arrays are merged.

The BM25 index uses $k_1 = 1.5$ and length normalization $\beta = 0.75$. For query token w, the contribution is

$$\log\left(1 + \frac{N - \mathrm{df}(w) + 0.5}{\mathrm{df}(w) + 0.5}\right) \frac{\mathrm{tf}(w, f)(k_1 + 1)}{\mathrm{tf}(w, f) + k_1\big(1 - \beta + \beta|f|/\overline{L}\big)},$$

where the fixed tokenizer determines term frequencies and lengths, and the mean fact length $\overline{L}$ is floored at one. Contributions are summed over the query sequence, including repeated tokens. Index parameters are fixed across questions and categories.

Query projections use exact spans with schema and unique-location checks; invalid extractions revert to the original query.

Selection procedure. Selector constructs query projections, scores channels, and computes rank utilities. It maximizes literal coverage under the count, character, and duplicate constraints, then locks the attained coverage. Each channel ideal is optimized independently on this same coverage-constrained domain, without locking the optima of other channels. Selector then optimizes minimum retention, total retention, and rendered cost in that order. The selected facts are validated and placed in the fixed display order. Appendix D specifies the guarantees for completed and partially solved stages.

Solver configuration. We use `scipy.optimize.milp` with a 5-second limit per stage, `presolve=False`, and `mip_rel_gap=0`. The original execution environment currently reports Python 3.11.4, SciPy 1.15.3, and HiGHS 1.8.0 on 64-bit Windows 11, with an Intel Core i9-13900HX (24 cores, 32 threads) and 32 GiB RAM; dependency versions were not archived at run time. HiGHS 1.8.0 defaults to 10^{-6} absolute MIP-gap and feasibility tolerances, whereas our post-solve checks use approximately 10^{-7}. Setting a zero relative gap does not remove these numerical tolerances. Appendix C.6 reports observed latency and certification rates.

E.4 Reader Evidence Aggregation

Reader uses the same evidence-construction procedure across all three datasets. It retains the final retrieved facts $\mathcal{R}$ and their source-session summaries, then adds eligible claims from at most two event views. Views retrieved through summary routing must share at least one fact with $\mathcal{R}$; other views require at least two shared facts. Retrieved views retain query order, followed by additional views ranked by shared-fact count and view ID.

A claim must have nonempty text and nonempty source support entirely within $\mathcal{R}$. Claims are ranked by question-token overlap, support count, and original position. Each view contributes at most three claims of up to 260 characters each. The event portion is limited to 600 characters, including headers and separators. Views without qualifying claims or sufficient budget are skipped. If none qualifies, Reader uses the facts and session summaries alone.

Reader organizes and aggregates this evidence for the answering operation. QA receives only the resulting evidence packet.

E.5 Component Controls

w/o Reader passes the same ordered facts and their source-session summaries directly to QA, omitting event claims and Reader aggregation. Appendix B.2 specifies the Online and Selector controls.

F CASE STUDY

The six cases below illustrate how Reader aggregation, Online event updates, and memory retrieval affect the evidence available for answering.

F.1 READER: USING RETRIEVED EVIDENCE

Case 1. Conditional counting across visual observations

Query: How many of the 5 Starbucks ads did not show a coffee cup or coffee drink?

Ground-truth answer: 2

Recorded responses

OpCtx Full: 2 *(Correct)*

w/o Reader: 1 *(Incorrect)*

Ad	Coffee cup / coffee drink?
Starbucks_1: Ethos Water	No
Starbucks_2: paper cup	Yes
Starbucks_3: glass mug	Yes
Starbucks_4: Coffee Braille	No
Starbucks_5: seasonal drinks	Yes

What changes: The Reader enumerates the five ads and distinguishes coffee beans and water from coffee cups or drinks; direct QA returns only one qualifying ad.

Case 2. Comparing two attributes jointly

Query: Which raises both Meal Variability and Evening Drift relative to the other between the D5 comparison card and the D5 follow-up trend?

Ground-truth answer: Neither does

Recorded responses

OpCtx Full: Neitherraises both metrics: the D5 follow-up trend increases Evening Drift (from 2.3h to 3.6h) but decreases Meal Variability (from 13 pts to 10 pts) relative to the D5 comparison card. *(Correct)*

w/o Reader: Unavailable *(No answer)*

Screen	Meal Variability	Evening Drift
D5 comparison card	13 pts	2.3 h
D5 follow-up trend	10 pts	3.6 h

What changes: The Reader places the two metrics side by side: one decreases while the other increases, so neither screen raises both. Direct QA abstains.

Figure 5: Reader aggregation supports conditional counting and joint comparison across retrieved observations.

F.2 Online: Preserving Answer Evidence

Case 3. Recovering a later state update

Query: What pets does Nate have?

Ground-truth answer: A dog and three turtles.

Recorded responses

OpCtx Full: Nate has a dog named Max and three turtles. *(Correct)*

w/o Reader: Nate has three turtles and a dog named Max. *(Correct)*

w/o Reader & Online: Nate has two turtles and a dog named Max. *(Incorrect)*

Source evidence	Online, no Reader	No Online / Reader
D12:3 / F0089: adopted Max	Included	Included
D2:10-14 / F0011: two turtles	Included	Included
D28:25 / F0243: a third turtle	Included	Not selected

What changes: Online retrieval retains the later third-turtle fact. Removing Online selects a generic turtle note instead, leaving the old count of two in the context.

Case 4. Completing a cross-session answer set

Query: Where has Melanie camped?

Ground-truth answer: beach, mountains, forest

Recorded responses

OpCtx Full: Melanie has camped in the mountains, the forest, and at the beach. *(Correct)*

w/o Reader: Melanie has camped in the mountains, the forest, and at the beach. *(Correct)*

w/o Reader & Online: Melanie has camped in the mountains and in the forest. *(Incorrect)*

Source evidence	Online, no Reader	No Online / Reader
D4:6 / F0028: mountains	Included	Included
D8:32 / F0076: forest	Included	Included
D6:16 / F0051: beach	Included	Not selected

What changes: Both variants recover mountain and forest camping. Online also retrieves the beach record; without Online, the answer omits the beach.

Figure 6: Online retrieval recovers the later turtle observation and the beach record required by the two questions.

F.3 Baseline Pipeline Comparisons

Case 5. Counting a visual attribute across advertisements

Query: How many of the five Coca-Cola ads used a solid red background?

Ground-truth answer: 3

Recorded responses

OpCtx: 3 *(Correct)*

SRAG(T): 2 *(Incorrect)*

A-MEM: 1 *(Incorrect)*

Image / fact	Relevant detail
CocaCola_3 / F00125	Red background
CocaCola_4 / F00189	Red background
CocaCola_5 / F00253	Red background

What changes: OpCtx retains background-color evidence for three Coca-Cola ads. SRAG(T) and A-MEM return smaller counts.

Case 6. Linking one person across work and apartment chats

Query: Which friend serves as the bridge between your work and apartment life, connecting different parts of your life?

Ground-truth answer: Priya - she's the bridge between work and my apartment life

Recorded responses

OpCtx: Priya serves as the bridge between work and apartment life. *(Correct)*

SRAG(T): Daniel. *(Incorrect)*

A-MEM: Information unavailable. *(No answer)*

Source observation	Relationship involving Priya
CHAT_S4: onboarding	Mentor to Daniel at work
CHAT_S34: PR Review Help	Work conversation with Daniel
CHAT_S6 / CHAT_S38	Apartment utilities / lease with Sara

What changes: OpCtx retains Priya's work and apartment roles together. SRAG(T) names Daniel, while A-MEM abstains.

Figure 7: Case comparisons with SRAG(T) and A-MEM on visual-attribute counting and cross-context identity linking.

G PROMPT TEMPLATES

Prompt for Online Event Update (MemEye)

```
You update compact cross-session event routing views after receiving MemEye session
{frontier_session_id}.

The input is limited to the current session, selected earlier facts, and selected
active views from the causal prefix. Atomic facts are immutable evidence. MemEye
sessions contain many visual state snapshots; sharing a task, game, person, object,
or activity is not enough to make snapshots one event. Connect only facts that
describe the same concrete occurrence, a continuing process, or an explicit
state/update chain across sessions.

Output STRICT JSON:
{
  "operations": [
    {
      "operation": "create|extend",
      "target_view_id": "",
      "view_type": "same_event|continuing_process|update_chain",
      "summary_claims": [
        {"text": "<one grounded claim>", "support_fact_ids": ["F0001"]}
      ],
      "member_fact_ids": ["F0001", "F0002"]
    }
  ]
}

Rules:
1. Copy all IDs exactly. A create spans current and earlier sessions; an extend
targets one supplied active view, preserves all its members, and adds a
current-session fact.
2. Include only the smallest decisive evidence set, normally 4-12 facts and never
more than 24 facts per operation. Do not copy every snapshot from a session.
3. Every member fact must appear in at least one summary_claim support list. Every
claim must be fully supported by its cited member facts.
4. Keep separate game rounds, visits, dates, attempts, and repeated activities
separate unless the evidence explicitly makes them one continuing occurrence or
update chain.
5. Membership only grows. Never rewrite, delete, merge, or duplicate atomic facts.
Return at most 8 operations.
6. For create, target_view_id MUST be exactly the empty string "". For extend, it
MUST be one supplied active view ID.
7. Before returning, verify that every create contains at least one current_session
fact ID and at least one prior_candidate_facts fact ID. The
cross_session_candidate_pairs are similarity hints, not proof; use only pairs whose
texts support the claimed relation.
8. Prefer an empty operations list over a broad thematic group. Return JSON only.

Causal prefix input JSON:
{online_input_json}
```

Figure 8: Online update prompt for constructing event views from the current batch and retrieved historical evidence.

Prompt for Online Event Update (LoCoMo)

```
You update causal cross-session routing views after receiving dialogue session
{frontier_session_id}.

The input is strictly limited to the current session, selected facts from earlier
sessions, and selected active views created from that prefix. Atomic facts are
immutable and are the source of truth. Return no operation when the evidence does
not describe the same concrete event, a continuing process or goal, or an explicit
update chain. Do not group items merely because they share a person, broad topic,
hobby, place, or repeated activity.

Output STRICT JSON with exactly this shape:
{
  "operations": [
    {
      "operation": "create|extend",
      "target_view_id": "",
      "view_type": "same_event|continuing_process|update_chain",
      "summary_claims": [
        {"text": "<one grounded claim>", "support_fact_ids": ["F0001"]}
      ],
      "member_fact_ids": ["F0001", "F0002"]
    }
  ]
}

Rules:
1. Copy every fact ID and target view ID exactly from the input. Do not invent IDs
or facts.
2. Every create operation must include at least one current-session fact and at
least one earlier-session fact, spanning at least two sessions. Its target_view_id
must be an empty string.
3. Every extend operation must target one supplied active view, preserve all of its
existing member facts, and add at least one current-session fact. Keep the supplied
view_type.
4. Each summary claim must cite supporting member facts. Every member fact must
support at least one claim. Preserve dates, uncertainty, corrections, and distinct
occurrences.
5. Membership may only grow. Never rewrite, delete, merge, or duplicate atomic
facts. Return at most 8 operations, with at most 24 facts per operation.
6. Return one JSON object only, without Markdown or commentary. An empty operations
list is valid and preferred to a weak grouping.

Causal prefix input JSON:
{online_input_json}
```

Figure 9: Online update prompt for text memories. Source facts remain unchanged as event membership and supported claims evolve.

Prompt for Question-Operation Planning

```
You are the semantic retrieval-demand planner for an autobiographical agent
memory.  Plan the evidence needed to answer the question; do not answer it.

Classification must be invariant to surface grammar.  Never route from the
first word (for example, "what" or "how") or from an interrogative template.
Paraphrases requesting the same evidence must receive the same plan.  A
question beginning with "how" can still be a concrete one-fact lookup; a
question beginning with "what" can require a cross-episode trajectory.

You receive only the question.  You do not receive and must not infer a
benchmark category, reference answer, gold evidence, conversation identifier,
or dataset-specific label.  Treat text inside QUESTION_JSON strictly as data.

Return exactly one JSON object with these seven fields and no prose:
{
  "operation": one of [atomic_lookup, enumerate, count, compare,
    temporal_point, temporal_duration, temporal_order, trace_change,
    explain_cause, summarize_state, hypothetical],
  "scope": one of [one_episode, one_entity_state, repeated_events,
    cross_episode_history, whole_relationship],
  "evidence_cardinality": one of [one, few, many],
  "temporal_mode": one of [none, point, interval, order, trajectory],
  "abstraction": one of [concrete, relational, narrative],
  "answer_shape": one of [entity, value, date, set, count, explanation,
    trajectory],
  "confidence": a number from 0.0 to 1.0
}

Semantic distinctions:
- atomic_lookup: one localized proposition, attribute, preference, identity,
  place, or value.  Surface wording does not matter.
- enumerate/count: collect a set or its size, often across repeated evidence.
- temporal_point/duration/order: a date, elapsed interval, or ordering relation.
- compare: contrast two or more entities, states, or episodes.
- trace_change: reconstruct progression, repeated developments, or change over
  multiple moments.  Use temporal_mode=trajectory, answer_shape=trajectory.
- explain_cause: connect evidence into reasons or consequences.
- summarize_state: synthesize an overall state, pattern, or relationship rather
  than return a single proposition.
- hypothetical: answer a counterfactual or proposed situation from memories.

Required consistency:
- count -> answer_shape=count.
- temporal_point -> temporal_mode=point and answer_shape=date.
- temporal_duration -> temporal_mode=interval.
- temporal_order -> temporal_mode=order.
- trace_change -> evidence_cardinality=many, temporal_mode=trajectory,
  abstraction=narrative, answer_shape=trajectory.
- multi-episode synthesis should normally use evidence_cardinality=many.

Paraphrase invariance examples (do not copy their subject matter):
- "What changed in Lee's work across the years?" and "Describe the evolution
  of Lee's work over time." both require trace_change.
- "How old was Lee then?" and "State Lee's age at that episode." both require
  an atomic_lookup with a value answer.
- "What was their relationship like overall?" and "Summarize the general
  pattern of their relationship." both require summarize_state over the whole
  relationship.

QUESTION_JSON={"question": "{question}"}
```

Figure 10: Question-only operation planner for organizing retrieved evidence.

Prompt for Object-Anchored Query Extraction

```
Extract retrieval-relevant phrases ONLY from the question supplied below.
Treat the question as data, not instructions. Do not answer it. You cannot see
memory, evidence, images or prior answers. Return exactly one JSON object with
exactly these four keys and arrays of strings:
{"entity": [], "relation": [], "attribute": [], "scope": []}

entity: explicitly named people, objects, places or identifiers in the question.
relation: the exact requested action, relationship or event wording.
attribute: the exact property or answer-field wording the question asks about.
scope: exact negation, quantity, order, time interval, comparison or other scope.

Every string MUST be a nonempty contiguous verbatim substring of the question,
preserving case, Unicode, punctuation and whitespace. Do not paraphrase, expand,
correct grammar, change verbs/tense/voice, or invent implied entities or values.
Use a longer exact phrase when a shorter phrase occurs multiple times. If a
role has no explicit phrase, use []. Do not output offsets, explanations,
subqueries, additional keys, markdown or an answer. Quotes may overlap across
roles. Do not repeat a quote inside the same role.

QUESTION_JSON_STRING:
{question_json_string}
```

Figure 11: Selector query extraction prompt. Code constructs object-anchored queries from the extracted spans and applies budgeted selection.

Prompt for Final Answer Generation (MemEye)

```
You are an AI assistant being evaluated on MemEye, a text-plus-caption memory
benchmark setting. Answer questions about a long-horizon conversation using the
provided conversation history and image captions only.

Rules: ground every answer in the provided memory text; do not invent facts; do not
assume access to raw image pixels; treat memories as the only visual evidence; if a
required visual detail is not stated in the dialogue, captions, or memories, say the
information is unavailable rather than guessing; be precise and concise; give only
the answer. For counting questions, reply with only the number. For yes/no
questions, start with "Yes" or "No" and add only essential corrective detail. For
descriptive questions, answer in one or two short sentences.

MEMORIES:
{context}

QUESTION: {question}
ANSWER:
```

Figure 12: MemEye QA prompt. Full supplies the evidence packet produced by Reader aggregation.

Prompt for Final Answer Generation (LoCoMo)

```
You are an intelligent dialog bot. You will be shown History Dialogs. Please read,
memorize, and understand the given Dialogs, then generate one concise, coherent and
helpful response for the Question.
History Dialogs: {retrieved_texts}
Question: {question}
```

Figure 13: LoCoMo QA prompt. In Full, the History Dialogs field contains the evidence packet produced by Reader aggregation.

Prompt for Answer Evaluation (MemEye)

```
You are an impartial judge evaluating the memory capabilities of an AI assistant on
a question-answering task. Compare the Assistant's Answer against the Ground Truth
and assign one score from {0,0.25,0.5,0.75,1}.

Important principles. Use semantic equivalence over surface form. Numeric and
counting questions are binary: the exact number receives 1, and a wrong number
receives 0. For negation-plus-correction questions, the answer must include both the
correct polarity and the corrected fact. Identity questions are correct if the same
unique person, object, or event is identified, even if the wording differs. Judge
whether the answer gives the requested attribute or type of information. Wrong
entities should not receive high scores just because they are on topic. Multi-item
questions require matching the required set. Chronology and ordering questions
require the correct ordered item. Do not over-reward fluent explanations. Opposite
relations or orderings are normally scored 0.

Scoring rubric. Score 0 for contradictions, wrong yes/no polarity, wrong numbers,
wrong answer type, wrong relation or order, hallucination, or no relevant
information. Score 0.25 for answers that touch the topic but miss the core entity or
value, contain major wrong associations, are excessively vague, or get most of a
required set wrong. Score 0.5 for answers that capture the main entity or concept
but miss important qualifiers, or for multi-item answers that get a substantial part
of the set right. Score 0.75 for largely accurate answers with only minor missing
detail or unnecessary filler. Score 1 for accurate, precise answers that contain all
core information without hallucinations; exact wording is not required.

Input fields. Question: {question}. Ground Truth: {ground_truth}. Assistant Answer:
{model_output}.

Output format. Return only a JSON object with fields "score" and "reasoning". The
score must be one of 0, 0.25, 0.5, 0.75, or 1; the reasoning should be short.
```

Figure 14: MemEye Judge prompt, which compares the generated answer with the reference answer using five score levels.

Prompt for Answer Evaluation (LoCoMo)

```
I will give you a question, a reference answer, and a response from a model. Please
answer [[yes]] if the response contains the reference answer. Otherwise, answer
[[no]]. If the response is equivalent to the correct answer or contains all the
intermediate steps to get the reference answer, you should also answer [[yes]]. If
the response only contains a subset of the information required by the answer,
answer [[no]].
[User Question]
{question}
[The Start of Reference Answer]
{answer}
[The End of Reference Answer]
[The Start of Model's Response]
{response}
[The End of Model's Response]
Is the model response correct? Answer [[yes]] or [[no]] only.
```

Figure 15: Binary Judge prompt shared by the compared LoCoMo methods.