PICORER: AGENTIC MEMORY, ONE CLUE AT A TIME

Anonymous authors
Paper under double-blind review

ABSTRACT

Agentic memory gives LLM agents control over how past experience is organized, retrieved, and used. Using this experience for a task requires deciding what information to acquire and how separate findings fit together. As the agent reads, newly discovered facts can change what it needs to seek next. Effective memory acquisition therefore demands adaptive exploration while preserving accumulated evidence. We present **Picorer**[1], a memory harness that acquires evidence one clue at a time. Its LLM policy revises a working state to guide search, while an independent ledger retains inspected passages for answering. Under at most eight search calls per query, Picorer raises MemoryAgentBench Overall from 41.60% to 49.17% with GPT-4o-mini, outperforming HippoRAG-v2 and MemGPT. With GPT-4.1-mini it leads the reproduced BEAM baselines at 72.52% and 60.49% on the 100K- and 10M-token archives, and with GPT-5-mini it reaches 90.00% on LoCoMo. Controlled interventions on multi-hop fact consolidation show that completing supporting relations yields larger accuracy gains than partial progress. Further reading can still improve accuracy after an independent model judges the evidence sufficient.

1 INTRODUCTION

Memory enables LLM agents to draw on past experience beyond their active context (Park et al., 2023; Packer et al., 2023). Agentic memory gives them control over how that experience is organized, retrieved, and used. Existing systems make experience retrievable by organizing it into structured memories (Xu et al., 2025; Kang et al., 2025; Liu et al., 2026). Yet a question does not always specify every fact needed to answer it. Reading one record can reveal a missing relation or identify an entity to investigate next. Effective memory use therefore requires the agent to refine what it seeks as it learns from what it reads. We study this process as *memory acquisition*: identifying, inspecting, and retaining information as the task's requirements become clearer.

MemoryAgentBench (Hu et al., 2026b) makes this challenge concrete. Its *multi-hop fact consolidation* task requires linking facts across a long archive and favoring newer information when facts conflict. Finding one relation can identify the entity to search for next, while answering requires combining relations acquired at different steps. Iterative retrieval uses these findings to guide further searches (Trivedi et al., 2023; Lee et al., 2024). Yet a working summary reflects only the agent's current understanding. Revising it may remove a detail whose relevance becomes clear after a later read. If that summary also supplies the answer model's evidence, adapting the search can discard information needed for reasoning. Acquisition therefore needs to separate what guides the next search from what remains available for answering.

We introduce **Picorer**, a memory harness built around this separation (Figure 1). It formulates acquisition as a sequence of decisions guided by a revisable working state. An LLM policy uses this state to decide what to search, which sources to inspect, and when to stop. The state captures what has been learned and what remains missing, so each observation can redirect acquisition. An independent ledger retains the inspected source text for the answer model. The agent can thus revise its understanding without discarding the evidence accumulated along the way.

[1] *Picorer* is French for "to peck at".

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.

We evaluate Picorer on the full MemoryAgentBench and BEAM (Tavakoli et al., 2026) suites and the non-adversarial split of LoCoMo (Maharana et al., 2024), covering factual and conversational memory with at most eight search calls per question. With GPT-4o-mini, Picorer raises MemoryAgentBench Overall from 41.60% for HippoRAG-v2 (Jiménez Gutiérrez et al., 2025) to 49.17%, also outperforming MemGPT (Packer et al., 2023). With GPT-4.1-mini, it leads the reproduced BEAM baselines (MemTensor, 2026) at both the 100K- and 10M-token archive scales, scoring 72.52% and 60.49%, respectively. On LoCoMo, it reaches 90.00% accuracy across 1,540 questions with GPT-5-mini. The same acquisition design thus supports memory use across task types and archive scales.

To explain these gains, we examine the evidence that reaches the answer model. On multi-hop fact consolidation with Qwen3.6-27B, Picorer supplies complete supporting chains more often than A-MEM, MemoryOS, and SimpleMem. Ablations show that the working state's note reduces repeated reads, while retention preserves relations found at different steps. Removing supporting relations sharply lowers accuracy, whereas completing the chain yields the largest gains. Further reads improve answers even after an independent model judges the available evidence sufficient.

Our contributions are:

1. **Memory harness.** We introduce Picorer, a memory harness that guides acquisition with a revisable working state and retains source evidence independently.
2. **Acquisition benefits.** We demonstrate that cumulative evidence retention improves multi-hop accuracy and a structured working note reduces redundant inspection.
3. **Sufficiency analysis.** We show that model judgments of sufficiency can precede decisive evidence, leaving substantial gains from continued acquisition.

2 Related Work

Agentic memory. Agentic memory gives agents control over how experience is retained and reused beyond the active context (Park et al., 2023; Packer et al., 2023; Zhong et al., 2024). A central concern is how to organize that experience for future use. MemoryOS and MIRIX use hierarchies (Kang et al., 2025; Wang & Chen, 2025), while A-MEM links notes (Xu et al., 2025) and graph memories connect related records (Gutiérrez et al., 2024; Rasmussen et al., 2025). Consolidation creates compact records (Chhikara et al., 2025; Salama et al., 2025; Fang et al., 2026; Ma et al., 2026; Wu et al., 2026), while distilling past attempts yields reusable guidance (Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2025; Ouyang et al., 2026). These representations make experience accessible, but which pieces are needed for an answer may become clear only as the agent reads.

Adaptive acquisition. This dependence on new observations motivates retrieval that can change course during reasoning (Yao et al., 2023; Trivedi et al., 2023; Asai et al., 2024; Li et al., 2026b; Shi et al., 2026). SimpleMem and EverMemOS adapt retrieval over constructed memories (Liu et al., 2026; Hu et al., 2026a). ReadAgent revisits original passages from compressed context (Lee et al., 2024). General Agentic Memory iterates search and synthesis, returning both integrated findings and associated source information (Yan et al., 2025). S2G-RAG makes remaining information gaps explicit and uses them to guide further retrieval (Li et al., 2026a). The agent must also carry useful information across these steps. Working-context compression (Hu et al., 2025), feedback from cited evidence (Tan et al., 2025), and source representation and ranking (Jin et al., 2026; Derehag et al., 2026) shape that selection. Picorer makes the distinction between search control and source retention explicit. Its working state combines an editable note with current observations to direct the next action. The harness independently accumulates every inspected passage for answering, so retention does not depend on what the policy keeps in its note.

Memory utility. The value of retained memory depends on how it influences subsequent decisions and answers (Jiang et al., 2026). Reinforcement learning approaches optimize memory operations with task rewards (Yu et al., 2026; Yan et al., 2026). MemoryArena tests whether stored experience guides later actions in interdependent tasks (He et al., 2026), while behavioral studies show that recalling past experience can also propagate errors (Xiong et al., 2026). In question answering, Sufficient Context distinguishes whether retrieved material supports an answer from whether the model

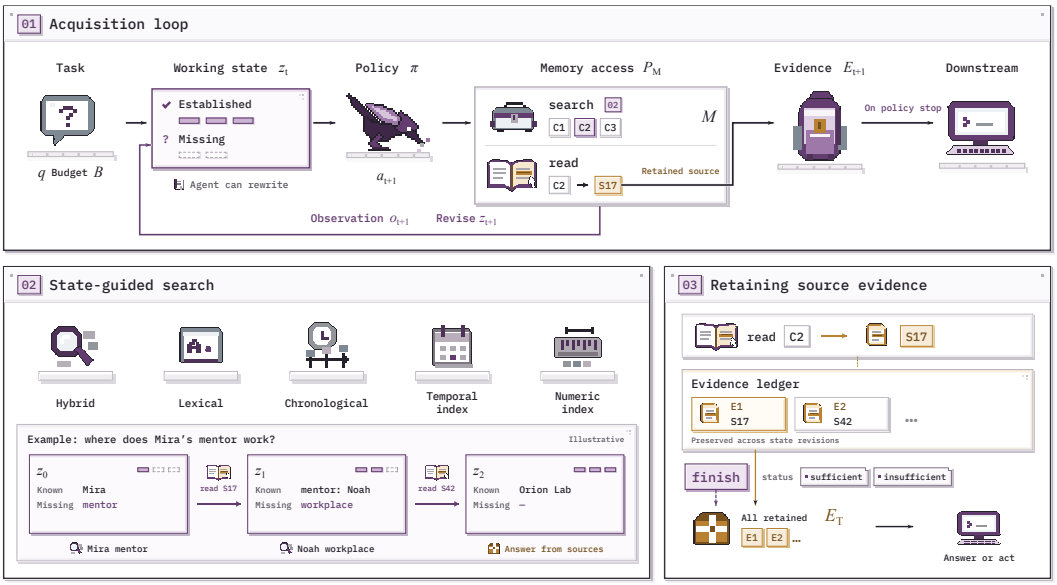

Figure 1: **Overview of Picorer.** The policy uses the working state to search and inspect memory. Returned observations update the state, while the ledger preserves read sources for the downstream model. The lower panels illustrate how a new clue changes the search target and how source evidence remains available across state revisions.

answers correctly (Joren et al., 2025). These perspectives connect memory utility to how information contributes to a task. During acquisition, that contribution can change as new observations complement what the agent has already found.

3 PICORER

We first formulate memory acquisition as sequential decision making. We then introduce the components of Picorer's architecture (Figure 1) and explain how they work together in the acquisition loop.

3.1 MEMORY ACQUISITION AS SEQUENTIAL DECISION MAKING

Given an archive $\mathcal{M}$ and question q, the agent acquires evidence for answering under a budget of B search calls. We adopt the *information-state* view of sequential decision making (Kaelbling et al., 1998; Subramanian et al., 2022) to represent the policy's evolving understanding of the task. Let $\tau_t = (a_i, o_i)_{i=1}^{t}$ be the action–observation history and E_t the retained source text, both initially empty. The working state is

$$z_t = \rho(q, \tau_t, E_t, B), \tag{1}$$

where ρ combines the latest working note and current observations with references to inspected sources and the remaining search budget. The policy chooses a search, a read, or termination ($\perp$):

$$a_{t+1} \sim \pi(\cdot \mid q, z_t), \qquad o_{t+1} \sim P_{\mathcal{M}}(\cdot \mid \tau_t, a_{t+1}), \tag{2}$$

where $P_{\mathcal{M}}$ returns tool observations for nonterminal actions. The next state incorporates the returned observation, allowing the policy to revise its search target or choice of sources.

Let $H(q, E)$ prepare the answer input from retained sources, and let $U(q, x)$ be the answer model's expected task score given input x. The acquisition objective is

$$\max_{\pi} \mathbb{E}[U(q, H(q, E_T))] \quad \text{s.t.} \quad N_{\text{search}}(\tau_T) \le B, \tag{3}$$

where T is the stopping time and N_{search} counts search calls. The objective measures acquisition through the answer supported by its collected evidence. Picorer implements π with an LLM guided by a *retrieval skill* that describes how to use the working state and memory interface (Appendix A.1).

3.2 Components of Picorer

Picorer combines a working state that guides acquisition, a memory interface that provides access to sources, and an independent ledger that retains inspected evidence for answering.

Revisable working state

The harness assembles z_t from the current observation, references to inspected sources, the remaining search budget, and a note written by the policy. The note distinguishes *Established* facts supported by read sources from *Missing* facts or relations. Before each action, the policy revises the note to preserve relevant findings, remove overturned claims, and specify unresolved relations. Each revision replaces the previous note, giving the policy a current account of the task rather than a growing transcript. A newly discovered entity can become the subject of the next query, while a resolved gap no longer needs further search.

Composable memory access

The memory interface lets the agent choose and combine retrieval operations to address missing information. Semantic search locates related content, lexical search matches specific names or phrases, and temporal and numerical operations filter or order records. A single search can combine multiple retrieval branches. Union collects their candidates, reciprocal-rank fusion (Cormack et al., 2009) combines their rankings, and intersection retains shared candidates. The policy can therefore combine broad semantic matches with precise lexical cues. Search returns previews that help the agent decide which sources to inspect. A read retrieves their original text and adds it to the ledger. Previews guide navigation, while read passages supply evidence.

Independent evidence ledger

The ledger preserves source text across working state revisions. When a read returns passages ΔE_{t+1}, the harness updates it as

$$E_{t+1} = \mathrm{Merge}_{\mathrm{src}}\big(E_t, \Delta E_{t+1}\big). \tag{4}$$

The merge combines overlapping passages from the same source while preserving their text and attribution. Search results and note revisions leave this record unchanged. The ledger records what the agent has read, while the note records its current understanding of that material. Earlier passages remain available for answering even when later findings lead the agent to revise or remove their summary from the note.

3.3 The Acquisition Loop

Acquisition begins with an empty ledger and a working note that identifies the information needed to answer the question. ❶ **Choose an action.** The policy uses the working state to decide whether to search, read, or finish. ❷ **Observe the result.** A search returns candidate previews. A read returns source text, which the harness also adds to the ledger. ❸ **Revise the state.** The agent updates established facts and remaining gaps in its note. The harness combines this note with the latest observation and source references to construct the next working state. The loop returns to ❶, allowing each observation to redirect acquisition. Searches and reads follow the current information need, so the agent can inspect several candidates before searching again.

❹ **Prepare the answer input.** When the policy judges the evidence adequate or further acquisition unhelpful within the remaining budget, it finishes the loop. The harness prepares $H(q, E_T)$ from the accumulated ledger for the downstream answer model. This input draws on sources collected throughout the trajectory, including passages no longer summarized in the final note. The skill and prompt are provided in Appendices A.1 and A.2.

4 EXPERIMENTS

We assess Picorer's effectiveness across three benchmarks, compare acquired evidence to explain differences between memory systems, and use ablations to identify how its components support acquisition.

4.1 SETUP

MemoryAgentBench (Hu et al., 2026b) evaluates accurate retrieval, test-time learning, long-range understanding, and selective forgetting. BEAM (Tavakoli et al., 2026) evaluates memory over long archives, and LoCoMo (Maharana et al., 2024) tests conversational memory. We follow each benchmark's scoring rules and use published or reproduced baseline configurations (MemTensor, 2026). Picorer uses the same backbone for acquisition and answering, with at most eight search calls per question.

The direct system comparison covers all 100 questions in each of three MemoryAgentBench tasks. *Multi-hop fact consolidation* requires linking facts across a 262K-token archive and favoring updates with higher serial numbers (Fact-MH in figures and tables). *Document question answering* uses RULER QA2 with a 421K-token archive. *Banking77* classifies queries into 77 intents using about 5,900 labeled examples. All systems use Qwen3.6-27B with shared embeddings and generation settings while retaining their native retrieval procedures. Implementation and scoring details appear in Appendix A.4.

4.2 OVERALL PERFORMANCE

Picorer performs strongly across the benchmarks, especially when an answer depends on locating specific information. On MemoryAgentBench, it achieves the highest Overall score among the listed GPT-4o-mini methods and improves on long-context inference with GPT-5-mini (Table 1). Although GPT-5-mini long-context inference remains stronger on long-range understanding, Picorer makes its largest gains in accurate retrieval and selective forgetting. Both tasks reward finding specific facts and selecting the applicable version within a large archive, which is what targeted acquisition provides.

Table 1: **MemoryAgentBench results.** Representative baselines are grouped by backbone. AR, TTL, LRU, and SF are accurate retrieval, test-time learning, long-range understanding, and selective forgetting. Overall is their unweighted mean. Bold marks the best score within the GPT-4o-mini and GPT-5-mini groups. Complete results appear in Table 12.

Backbone	Method	AR	TTL	LRU	SF	Overall
GPT-4o-mini	Long context	49.20%	**48.60%**	**46.20%**	25.00%	42.20%
	BM25	60.50%	44.50%	35.60%	25.50%	41.50%
	HippoRAG-v2	65.10%	35.80%	36.20%	29.50%	41.60%
	MemGPT	34.30%	40.80%	22.40%	15.50%	28.30%
	Picorer	**73.35%**	47.36%	44.97%	**31.00%**	**49.17%**
GPT-5-mini	Long context	74.40%	48.60%	**66.20%**	53.00%	60.60%
	Picorer	**85.27%**	**50.38%**	52.32%	**56.50%**	**61.12%**
GPT-4.1-mini	MIRIX	63.00%	35.70%	40.50%	11.50%	37.70%
Qwen3.6-27B	**Picorer**	85.18%	52.20%	59.03%	65.00%	65.35%

The benefits extend to long archives and conversational memory. With GPT-4.1-mini, Picorer leads the reproduced BEAM baselines, Viking, Hindsight, and MemOS, at both archive scales and remains competitive on LoCoMo (Table 2). These results show that the same acquisition procedure supports both factual and conversational memory.

Table 2: **BEAM scores and LoCoMo accuracy.** Reference values are OmniMemEval reproductions (MemTensor, 2026); bold marks the best GPT-4.1-mini result. Full breakdowns are in Tables 14 and 15.

Backbone	Method	BEAM-100K	BEAM-10M	LoCoMo
GPT-4.1-mini	Viking	70.76%	58.14%	69.33%
	Hindsight	70.22%	59.75%	81.99%
	MemOS	66.87%	56.75%	**88.83%**
	Picorer	**72.52%**	**60.49%**	86.95%
GPT-5-mini	Picorer	76.45%	67.06%	90.00%
Qwen3.6-27B	Picorer	76.67%	66.88%	88.57%

4.3 Comparison with Other Memory Harnesses

We compare Picorer with three memory harnesses to locate the source of its gains. A-MEM retrieves linked notes (Xu et al., 2025), MemoryOS searches a hierarchy of recent and long-term memories (Kang et al., 2025), and SimpleMem adaptively retrieves consolidated statements (Liu et al., 2026). Picorer instead revises its working state as it inspects sources and retains the read passages for answering. It lies on the Pareto frontier of accuracy and token cost across all three tasks, with its largest accuracy advantage on multi-hop fact consolidation (Figure 2a).

We examine 64 of the 100 multi-hop questions whose annotated support chains have no conflict with the archive. Picorer places the complete chain in the answer input more often than any baseline (Figure 2b). On the 17 questions where Picorer has the full chain and SimpleMem does not, Picorer answers 15 correctly versus 1 for SimpleMem. Its successive reads assemble the relations needed for the answer, not just related records (Appendix A.4).

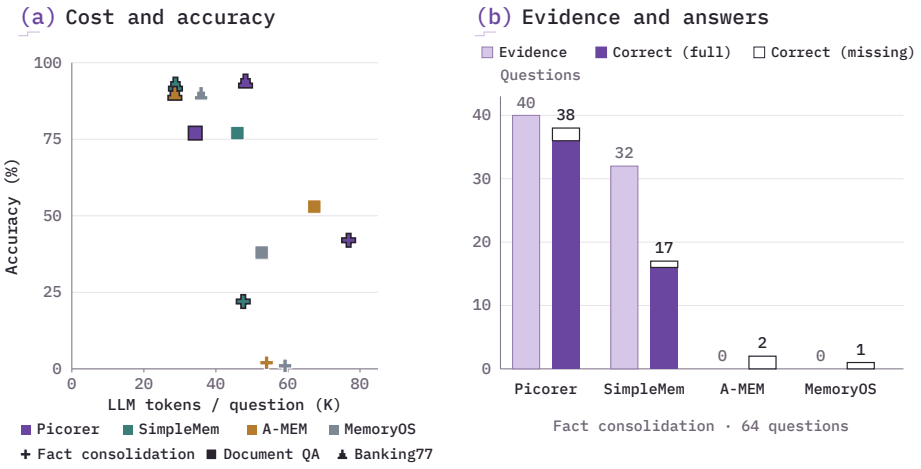

Figure 2: **Answer quality and acquired evidence.** (a) Accuracy against model input-plus-output tokens per question, including construction cost averaged over 100 queries. Colors identify systems, shapes identify tasks, and outlines mark each task's Pareto frontier. (b) Among the 100 multi-hop fact-consolidation questions, 64 have annotated chains with no conflict in the archive. The left bar counts complete chains in the answer input. The right bar counts correct answers, with dark and white segments indicating presence and absence of the complete chain. Coverage matches normalized reference statements and verified equivalent phrasings.

4.4 System Ablations

Picorer needs to preserve evidence across reads. In a separate retention study, giving the answer model only the latest read cuts multi-hop accuracy from 48% to 24%, while single-hop accuracy remains at 92% (Appendix C). The difference appears when an answer must combine facts found at different steps.

The working state's note records what the agent has established and what remains missing. We remove only this note, keeping the current observations, source references, and remaining budget in the state. Accuracy barely changes, but the share of repeated source blocks in subsequent reads rises

from 2.4% to 33.5%, and complete supporting chains take more tool steps to assemble (Table 3 and Figure 3). The free-form-note variant also repeats sources more often. When a candidate preview reveals a missing fact, the agent is more likely to read that candidate (Appendix D.4.1).

Combining retrieval results within a search also matters. Without this composition, document question-answering accuracy falls from 77% to 71% at nearly the same token cost (Table 3).

Table 3: **Working-state and memory-access ablations.** All variants use the same frozen memory and 100 questions per task, with up to eight search calls. Acc. is accuracy (%). Queries is the mean number of executed retrieval queries (a call may contain several). Repeats is the percentage of blocks shown after each question's first successful read that revisit a source without adding support on the 64-question multi-hop subset. Tokens is mean acquisition and answer tokens, in thousands.

	Fact-MH				RULER QA2		
Picorer variant	**Acc.↑**	**Queries**	**Repeats**	**Tokens**	**Acc.↑**	**Queries**	**Tokens**
Picorer	**42**	10.39	2.4	76.8	**77**	5.28	34.3
Free-form note	40	11.12	7.5	75.0	74	5.40	35.2
w/o note	41	11.69	33.5	77.4	74	4.08	30.1
w/o search composition	39	10.11	4.8	70.1	71	5.53	34.5

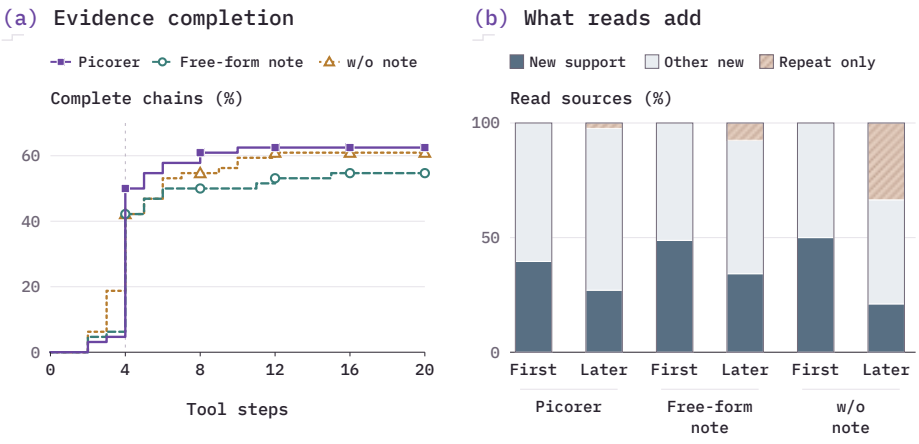

Figure 3: **Evidence accumulation and repeated access.** On the same 64 fact-consolidation questions, (a) shows the fraction with complete reference support by each tool step. In (b), First denotes each question's first successful read, and Later denotes all subsequent successful reads. Source blocks shown in each phase are classified as new reference support, other unseen sources, or repeats without new support. Counts and definitions are in Appendix B.2.

Keeping the original passages also separates acquisition from answer formatting. On multi-hop fact consolidation, with acquired sources fixed, question-directed summaries outperform whole-record selection under a 4K input cap, while original records are stronger at 8K (Table 18). Compression helps when space is scarce, but the ledger preserves the option to use source text.

5 Further Analysis

Section 4 links Picorer's gains to acquiring and retaining supporting relations. We now test why retaining intermediate evidence matters: does a new finding help immediately, or only when another relation completes its support? We also examine whether the model's sufficiency judgment recognizes this distinction.

We use the 262K-token multi-hop fact-consolidation task and Qwen3.6-27B. For question q, let E be the read evidence, G_q the annotated supporting relations, and $G_q(E)$ those present in E. We distinguish *reference coverage* R, *judged sufficiency* S, and *answer accuracy* Y:

$$R(E) = \frac{|G_q(E)|}{|G_q|}, \qquad S(E) = \sigma\big(m(E)\big), \qquad Y(E) = \frac{1}{3}\sum_{k=1}^{3} \mathbf{1}\big[a_k(q,E) = a_q^\star\big]. \tag{5}$$

An independent observer sees only (q, E) and judges whether it supports a correct answer. The margin m averages the logit difference between *sufficient* and *insufficient* under two prompts with reversed 0/1 label meanings, and σ is the sigmoid function. Each a_k answers from (q, E) with a fixed seed and is scored by normalized exact match to the reference answer $a_q^\star$. Paired conditions share an answer budget. Results average observations within each question before averaging questions.

5.1 Evidence Completion

We first test whether supporting relations are useful together. On 40 questions, we take the first read snapshot containing the full reference chain with no conflicting newer fact in that snapshot. We remove the first relation, final relation, both, or unrelated text matching the combined deletion in length. All visible versions of a removed relation are deleted. Removing either relation sharply lowers accuracy, while removing unrelated text preserves it (Table 4a). On 27 questions, all three answers are correct with full evidence and wrong under either single deletion. These relations contribute through their combination, explaining why retaining earlier reads matters in Section 4.4.

We then test when a new read makes retained evidence useful. For 103 reads from 71 questions, we compare the actual read with an unrelated addition matched within 10% in token count. A further condition adds all missing reference facts, with unrelated padding, to the same prior ledger. Table 4b separates unchanged coverage, partial progress, and chain completion. Partial progress raises S most but barely improves accuracy. Completing the chain produces the largest answer gain, while supplying missing facts restores accuracy even in the partial group. A read can therefore make progress without yet enabling an answer. Its value depends on the complementary evidence acquired next.

Table 4: **Evidence completion and judged sufficiency.** (a) Remove relations or unrelated text from complete evidence on 40 questions. (b) Compare actual reads with matched unrelated additions (103 reads, 71 questions). ΔS and ΔY are actual minus unrelated. Missing-fact Y measures accuracy after adding all missing reference facts to the prior ledger. Questions receive equal weight within each group and may appear in more than one group. Accuracy is in %.

(a) Removing supporting relations

Removed	None	First relation	Final relation	Both	Unrelated text
Sufficiency S	0.884	0.328	0.724	0.214	0.892
Accuracy Y	90.0	10.0	1.7	0.0	91.7

(b) Adding supporting relations

Change in support	Questions	ΔS	ΔY (pp)	Actual Y	Missing-fact Y
Unchanged	19	+0.191	+5.3	10.5	85.4
Increased, incomplete	45	+0.376	+3.0	3.0	94.1
Completed chain	25	+0.220	+72.0	74.7	94.7

5.2 Sufficiency Judgments

Judged sufficiency can rise before the evidence supports an answer. In Table 4a, both single deletions leave the same reference coverage, yet S remains much higher when the final relation is missing, despite lower accuracy. We test this gap on 217 natural snapshots from 98 questions by fitting logistic predictors of Y from m, R, or both. Five-fold validation holds out whole questions. Reference coverage predicts accuracy more reliably than the observer margin (Table 5a).

A high judgment also leaves substantial value in further reading. For reads whose prior score meets the thresholds in Table 5b, actual sources still improve accuracy over matched unrelated additions. Judged sufficiency can still precede decisive support. This clarifies the role of Picorer's two records: the working state's note tracks missing relations to guide search, while the ledger preserves the evidence those relations will complete. Full protocols and trajectory results appear in Appendix D.

Table 5: **Judged sufficiency and useful further reading.** (a) Prediction of answer accuracy on held-out questions. (b) Accuracy gains from actual reads over matched unrelated additions, grouped by the score before reading. Q counts questions.

(a) Held-out prediction

Predictor	Log loss ↓
Observer margin m	0.446
Reference coverage R	0.255
Coverage and margin (R, m)	0.242

(b) Value of another read

Prior score	Q	Reads	Gain (pp)
All	71	103	+22.5
$S \geq 0.8$	20	23	+41.7
$S \geq 0.9$	9	10	+22.2

6 Limitations

Picorer uses a textual working note and a prompted acquisition policy, so its behavior depends on the backbone that writes that note. We evaluate three benchmarks with four backbones, one of which is shared with the controlled system comparison. The gains we report are specific to these settings and are not a general claim about memory systems. The acquisition loop is capped at eight search calls per query, so the results characterize a bounded policy rather than the value of unrestricted search. Construction and query costs differ across tasks, and Picorer trades additional reads for accuracy rather than optimizing that trade-off end to end. Our interventions measure one effect on 40 questions from a single task of one benchmark, and they isolate how completing a supporting relation changes answers. The sufficiency predictors are fit on held-out questions from the same distribution. These measurements establish that the judgment can precede the support it indicates, but they do not establish how often that gap appears in deployment, where retrieval quality, corpus freshness, and query mix all differ from the benchmarks we study.

7 Conclusion

In agentic memory, what an agent needs to retrieve can change as it reads. A relevant record may expose a missing relation rather than resolve the question. Picorer makes this process explicit. A revisable working state records what is established and what remains missing, and it directs the next search. An independent ledger preserves the original passages already read, so revising that state cannot discard evidence a later answer may need.

The experiments connect this design to both performance and behavior. Picorer's advantage on multi-hop fact consolidation is accompanied by more complete supporting relations in the answer input. Ablations separate the two components: retaining earlier reads is what lets successive relations be combined, while the working state's note reduces repeated inspection. Controlled interventions explain why the combination matters. Removing either the first or the final relation sharply reduces accuracy, yet judged sufficiency remains high when the final relation is missing. Reads that add only partial support raise that judgment most but improve accuracy little. Completing the chain produces the largest gain. The judgment can arrive before the support it seems to indicate.

These results place the value of a memory in the evidence it joins and the reads it enables. A record that is unhelpful alone can become decisive once another read completes its support. Acquisition is better evaluated by how findings complement one another than by how each one scores in isolation. This points to what memory policies need to judge as they read: complementarity with the evidence already retained, rather than the apparent confidence of a single step.

An open question is whether these gains require explicitly verbalizing missing information or can arise from a learned internal state. Drawing on latent reasoning and memory (Hao et al., 2025; Zhang et al., 2026), future work could learn a latent working state that integrates new observations and guides retrieval, while the source ledger stays explicit. The separation between search control and evidence retention would then survive without a written note. Building on learned memory management (Yu et al., 2026), state updates and acquisition actions could be optimized jointly for eventual answer quality. Our interventions motivate a different reward as well. Reads that supply only partial support rarely help on their own, so rewarding them for what they enable together with later evidence could support stopping decisions that weigh the expected benefit of further acquisition against its cost.

Reproducibility Statement

The appendices specify the evaluation protocols, component controls, and acquisition studies. We will release the code, prompts, configurations, per-question results, and experiment data, together with the scripts that generate the figures and tables.

Ethics Statement

Our experiments use existing memory benchmarks. In practical deployments, persistent histories may contain personal or outdated information. Picorer preserves source references so acquired evidence can be inspected. Data access, retention, correction, and deletion should follow the requirements of the application and the people whose records are stored.

AI Use Statement

Generative AI assisted with research framing and experimental design, implementation of experiment and analysis code, data organization, experiment auditing, interpretation of results, drafting and editing, and preparation of figures. The authors remain responsible for the correctness, originality, and attribution of all submitted content. The visualization workflow followed procedural guidance from Scientific Agent Skills (Kassis et al., 2026).

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. In *ECAI 2025*. IOS Press, 2025. doi: 10.3233/FAIA251160. URL https://doi.org/10.3233/FAIA251160.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 758–759. ACM, 2009. doi: 10.1145/1571941.1572114.

Jesper Derehag, Carlos Calva, and Timmy Ghiurau. SmartSearch: How ranking beats structure for conversational memory retrieval. *arXiv preprint arXiv:2603.15599*, 2026. URL https://arxiv.org/abs/2603.15599.

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 *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=dyJ0GWpjJB.

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, pp. 59532–59569, 2024. doi: 10.52202/079017-1902.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *Conference on Language Modeling*, 2025. URL https://arxiv.org/abs/2412.06769.

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. Benchmarking agent memory in interdependent multi-session agentic tasks. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=JHYmxqS9Jv.

Chuanrui Hu, Xingze Gao, Zuyi Zhou, Dannong Xu, Yi Bai, Xintong Li, Hui Zhang, Tong Li, Chong Zhang, Lidong Bing, and Yafeng Deng. EverMemOS: A self-organizing memory operating system for structured long-horizon reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 45836–45853. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.acl-long.2125. URL https://aclanthology.org/2026.acl-long.2125/.

Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. HiAgent: Hierarchical working memory management for solving long-horizon agent tasks with large language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32779–32798. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1575. URL https://aclanthology.org/2025.acl-long.1575/.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in LLM agents via incremental multi-turn interactions. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=DT7JyQC3MR.

Dongming Jiang, Yi Li, Songtao Wei, Jinxin Yang, Ayushi Kishore, Alysa Zhao, Dingyi Kang, Xu Hu, Feng Chen, Qiannan Li, and Bingzhe Li. Anatomy of agentic memory: Taxonomy and empirical analysis of evaluation and system limitations, 2026. URL https://arxiv.org/abs/2602.19320.

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.

Zhengda Jin, Bingbing Wang, Jing Li, Ruifeng Xu, and Min Zhang. Mitigating provenance-role collapse in long-term agents via typed memory representation. *arXiv preprint arXiv:2605.25869*, 2026. URL https://arxiv.org/abs/2605.25869.

Hailey Joren, Jianyi Zhang, Chun-Sung Ferng, Da-Cheng Juan, Ankur Taly, and Cyrus Rashtchian. Sufficient context: A new lens on retrieval augmented generation systems. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Jjr2Odj8DJ.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134, 1998. doi: 10.1016/S0004-3702(98)00023-X.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25961–25970. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1318.

Timothy Kassis, Vinayak Agarwal, Yuhuan He, Darshil Patel, and Aubrey M. Brueckner. Scientific agent skills: A library of procedural knowledge for research agents. *arXiv preprint arXiv:2609.00065*, 2026. doi: 10.48550/arXiv.2609.00065. URL https://arxiv.org/abs/2609.00065.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. A human-inspired reading agent with gist memory of very long contexts. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 26396–26415. PMLR, 2024. URL https://proceedings.mlr.press/v235/lee24c.html.

Minghan Li, Junjie Zou, Xinxuan Lv, Chao Zhang, and Guodong Zhou. S2G-RAG: Structured sufficiency and gap judging for iterative retrieval-augmented QA. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25846–25862. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.acl-long.1185. URL https://aclanthology.org/2026.acl-long.1185/.

Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. Towards deep conversational recommendations. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/800de15c79c8d840f4e78d3af937d4d4-Abstract.html.

Xingyu Li, Rongguang Wang, Yuying Wang, Mengqing Guo, Chenyang Li, Tao Sheng, Sujith Ravi, and Dan Roth. PAR^2-RAG: Planned active retrieval and reasoning for multi-hop question answering. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 1727–1737. Association for Computational Linguistics, 2026b. doi: 10.18653/v1/2026.acl-industry.118. URL https://aclanthology.org/2026.acl-industry.118/.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. SimpleMem: Efficient lifelong memory for LLM agents. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=oBgLvd5YC6.

Wenquan Ma, Jiayan Nan, and WenLong Wu. What deserves memory: Adaptive memory distillation for LLM agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 34789–34812. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1607.

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. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.747.

MemTensor. OmniMemEval: Evaluation framework for benchmarking memory systems. https://github.com/MemTensor/OmniMemEval, 2026. GitHub repository.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. ReasoningBank: Scaling agent self-evolving with reasoning memory. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/980ea04d23d1f6908964eba2a74afe45-Abstract-Conference.html.

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

Joon Sung Park, Joseph C. O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22, 2023. doi: 10.1145/3586183.3606763.

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.

Rana Salama, Jason Cai, Michelle Yuan, Anna Currey, Monica Sunkara, Yi Zhang, and Yassine Benajiba. MemInsight: Autonomous memory augmentation for LLM agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 33136–33152. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1683.

Yuling Shi, Maolin Sun, Zijun Liu, Mo Yang, Yixiong Fang, Tianran Sun, and Xiaodong Gu. Reasoning in trees: Improving retrieval-augmented generation for multi-hop question answering. In *Companion Proceedings of the ACM Web Conference 2026*, pp. 626–635. Association for Computing Machinery, 2026. doi: 10.1145/3774905.3794658. URL https://doi.org/10.1145/3774905.3794658.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Jayakumar Subramanian, Amit Sinha, Raihan Seraj, and Aditya Mahajan. Approximate information state for approximate planning and reinforcement learning in partially observed systems. *Journal of Machine Learning Research*, 23(12):1–83, 2022. URL https://jmlr.org/papers/v23/20-1165.html.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, Anand Rajan Iyer, Tianlong Chen, Huan Liu, Chen-Yu Lee, and Tomas Pfister. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8416–8439. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.413.

Mohammad Tavakoli, Alireza Salemi, Carrie Ye, Mohamed Abdalla, Hamed Zamani, and J. Mitchell. Beyond a million tokens: Benchmarking and enhancing long-term memory in LLMs. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/d7f0cfa0fe759b033d5262e1bb7d4065-Abstract-Conference.html.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10014–10037. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.557. URL https://aclanthology.org/2023.acl-long.557/.

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

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63897–63911. PMLR, 2025. URL https://proceedings.mlr.press/v267/wang25bx.html.

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://openreview.net/forum?id=pZiyCaVuti.

Zhaofen Wu, Hanrong Zhang, Fulin Lin, Wujiang Xu, Xinran Xu, Yankai Chen, Henry Peng Zou, Shaowen Chen, Weizhi Zhang, Xue Liu, Philip S. Yu, and Hongwei Wang. GAM: Hierarchical graph-based agentic memory for LLM agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 34647–34664. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1600. URL https://aclanthology.org/2026.acl-long.1600/.

Zidi Xiong, Yuping Lin, Wenya Xie, Pengfei He, Zirui Liu, Jiliang Tang, Himabindu Lakkaraju, and Zhen Xiang. How memory management impacts LLM agents: An empirical study of experience-following behavior. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 623–645. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.27.

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*, volume 38, pp. 20004–20031, 2025. doi: 10.52202/085713-0593.

B. Y. Yan, Chaofan Li, Hongjin Qian, Shuqi Lu, and Zheng Liu. General agentic memory via deep research, 2025. URL https://arxiv.org/abs/2511.18423.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schuetze, Volker Tresp, and Yunpu Ma. Memory-R1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12805–12825. Association for Computational Linguistics,

2026. doi: 10.18653/v1/2026.acl-long.583. URL https://aclanthology.org/2026.acl-long.583/.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Yi Yu, Liuyi Yao, Yuexiang Xie, Qingquan Tan, Jiaqi Feng, Yaliang Li, and Libing Wu. Agentic memory: Learning unified long-term and short-term memory management for large language model agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21457–21483. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.981.

Guibin Zhang, Muxin Fu, and Shuicheng Yan. MemGen: Weaving generative latent memory for self-evolving agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/26407588dfb08e48c459c074ab6adb7d-Abstract-Conference.html.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. ∞Bench: Extending long context evaluation beyond 100k tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15262–15277, 2024. doi: 10.18653/v1/2024.acl-long.814. URL https://aclanthology.org/2024.acl-long.814/.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 19632–19642, 2024. doi: 10.1609/AAAI.V38I17.29936.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 19724–19731, 2024. doi: 10.1609/aaai.v38i17.29946.

Zexuan Zhong, Zhengxuan Wu, Christopher Manning, Christopher Potts, and Danqi Chen. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 15686–15702. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.971. URL https://aclanthology.org/2023.emnlp-main.971/.

Appendix Contents

A Evaluation Setup and Reproductions 15
A.1 Retrieval Skill 15
A.2 Acquisition Prompt 15
A.3 Common Evaluation Protocol 16
A.4 Matched Three-Task Reproduction 16
A.5 Complete Benchmark Results 19
B Acquisition Controls and Evidence Delivery 23
B.1 Working-State and Memory-Access Ablations 23
B.2 Evidence Acquisition Over the Trajectory 24
B.3 Fixed-Source Delivery Under Restricted Input Caps 24
C Evidence-Ledger Retention 25
C.1 Cumulative Versus Latest-Read Retention 25
D Further Analysis: Protocols and Supplementary Results 25
D.1 Read-Ledger Interventions 26
D.2 Natural Trajectories and Held-Out Diagnostics 27
D.3 Native Policy Measurements 28
D.4 Preview and Constructed-Evidence Interventions 29

A Evaluation Setup and Reproductions

A.1 Retrieval Skill

The acquisition agent receives the following skill instructions. The harness manages source retention and final-answer input assembly.

RETRIEVAL SKILL

Locate source memories that may help the downstream model answer the caller. Do not answer the caller yourself.

- Use the available search operators to discover candidates, and let observations determine whether to change the query, operator, or direction.
- Search spans all source roles in the current scope; start with `queries` only, adding `order` or `maxPerSession` when source sequence or conversation breadth matters. When semantic descriptions and exact phrases are both useful, issue one hybrid primary path plus a lexical `branch` and `union` rather than serializing those paths. Use `rrf` for fused ranking and `intersection` only when every branch must match.
- If another ranked page remains and a required aspect is still missing, use `search_more` before rewriting the query, and stop paging once direct evidence suffices.
- Read candidates whose sources may be useful downstream. Useful evidence may be direct, analogous, or distributed and need not repeat the caller's answer verbatim, so read only sources worth returning, since every read source enters the final package. While retrieval continues, keep only supported facts and remaining gaps in `workingMemory`.
- Finish when the source package is useful enough to hand off, when further search is unlikely to improve it, or when the budget is exhausted.
- Call `finish` by itself after observing the preceding results; source identity, citations, hashes, provenance, and final formatting are owned by the harness.

A.2 Acquisition Prompt

The fixed system prompt and required working-note instruction are typeset below. The runtime inserts an operator catalog from the available retrieval interface before the skill in Appendix A.1.

SYSTEM PROMPT

You are Picorer: a memory retrieval agent.

Your task is to locate source memories that may help a downstream model answer the caller's question. Do not answer or format the caller's final response.

Tool contract.

- `search` returns navigation candidates across all source roles available in the current scope. Queries alone use the default retriever, while optional branches, ordering, and session diversity form one inline retrieval program. Source roles are harness-owned metadata, not search arguments, and previews are not source evidence.
- `search_more` reveals another bounded page from the most recent search without changing its operator or queries.
- `read` takes candidate handles from the current search result and adds bounded exact source evidence to the final source package, so read useful candidates before moving to another search. If `workingMemory` is available, keep only the supported facts and remaining gaps in plain text; the harness owns source handles, read receipts, and context retirement.
- `finish` reports status, and `evidenceSummary` is an optional audit note that should normally be omitted. The harness automatically commits every exact source returned by `read` and generates citations, hashes, provenance, and answer-package formatting. Observe prior tool results, then call `finish` as the only tool call in that assistant turn.
- The harness owns source identity, provenance, package limits, and formatting.

Context policy: working-memory-rewrite. Every tool call, including the first search and `finish`, must submit a nonempty `workingMemory` string of at most 1,600 characters. It replaces the current note, so omission and null are invalid and an unchanged state is resubmitted. Write only the current state in plain text:

- Established: facts that are supported by source text you have read.
- Missing: facts or relationships still needed to answer the question.

Preserve every still-relevant fact and gap when replacing the note, and remove a fact when later evidence overturns it. Do not put candidate handles, evidence handles, source IDs, search history, or a reasoning transcript in the note.

Search results are navigation for the current decision, so read a promising candidate before moving to another search. The harness keeps source identity, exact read evidence, compact read receipts, audit history, and the final evidence package, and after your next successful tool action it removes prior raw tool results from active context, while a failed action keeps them visible for correction.

Use the Missing facts to choose the next search, and treat a relationship as established only after reading source text that states it rather than filling a missing relationship from world knowledge. Decide whether the acquired sources are adequate for answering, and if the remaining budget cannot resolve an important gap, finish insufficient. The final `workingMemory` note is required.

A.3 Common Evaluation Protocol

Each Picorer question permits at most 8 search calls, excluding reads and state updates. The same backbone acquires evidence and answers. Broad reference baselines retain their published configurations and budgets.

We use each benchmark's prompts and scoring rules. MemoryAgentBench includes ReDial Recall@5 (Li et al., 2018), and its InfinityBench summarization task uses the benchmark's LLM-judged F1 (Zhang et al., 2024). LongMemEval (S*) reformulates LongMemEval as 5 shared histories with 300 questions (Hu et al., 2026b), using a judge that depends on the question type (Wu et al., 2025). BEAM reports a normalized task score as a percentage with partial credit for rubric items. Event ordering combines normalized Kendall's τ_b with event coverage and halves the score when fewer than 2 events are identified. LoCoMo reports answer accuracy.

A.4 Matched Three-Task Reproduction

Table 6 uses all 100 questions from each task in Section 4.1: multi-hop fact consolidation over 262K tokens, document question answering over 421K tokens, and Banking77 intent classification over about 5,900 labeled examples. Question identities and order are unchanged. All systems use Qwen3.6-27B, text-embedding-v4 with 1,024 dimensions, and 1,024-token source chunks. Ingestion uses thinking off and temperature 0. Query generation uses thinking on, temperature 1, top-p 0.95, and top-k 20. The output allowance is the remaining model context after the actual input, less a 256-token reserve.

Each system retains its native retrieval procedure. Picorer permits 8 search calls with separate reads. SimpleMem permits 8 semantic, keyword, or structured retrieval attempts. A-MEM retrieves 10 root records and follows links before packing the answer context. MemoryOS searches its recent and long-term memories. Retrieval work and token use are measured outcomes.

Table 6: **Same-backbone memory-system comparison.** Accuracy (%, higher is better) on all 100 questions in each task. All systems use Qwen3.6-27B with shared embedding and generation settings. Each retains its native memory and retrieval procedures.

Method	Fact consolidation Multi-hop, 262K	Document QA 421K	Banking77 77 intents
A-MEM	2	53	90
MemoryOS	1	38	90
SimpleMem	22	**77**	93
Picorer	**42**	**77**	**94**

Fact consolidation and document question answering use normalized substring matching. Banking77 accepts only a number or *label:* followed by a number, after whitespace and case normalization. Context overflow and exhausted output budgets count as incorrect. Service and parser errors were repaired only for affected questions using the same frozen memory, with original attempts preserved. SimpleMem's document and Banking77 stores were rebuilt from recorded ingestion output to preserve source fields, without new ingestion generation. This comparison and the retention study in Appendix C are separate runs.

A.4.1 Paired Outcomes

Table 7 pairs Picorer and SimpleMem responses by question identity, including capacity failures as incorrect answers.

Table 7: **Paired Picorer–SimpleMem outcomes.** Picorer only and SimpleMem only count questions answered correctly by that method alone. Difference is in percentage points.

Task	Both correct	Picorer only	SimpleMem only	Both wrong	Difference
Fact consolidation	13	29	9	49	+20
Document QA	74	3	3	20	0
Banking77	90	4	3	3	+1

A.4.2 Construction and Query Cost

Table 8 separates construction, acquisition, and answering costs over 100 questions per task. Totals include capacity failures and exclude superseded attempts. Construction cost is amortized over 100 queries. Picorer's construction uses embeddings and source indices but no language-model generation.

Table 8: **Memory construction and query token use.** Qwen3.6-27B input-plus-output tokens for construction and 100 questions per task. Query usage includes acquisition and answering. Embedding tokens are separate. The last column adds construction/100 to mean query usage.

Task	Method	Build LM (M tokens)	Build embedding (K tokens)	Query LM (K tokens per question)	Build/100 + query (K LM tokens per question)
Fact consolidation	A-MEM	3.150	1109.694	22.567	54.070
	MemoryOS	2.943	333.258	29.767	59.196
	SimpleMem	2.224	241.584	25.354	47.589
	Picorer	0.000	337.859	76.842	76.842
Document QA	A-MEM	4.817	2223.994	19.143	67.313
	MemoryOS	2.902	468.094	23.668	52.690
	SimpleMem	2.852	455.199	17.448	45.968
	Picorer	0.000	466.274	34.252	34.252
Banking77	A-MEM	1.096	353.564	17.642	28.602
	MemoryOS	1.130	127.700	24.594	35.891
	SimpleMem	0.992	243.488	18.848	28.766
	Picorer	0.000	113.281	48.219	48.219

A.4.3 Retrieval Traces and Evidence Coverage

On fact consolidation, A-MEM expands 10 roots to a mean of 55.17 candidates and packs 10.97 records per answer. MemoryOS exhausts its output budget on 21 questions. Picorer averages 3.89 searches and 2.31 reads on this task, versus 1.75 and 1.48 on document question answering. Its native completion status is `sufficient` on 86 fact-consolidation questions, while 42 answers are correct.

For Figure 2b, a complete chain requires every annotated subject–relation–object statement in the answer input. Matching normalizes spelling, punctuation, and country-name articles and accepts verified rewrites that preserve the entities and relation. The rules are identical across systems and independent of correctness. We inspect Picorer's final read sources, SimpleMem's retrieved statements, A-MEM's packed context, and MemoryOS's answer-time memory messages. Search previews are excluded. Table 9 reports the 64-question subset with no support-chain conflict and the full 100-question sensitivity analysis. The data-based partition is defined below.

Table 9: **Complete supporting chains by system.** A chain counts as complete when every annotated hop appears in the memory shown to the answerer. The upper block covers the 64 questions with no support-chain conflict. The lower block uses all 100 questions and the original annotations. Correct with chain counts questions with both a complete chain and a correct answer.

System	Full chain	Correct	Correct with chain
64 questions with no conflict			
Picorer	40	38	36
SimpleMem	32	17	16
A-MEM	0	2	0
MemoryOS	0	1	0
All 100 questions			
Picorer	48	42	38
SimpleMem	38	22	20
A-MEM	0	2	0
MemoryOS	1	1	0

Observed source handoff. Table 10 checks each retained source against the assembled answer prompt by content hash. Source counts are summed across questions, not deduplicated across the archive.

Table 10: **Original Picorer handoff under the 32K context contract.** Each task has 100 questions. Exact and expanded count retained source items in the assembled prompt, including prompts blocked before dispatch. Blocked and dispatched count questions, not source items.

Task	Retained	Exact	Expanded	Missing	Blocked	Dispatched
Multi-hop	550	510	40	0	3	97
Document QA	232	232	0	0	0	100
Banking77	698	658	40	0	0	100

Input overflow blocks dispatch without trimming records. Of the 97 fact-consolidation requests dispatched, 1 exhausts its output allowance. Parent expansion occurs on 2 fact-consolidation and 2 Banking77 questions.

A.4.4 Fact-Consolidation Data Strata

The archive check in Appendix D.3 partitions the 100 questions into 64 with no template or serial conflict and 36 with a conflict. All 267 source chunks and 100 question identities match those in this reproduction. Table 11 retains the official answers and scoring rules for both groups.

Table 11: **Fact consolidation scores by a deterministic data diagnostic.** Correct answers out of the stated question count, using the official labels. The diagnostic tests template/serial chain conflicts, not general semantic validity.

Diagnostic group	n	Picorer	SimpleMem	A-MEM	MemoryOS
No conflict	64	38	17	2	1
Conflict	36	4	5	0	0
All questions	100	42	22	2	1

Exclusive Picorer/SimpleMem successes are 25/4 in the no-conflict group and 4/5 in the conflict group.

A.5 Complete Benchmark Results

A.5.1 MemoryAgentBench

Table 12 lists reference configurations and scores from Hu et al. (2026b). Table 13 gives Picorer's task-level results and the LongMemEval breakdown.

Table 12: **Complete MemoryAgentBench results.** Published baselines follow Hu et al. (2026b). Bold marks the highest score in each column across all listed configurations.

Method	Backbone / context	AR	TTL	LRU	SF	Overall
Long-context agents						
Long context	GPT-4o (128K)	58.10%	50.00%	54.90%	32.50%	48.80%
Long context	GPT-4o-mini (128K)	49.20%	48.60%	46.20%	25.00%	42.20%
Long context	Claude-3.7-Sonnet (200K)	59.70%	**53.90%**	62.20%	22.50%	49.60%
Long context	GPT-5-mini (400K)	74.40%	48.60%	**66.20%**	53.00%	60.60%
Long context	GPT-4.1-mini (1M)	71.80%	46.20%	49.10%	20.50%	46.90%
Long context	Gemini-2.0-Flash (1M)	65.10%	46.40%	41.60%	16.50%	42.40%
Simple and embedding-based RAG						
BM25	GPT-4o-mini	60.50%	44.50%	35.60%	25.50%	41.50%
Contriever	GPT-4o-mini	33.90%	42.90%	29.80%	12.50%	29.80%
Text-Embed-3-Small	GPT-4o-mini	53.80%	42.70%	36.30%	15.50%	37.10%
Text-Embed-3-Large	GPT-4o-mini	54.60%	44.30%	37.30%	16.00%	38.00%
Qwen3-Embedding-4B	GPT-4o-mini	54.70%	45.10%	37.00%	16.00%	38.20%
Structure-augmented RAG						
RAPTOR	GPT-4o-mini	36.80%	35.90%	27.90%	7.50%	27.00%
GraphRAG	GPT-4o-mini	40.90%	24.80%	19.90%	8.00%	23.40%
MemoRAG	GPT-4o-mini	34.50%	45.10%	30.00%	14.00%	30.90%
HippoRAG-v2	GPT-4o-mini	65.10%	35.80%	36.20%	29.50%	41.60%
Mem0	GPT-4o-mini	32.60%	21.20%	20.70%	10.00%	21.10%
Cognee	GPT-4o-mini	28.30%	22.80%	16.00%	15.50%	20.60%
Zep	GPT-4o-mini	37.50%	37.50%	16.20%	5.00%	24.00%
Agentic memory						
Self-RAG	GPT-4o-mini	33.60%	12.20%	18.10%	11.00%	18.70%
MemGPT	GPT-4o-mini	34.30%	40.80%	22.40%	15.50%	28.30%
MIRIX	GPT-4o-mini	47.50%	24.10%	25.40%	8.00%	26.20%
MIRIX	GPT-4.1-mini	63.00%	35.70%	40.50%	11.50%	37.70%
Picorer (ours)						
Picorer	GPT-4o-mini	73.35%	47.36%	44.97%	31.00%	49.17%
Picorer	GPT-5-mini	**85.27%**	50.38%	52.32%	56.50%	61.12%
Picorer	Qwen3.6-27B	85.18%	52.20%	59.03%	**65.00%**	**65.35%**

Table 13: **MemoryAgentBench diagnostics.** Task-level scores and accuracy by LongMemEval question type. n counts questions. ReDial uses Recall@5 and InfinityBench summarization uses LLM-judged F1. Bold marks the best result in each row.

Task-level results

Task	n	Qwen3.6-27B	GPT-4o-mini	GPT-5-mini
Banking77	100	**94.00%**	89.00%	92.00%
CLINC150	100	**98.00%**	88.00%	96.00%
NLU	100	**87.00%**	84.00%	**87.00%**
TREC Coarse	100	**90.00%**	72.00%	89.00%
TREC Fine	100	**78.00%**	61.00%	64.00%
Single-hop QA	100	**92.00%**	82.00%	**92.00%**
Multi-hop QA	100	76.00%	67.00%	**77.00%**
EventQA	500	**96.40%**	80.40%	91.40%
Detective-QA	71	**80.28%**	63.38%	76.06%
Single-hop fact consolidation	100	**90.00%**	52.00%	84.00%
Multi-hop fact consolidation	100	**40.00%**	10.00%	29.00%
LongMemEval (S*)	300	76.33%	64.00%	**80.67%**
ReDial R@5	200	15.00%	**15.92%**	15.17%
InfinityBench summarization	100	**37.79%**	26.56%	28.58%

LongMemEval (S*)

Question type	n	Qwen3.6-27B	GPT-4o-mini	GPT-5-mini
User fact	45	**93.33%**	88.89%	**93.33%**
Assistant fact	30	**96.67%**	93.33%	**96.67%**
Preference	30	20.00%	23.33%	**40.00%**
Temporal	75	81.33%	57.33%	**86.67%**
Multi-session	75	69.33%	48.00%	**76.00%**
Knowledge update	45	**86.67%**	84.44%	82.22%
Overall	300	76.33%	64.00%	**80.67%**

A.5.2 BEAM AND LOCOMO

Tables 14 and 15 report the BEAM capability breakdown and LoCoMo's 1,540 non-adversarial questions. BEAM's 100K- and 10M-token settings use distinct question sets. Baselines follow OmniMemEval (MemTensor, 2026).

Table 14: **BEAM scores by system and capability.** Baseline systems use GPT-4.1-mini. Picorer backbones are listed explicitly. Archives contain 100K or 10M tokens. Bold marks the best result in each column of the system comparison and within each archive scale for each capability.

Overall comparison

System	100K tokens	10M tokens
Mem0	70.41%	43.33%
Zep / Graphiti	68.70%	56.11%
Supermemory	65.98%	52.49%
Viking	70.76%	58.14%
Cognee	59.30%	56.02%
Letta	69.22%	52.30%
Hindsight	70.22%	59.75%
EverOS	58.64%	47.73%
MemMachine	64.80%	51.90%
mem9	65.75%	57.30%
MemOS	66.87%	56.75%
Picorer (GPT-4.1-mini)	72.52%	60.49%
Picorer (GPT-5-mini)	76.45%	**67.06%**
Picorer (Qwen3.6-27B)	**76.67%**	66.88%

Capability breakdown

	100K tokens			10M tokens		
Capability	Qwen3.6 27B	GPT-4.1 mini	GPT-5 mini	Qwen3.6 27B	GPT-4.1 mini	GPT-5 mini
Abstention	85.00%	**87.50%**	67.50%	80.00%	82.50%	**95.00%**
Contradiction	85.62%	69.38%	**86.25%**	64.38%	62.50%	**72.50%**
Event ordering	27.56%	29.01%	**45.76%**	27.10%	31.28%	**31.50%**
Info. extraction	**91.41%**	90.73%	89.74%	**80.00%**	75.00%	72.50%
Instruction	**89.38%**	85.00%	87.50%	**76.25%**	68.75%	75.00%
Knowledge update	**72.50%**	55.00%	68.75%	**78.75%**	67.50%	72.50%
Multi-session	**70.86%**	70.73%	70.10%	39.21%	23.58%	**45.83%**
Preference	88.12%	**93.12%**	92.50%	**78.75%**	68.33%	72.08%
Summarization	**80.04%**	75.40%	76.35%	**86.88%**	84.19%	86.17%
Temporal	76.25%	69.38%	**80.00%**	**57.50%**	41.25%	47.50%
Overall	**76.67%**	72.52%	76.45%	66.88%	60.49%	**67.06%**

Table 15: **LoCoMo accuracy by question category.** Reproduced baseline values are taken from OmniMemEval (MemTensor, 2026), and Picorer rows report our evaluations. Overall accuracy weights all questions equally: 841 single-hop, 282 multi-hop, 321 temporal, and 96 open-domain questions. Bold marks the best accuracy in each column.

System	Single-hop	Multi-hop	Temporal	Open-domain	Overall
Mem0	81.09%	76.12%	77.15%	54.17%	77.68%
Zep	65.36%	68.79%	55.56%	63.54%	63.83%
Supermemory	75.39%	77.07%	67.60%	66.67%	73.53%
Viking	78.04%	73.29%	48.81%	50.00%	69.33%
Cognee	87.99%	78.84%	81.83%	63.19%	83.48%
Letta	87.99%	76.24%	53.48%	63.54%	77.12%
Hindsight	88.98%	78.84%	73.52%	58.33%	81.99%
Memori	47.32%	44.09%	22.53%	43.75%	41.34%
EverOS	86.80%	77.78%	84.11%	57.29%	82.75%
MemMachine	83.47%	53.19%	71.96%	57.29%	73.90%
mem9	79.27%	62.88%	73.62%	55.90%	73.64%
MemoryLake	70.87%	75.30%	79.75%	54.17%	72.49%
Backboard.io	25.09%	22.34%	13.40%	29.17%	22.40%
MemOS	**92.51%**	88.65%	85.05%	69.79%	88.83%
Picorer (GPT-4.1-mini)	92.03%	79.79%	85.98%	66.67%	86.95%
Picorer (GPT-5-mini)	92.27%	**89.36%**	88.16%	**78.12%**	**90.00%**
Picorer (Qwen3.6-27B)	92.39%	87.94%	**88.47%**	57.29%	88.57%

B Acquisition Controls and Evidence Delivery

These controls use all 100 questions from each of the multi-hop fact-consolidation and document-QA tasks. They retain the matched comparison's frozen memory, model settings, evidence ledger, 8-search budget, source scope, and candidate presentation. Only the named component changes. Appendix C reports a separate retention study.

B.1 Working-State and Memory-Access Ablations

Free-form note replaces the Established/Missing instruction with a free-form summary under the same 1,600-character limit. *Without note* removes note generation and replay but retains current observations, sources, and the remaining budget. *Without search composition* disables branches and result fusion within a call. It retains all 5 base operators, multiple queries to one operator, and operator changes between calls. Table 3 reports accuracy and token use. Table 16 adds source access and elapsed time.

Table 16: **Source access and elapsed time in the acquisition controls.** S/R gives mean search/read calls. Sources gives mean distinct source IDs read per question. Repeats counts later emissions of an already-read ID across 100 questions, with affected questions in parentheses. Time is mean query wall time in seconds, including the shared model queue and transport.

Task	Picorer variant	S/R	Sources	Repeats	Time
Multi-hop	Picorer	3.89/2.31	5.50	7 (6)	266.2
	Free-form note	4.15/2.52	4.53	16 (10)	265.5
	w/o note	4.44/2.39	4.76	124 (42)	268.4
	w/o search composition	3.80/2.38	5.19	10 (6)	249.4
Document QA	Picorer	1.75/1.48	2.32	44 (7)	104.4
	Free-form note	1.89/1.59	2.38	53 (11)	94.3
	w/o note	1.58/1.49	2.41	58 (9)	79.6
	w/o search composition	1.85/1.53	2.44	60 (8)	105.7

B.2 Evidence Acquisition Over the Trajectory

Figure 3 uses the 64 questions with no support-chain conflict, selected independently of model answers. A chain is complete once every reference statement has appeared in a successful read. After a trajectory stops, its final evidence state is carried forward, keeping the curve denominator at 64. Table 17 also reports all 100 questions using the same matching rules.

Displayed source blocks are classified in order. *New support* adds a previously unseen reference statement. *Other new* introduces a source ID without adding a reference statement. *Repeat only* repeats a source ID without new support. Counts reset between questions. The first read is the first successful read action, and later reads are subsequent successful reads. Figure proportions pool source blocks within each condition and phase.

Table 17: **Chain completion and later-read allocation.** Chains gives counts complete by tool step 4 and at termination. Later sources gives new support / other new / repeat-only counts. The all-100 rows retain the 36 reference-conflict flags and report literal reference matching for sensitivity.

Questions	Picorer variant	Chains: 4 / final	Later sources
No conflict (64)	Picorer	32 / 40	46 / 120 / 4
	Free-form note	27 / 35	50 / 85 / 11
	w/o note	27 / 39	39 / 84 / 62
	w/o search composition	36 / 41	52 / 88 / 7
All 100	Picorer	36 / 48	66 / 253 / 7
	Free-form note	31 / 42	71 / 183 / 16
	w/o note	30 / 45	59 / 182 / 124
	w/o search composition	40 / 48	76 / 224 / 10

B.3 Fixed-Source Delivery Under Restricted Input Caps

This study fixes Picorer's exact retained excerpts and changes their selection or representation for answering. Caps of 4,096 and 8,192 tokens apply to the entire answer input. Without these caps, all excerpts fit the 32K context for 98 multi-hop and 100 document questions. This check excludes the parent expansion in Appendix A.4.3.

Cumulative selects the longest prefix of whole records in first-read order. *Recent* selects records in reverse read order, then restores their original order. Neither splits records. *Summary* generates question-directed text from the question and exact excerpts alone. It uses no reference answer, prior answer, working note, new source, or expanded parent. Oversized summaries are neither truncated nor regenerated. The 8K summary condition reuses the 4K summary and answer by design.

Each condition covers 100 questions, and identical requests share one response. The output allowance is $32{,}768 - \text{input tokens} - 256$. For multi-hop summarization, 2 inputs exceed capacity before dispatch and 1 output exhausts its allowance. All 3 count as incorrect, and consumed tokens remain in the totals in Table 18.

Table 18: **Answering with fixed sources under input limits.** Every row covers 100 questions. The cap limits final-answer input. Accuracy is in percent. Tokens are mean Qwen3.6-27B input-plus-output tokens in thousands for delivery and answering, excluding acquisition. The Sources column counts whole records selected for answering or given to summarization (†). The 8K summary rows reuse the 4K answers (∗).

Task / cap	Representation	Acc. ↑	Capacity failures	Sources	Tokens
Multi-hop / 4K	Cumulative	17	0	2.41	6.602
	Recent	21	0	2.48	6.744
	Summary	39	3	5.50†	11.730
Multi-hop / 8K	Cumulative	41	0	4.03	8.508
	Recent	40	0	4.06	8.363
	Summary*	39	3	5.50†	11.730
Document QA / 4K	Cumulative	78	0	2.12	3.170
	Recent	75	0	2.12	3.175
	Summary	74	0	2.32†	6.164
Document QA / 8K	Cumulative	78	0	2.32	3.369
	Recent	78	0	2.32	3.369
	Summary*	74	0	2.32†	6.164

C Evidence-Ledger Retention

C.1 Cumulative Versus Latest-Read Retention

Latest-read retention replaces the cumulative ledger with text from only the most recent read. All conditions use the same 100 multi-hop fact-consolidation questions, Qwen3.6-27B, prompts, archive, and 8-search budget. This is an independent run from the matched reproduction in Table 6. Table 19 also includes *stateless context*, which removes cross-turn acquisition context, and *hybrid-only search*, which permits only hybrid retrieval. The latter restricts both the base operators and their composition.

Table 19: **Evidence retention and component controls on multi-hop fact consolidation.** All conditions use Qwen3.6-27B and the same 100 questions. Accuracy and search means include all 100 questions, with failed questions scored incorrect. Bold marks the best accuracy.

Condition	Accuracy ↑	Failed questions	Mean search calls
Picorer	**48.00%**	0	3.44
Stateless context	45.00%	1	3.94
Latest-read retention	24.00%	2	3.35
Hybrid-only search	46.00%	0	3.31

Table 20 gives paired outcomes on the multi-hop task and its 100-question single-hop counterpart. The 2 empty completions under latest-read retention are also incorrect under cumulative retention.

Table 20: **Paired outcomes for cumulative and latest-read retention.** Each row contains 100 questions. The middle columns count answers correct under only the named retention rule.

Task	Both correct	Cumulative only	Latest-read only	Both wrong
Multi-hop	20	28	4	48
Single-hop	88	4	4	4

D Further Analysis: Protocols and Supplementary Results

All studies use multi-hop fact consolidation over 262K tokens and Qwen3.6-27B. Appendices D.1–D.2 document the independent observer in Section 5. Appendices D.3–D.4 cover native policy measurements and preview interventions.

D.1 READ-LEDGER INTERVENTIONS

Reference coverage R, judged sufficiency S, and answer accuracy Y follow Equation 5. Both the observer and answerer receive only the question and read source text. Previews, notes, tool history, and reference annotations are excluded. The observer uses thinking off, temperature 0, top-p 1, top-k -1, seed 20260924, and a 1-token output limit. Two prompts reverse the meanings of labels 0 and 1. Their mean logit margin determines S, a conditional preference between labels.

The answerer uses thinking on, temperature 1, top-p 0.95, top-k 20, and seeds 202609240, 202609241, and 202609242. Answers are scored by normalized exact match. Paired conditions share an output allowance of 32,768 tokens minus the longest answer input and a 256-token reserve.

Removing supporting relations. We take the first read snapshot containing the full reference chain and construct 5 conditions: full evidence, first relation removed, final relation removed, both removed, and unrelated text removed. Relation deletions remove all visible versions of the same cloze, including outdated values. Other text and source order remain fixed. The unrelated deletion matches the combined deletion in character count.

Of 47 eligible questions, 7 have newer facts in that snapshot that conflict with the chain, leaving 40 in Figure 4. Selection uses recorded facts, not model judgments or answers. Single deletions leave equal reference coverage. First-relation deletion also removes the starting entity in 39 pairs. Excluding 2 question–chain mismatches leaves 38 questions, with 92.1% accuracy under full evidence and 1.8% after final-relation deletion.

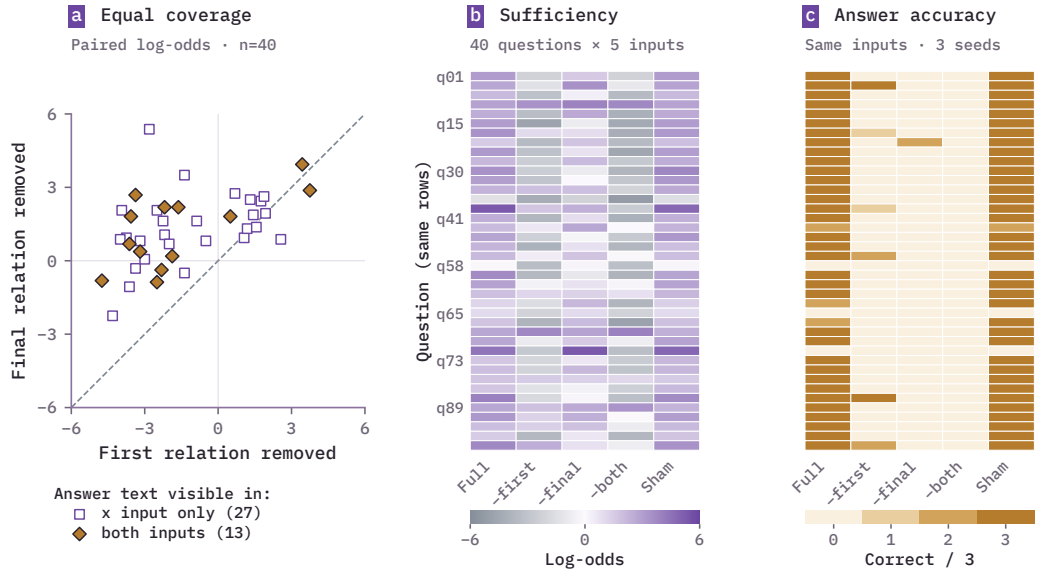

Figure 4: **Per-question effects of removing supporting relations.** Left: one point per question compares the observer margin after first- and final-relation deletion. Center and right: the same 40 questions in index order, under all 5 conditions, show the aligned observer margin and correct answers out of 3. The original input and unrelated deletion serve as controls.

Adding real reads and matched controls. We identify 223 reads that change the ledger across 99 questions. Each has 4 inputs: the prior ledger, the actual updated ledger, the prior ledger plus unrelated facts, and the prior ledger plus missing reference facts and unrelated padding. Additions use whole archive facts. Unrelated facts and padding exclude reference entities, answers, and relations. Before generation, 106 reads meet the token-matching tolerance of 10%. Complete measurements are available for 103 reads from 71 questions. The remaining 3 are recorded as unavailable, not incorrect.

Reads are grouped by the change in R. Partial progress increases R but leaves it below 1. Completion raises it to 1. Unchanged coverage includes both incomplete and already complete ledgers.

Within each group, we average reads per question and then weight questions equally. A question can enter multiple groups. Table 21 reports all conditions.

Table 21: **Answer accuracy under controlled additions to the same ledger.** Before uses the prior ledger. The other columns add unrelated facts, the actual read, or all missing reference facts with unrelated padding. Entries are accuracy (%). Questions have equal weight within each group and can contribute to more than one group.

Change in support	Questions	Reads	Before	Unrelated	Actual	Missing facts
Unchanged	19	26	5.3	5.3	10.5	85.4
Increased, incomplete	45	52	0.0	0.0	3.0	94.1
Completed chain	25	25	4.0	2.7	74.7	94.7

Reference consistency and conflicting facts. Restricting to answers consistent with the newest archive facts and excluding the 2 question–chain mismatches leaves 26 questions with partial reads and 18 with completing reads. Actual-read/missing-fact accuracy is 5.1%/98.7% and 96.3%/98.1%, respectively.

For 95 questions, a separate control inserts an alternative terminal fact into a complete chain. It crosses an older or newer serial number with placement before or after the chain. Each condition is scored against its own implied answer. Accuracy ranges from 97.5% to 98.2% across the 4 conditions.

D.2 Natural Trajectories and Held-Out Diagnostics

The 869 acquisition states contain 323 unique ordered evidence snapshots. Unchanged inputs share measurements. Figure 5 shows the 99 questions that acquire nonempty evidence, including their initial empty ledgers. Columns count evidence changes, not tool calls or elapsed time.

At the first positive observer margin, 67 of 82 questions have no correct answer across the 3 seeds. Of these 67, 26 later reach majority correctness. After excluding inconsistent reference labels and 2 question–chain mismatches, the corresponding counts are 52, 40, and 26. On all 37 questions that reach both full reference coverage and majority correctness, the first such snapshots coincide.

Prediction protocol. We fit logistic calibrators on 217 evaluable nonempty snapshots from 98 questions. 5-fold validation holds out whole questions. Features are standardized using the training fold. Slopes use an L_2 penalty of 1 and the intercept is unpenalized. Each question has total weight 1. The target is mean correctness across the 3 seeds. Table 22 averages held-out losses within each question and then across questions.

Table 22: **Predicting answer success on held-out questions.** Lower log loss and Brier score are better. All rows use the same 217 nonempty snapshots from 98 questions. The raw score S is uncalibrated. The other evidence-based rows use logistic calibration.

Predictor	Log loss	Brier score
Training answer-rate prior	0.529	0.172
Raw sufficiency score S	1.177	0.367
Observer margin m	0.446	0.148
Reference coverage R	0.255	0.068
Coverage and margin (R, m)	0.242	0.069

Reads after high observer scores. Table 23 applies thresholds to S before each read. Token matching, output budgets, and question weighting are unchanged.

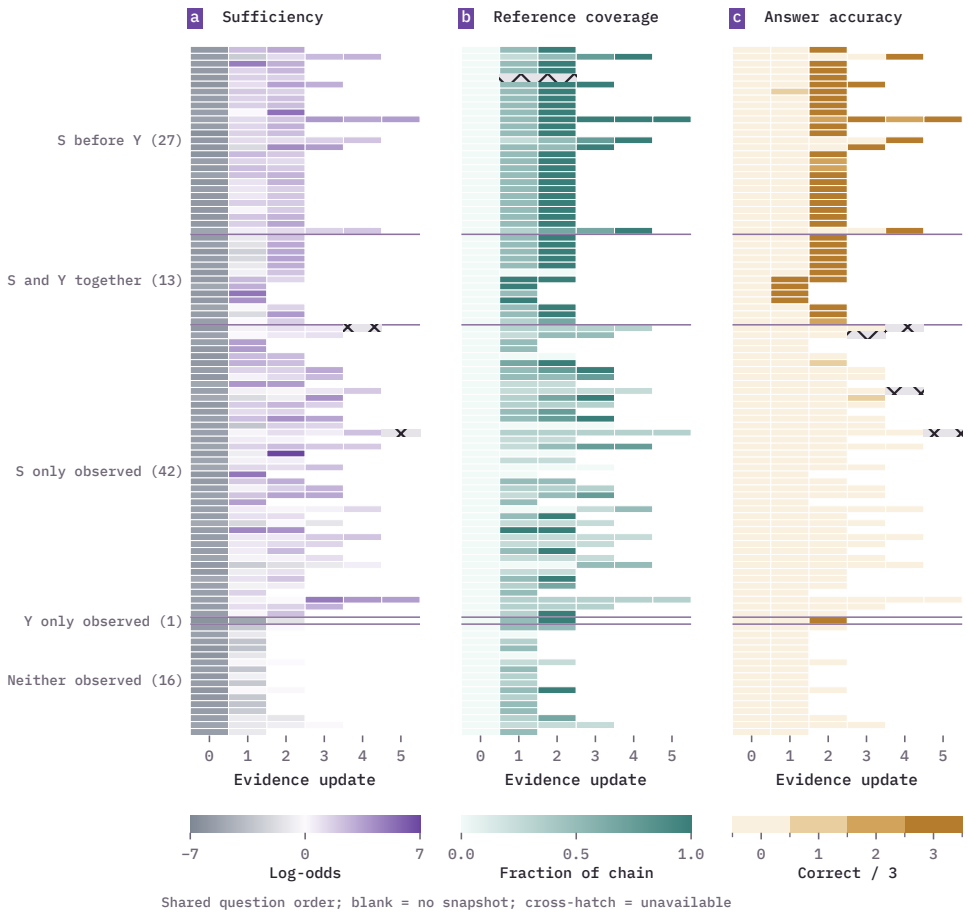

Figure 5: **Judgment, coverage, and answers along the same trajectories.** Each row is one question across all 3 panels. Rows are grouped by the first positive sufficiency margin and first majority-correct snapshot, then sorted by question index. Every observed snapshot appears without interpolation. Blank tails mark the end of the recorded trajectory. Cross-hatching marks unavailable measurements.

Table 23: **Answer gains after high observer scores.** Thresholds refer to S before the read. Gains compare actual reads with unrelated additions, giving each question equal weight. The last row excludes inconsistent reference labels and question–chain mismatches (Appendix D.1).

Prior judgment	Questions	Reads	Answer gain (pp)
All	71	103	+22.5
$S \geq 0.8$	20	23	+41.7
$S \geq 0.9$	9	10	+22.2
$S \geq 0.8$, consistent labels	12	13	+66.7

D.3 Native Policy Measurements

This dataset contains 2,150 states from 3 retrieval runs over 100 questions. The 64-question subset with no support-chain conflict contains 1,181 states and consists of 49 two-hop, 7 three-hop, and 8 four-hop questions. Table 26 also reports all 100 questions.

Reference mapping. We match question text to its source MQuAKE case (Zhong et al., 2023). Each annotated hop's cloze and answer is mapped to a numbered archive fact after normalizing case, whitespace, and a trailing period. The check flags 36 questions where a required fact is absent or

superseded by a later serial number for the same cloze. Separate inspection identified question–chain mismatches at indices 65 and 95 within the no-conflict subset. Benchmark scores retain all original questions and labels.

Native sufficiency score. We retain the captured policy input and tool schema, append the common prefix of a `finish` call with status `sufficient` or `insufficient`, and score the first divergent token. Averaging log-probability margins across 2 replicas gives

$$m_t^\pi = \frac{1}{2}\sum_{j=1}^{2}\big(\ell_{\mathrm{suf},j} - \ell_{\mathrm{ins},j}\big), \qquad S_t^\pi = \frac{1}{1+\exp(-m_t^\pi)}. \tag{6}$$

This score measures status preference conditional on a finish call. Reference coverage R is computed offline and never supplied to the policy.

Generation and aggregation. Answers use temperature 0.7, top-p 0.8, low reasoning effort, a 16,384-token output limit, and 5 fixed-seed samples per condition. Preview interventions use 4 next-action samples with a 2,048-token limit. Repeated observations are averaged within each question before questions are weighted equally. Empty answers trigger the archived same-evidence, no-thinking fallback. These are recovered measurements rather than first-attempt outcomes.

D.4 Preview and Constructed-Evidence Interventions

D.4.1 Candidate Previews

Only candidate previews change. The question, note, candidate references, read receipts, ledger, and budget remain fixed. Unrelated previews contain archive facts without the answer or its entities. Relevant previews replace the same amount of text with a missing supporting fact. Candidate references are unchanged, so inserted facts may differ from the referenced source.

Tables 24 and 25 cover 564 paired states from 64 questions, with 4 next-action samples per condition. Samples are averaged within states and then questions. The full population contains 1,072 pairs after excluding 6 truncated responses and their pairs.

Table 24: **Next actions under candidate-preview replacement.** Action rates averaged across 64 equally weighted questions. Other combines the remaining action classes.

Next action	Unrelated preview	Relevant preview
Read	3.37%	84.20%
Search	91.28%	14.01%
Finish	5.01%	1.40%
Other	0.33%	0.39%

Table 25: **Preview effects before and after evidence is retained.** Next-action rates, with questions weighted equally within each group. Empty evidence: 269 states, 64 questions. Nonempty evidence: 295 states, 42 questions.

Evidence	Action	Unrelated preview	Relevant preview
Empty	Read	4.33%	95.07%
	Search	95.57%	4.84%
Nonempty	Read	0.45%	65.55%
	Search	82.83%	28.19%

D.4.2 Evidence Content

Every subset of the reference chain is tested in forward and reverse order. Unrelated facts replace omitted facts, preserving source-slot count and approximately matching word count. Conditions share a constructed tool history and receive 5 answers scored by normalized substring matching.

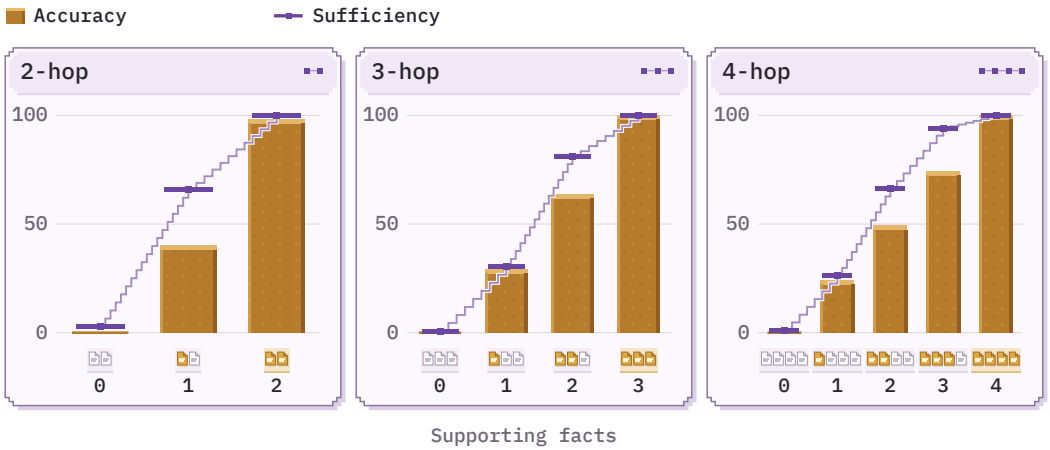

Figure 6: **Evidence content at fixed source-slot count.** Supporting facts replace unrelated facts in fixed-slot evidence sets. Values average the tested subsets and source orders at each count. Gold bars show answer accuracy and purple marks show the native sufficiency score S^{π} on a common 0–100% scale. Source tiles mark supporting-fact counts. The panels contain 49 two-hop, 7 three-hop, and 8 four-hop questions.

Table 26: **Answer accuracy with partial and complete constructed chains.** The all-question result includes the 36 questions with annotation conflicts.

Questions	One fact missing	Complete chain
No chain conflict ($n = 64$)	46.90%	**98.59%**
All questions ($n = 100$)	54.95%	**98.90%**

Subsets and orders are averaged within each question at each support count (Figure 6). The one-fact-missing condition in Table 26 averages over all omitted hops. Partial inputs can still contain the terminal fact and its answer.

The completion gain is 51.69 percentage points under substring matching and 51.61 under strict exact match. Using only available non-fallback answers gives 50.78 points over 298 of the 302 paired subset/order comparisons.

Native-score prediction protocol. Table 27 uses 1,181 states from 64 questions. Answers use retained text without parent expansion. Logistic models use 5-fold validation by question, an intercept, the listed interactions, equal question weights, and fixed L_2 penalty $\alpha = 10^{-5}$. For mean correctness y over 5 answers and prediction p, log loss is $-y \log p - (1 - y) \log(1 - p)$ and Brier score is $y(1 - p)^2 + (1 - y)p^2$. Losses are averaged within and then across questions. R uses annotations. The native score S^{π} uses the captured policy input.

Table 27: **Predicting answer correctness from retained evidence.** Five-fold validation grouped by question. Lower is better. Joint models include an interaction and use the native score S^{π} or scaled native margin $m^{\pi}/10$. R is annotation-derived and available only for offline analysis.

Predictors	Log loss ↓	Brier score ↓
R	0.135	0.033
S^{π}	0.423	0.127
R, S^{π}, RS^{π}	0.141	0.036
$R, m^{\pi}/10, Rm^{\pi}/10$	0.136	0.035