EMBER: Efficient Memory via Budgeted Evidence Retention for Long-Horizon Agents

Yilong Li and Suman Banerjee
University of Wisconsin–Madison
{yli758,suman}@wisc.edu

Tong Che
NVIDIA Research
tongc@nvidia.com

Abstract

Long-horizon agents cannot treat memory as repeated raw-log access. Raw histories may be archived, but using them for future requests still requires retrieval, rereading, and context tokens; when retrieval misses answer evidence, systems often pay again by expanding the search. We study *budgeted evidence survival*: under a fixed query-time memory budget, which source evidence should remain available, retrievable, and answer-bearing after ingestion? We instantiate this problem as *Budgeted Pre-Query Retention*, where memory is written before the future query is known and later read without returning to the full raw stream. We introduce EMBER, a learned retention policy that builds a compact, source-backed evidence state. Instead of storing summaries, predicted future questions, or a full raw-log substrate, EMBER stores evidence capsules: source excerpts paired with retrieval keys and update metadata. Post-query outcome feedback trains the writer to preserve evidence through the full chain from ingestion to retrieval to answer generation. On LongMemEval-RR, our LongMemEval-derived retained-evidence protocol, EMBER-14B reaches 0.3017 F1 at the uniform 8192-token comparison point, compared with 0.1765 for the best non-EMBER budgeted baseline. Across retained-token budgets, EMBER improves F1, Retain-Recall, and Read-Recall, showing that effective memory comes not from rereading more, but from making the right evidence survive.

1 Introduction

Long-horizon agents cannot treat memory as a free raw-log archive. The expensive part is not simply retaining bytes; it is keeping past evidence usable by a language model. Documents, conversations, files, and tool traces must be indexed, retrieved, reranked, and often reread through model context. These operations recur across queries. For both personal and organizational use, token budget is therefore a practical constraint: every missed memory write can force the system to broaden search, retrieve larger chunks, or send more history back through the model.

This cost changes the object of memory. Full-log RAG can postpone the choice of what matters because the corpus remains available at read time. Query-visible memory agents face a different, easier problem: they see the target task before deciding what to rewrite, retrieve, or route, as in MemAgent [Yu et al., 2026] and learned memory-operation systems optimized for downstream QA [Yan et al., 2025, Wang et al., 2025]. Persistent agents cannot always rely on either shortcut. They write memory while the future request is still unknown, and the full raw stream cannot always remain query-time usable. This turns memory into *budgeted evidence survival*: the goal is not to search more history, but to decide which source evidence should remain available, recoverable, and useful under a fixed query-time memory budget. We instantiate this objective as *Budgeted Pre-Query Retention*. The agent builds memory during ingestion, keeps only a fixed budget of source evidence, and later answers without returning to the full stream. In this protocol, pre-query writing is the timing constraint, while budgeted evidence survival is the memory objective. Efficiency therefore means

Preprint.

memory-token efficiency: the same retained-source budget should preserve more useful evidence, make that evidence easier to recover, and improve final answer quality.

EMBER addresses this tradeoff by learning what source evidence is worth keeping. It does not store predicted future questions, nor does it summarize the past indiscriminately. Instead, it identifies source spans likely to support future answers, attaches retrieval keys that make them findable, and stores compact evidence capsules under the retained-memory budget. On LongMemEval-RR, which keeps the LongMemEval histories, questions, and answers fixed while changing the resource regime [Wu et al., 2025], EMBER-14B reaches 0.3017 F1 at the shared 8192-token budget, compared with 0.1765 for the strongest budgeted non-EMBER baseline.

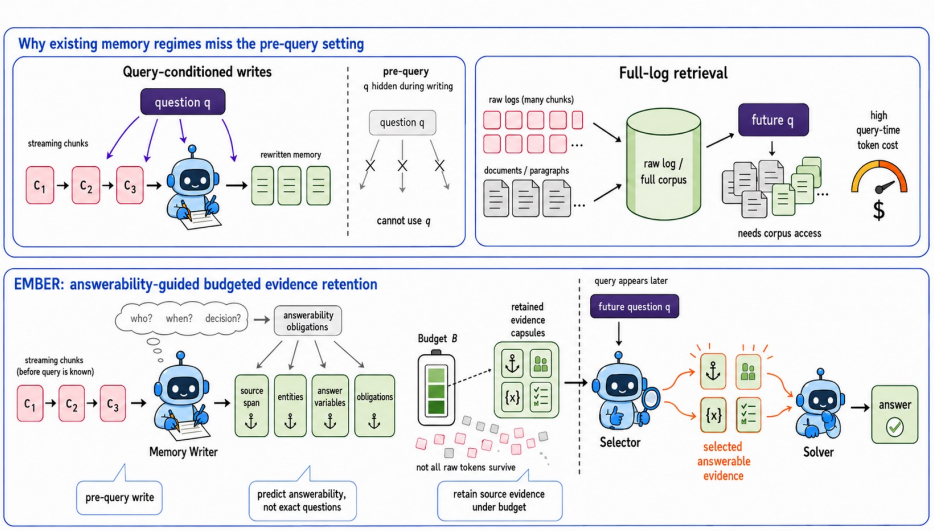

Figure 1: EMBER improves evidence survival under a retained-source budget: answerability probes select source spans, and retrieval keys keep them searchable at read time.

Retain-Recall asks whether gold evidence remains after ingestion; Read-Recall asks whether that evidence reaches the reader after the query becomes available. Together with F1, these metrics separate write-time retention from read-time access.

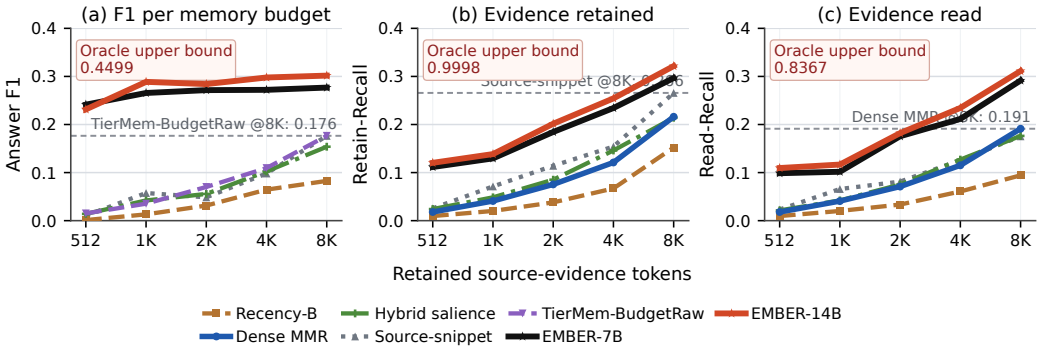

Figure 2: **Memory-token efficiency frontier.** We vary the retained-token budget and report F1, Retain-Recall, and Read-Recall. Table 1 fixes $B = 8192$ for the cross-method comparison; this figure shows the full budget frontier. Dashed references mark the best non-EMBER budgeted value at $B = 8192$ in each panel; Oracle is the in-panel upper bound.

Among budgeted evidence-survival baselines, TierMem [Zhu et al., 2026] with budgeted raw fallback gives the top F1, essentially tied with the source-snippet heuristic, while source snippets give the strongest Retain-Recall. EMBER-14B improves over both at this common max-budget point: it retains more gold evidence, retrieves more of that evidence at read time, and reaches 0.3017 F1. Figure 2 shows that this is not a single-budget effect: across retained-token budgets, EMBER turns the

Table 1: Main results on LongMemEval-RR at $B = 8192$ retained source tokens and top-$k = 10$. F1 is reported as mean $\pm$ half-width of the 95% nonparametric bootstrap confidence interval over 500 examples. Budgeted rows write memory before the query is known and have no raw-log access at read time; full-log and query-visible rows are reference categories. Oracle retention packs gold-annotated evidence into the same budget.

Method	Query visible at write time?	Full raw-log access?	Memory budget	Retain-Recall ↑	Read-Recall ↑	F1 ↑
Full-log references						
Full raw-log RAG [Lewis et al., 2020]	No	Yes	full log	1.0000	0.6082	0.3645±0.042
TierMem with full raw logs [Zhu et al., 2026]	No	Yes	full log	1.0000	0.6152	0.4541±0.044
Query-visible reference						
MemAgent-7B [Yu et al., 2026]	Yes	No	native	N/A	N/A	0.4633±0.044
Upper Bound						
Oracle retention	No	No	8192	0.9998	0.8367	0.4499±0.044
Budgeted evidence-survival baselines						
Random-B	No	No	8192	0.0612	0.0425	0.0350±0.016
Recency-B	No	No	8192	0.1521	0.0943	0.0827±0.024
Reservoir-B	No	No	8192	0.0753	0.0615	0.0498±0.019
Hybrid salience	No	No	8192	0.2140	0.1763	0.1538±0.032
Summary-only	No	No	B-equiv.	N/A	N/A	0.1156±0.028
MemAgent-7B (adapted) [Yu et al., 2026]	No	No	8192	N/A	N/A	0.1311±0.030
Source-snippet heuristic	No	No	8192	0.2657	0.1754	0.1763±0.033
Generated-query indexing [Nogueira et al., 2019]	No	No	8192	0.2021	0.1423	0.1205±0.029
TierMem with budgeted raw fallback [Zhu et al., 2026]	No	No	8192	0.1246	0.0997	0.1765±0.033
Memory-R1 (adapted) [Yan et al., 2025]	No	No	8192	0.1714	0.1464	0.1565±0.032
EMBER						
EMBER-7B (Qwen2.5-7B)	No	No	8192	**0.2966**	**0.2915**	**0.2768**±0.039
EMBER-14B (Qwen2.5-14B)	No	No	8192	**0.3215**	**0.3112**	**0.3017**±0.040

Notes. Oracle retention packs annotated gold-evidence turns into the same source-token budget; it uses gold evidence labels during retention but no answer text at read time. Appendix Table 13 gives extended reader diagnostics where logged; EMBER rows report only the synchronized F1/Retain/Read run used in the main comparison. Appendix C.4 defines the metrics formally.

same memory allowance into stronger evidence retention, read-time access, and answer F1. Oracle retention reaches 0.4499 F1 under the 8192-token budget, marking the remaining headroom from better source selection.

Contributions. Our contributions can be summarized as follows:

- We formalize budgeted evidence survival as Budgeted Pre-Query Retention and instantiate it as LongMemEval-RR, with metrics that separate retention from read-time access.
- We introduce an answerability-guided retention method that stores source excerpts with retrieval keys rather than generated summaries or future questions.
- We train the retention policy with answer-time outcome feedback, enabling the agent to preserve evidence that remains useful under a fixed query-time usable memory budget.

EMBER shifts memory construction from storing what happened to preserving what will remain answerable. Before any query is observed, it selects a compact set of source-backed evidence capsules that can still be retrieved and used under the read-time memory budget.

2 Problem Formulation

This section defines the episode timing, memory state, budget constraint, and learning objective used throughout the paper.

2.1 Streaming Episodes

Each episode is drawn from a distribution D and provides a stream $c_{1:K} = (c_1, \ldots, c_K)$, a query q, and a target answer y. During ingestion, the agent observes the stream in order and must retain memory without access to q. After ingestion, q becomes available, and the agent produces an answer $\hat{y}$ from its active context and retained memory.

2.2 Pre-Query Online Memory

The timing of query access is part of the problem definition. At write time, the agent can condition only on the observed stream prefix and its current memory state. At read time, it may condition on q to retrieve or select stored evidence. This pre-query memory setting captures online agents whose future user requests are unknown when memory is built.

2.3 Memory State

At step t, the agent maintains an active context w_t and an external memory state S_t. The active context is the bounded working state available for immediate computation. The external memory state is a collection of retained evidence capsules that persists beyond the active context within the episode. A retained-memory budget B_{ret} limits the retained evidence available after ingestion:

$$|S_K|_{\text{tok}} \leq B_{\text{ret}}.$$

Throughout the paper, retained-memory budget, retained-source budget, retained-token budget, and query-time usable memory budget refer to this same constraint: the number of retained source-excerpt tokens available at read time. The desired memory state is evidence-grounded: it should expose retrievable handles for future queries while preserving source evidence that can be returned to the reader.

2.4 Objective

The agent's objective is to learn a policy that maps streaming observations into budgeted retained memory and later answers from that memory:

$$S_K, w_K \leftarrow \pi(c_{1:K}), \qquad \hat{y} \sim \pi(\cdot \mid q, w_K, S_K).$$

Performance is measured by task success, such as answer correctness or F1, under the query-time usable memory budget and pre-query protocol. Section 3 instantiates this policy with two interfaces: a pre-query memory writer that builds the retained evidence cover during ingestion, and a reader that queries this cover after the target query is known.

3 Method

3.1 Retention Policy Overview

EMBER learns budgeted evidence survival. During ingestion, the writer decides which source spans should remain in memory under a fixed retained-source budget. Its state is a retained evidence cover: the writer observes the stream online, proposes evidence capsules, and maintains a cover S_t under the budget constraint

$$|S_t|_{\text{tok}} \leq B_{\text{ret}}.$$

The write path has three parts. Answerability probes identify source spans that may become useful later. Evidence capsules pair those spans with retrieval keys. A budget layer admits, consolidates, or rejects proposals under the retained-token budget. At read time, the reader retrieves from this cover and answers from preserved source evidence.

3.2 Answerability Probes

The write-time problem is not topical salience: a relevant passage may still miss the fact, relation, date, or update needed later. EMBER uses answerability probes, schema-constrained writer decisions that identify what future needs a chunk may support. A probe is not stored as a predicted query; it specifies what source evidence should remain and how it should be found. The policy emits a title, entity anchors, surface and intent retrieval keys, an update mode, and optional update targets. In extractive mode, the writer selects a retrieval handle, update intent, and bounded source-backed excerpt under budget; the excerpt is later returned verbatim to the reader. Table 2 summarizes the write path, and Appendix A gives the schema.

Table 2: Answerability-probe pipeline for one incoming chunk. The policy emits retrieval and update fields; retained memory stores source evidence bound to those fields.

Step	Decision	Materialized field or action
Evidence test	Keep or skip the chunk	`update_mode=insert/merge/overwrite/skip`
Anchor selection	Which entities, dates, or relations identify the fact	`title`, `entities`
Retrieval-key generation	How a future query may retrieve the evidence	`retrieval_keys_surface`, `retrieval_keys_intent`
Source binding	Which verbatim source excerpt carries the evidence	bounded `source_snippet/focused_source`
Budget update	Add, consolidate, replace, or reject under budget	BUDGETUPDATE and update targets

The GRPO update in Section 3.6 is applied to these emitted memory-control tokens. The stream text is the writer's input, the future query is hidden during writing, and the source excerpt is bound from the current chunk rather than invented by the writer.

3.3 Source-Backed Evidence Capsules

The retained cover consists of evidence capsules: a bounded source excerpt, retrieval keys, and token cost. The excerpt preserves stream evidence verbatim, while keys such as entities and intent descriptors make it findable after the query is known. The main budget counts source-excerpt tokens; retrieval metadata is reported separately in Appendix Table 9. Appendix A.1 gives the structured capsule action schema. The stored object is the capsule, not a free-form summary and not a generated future question.

3.4 Budgeted Retention

For each incoming chunk, the writer may propose candidate capsules M_t. The retained cover is updated by a budget-accounting layer:

$$S_t \leftarrow \mathrm{BudgetUpdate}(S_{t-1}, M_t; B_{\mathrm{ret}}).$$

BUDGETUPDATE is deterministic accounting over learned proposals: it checks source grounding, counts retained source tokens, and applies valid updates. During training rollouts, it also records any over-budget mass against the sampled budget, and the reward penalizes that overrun. Reported evaluations enforce the fixed retained budget exactly: the memory used at read time satisfies $|S_K|_{\mathrm{tok}} \leq B_{\mathrm{ret}}$. The learned component is the proposal itself: which spans to preserve, how much source context to include, and which retrieval keys to attach. Appendix A gives the structured action schema, and Appendix B.1 describes the concrete update modes, budget checks, and consolidation rules used to maintain the cover.

3.5 Read-Time Evidence Selection from the Retained Cover

After the query is known, EMBER reads only from the retained evidence cover S_K. The reader first expresses its information need as short retrieval queries $u_{1:n}$:

$$C \leftarrow \mathrm{Retrieve}(S_K, u_{1:n}),$$

where C contains top-k candidate capsules from the retained cover. The retriever never searches the full raw stream outside the retained-memory budget, so answerability now depends on what survived ingestion.

The memory policy then selects retained evidence for the reader:

$$r \sim \pi^{\mathrm{MM}}(\cdot \mid q, w, C), \qquad \hat{y} \sim \pi^{\mathrm{TS}}(\cdot \mid q, w, r).$$

Here r is selected evidence from preserved source excerpts, not a regenerated summary. This read path defines our evidence metrics: Retain-Recall measures whether gold evidence survives the pre-query budget update, and Read-Recall measures whether it reaches the reader after the query is known. Failure at either stage lowers F1. Outcome-gated training, described next, aligns retention with downstream answer success.

3.6 Answer-Gated Evidence-Chain Training

EMBER does not optimize final-answer reward alone. It trains the memory writer with an answer-gated evidence-chain objective: answer-bearing evidence should survive ingestion, remain readable at query time, and support the final answer under a sampled retained-memory budget. This aligns training with the same Survive–Read–Answer bottlenecks measured by Retain-Recall, Read-Recall, and answer F1. A rollout first commits a retained evidence cover under B_{ret} before the query is known; after the query becomes available, the reader retrieves from this cover, selects evidence, and answers. The training signal assigns final answer feedback to the earlier retention decisions, rather than only to the reader. Section 4 describes the training curriculum; Appendix C.9 gives the episode construction, budget sampling, reward instrumentation, and optimizer details.

Objective. Each rollout i produces a trajectory τ_i and an answer $\hat{y}_i$. Let $Q_i = \mathrm{Verify}(\hat{y}_i, y_i) \in [0, 1]$ denote final answer quality. We implement the chain with a multiplicative answer gate. The survival term is retained evidence coverage E_i; the readability terms are lookup rank score L_i and selection purity P_i; and W_i rewards valid source-preserving writes. All auxiliary terms are normalized to $[0, 1]$. The rollout reward is

$$\mathcal{R}_i \;=\; Q_i \left(\alpha_Q + \alpha_E E_i + \alpha_L L_i + \alpha_P P_i + \alpha_W W_i\right) \;-\; \lambda_{\mathrm{budget}} \frac{\max(0, |S_i|_{\mathrm{tok}} - \tilde{B}_i)}{\tilde{B}_i}\,.$$

Algorithm 1 Answer-Gated Evidence-Chain Training

Require: Dataset $\mathcal{D}$, retained-evidence budget grid $\mathcal{G}_B = \{512, 1024, 2048, 4096, 8192\}$, rollout batch size G

for each training batch **do**
  for $i = 1$ to G **do**
    Sample episode $(c^{(i)}_{1:K}, q_i, y_i) \sim \mathcal{D}$
    Sample retained-evidence budget $\tilde{B}_i \sim \mathrm{Uniform}(\mathcal{G}_B)$
    $(S_i, w_i, \tau_i) \leftarrow \textsc{IngestRetain}(c^{(i)}_{1:K}, \tilde{B}_i)$ ▷ withhold q_i during ingestion
    $u_i \leftarrow \textsc{Query}(q_i, w_i)$; $\mathcal{C}_i \leftarrow \textsc{Retrieve}(S_i, u_i)$
    $r_i \leftarrow \textsc{SelectEvidence}(q_i, w_i, \mathcal{C}_i)$
    $\hat{y}_i \leftarrow \textsc{Answer}(q_i, w_i, r_i)$
    Compute answer-gated evidence-chain reward $\mathcal{R}_i$
  end for
  $\bar{\mathcal{R}} \leftarrow \frac{1}{G}\sum_{i=1}^{G} \mathcal{R}_i$
  $\hat{A}_i \leftarrow \mathcal{R}_i - \bar{\mathcal{R}}$ for each $i \in \{1, \ldots, G\}$
  Update the trainable memory-control decision from each τ_i with a GRPO-style clipped update
end for

Final answer quality gates auxiliary evidence rewards, so survival and readability proxies receive credit only when they support the eventual answer. The last term is the budget penalty: if the retained cover S_i exceeds the sampled budget $\tilde{B}_i$, the overrun is normalized by $\tilde{B}_i$ and subtracted from the reward. Budget therefore becomes part of the evidence-chain objective: extra source evidence must improve answer-supported survival or readability enough to pay for its retained-token cost. This soft penalty is used during training rollouts to give the policy an explicit budget-aware signal across the frontier; reported evaluations use the fixed retained-budget protocol for comparison.

Group-relative memory-policy update. We use a standard GRPO-style update, applied to pre-query memory-control trajectories. For a rollout group of size G, we compute the group-relative advantage

$$\hat{A}_i \;=\; \mathcal{R}_i \;-\; \frac{1}{G}\sum_{j=1}^{G} \mathcal{R}_j.$$

Let $\mathcal{T}_i$ denote the tokens of the trainable memory-control decision updated for rollout i. We then apply a standard GRPO-style clipped update to those exposed memory-control tokens. The optimizer is standard; the method places the learning signal on the memory-policy interface, where pre-query trajectories receive outcome-gated evidence rewards.

Algorithm 1 summarizes the rollout and group-relative update. Appendix C.9 defines the auxiliary reward terms, episode format, budget sampling, coefficient selection, token-level objective, and implementation details.

4 Experiments

Training Details. We fine-tune Qwen2.5-7B and Qwen2.5-14B [Yang et al., 2024] memory-policy backbones on external pre-query episodes, keeping the retriever fixed during rollouts. Stage I uses RULER-HotpotQA [Hsieh et al., 2024] for controlled multi-hop evidence preservation and converges in about 500 optimization steps. Stage II then runs a 100-step multi-session continuation from MuSiQue [Trivedi et al., 2022] and 2WikiMultiHopQA [Ho et al., 2020], with Stage III hard cases for stale facts, time-scoped evidence, and related-but-unanswerable contexts folded into the same continuation. Across all stages, the query is hidden during memory writing, and the retained-memory budget is sampled from $\{512, 1024, 2048, 4096, 8192\}$ so the writer learns a budget frontier rather than a single operating point. Checkpoints and reward coefficients are selected on held-out external pre-query validation episodes, not on LongMemEval-RR or MultiQ-LongMemEval-RR; the selected coefficient vector $(\alpha_Q, \alpha_E, \alpha_L, \alpha_P, \alpha_W) = (0.45, 0.25, 0.15, 0.10, 0.05)$ is fixed for all reported evaluations. Appendix C.9 gives the episode format, optimization settings, coefficient sensitivity, seed stability, and reward-term ablations.

Benchmarks. LongMemEval-RR [Wu et al., 2025] is the main external long-horizon memory evaluation. RULER-HotpotQA [Hsieh et al., 2024] is the controlled, training-adjacent stress test. MultiQ-LongMemEval-RR and ablations then probe memory reuse and component effects.

- **LongMemEval-RR** is our LongMemEval-derived Budgeted Pre-Query Retention protocol [Wu et al., 2025], not an official LongMemEval benchmark name. It preserves the original histories, questions, and answer targets; only the resource regime changes. Methods ingest sessions before the query is known, retain a fixed budget of source evidence, and answer from retained memory instead of returning to the raw stream. This evaluates a natural deployment constraint for long-horizon agents: the task distribution stays fixed, while query-time raw-log access is no longer available. All LongMemEval-derived protocols are used only for evaluation.
- **RULER-HotpotQA** [Hsieh et al., 2024] is a controlled, training-adjacent stress test for evidence survival when memory is written before the question. Evaluation checks whether retained memory preserves clean multi-hop support chains.

Baselines. We compare three baseline categories: full-log systems with raw-history access, query-visible memory agents, and budgeted evidence-survival methods. EMBER belongs to the last category: it writes memory before the query is known and answers only from retained memory. On RULER-HotpotQA, we compare with context-only Qwen models, fixed-retriever vanilla RAG, and a pre-query adaptation of MemAgent; Appendix C.3 reports its adaptation curve. Because MemAgent was designed for query-visible in-context memory rewriting rather than source-evidence retention, this adapted version serves as a diagnostic, protocol-matched baseline.

LongMemEval-RR Main Results. Table 1 evaluates the evidence-survival chain at $B = 8192$, the largest retained-token budget used by the budgeted baselines. We use this common max-budget anchor to give every budgeted baseline its largest memory allowance. At this anchor, EMBER-14B reaches 0.3017 F1, compared with the top non-EMBER budgeted F1 baseline at 0.1765, a +0.125 F1 gain (95% paired bootstrap CI: [+0.081,+0.169]). Figure 2 shows the corresponding memory-token efficiency frontier: across retained-token budgets, EMBER converts the same source budget into higher F1, Retain-Recall, and Read-Recall than budgeted baselines. The gain is not only accuracy at the largest budget: at $B = 512$, EMBER-14B already reaches 0.2314 F1, above the strongest non-EMBER budgeted baseline at $B = 8192$. This uses one sixteenth of the retained source budget, saving about 7.7K source tokens in the query-time memory state. This is the relevant cost axis for long-horizon agents: if retained memory misses the useful evidence, the system must retrieve or reread larger raw contexts at query time. This pattern matches the training design, where the writer samples multiple budgets and learns to choose evidence under both tight and loose memory limits. Oracle retention reaches 0.4499 F1 and 0.8367 Read-Recall with the same 8192-token source budget; this leaves clear headroom and localizes the remaining loss to evidence selection under budget. Appendix Table 10 reports the full F1 sweep, and Appendix Table 9 reports the serialized metadata overhead for retained-memory states.

Mechanism Ablation: Survive, Read, Answer. Table 3 follows evidence through the pipeline: survival after ingestion, access at read time, and final answer F1. The ablations show that source evidence, answerability probes, and retrieval keys are coupled. Without probes, the writer keeps weaker source evidence. Without retrieval keys, more source evidence survives, but less reaches the reader. Full EMBER resolves both failures: probes choose source spans, retrieval keys make them recoverable, and outcome RL aligns the retained cover with final answer quality. The no-RL row uses the same probe-and-capsule interface as EMBER; its gap to full EMBER isolates the effect of answer-gated training on write-time retention choices, rather than a change in the reader or prompt format.

Table 3: LongMemEval-RR mechanism ablations at $B = 8192$. Columns track retention after ingestion, read-time access, and answer F1.

Variant	Mechanism removed	Retain-Recall	Read-Recall	F1	Failure mode
EMBER full	–	0.3215	0.3112	0.3017	Missed evidence / Aggregation errors
EMBER w/o outcome RL	Delayed outcome training	0.2651	0.2481	0.2242	Weak outcome alignment
No answerability probes	Write-time answerability cues	0.2426	0.2075	0.2031	Poor source selection
No retrieval keys	Retrieval keys	0.3032	0.2162	0.2089	Retained source is hard to retrieve/use
Generated-query indexing [Nogueira et al., 2019]	Answerability-grounded cues	0.2021	0.1423	0.1205	Query-like keys do not preserve answerability
Summary-only	Source traceability	N/A	N/A	0.1156	Compressed memory loses evidence
Oracle retention	Learned retention	0.9998	0.8367	0.4499	Upper bound

MultiQ Coverage Ablation. MultiQ-LongMemEval-RR tests whether one frozen retained memory can serve multiple future tasks. Table 4 shows the failure mode of generic budgeted retention: strong heuristics preserve only 10–12% of future evidence at a 10% budget and leave coverage balance at zero. EMBER-14B preserves a broader cover and reaches 0.2767 F1, compared with 0.0824 for the strongest heuristic baseline.

Table 4: MultiQ-LongMemEval-RR retention. One retained memory serves multiple future queries over the same history. Values report the 10% retained source-token budget with top-$k = 10$ retrieval. Coverage balance is the average weakest-query Retain-Recall.

Method	Budget	Mean-query Retain-Recall	Read-Recall	Group Retain-Recall	Coverage balance	F1
Recency-B	10%	0.1200	0.1200	0.1213	0.0000	0.0451
Reservoir-B	10%	0.1157	0.1157	0.1082	0.0000	0.0483
Dense MMR	10%	0.1131	0.1131	0.1141	0.0000	0.0614
Hybrid salience	10%	0.1063	0.1063	0.1071	0.0000	0.0824
Oracle retention	10%	0.6275	0.6275	0.6083	0.0517	0.3982
EMBER-7B	10%	0.2752	0.2691	0.3218	0.0312	0.2617
EMBER-14B	10%	0.3081	0.3051	0.3412	0.0356	0.2767

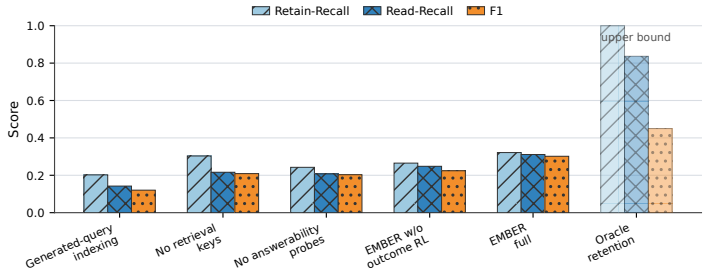

Figure 3: Evidence-loss chain on LongMemEval-RR. Each variant is evaluated at $B = 8192$ with the same reader. Bars show where evidence is lost: ingestion, read-time access, or answer use.

Controlled Pre-Query QA. RULER-HotpotQA is a controlled, training-adjacent stress test; LongMemEval-RR remains the main external memory evaluation. The task asks whether pre-query writing preserves a multi-hop support chain. Table 5 compares the main baseline families at the 28K operating point. EMBER-14B reaches 0.8412 ± 0.033 F1, above the strongest vanilla RAG configuration at 0.7772 ± 0.037 F1; Appendix C.8 reports the full length sweep.

Table 5: Controlled, training-adjacent pre-query QA on RULER-HotpotQA at 28K tokens. Appendix C.8 reports the full length sweep. F1 is reported as mean ± 95% confidence-interval half-width.

Group	Method	Query visible at write time?	Full source at read time?	F1 ↑
Context-only	Qwen3-8B	No	Yes	0.7108 ± 0.041
Full-source retrieval	Vanilla RAG, Qwen3-8B, k=8	No	Yes	0.7772 ± 0.037
Query-visible reference	MemAgent-7B [Yu et al., 2026]	Yes	No	0.7865 ± 0.037
Query-visible reference	Memory-R1 [Yan et al., 2025]	Yes	No	0.8165 ± 0.035
Budgeted evidence survival	MemAgent-7B (pre-query adapted) [Yu et al., 2026]	No	No	0.2606 ± 0.039
Budgeted evidence survival	Memory-R1 (pre-query adapted) [Yan et al., 2025]	No	No	0.3124 ± 0.041
Budgeted evidence survival	Source-snippet retention	No	No	0.7353 ± 0.039
Budgeted evidence survival	**EMBER-7B** (Qwen2.5-7B)	No	No	$\mathbf{0.8195 \pm 0.034}$
Budgeted evidence survival	**EMBER-14B** (Qwen2.5-14B)	No	No	$\mathbf{0.8412 \pm 0.033}$
Δ vs strongest Vanilla RAG				**+0.0640**

Notes. Budgeted evidence-survival rows write memory before the question is known and answer without full-source access. Full-source and query-visible rows are reference categories; all values report answer F1.

5 Related Work

Agent memory and learned memory operations. Persistent-memory systems store long-lived interaction records or external memory substrates for later use [Park et al., 2023, Zhong et al., 2023, Packer et al., 2023, Wang et al., 2023, 2024, Chhikara et al., 2025, Anokhin et al., 2024, Xu et al., 2025]. Recent learned memory-operation methods train agents to rewrite, update, revisit, or use memory entries [Yu et al., 2026, Yan et al., 2025, Wang et al., 2025, Huo et al., 2026, Shi et al., 2026]. MemAgent and Memory-R1 are the closest examples: they make memory writing an explicit learned action over entries or rewrites. These works leave open the evidence-survival question: under a fixed retained-memory budget, which source evidence should remain usable after ingestion?

RAG, tiered memory, and budgeted retrieval. Retrieval-augmented and tiered-memory systems reduce read-time cost by indexing, routing, or compressing access to stored information [Lewis et al., 2020, Zhang et al., 2026, Fang et al., 2025, Zhu et al., 2026]. TierMem is the sharpest contrast: it routes among summaries and can escalate to an immutable raw-log store when summary evidence is insufficient. Budgeted evidence survival instead asks which source evidence enters the retained memory during ingestion, when full-log fallback is outside the query-time budget. The question is not how to route over an accessible corpus, but how to make a small retained memory remain answer-bearing.

Query prediction, summarization, and context compression. Document expansion, self-questioning, and evidence-utility methods improve retrieval or memory access through predicted queries, self-generated probes, or utility signals [Nogueira et al., 2019, Yang et al., 2026, Ma et al., 2025, Jain and Vedam, 2026]. Extractive and query-focused summarization select source spans under a length budget or for a known information need [Liu, 2019, Dong et al., 2018, Dang, 2006]: the objective is document coverage or relevance to an observed query. Related context- and cache-compression work reduces read-time text or hidden-state footprint. These directions reduce the cost of using an available context or corpus. EMBER instead learns budgeted evidence survival: it stores source excerpts with write-time answerability cues so the retained memory remains available, retrievable, and answer-bearing under the query-time budget.

6 Conclusion

This paper studies pre-query memory writing under a fixed retained-token budget. EMBER uses answerability probes to select source spans before the query is known and trains the retention policy with post-query outcome feedback. Across LongMemEval-RR and MultiQ-LongMemEval-RR, this policy retains more evidence needed for future answers and achieves higher F1 than generic retention and budgeted raw-fallback baselines. Appendix C.1 discusses how this retained-evidence view extends to persistent agent memory.

7 Limitations and Broader Impact

We evaluate EMBER on controlled long-horizon memory benchmarks, not deployed assistant systems. Scores depend on evidence-label granularity, retained-token budget, and the reader/retriever; deployments may add variation from user-specific histories, evolving preferences, privacy constraints, and memory drift. We estimate training-seed variance with three EMBER-7B runs. Selective retention can reduce query-time readable history, making memory more compact and auditable, but it can also preserve sensitive, stale, or misleading evidence. Deployed systems should provide inspection and deletion controls, respect user intent before preserving private evidence, and treat retained memory as auditable application state rather than invisible model context.

References

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=k5nIOvYGCL.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schütze, Volker Tresp, and Yunpu Ma. Memory-r1:

Enhancing large language model agents to manage and utilize memories via reinforcement learning. *arXiv preprint arXiv:2508.19828*, 2025.

Yu Wang, Ryuichi Takanobu, Zhiqi Liang, Yuzhen Mao, Yuanzhe Hu, Julian McAuley, and Xiaojian Wu. Mem-α: Learning memory construction via reinforcement learning. *arXiv preprint arXiv:2509.25911*, 2025.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations*, 2025.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, 2020.

Qiming Zhu, Shunian Chen, Rui Yu, Zhehao Wu, and Benyou Wang. From lossy to verified: A provenance-aware tiered memory for agents. *arXiv preprint arXiv:2602.17913*, 2026. URL https://arxiv.org/abs/2602.17913.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*, 2019.

An Yang et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. doi: 10.48550/arXiv.2412.15115. URL https://arxiv.org/abs/2412.15115.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? In *Conference on Language Modeling*, 2024.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 2022.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, 2020.

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

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. *arXiv preprint arXiv:2305.10250*, 2023.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. *arXiv preprint arXiv:2306.07174*, 2023.

Yu Wang, Yifan Gao, Xiusi Chen, Haoming Jiang, Shiyang Li, Jingfeng Yang, Qingyu Yin, Zheng Li, Xian Li, Bing Yin, Jingbo Shang, and Julian McAuley. Memoryllm: Towards self-updatable large language models. *arXiv preprint arXiv:2402.04624*, 2024.

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.

Petr Anokhin, Nikita Semenov, Artyom Sorokin, Dmitry Evseev, Andrey Kravchenko, Mikhail Burtsev, and Evgeny Burnaev. Arigraph: Learning knowledge graph world models with episodic memory for llm agents. *arXiv preprint arXiv:2407.04363*, 2024.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*, 2025.

Yupeng Huo, Yaxi Lu, Zhong Zhang, Haotian Chen, and Yankai Lin. Atommem: Learnable dynamic agentic memory with atomic memory operation. *arXiv preprint arXiv:2601.08323*, 2026. URL https://arxiv.org/abs/2601.08323.

Yaorui Shi, Yuxin Chen, Siyuan Wang, Sihang Li, Hengxing Cai, Qi Gu, Xiang Wang, and An Zhang. Look back to reason forward: Revisitable memory for long-context llm agents. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=1cymflI2Lh.

Haozhen Zhang, Haodong Yue, Tao Feng, Quanyu Long, Jianzhu Bao, Bowen Jin, Weizhi Zhang, Xiao Li, Jiaxuan You, Chengwei Qin, and Wenya Wang. Learning query-aware budget-tier routing for runtime agent memory. *arXiv preprint arXiv:2602.06025*, 2026. URL https://arxiv.org/abs/2602.06025.

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. *arXiv preprint arXiv:2510.18866*, 2025.

Chengyuan Yang, Zequn Sun, Wei Wei, and Wei Hu. Beyond static summarization: Proactive memory extraction for llm agents. *arXiv preprint arXiv:2601.04463*, 2026. URL https://arxiv.org/abs/2601.04463.

Wenquan Ma, Jiayan Nan, Wenlong Wu, and Yize Chen. What deserves memory: Adaptive memory distillation for llm agents. *arXiv preprint arXiv:2508.03341*, 2025.

Siddharth Jain and Venkat Narayan Vedam. Cue-r: Beyond the final answer in retrieval-augmented generation. *arXiv preprint arXiv:2604.05467*, 2026. URL https://arxiv.org/abs/2604.05467.

Yang Liu. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*, 2019.

Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung. Banditsum: Extractive summarization as a contextual bandit. *arXiv preprint arXiv:1809.09672*, 2018.

Hoa Trang Dang. DUC 2005: Evaluation of question-focused summarization systems. In *Proceedings of the Workshop on Task-Focused Summarization and Question Answering*, pages 48–55, Sydney, Australia, 2006. Association for Computational Linguistics.

Daya Guo et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Bowen Jin et al. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings, 2023.

A Prompt Templates (Schema-Constrained JSON)

A.1 Memory Writer Prompt

```
SYSTEM: You are a memory writer. Output ONLY valid JSON.
GOAL: Convert the current stream chunk into source-evidence capsule actions
and a compact residual context.
CONSTRAINTS:
- produce source-evidence capsule actions when the chunk contains durable
    evidence
- emit a skip action when the chunk adds no evidence worth retaining
- each action uses insert / merge / overwrite / skip
- keep retrieval keys concise
- preserve source evidence needed for future answers through source excerpts
- residual active context must be compact
INPUTS:
- current_chunk: ...
- candidate_context: ...
- related_capsules:
  [ {id, title, entities, retrieval_keys, source_excerpt, version, update_mode
    }, ... ]
- retained_memory_state: ...
- usage_signals: ...
- budget_state: ...
OUTPUT JSON SCHEMA:
{
  "memory_items": [
    {
      "title": "...",
      "entities": ["...", "..."],
      "retrieval_keys_surface": ["...", "..."],
      "retrieval_keys_intent": ["...", "..."],
      "focused_source": "...",
      "update_mode": "insert|merge|overwrite|skip",
      "merge_target_id": "..."
    }
  ],
  "residual_context": {
    "pointers": ["...", "..."],
    "active_frontier": ["...", "..."],
    "residual_local_context": ["...", "..."]
  }
}
NOTE: This JSON is the memory-writer action format. The stored form is a
retrieval-indexed source-evidence capsule materialized from these fields.
For \texttt{skip}, the writer emits only the update mode and no source payload.
```

A.2 Task-Solver Prompt (Query Generation)

```
SYSTEM: You are a retrieval query generator. Output ONLY JSON.
GOAL: Produce 1-3 short retrieval queries for the current question.
CONSTRAINTS:
- each query <= 12 tokens
- diversify queries: entity-based + intent-based
OUTPUT:
{"queries": ["...", "..."]}
```

A.3 Memory Writer Evidence Selection Prompt

```
SYSTEM: You are a memory writer selecting retained source evidence.
Output ONLY JSON.
GOAL: From retrieved memory items, select the preserved source evidence
that should return to the task solver.
CONSTRAINTS:
- select the most relevant item ids for the given query
- avoid redundant or weakly supported items
- do not rewrite or compress the source excerpts
- the answer prompt will receive the selected source excerpts verbatim
INPUTS:
- query: ...
- active_context: ...
- retrieved_candidates: [ {id, title, entities, retrieval_keys, source_excerpt
    }, ... ]
OUTPUT JSON SCHEMA:
{
  "selected_ids": ["id1", "id2"]
}
NOTE: The task solver sees the source excerpts attached to the selected
items. The memory writer uses the retained-memory structure to judge
which retrieved evidence is authoritative and non-redundant.
```

B Implementation Notes

B.1 Update Modes for the Retained Evidence Cover

The writer proposes each capsule with an `update_mode` field. The budget-accounting layer executes the proposal deterministically:

`insert`
: Adds a new capsule to the cover. The layer builds a stored item from the proposed source excerpt, anchors, and retrieval keys, indexes it, and charges its source-token cost against B_{ret}. If admission would exceed the budget, the proposal is rejected.

`merge`
: Updates an existing capsule identified by `merge_target_id`. New facts are appended and deduplicated; the title, entities, and retrieval keys are replaced with the proposal's values. The source excerpt is overwritten only when the proposal supplies a new one; otherwise the original excerpt is preserved. The merged capsule is re-embedded and re-indexed.

`overwrite`
: Replaces the target capsule entirely: all fields are set to the proposed values, the version counter increments, and the index is rebuilt. This mode is used when the writer determines that the existing capsule is stale or superseded.

`skip`
: No modification to the cover. The writer emits `skip` when the current observation adds no evidence worth retaining. The layer validates that no target ID or payload accompanies a skip action.

Budget rejection. Every non-skip proposal is charged a source-token cost equal to the length of its source excerpt. If the cumulative retained tokens after the proposed update would exceed B_{ret}, the proposal is rejected and the cover remains unchanged. During training rollouts, over-budget proposals incur a cost penalty in the reward; during evaluation, the budget is enforced exactly.

Deduplication and consolidation. Before insertion, the layer checks for near-duplicate capsules using embedding cosine similarity. When a proposed capsule overlaps substantially with an existing

one, the writer is expected to use `merge` or `overwrite` rather than `insert`. Invalid references (e.g., a `merge` targeting a nonexistent ID) trigger a validation error and the proposal is dropped.

B.2 General Implementation Practices

- Strict JSON validation on every writer output; invalid outputs are resampled and resamples are counted as cost when desired.
- A lightweight dedup index (embeddings + hashing) identifies near-duplicate candidates for the merge and overwrite modes above.
- The retriever remains frozen during RL rollouts for training stability and a fixed read-side interface.

C Appendix

C.1 Retained Evidence State

Long-horizon memory is more than an index, a summary, or an extended context; it is a retained evidence state built before the next request arrives. Retain-Recall and Read-Recall make this state measurable by separating write-time evidence loss from read-time access failure and answer-time reasoning failure. This decomposition lets Budgeted Pre-Query Retention test memory construction rather than only retrieval quality.

Persistent assistants, coding agents, and tool-using systems accumulate dialogue, files, edits, and observations before the next request arrives. EMBER does not require these systems to archive everything as raw context; it provides a learnable interface for deciding which source evidence should remain available under a query-time memory budget.

C.2 Vanilla RAG K-Curve Control

Table 6: Vanilla RAG K-curve on the 28K-token pre-query RULER-HotpotQA setting, separate from the LongMemEval-RR retained-budget protocol. All rows use the Qwen2.5-7B backbone and BGE-small [Xiao et al., 2023] retriever. The control shows that EMBER's extractive read-select gain is not explained by reduced context size alone. Values report answer F1 in percentage points.

Method	**Result**
Qwen2.5-7B + vanilla RAG ($k = 3$)	62.28
Qwen2.5-7B + vanilla RAG ($k = 8$)	73.08
Qwen2.5-7B + vanilla RAG ($k = 15$)	74.84
Qwen2.5-7B + extractive read-select	76.95

C.3 MemAgent Retraining for Pre-Query Writing

For the MemAgent baseline, we follow the MemAgent in-context memory algorithm but train and evaluate it under the same pre-query protocol used for EMBER. During memory-writing turns, the model receives the stream chunk and its current in-context memory state, but not the downstream question. The question becomes available only after memory construction, when the model retrieves or uses the maintained memory to answer. This is the pre-query adapted MemAgent baseline used in the main comparison. The adaptation tests whether MemAgent's in-context rewriting interface can operate under the same timing constraint; it is not the native query-visible MemAgent setting. In the held-out curve below, retraining raises validation F1 from 0.0180 to 0.1330 and lowers unknown answers from 0.9860 to 0.8140, but the adapted interface remains weaker than source-evidence retention methods in the main comparison.

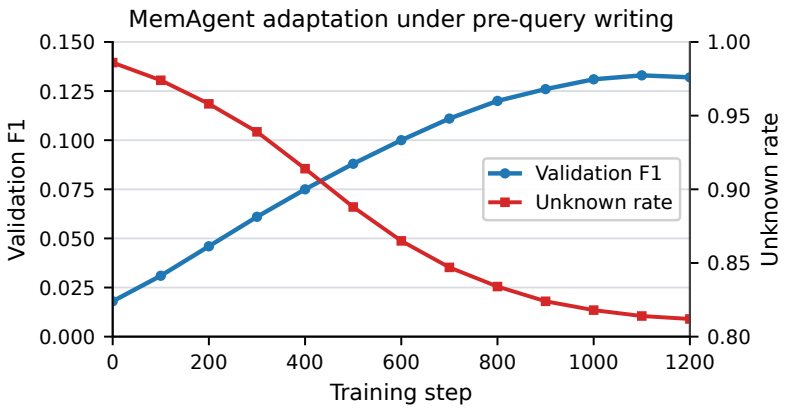

Figure 4: Retraining curve for MemAgent adapted to pre-query writing. The curve tracks the held-out pre-query objective during adaptation: memory is written before the target question is known, then evaluated after the question becomes available.

C.4 LongMemEval-RR Protocol and Extended Tables

LongMemEval-RR is a derived evaluation protocol built on LongMemEval [Wu et al., 2025]. It reuses LongMemEval histories and questions, keeps the answer targets fixed, and changes only access timing and the retained-memory budget. Methods ingest sessions online before the target query is known; after ingestion, they may use only a fixed budget B of retained source evidence. The full raw history is outside the query-time usable memory budget, and answers must be produced from retained memory. We report Retain-Recall, Read-Recall, answer F1, Sub-EM, and budget-normalized metrics. The offline retention probe measures whether gold evidence survives the retained-memory budget. Table 13 reports GPT-4o reader evaluations after query-time retrieval. We release the construction script, budget settings, chunk/span rules, retained-memory wrapper, and evaluator with the benchmark artifacts.

Retention metrics. For each query q, let $G(q)$ denote the gold source-evidence units, $M(q)$ the source-evidence units retained after ingestion, and $R(q)$ the units retrieved for the reader after the query becomes available. We measure the two points at which answer evidence can be lost:

$$\mathrm{RetainRecall}(q) = \frac{|G(q) \cap M(q)|}{|G(q)|}, \qquad \mathrm{ReadRecall}(q) = \frac{|G(q) \cap R(q)|}{|G(q)|}.$$

Retain-Recall measures retention: how much gold evidence remains in memory after pre-query ingestion. Read-Recall measures answer-time access: how much gold evidence reaches the reader context after query-time retrieval. For budgeted methods, retrieval is performed from retained memory, so Read-Recall is upper-bounded by Retain-Recall. Low Retain-Recall identifies a write-time retention failure; high Retain-Recall with low Read-Recall identifies a retrieval or ranking failure; high Read-Recall with low F1 identifies a reader-side reasoning or aggregation failure. Methods that do not retain source-traceable units, such as abstractive summaries or native MemAgent memory entries, report N/A for these source-evidence metrics.

Statistical reporting. Main reader tables define their uncertainty notation in the corresponding captions. LongMemEval-RR reports bootstrap confidence half-widths; the RULER-HotpotQA headline table reports 95% confidence-interval half-widths. Extended deterministic diagnostics report point estimates.

Table 7: Retention sweep and fixed-budget evidence results on LongMemEval-RR (turn-level, top-$k = 10$). Values are reported across retained source-evidence token budgets.

Method	**Metric**	B=512	B=1024	B=2048	B=4096	B=8192
Recency-B	Retain-Recall	0.0096	0.0202	0.0380	0.0670	0.1521
Recency-B	Read-Recall	0.0096	0.0202	0.0335	0.0607	0.0943
Dense MMR	Retain-Recall	0.0180	0.0409	0.0754	0.1210	0.2161
Dense MMR	Read-Recall	0.0180	0.0407	0.0709	0.1151	0.1912
Hybrid salience	Retain-Recall	0.0241	0.0485	0.0852	0.1460	0.2140
Hybrid salience	Read-Recall	0.0213	0.0421	0.0752	0.1270	0.1763
Source-snippet heuristic	Retain-Recall	0.0256	0.0712	0.1141	0.1541	0.2657
Source-snippet heuristic	Read-Recall	0.0231	0.0658	0.0812	0.1258	0.1754
Oracle retention	Retain-Recall	0.9608	0.9744	0.9839	0.9960	0.9998
Oracle retention	Read-Recall	0.9576	0.9602	0.9219	0.8842	0.8367
EMBER-7B	Retain-Recall	0.1116	0.1295	0.1845	0.2343	0.2966
EMBER-7B	Read-Recall	0.0987	0.1017	0.1756	0.2116	0.2915
EMBER-14B	Retain-Recall	0.1203	0.1385	0.2016	0.2543	0.3215
EMBER-14B	Read-Recall	0.1095	0.1167	0.1821	0.2345	0.3112

Table 8: Memory-token efficiency on LongMemEval-RR (turn-level, GPT-4o reader, top-$k = 10$). Table 1 uses $B = 8192$ as the fixed cross-method comparison point. F1 per 1K retained source tokens is computed only for budgeted methods with source-traceable retained evidence. Full-log reference systems are excluded because their accessible raw-log denominator is not the retained-memory budget.

Method	**Retained source tokens**	**Retain-Recall**	**Read-Recall**	**F1**	**F1 / 1K retained source tokens**
Random-B	8192	0.0612	0.0425	0.0350	0.0043
Recency-B	8192	0.1521	0.0943	0.0827	0.0101
Reservoir-B	8192	0.0753	0.0615	0.0498	0.0061
Hybrid salience	8192	0.2140	0.1763	0.1538	0.0188
Source-snippet heuristic	8192	0.2657	0.1754	0.1763	0.0215
Generated-query indexing [Nogueira et al., 2019]	8192	0.2021	0.1423	0.1205	0.0147
TierMem-BudgetRaw [Zhu et al., 2026]	8192	0.1246	0.0997	0.1765	0.0215
Memory-R1 (pre-query adapted) [Yan et al., 2025]	8192	0.1714	0.1464	0.1565	0.0191
Oracle retention	8192	0.9998	0.8367	0.4499	0.0549
EMBER-7B	8192	0.2966	0.2915	0.2768	0.0338
EMBER-14B	8192	0.3215	0.3112	0.3017	0.0368

Table 9: Retained-memory accounting at the $B = 8192$ LongMemEval-RR operating point. All rows use the same 8192-token source cap. The main protocol charges retained source-evidence tokens because source excerpts are the answer-bearing content. Retrieval metadata is serialized as index text and reported separately; it is not returned as answer content, and the reader receives source excerpts selected through the index. For EMBER, this overhead is at most 20.2% of the source-token cap.

Method	**Source-token cap**	**Avg metadata tokens**	**Total serialized tokens**	**Metadata/source ratio**	**F1**
Source-snippet heuristic	8192	0	8192	0.0%	0.1763
Generated-query indexing over snippets [Nogueira et al., 2019]	8192	1657	9849	20.2%	0.1205
EMBER-7B	8192	1243	9435	15.2%	0.2768
EMBER-14B	8192	1657	9849	20.2%	0.3017

Table 10: Reader F1 budget sweep on LongMemEval-RR (turn-level, GPT-4o reader, top-$k = 10$). Each cell reports a matched reader evaluation under the retained source-evidence budget.

Method	B=512	B=1024	B=2048	B=4096	B=8192
Recency-B	0.0012	0.0131	0.0314	0.0645	0.0827
Hybrid salience	0.0141	0.0420	0.0556	0.1024	0.1538
Source-snippet heuristic	0.0104	0.0575	0.0485	0.0981	0.1763
TierMem-BudgetRaw [Zhu et al., 2026]	0.0151	0.0353	0.0701	0.1094	0.1765
Oracle retention	0.4843	0.5013	0.5218	0.4632	0.4499
EMBER-7B	0.2413	0.2656	0.2715	0.2721	0.2768
EMBER-14B	0.2314	0.2887	0.2841	0.2978	0.3017

Interpreting the F1 frontier. The F1 sweep tracks the same efficiency pattern as the evidence metrics: EMBER remains ahead across retained-token budgets and reaches its strongest reported F1 at the shared $B = 8192$ comparison point. Retain-Recall and Read-Recall in Table 7 show why the frontier improves: as the budget grows, more gold evidence survives ingestion and reaches the reader. The low-budget end is also important for deployment: EMBER-14B at $B = 512$ reaches 0.2314 F1, already above the strongest non-EMBER budgeted baseline at $B = 8192$ (0.1765). In retained-memory terms, this is the same answer pipeline using 512 rather than 8192 charged source tokens, a reduction of 7680 tokens, or about 94%, per query-time memory state. These savings matter because missed retained evidence usually shifts cost back to read time: the system must retrieve broader chunks, rerank more candidates, or reread more raw history through the model context.

Table 11: Offline retention probe on LongMemEval-RR (top-$k = 10$). The table reports the strongest fair heuristic at each retained-token budget under the offline probe. Randomized baselines are averaged over five seeds, with confidence intervals reported as 95% CIs when shown; dense baselines use deterministic hashing embeddings. Session-level rows report oracle retained recall only, so oracle retrieved recall is marked n/a.

Granularity	**Budget**	**Best fair heuristic**	**Retain-Recall**	**Read-Recall**	**Oracle Retain-Recall**	**Oracle retrieved (Read-Recall)**
Session	512	Source-snippet heuristic	0.0150	0.0150	0.0270	n/a
Session	1024	Source-snippet heuristic	0.0180	0.0180	0.0810	n/a
Session	2048	TF-IDF salience	0.0358	0.0358	0.2790	n/a
Session	4096	Recency-B	0.0679	0.0679	0.7069	n/a
Session	8192	Hybrid salience	0.1297	0.1297	0.9776	n/a
Turn	512	Dense MMR	0.0180	0.0180	0.9608	0.9576
Turn	1024	Dense MMR	0.0409	0.0407	0.9744	0.9602
Turn	2048	Dense MMR	0.0754	0.0709	0.9839	0.9219
Turn	4096	Dense MMR	0.1210	0.1151	0.9960	0.8842
Turn	8192	Dense MMR	0.2161	0.1912	0.9998	0.8367

Table 12: Hybrid-salience retention probes on LongMemEval-RR. Values summarize hybrid-salience retention across single-query and multi-query retained-memory settings.

Setting	**Granularity**	**Budget**	**Top-k**	**Retain-Recall**	**Read-Recall**	**Coverage balance**
LongMemEval-RR	Turn	8192	10	0.2140	0.1763	n/a
LongMemEval-RR	Session	8192	10	0.1297	0.1297	n/a
MultiQ-LongMemEval-RR	Group	10%	10	0.1063	0.1063	0.0000

Table 13: Extended GPT-4o reader evaluation on LongMemEval-RR, derived from LongMemEval histories and questions [Wu et al., 2025]. Methods ingest turn-level histories before the query is known, then answer after top-k retrieval from either retained memory or full raw logs. Budgeted rows use $B = 8192$ retained source tokens unless noted. Full raw-log RAG and TierMem-FullRaw keep the full raw history at read time. Oracle retention is an upper-bound policy that packs annotated gold-evidence turns into the same retained source-token budget before answer-time retrieval; it uses gold evidence labels during retention but no answer text at read time. Unknown is the fraction of reader outputs that abstain or cannot produce an answer. MemAgent rows store rewritten memory text rather than retained source units, so source-evidence recall metrics are N/A. EMBER rows report the synchronized F1/Retain/Read run used in the main comparison; Sub-EM and Unknown are not reported for those rows.

Method	Budget	Top-k	F1	Sub-EM	Unknown	Retain-Recall	Read-Recall
Full raw-log RAG [Lewis et al., 2020]	full raw	3	0.2567	0.2260	0.6180	1.0000	0.3949
Full raw-log RAG [Lewis et al., 2020]	full raw	5	0.3004	0.2640	0.5480	1.0000	0.4814
Full raw-log RAG [Lewis et al., 2020]	full raw	10	0.3645	0.3140	0.4340	1.0000	0.6082
TierMem-FullRaw [Zhu et al., 2026]	full raw	10	0.4541	0.4600	0.4600	1.0000	0.6152
Oracle retention	8192	3	0.3786	0.3420	0.4680	0.9998	0.6802
Oracle retention	8192	5	0.4104	0.3600	0.4100	0.9998	0.7527
Oracle retention	8192	10	0.4499	0.3980	0.3680	0.9998	0.8367
Random-B	8192	10	0.0350	0.0129	0.9251	0.0612	0.0425
Recency-B	8192	10	0.0827	0.0756	0.8756	0.1521	0.0943
Reservoir-B	8192	10	0.0498	0.0275	0.9214	0.0753	0.0615
Hybrid salience	8192	10	0.1538	0.1264	0.7653	0.2140	0.1763
Summary-only	B-equiv.	10	0.1156	0.0967	0.8251	N/A	N/A
Source-snippet heuristic	8192	10	0.1763	0.1285	0.7852	0.2657	0.1754
Generated-query indexing [Nogueira et al., 2019]	8192	10	0.1205	0.1065	0.7402	0.2021	0.1423
TierMem-BudgetRaw [Zhu et al., 2026] (Recency-B)	8192	10	0.1765	0.1476	0.7126	0.1246	0.0997
MemAgent-7B (pre-query adapted) [Yu et al., 2026]	8192	10	0.1311	N/A	0.8180	N/A	N/A
EMBER-7B	8192	10	0.2768	N/A	N/A	0.2966	0.2915
EMBER-14B	8192	10	0.3017	N/A	N/A	0.3215	0.3112
TierMem-BudgetRaw [Zhu et al., 2026] (Recency-B)	8192	3	0.0503	0.0480	0.9300	0.1246	0.0731
TierMem-BudgetRaw [Zhu et al., 2026] (Dense MMR)	8192	3	0.0248	0.0180	0.9560	0.0303	0.0208
TierMem-BudgetRaw [Zhu et al., 2026] (Hybrid salience)	8192	3	0.0224	0.0200	0.9720	0.0171	0.0130
TierMem-BudgetRaw [Zhu et al., 2026] (Source-snippet)	8192	3	0.0195	0.0160	0.9700	0.0147	0.0134
MemAgent-7B (query-visible) [Yu et al., 2026]	native	3	0.4633	0.4700	0.4200	N/A	N/A

Table 14: TierMem-BudgetRaw breakdown on LongMemEval-RR with GPT-4o reader, turn-level retention, $B = 8192$, and top-$k = 3$ query-time retrieval. Values report answer F1 by question type.

Question type	F1
Knowledge update	0.1995
Multi-session	0.0030
Temporal reasoning	0.0000
Single-session user	0.0786

Table 15: Diagnostic retained-budget adaptations of existing memory methods. Adapted variants ingest the same stream before the query is known, enforce the same retained-memory budget, and answer only from retained memory. These rows use a Qwen2.5-14B reader and are not mixed with the GPT-4o Table 1 comparison.

Method	Original assumption	RR adaptation	Budget enforcement	Raw-log access?	Reader	Retain-Recall	F1
Memory-R1 [Yan et al., 2025]	Learned memory operations	RR retained-evidence adaptation	Hard retained budget	No	Qwen2.5-14B	0.1701	0.1524
Mem-α [Wang et al., 2025]	QA-reward memory construction	RR retained-evidence adaptation	Hard retained budget	No	Qwen2.5-14B	0.2341	0.1731
AtomMem [Huo et al., 2026]	Atomic memory editing	RR retained-evidence adaptation	Hard retained budget	No	Qwen2.5-14B	0.1792	0.1568

Table 16: Multi-query LongMemEval-RR retention probe. A single retained memory is shared by multiple future queries from the same history. Values report mean per-query retained evidence recall, retrieved evidence recall, and coverage balance under retained source-token budgets. For a query bundle g with queries $q_{g,1:K}$, coverage balance is $\min_j \mathrm{RetainRecall}(q_{g,j})$, averaged over bundles; it measures whether one frozen memory supports all future queries rather than only the easiest subset.

Top-k	Budget	Best fair retention	Mean query Retain-Recall	Read-Recall	Coverage balance	Oracle mean query Retain-Recall
3	1%	Recency-B	0.0147	0.0147	0.0000	0.0950
3	2%	Random-B	0.0265	0.0265	0.0000	0.1397
3	5%	Recency-B	0.0650	0.0645	0.0000	0.3386
3	10%	Recency-B	0.1200	0.1100	0.0000	0.6275
10	1%	Recency-B	0.0147	0.0147	0.0000	0.0950
10	2%	Random-B	0.0265	0.0265	0.0000	0.1397
10	5%	Recency-B	0.0650	0.0650	0.0000	0.3386
10	10%	Recency-B	0.1200	0.1200	0.0000	0.6275

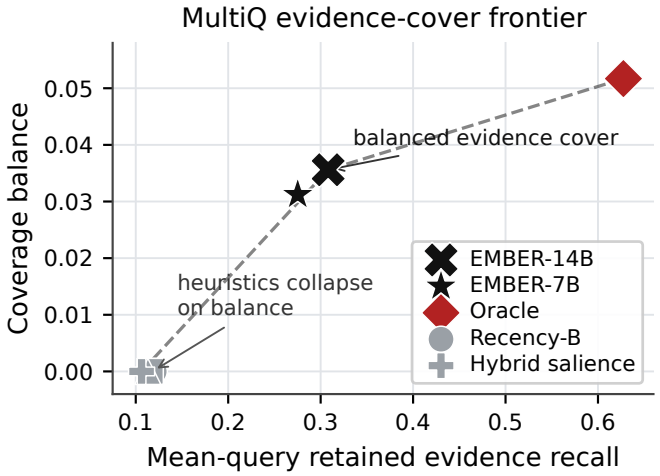

Figure 5: MultiQ coverage frontier on LongMemEval-RR. At the same 10% retained source-token budget, heuristic retention methods preserve fragments of future evidence but fail to cover all future queries. EMBER shifts the retained memory toward a more balanced evidence cover while also increasing mean-query evidence recall.

Table 17: Implementation and training infrastructure used for LongMemEval-RR rollouts.

Component	Implementation choice	Why it matters
Rollout engine	vLLM rollout server	Supports batched pre-query episodes.
Retriever / index	GPU-resident embedding index	Keeps retrieval latency controlled during budget sweeps.
Episode format	Online ingest followed by post-ingestion queries	Matches the retained-budget protocol.
Budget accounting	Soft over-budget penalty during training; hard retained budget at evaluation	Makes retained evidence cost comparable across methods.
RL interface	Selected trainable decision turn	Assigns answer-time outcome feedback to the memory-control action.

C.5 Compute, Assets, and Release Notes

Table 18 summarizes the execution paths used by the reported experiments. Reader evaluations use GPT-4o through Azure OpenAI with deterministic decoding. Retention probes and LongMemEval-RR reader runs share the same retained-budget wrapper and evaluation scripts. The training and rollout jobs use two GPU configurations: H200 SXM $\times 4$ and H100 NVL $\times 8$. The anonymized supplementary artifact includes the construction scripts, budget settings, evaluation commands, and wrappers needed to reproduce the reported protocol.

Table 18: Compute disclosure for the reported experiments. Exact wall-clock time depends on batching and API throughput; the released scripts specify the commands and operating points used for each run.

Experiment family	Compute path	Reported operating point
LongMemEval-RR retention sweeps	CPU preprocessing plus embedding retrieval/indexing	$B \in \{512, 1024, 2048, 4096, 8192\}$, top-$k = 10$
LongMemEval-RR reader evaluation	Azure GPT-4o reader, temperature 0	turn-level, $B = 8192$, top-$k = 10$
EMBER training	vLLM rollout server on H200 SXM $\times 4$ or H100 NVL $\times 8$	Stage I converges around 500 steps; Stage II/III continuation trains for 100 steps
MultiQ-LongMemEval-RR	Offline retained-evidence probe on CPU/GPU preprocessing path	multi-query retained-budget setting, 1–10% retained budget

Table 19 lists the main external assets. We cite the upstream sources and do not redistribute third-party model weights or benchmark data beyond derived protocol scripts and evaluation metadata. Users of the released artifact should obtain each upstream dataset or model under its own license and terms.

Table 19: External assets used by the paper and how they are used.

Asset	Use in this paper	License / terms handling
LongMemEval [Wu et al., 2025]	Source histories and questions for LongMemEval-RR	Cited upstream; users obtain upstream data under its terms.
RULER / HotpotQA / MuSiQue / 2WikiMultiHopQA [Hsieh et al., 2024, Trivedi et al., 2022, Ho et al., 2020]	Controlled pre-query QA training and stress tests	Cited upstream; benchmark data are not redistributed in the paper.
Qwen2.5 / Qwen3	Writer and reader backbones	Upstream model sources are credited; third-party weights are not redistributed.
BGE-small [Xiao et al., 2023]	Embedding retriever	Cited upstream; used as an external embedding model.
GPT-4o / Azure OpenAI	Reader evaluation	Used through the hosted API; model weights are not redistributed.
Prior memory systems [Yu et al., 2026, Yan et al., 2025, Wang et al., 2025, Zhu et al., 2026]	Baselines and comparison regimes	Cited upstream; adaptations are evaluated under the retained-budget protocol.

C.6 Full Rollout Procedure

Algorithm 2 expands the concise method description into the full execution procedure used by the rollout engine. The main text focuses on the retention mechanism; this appendix records the implementation sequence: online ingest, budgeted cover update, query-time retrieval from retained memory, evidence selection, answer generation, and trajectory logging for outcome training.

Here B_{work} denotes the active-context capacity used to trigger a write step when the streaming buffer grows too large. It is an implementation threshold for w, not the retained-memory budget reported in our experiments. All budgeted comparisons vary or fix B_{ret}.

Algorithm 2 EMBER Rollout with Budgeted Pre-Query Evidence Retention

Require: Stream $c_{1:K}$, query q, active-context budget B_{work}, retained-evidence budget B_{ret}, initial cover S_0, memory policy π^{MM}, solver policy π^{TS}
Ensure: Answer $\hat{y}$, trajectory τ, retained evidence cover S

$w \leftarrow \emptyset,\ S \leftarrow S_0,\ \tau \leftarrow ()$
Phase 1: Pre-query evidence retention
for $k = 1$ to K **do**
  $w \leftarrow \text{Append}(w, c_k)$
  while $|w| > B_{\text{work}}$ **do**
    $C^{\text{maint}} \leftarrow \text{LocalMemoryState}(S, w)$
    $(M, w') \sim \pi^{\text{MM}}(\cdot \mid c_k, w, C^{\text{maint}})$
    $S \leftarrow \text{BudgetUpdate}(S, M; B_{\text{ret}})$
    $w \leftarrow w'$
    Append the memory-control segment and log-probs to τ
  end while
end for
Phase 2: Answering from retained evidence
$u_{1:n} \sim \pi^{\text{TS}}(\cdot \mid q, w)$
Append q to $u_{1:n}$ if it is not already included
$C \leftarrow \text{Retrieve}(S, u_{1:n})$ ▷ retrieve only from retained evidence
$r \sim \pi^{\text{MM}}(\cdot \mid q, w, C)$ ▷ select retained evidence
Append the evidence-selection segment and log-probs to τ
$\hat{y} \sim \pi^{\text{TS}}(\cdot \mid q, w, r)$
Append solver query and answer segments to τ
return $\hat{y},\ \tau,\ S$

C.7 LongMemEval-S Ablation Settings

The LongMemEval-S representation ablations compare memory representations and interfaces for online ingestion. These settings are not a fully nested ablation ladder: they test different paths from streamed sessions to query-time evidence. **Online notes** asks the LLM to write generic notes before the query is known, stores the generated note text, retrieves those notes, and answers from them. This setting represents online writeable memory with weak evidence grounding: the model may organize the stream, but it can also discard source details needed later. **Source only** stores preserved source snippets directly and retrieves them at query time. It keeps answer evidence available but lacks retrieval metadata such as titles, entities, or keys. **Schema + source** augments preserved source snippets with retrieval metadata, testing whether source evidence becomes more useful future memory when paired with an explicit schema. Thus, source only is contained in schema + source, but online notes is a separate generated-memory path rather than a subset of either source-retention setting.

C.8 Pre-Query RULER-HotpotQA Length Sweep

Table 20: Pre-query RULER-HotpotQA length sweep across input lengths up to 3.5M tokens. All methods commit memory or indexes before the question is known and see the question only at retrieval or answer time. MemAgent rows use the pre-query retraining protocol described in Appendix C.3. Values report answer F1 on a 0–1 scale.

Model	Method	7K	14K	28K	56K	112K	224K	448K	896K	1.75M	3.5M
Qwen2.5-7B-Instruct	Context only	0.6366	0.5739	0.5237	0.0025	0.0023	0.0000	0.0014	0.0071	0.0022	0.0000
Qwen2.5-7B-Instruct	Vanilla RAG ($k = 15$)	0.7553	0.7514	0.7484	0.7416	0.7334	0.7279	0.7089	0.6633	0.6525	0.6285
Qwen2.5-7B-Instruct	Vanilla RAG ($k = 8$)	0.7331	0.7282	0.7308	0.7203	0.7142	0.7124	0.7032	0.6542	0.6401	0.6197
Qwen2.5-14B-Instruct	Context only	0.7433	0.7071	0.6447	0.0026	0.0047	0.0000	0.0030	0.0069	0.0025	0.0000
Qwen2.5-14B-Instruct	Vanilla RAG ($k = 15$)	0.7570	0.7382	0.7369	0.7532	0.7361	0.7395	0.6836	0.6793	0.6395	0.5647
Qwen2.5-14B-Instruct	Vanilla RAG ($k = 8$)	0.7412	0.7281	0.7458	0.7401	0.7156	0.7215	0.6645	0.6638	0.6196	0.5432
Qwen3-8B	Context only	0.7671	0.7570	0.7108	0.0026	0.0029	0.0017	0.0039	0.0076	0.0024	0.0000
Qwen3-8B	Vanilla RAG ($k = 15$)	0.7817	0.7685	0.7713	0.7653	0.7584	0.7715	0.7527	0.7429	0.6990	0.5563
Qwen3-8B	Vanilla RAG ($k = 8$)	0.7707	0.7542	0.7772	0.7587	0.7412	0.7653	0.7392	0.7245	0.6775	0.5502
Qwen3-14B	Context only	0.6880	0.6846	0.5478	0.0075	0.0027	0.0000	0.0087	0.0036	0.0028	0.0000
Qwen3-14B	Vanilla RAG ($k = 15$)	0.7821	0.7627	0.7755	0.7723	0.7603	0.7644	0.7010	0.7312	0.6330	0.5354
Qwen3-14B	Vanilla RAG ($k = 8$)	0.8017	0.7712	0.7542	0.7765	0.7514	0.7556	0.7125	0.7245	0.6225	0.4914
MemAgent-7B	retrained for pre-query writing	0.2929	0.2796	0.2606	0.2249	0.2597	0.2665	0.3320	0.2363	0.1500	0.2125
MemAgent-14B	retrained for pre-query writing	0.2981	0.2814	0.2069	0.2396	0.2208	0.2494	0.2476	0.2181	0.1631	0.1169
EMBER-7B	**Learned memory policy**	0.8342	0.8343	0.8195	**0.8249**	0.8116	0.8013	0.7856	0.7907	0.7927	**0.7856**
EMBER-14B	**Learned memory policy**	**0.8542**	**0.8613**	**0.8412**	0.8214	**0.8212**	**0.8203**	**0.8023**	**0.7945**	**0.8006**	0.7851

C.8.1 W0 Diagnostic Probe

To localize the bottleneck of the trained policy, we use a fixed-action diagnostic rollout with the trained checkpoint. During rollout, each memory write is instrumented with provenance tags and measured whether each stored item was later recovered by retrieval, whether it was selected into the recall context, and whether it contained the gold evidence span. Table 21 reports the resulting stage-wise metrics.

Table 21: W0 training diagnostic rollout on held-out episodes, reported separately from the LongMemEval-RR retained-budget benchmark. Retrieval recall is saturated under the default top-k setting, while losses appear in write quality and read-side selection.

Stage	Value	Interpretation
Written evidence coverage	47.8%	About half of written items contain gold evidence.
Retrieval hit rate	100.0%	The retriever recovers written memory at default top-k.
Read-selection hit rate	41.7%	Read selection keeps a minority of written items.
Answer-context gold rate	77.3%	Some answers still lack selected gold evidence.

The diagnostic shows that retrieval recall is not the current bottleneck: every stored item was recovered by the retriever under the default top-k setting. The remaining loss comes from write quality and read-side selection. Only 47.8% of written items contain the gold evidence span, only 41.7% of written items are selected into the recall context, and 22.7% of episodes still reach the answer stage without selected gold evidence. This points to write quality and read-side selection rather than simply increasing retrieval capacity.

C.9 Training and Reward Details

Algorithm 1 in Section 3.6 summarizes the training rollout and group-relative update. This appendix gives the instrumentation, episode construction, coefficient selection, and token-level objective used by that algorithm.

The auxiliary reward terms in Section 3.6 are computed from rollout instrumentation. Evidence coverage E_i measures whether the stored or selected evidence contains gold support. The rank score L_i rewards placing gold evidence high in the retrieved set. Selection purity P_i measures how much of the selected evidence is useful rather than distracting. Write utility W_i rewards valid memory writes that preserve future-use evidence. All auxiliary terms are normalized to $[0, 1]$ before being gated by final answer quality Q_i. The budget penalty is $\lambda_{\text{budget}} \max(0, |S_i|_{\text{tok}} - \tilde{B}_i)/\tilde{B}_i$, which charges the policy for source evidence retained beyond the sampled budget.

Training data construction. The curriculum trains the retention interface, not a standalone reader. It has two optimizer-length blocks. Stage I uses RULER-HotpotQA to create controlled long-context streams with support evidence and distractors and converges in about 500 optimization steps. Stage II is a 100-step multi-session continuation: it converts MuSiQue and 2WikiMultiHopQA examples into episodes by assigning support facts to timestamped sessions, inserting distractor sessions, and preserving support unit identifiers for reward computation. Stage III hard cases are included in this 100-step continuation rather than run as a separate optimizer block. These transformations add update, temporal, and abstention episodes. In update episodes, an early session states an old value and a later session corrects it; the writer must preserve the evidence that determines the current answer. In temporal episodes, the query depends on when an event, deadline, or preference holds. In abstention episodes, the stream contains topically related content but no source evidence sufficient to answer the query. These cases train the writer to prefer gold source evidence over merely topical context. LongMemEval histories, questions, and evidence labels are not used for training; all LongMemEval-derived protocols are held out for evaluation.

Reward coefficient selection. We select $(\alpha_Q, \alpha_E, \alpha_L, \alpha_P, \alpha_W) = (0.45,\ 0.25,\ 0.15,\ 0.10,\ 0.05)$ on a held-out validation split constructed from the external training sources: RULER-HotpotQA, MuSiQue, and 2WikiMultiHopQA-derived episodes. These validation episodes are disjoint from training episodes and from the LongMemEval-RR and MultiQ-LongMemEval-RR evaluation sets. All reported results use this frozen coefficient vector.

Table 22: Reward-coefficient sensitivity on LongMemEval-RR for EMBER-7B at $B = 8192$. The default coefficient vector is selected on external validation episodes; the alternatives vary the reward emphasis without test-set tuning.

Reward coefficients $(\alpha_Q, \alpha_E, \alpha_L, \alpha_P, \alpha_W)$	**Retain-Recall** ↑	**Read-Recall** ↑	**F1** ↑
(0.45, 0.25, 0.15, 0.10, 0.05)	**0.2966**	**0.2915**	**0.2768**
(0.50, 0.20, 0.15, 0.10, 0.05)	0.2917	0.2856	0.2733
(0.40, 0.30, 0.15, 0.10, 0.05)	0.2989	0.2854	0.2722
All equal: (0.20, 0.20, 0.20, 0.20, 0.20)	0.2876	0.2711	0.2552
Top non-EMBER budgeted F1 baseline	0.1246	0.0997	0.1765

The sweep shows that EMBER is stable under small coefficient changes. Nearby answer-heavy and evidence-heavy variants stay within 0.005 F1 of the reported configuration, while equal weighting is lower. We therefore use the rounded validation-selected vector for all reported runs.

Table 24 ablates the same objective by removing individual reward signals. The default objective gives the strongest full-chain performance. Final-answer-only training loses both retention and read-time access, while removing E, L, or P damages different points of the Survive–Read–Answer chain. This supports the training design used in the main experiments: final answer quality gates the reward, but evidence-retention and readability signals are needed to assign credit to pre-query memory decisions.

For each training batch, we sample a budget $B_{\text{ret}} \in \{512, 1024, 2048, 4096, 8192\}$ for each episode rather than training only at the 8192-token operating point. This exposes the writer to the same

Table 23: Training-seed stability for EMBER-7B on LongMemEval-RR at $B = 8192$. Each seed uses the same reward coefficients, training curriculum, budget sampling, and checkpoint selection on held-out external pre-query validation episodes. LongMemEval-RR is used only for final evaluation. We use EMBER-7B as the lower-cost replicated setting for estimating training-seed variance; EMBER-14B is reported as a single-seed main model.

Seed	**Retain-Recall ↑**	**Read-Recall ↑**	**F1 ↑**
0	0.2966	0.2915	0.2768
1	0.2912	0.2856	0.2705
2	0.2954	0.2926	0.2801
Mean ± std	0.2944±0.0028	0.2899±0.0038	0.2758±0.0049
TierMem-BudgetRaw (best F1 baseline)	0.1246	0.0997	0.1765

Table 24: Reward-term ablation on LongMemEval-RR for EMBER-7B at $B = 8192$. All rows use the same pre-query retention protocol and reader. The ablations compare the full answer-gated objective with final-answer-only training and single-term removals of retained-evidence coverage E, lookup/readability score L, and selection purity P.

Reward	**Retain-Recall ↑**	**Read-Recall ↑**	**F1 ↑**
Default	**0.2966**	**0.2915**	**0.2768**
Final-answer Only	0.2666	0.2415	0.2361
w/o E	0.2556	0.2416	0.2354
w/o L	0.2712	0.2448	0.2322
w/o P	0.2759	0.2571	0.2425

retained-memory frontier used in Figure 2: the policy must learn which evidence survives when the budget is tight as well as when the cover is larger.

Validation and checkpoint selection. Checkpoint selection uses held-out pre-query validation episodes constructed from the same external sources and transformations as training. LongMemEval-RR and MultiQ-LongMemEval-RR are used only after checkpoint selection, so the main external evaluation does not tune the retention policy on LongMemEval histories, questions, or support labels.

Episode and batch format. Training and evaluation share a single episode schema. Each episode contains the pre-query stream, the hidden query, the target answer, annotated support units when available, the retained-memory budget, and a coarse task type:

```
{
  "stream": [
    {"session_id": 1, "timestamp": "...", "text": "..."},
    {"session_id": 2, "timestamp": "...", "text": "..."}
  ],
  "hidden_query": "...",
  "answer": "...",
  "support_units": [...],
  "budget": 512 | 1024 | 2048 | 4096 | 8192,
  "task_type": "single-hop | multi-hop | update | temporal | abstention"
}
```

For the GRPO-style update in Section 3.6, the old-policy importance ratio for an updated token is

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid \mathrm{ctx}_{i,<t})}{\pi_{\theta_{\mathrm{old}}}(o_{i,t} \mid \mathrm{ctx}_{i,<t})}.$$

For the exposed memory-control tokens $\mathcal{T}_i$, we use the clipped objective

$$\mathcal{J}(\theta) = \mathbb{E}_{i,t\in\mathcal{T}_i}\Big[\min\Big(\rho_{i,t}(\theta)\hat{A}_i,\ \mathrm{clip}\big(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon\big)\hat{A}_i\Big) - \beta\,\mathrm{KL}\big(\pi_\theta \| \pi_{\mathrm{ref}}\big)\Big].$$

The reported veRL integration exposes one selected trainable decision turn per rollout to the PPO/-GRPO update, while the reward is computed from the full memory trajectory. Thus the implementation applies standard clipped GRPO to the exposed memory-control decision; it does not introduce a new optimizer.

Table 25: Training configuration for the reported EMBER runs.

Setting	**Value**
Stage I data	RULER-HotpotQA pre-query online-memory episodes
Stage II data	MuSiQue and 2WikiMultiHopQA multi-session retention episodes
Stage III data	Update, temporal, and abstention hard-case transformations
Stage I convergence	about 500 optimization steps
Stage II/III continuation	100 optimization steps
Learning rate	1×10^{-6}
PPO epochs	8
KL coefficient	0.001
Optimizer	GRPO-style outcome RL with clipped PPO update

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the setting, method, and measured gains under Budgeted Pre-Query Retention. The main quantitative claims are reported in Table 1, Figure 2, and Section 4.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 7 discusses the benchmark scope, annotation and budget assumptions, and deployment risks for persistent agent memory.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete and correct proof?

   Answer: [N/A]

   Justification: The paper is empirical and does not present formal theoretical results.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper?

   Answer: [Yes]

   Justification: Section 3 describes the retention policy, Section 3.6 gives the training objective, and Appendix C.4 defines the LongMemEval-RR protocol, metrics, budgets, and evaluation wrapper.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Appendix C.4 specifies the derived-protocol construction, retained-budget settings, chunk/span rules, answer-time wrapper, and evaluator. Appendix C.5 states that the anonymized supplementary artifact includes the construction scripts, evaluation commands, and wrappers.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 describes the benchmark hierarchy, main baselines, and reader settings. Appendix C.9 reports the training configuration, and Appendix C.4 reports the protocol and extended tables.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Main F1 tables define uncertainty in their captions: LongMemEval-RR reports bootstrap confidence half-widths, while the RULER-HotpotQA headline table reports 95% confidence-interval half-widths. The primary LongMemEval-RR gain reports a paired bootstrap confidence interval. Randomized retention probes average five seeds; extended deterministic diagnostics report point estimates.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources needed to reproduce the experiments?

   Answer: [Yes]

Justification: Appendix Table 17 summarizes the rollout engine, GPU-resident retriever, hard budget enforcement, and training interface. Appendix Table 18 reports the compute paths, operating points, and GPU configurations, including H200 SXM $\times 4$ and H100 NVL $\times 8$.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics?

   Answer: [Yes]

   Justification: The work uses public benchmarks and standard language-model evaluation protocols. It does not involve deception, private user-data collection, or human-subject intervention.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Section 7 discusses both the positive impact of reducing unnecessary memory/context and the risks of persistent source-evidence retention, including sensitive-history retention and user control.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse?

    Answer: [N/A]

    Justification: The paper does not release a new general-purpose pretrained language model or a high-risk scraped dataset.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets used in the paper properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Appendix Table 19 lists the main external assets, how they are used, and how upstream licenses or service terms are handled. The paper cites the creators of the datasets, model families, retrieval components, and baseline methods used in training and evaluation.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Appendix C.4 documents LongMemEval-RR, including its source data, construction rules, retained-budget settings, online ingest protocol, metrics, and evaluation wrapper. Appendix C.5 describes the anonymized supplementary artifact and upstream asset handling.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation?

    Answer: [N/A]

    Justification: The work does not conduct crowdsourcing or human-subject experiments.

15. **Institutional review board approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board approvals were obtained?

    Answer: [N/A]

    Justification: The work does not conduct human-subject research.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research?

    Answer: [Yes]

Justification: The paper describes the LLM-based memory writer, reader, retrieval prompts, Qwen backbones, GPT-4o reader evaluation, and prompt templates in the method, experiments, and appendices.