TOKENGATE: WRITE ADMISSION AND BUDGETED RETRIEVAL FOR LONG-TERM AGENT MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term memory enables language model agents to sustain continuity across interactions, yet accumulating histories impose escalating storage demands and recurrent token expenditure during inference. The central challenge is to regulate persistent memory growth while allocating a finite context budget to useful evidence. Conventional top-k retrieval imposes a cardinality constraint rather than an explicit token bound, leaving memory accumulation at write time unaddressed. We introduce TokenGate, a modular memory controller that couples learned admission control with query-conditioned evidence selection under a hard token budget. The admission model is trained on twelve structured views of each source candidate, while retrieval ranking combines question–memory relevance, semantic similarity, and a length penalty. On two LoCoMo test conversations with corrected inputs, filtering and learned admission reduce stored memory-body tokens by 41.7% relative to retaining all candidates, before offline maintenance. Compared with the store-all baseline (BASE), the full configuration with offline maintenance (FULL) has lower ordinary-question F1 in A-MEM and Letta and higher total counted LLM input and output tokens in Mem0 and Letta. In a separate comparison limited to Mem0 and Letta, with admitted stores and candidate pools fixed, reranking with speaker and observation-date metadata (RST) improves ordinary-question F1 over native ranking by 5.21 and 2.66 percentage points, respectively, within a 512-token memory budget. On 5,000 teacher-labelled admission examples derived from LongMemEval, twelve-view training achieves an AUROC of .823, compared with .811 for a matched native-example repetition control. The results support separate control of stored text and evidence selection within a fixed retrieval budget, with backend-dependent effects on answer quality and counted LLM-token cost.

1 INTRODUCTION

Language model agents that operate across multiple sessions must recover information that is no longer present in the active context. Personal assistants, for example, need access to user preferences, earlier decisions, and evolving task constraints. Persistent memory provides this continuity by retaining information outside the model's immediate context and retrieving it when needed (Packer et al., 2023; Chhikara et al., 2025). As interaction histories accumulate, memory management becomes a resource-allocation problem: the system must determine both what deserves continued storage and what warrants inclusion in subsequent prompts.

This resource burden has two distinct components. At write time, retaining redundant or transient records expands the memory store, even when those records never support a later response. At read time, retrieved memories consume prompt tokens whenever they are supplied to the language model, creating a recurring cost across interactions. A larger context window relaxes the input limit but does not determine whether retaining or reusing a memory is worthwhile. We therefore focus on *reducing retained-memory tokens and explicitly bounding the memory tokens supplied to each query*.

Controlling these costs is difficult because admission and retrieval operate under different information conditions. Admission must assess a candidate before future queries are known, so an apparently minor detail may later become essential. Retrieval can condition on the current question, but

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.

must select among records with different lengths and overlapping relevance. A fixed top-k constrains record cardinality without enforcing a precise token ceiling. Furthermore, relevance to an individual benchmark question need not coincide with value for long-term retention. Effective memory control must therefore distinguish admission value, query relevance, and token cost.

Existing research offers several complementary mechanisms. MemGPT manages information across memory tiers, Mem0 extracts and consolidates conversational information, and A-MEM organizes memories through linked notes and memory evolution (Packer et al., 2023; Chhikara et al., 2025; Xu et al., 2025). Prompt compression methods such as LLMLingua-2 reduce the text supplied to a model (Pan et al., 2024). A-MAC explicitly models admission using interpretable factors and an LLM-assisted utility assessment (Zhang et al., 2026a). These approaches establish memory efficiency as an active research problem, while addressing different decisions within the memory lifecycle.

The design question we address is how to connect local admission decisions with direct token control at retrieval. Compressing selected text does not by itself specify which records should enter persistent storage, while admission alone does not determine how a later query should use its context budget. Repeated generative judgments can also add inference work to the memory-management process. Our approach makes the write decision and the retrieval constraint explicit at the interfaces of an existing memory backend.

We propose **TokenGate**, a modular controller combining a trained admission classifier with budgeted evidence selection. Its admission component, denoted **A1**, is our lightweight binary memory classifier, obtained by adapting Qwen2.5-0.5B with low-rank adaptation (LoRA) (Yang et al., 2024; Hu et al., 2022). We train this model on augmented candidates from the LoCoMo training split (Maharana et al., 2024) and select its operating threshold using development data. Given a candidate's text, the model produces a keep score that determines whether the candidate is admitted to storage. This formulation enables a local classification decision without generating a free-form assessment for every candidate.

At retrieval time, TokenGate ranks the backend's candidate memories using question–memory relevance, semantic similarity, and a length penalty, then selects complete records subject to a hard token budget. Token accounting applies to the fully rendered evidence, including metadata and separators. Our separate Mem0/Letta reranking study uses the pretrained Qwen3-Reranker-0.6B with speaker and observation-date metadata, without additional fine-tuning (Zhang et al., 2025). Admission determines what remains available, while budgeted retrieval determines how that evidence uses the available context.

We evaluate TokenGate through corrected-input integrations with Mem0, A-MEM, and Letta, and through a separate controlled reranking study on the same two LoCoMo test conversations. Across the three backend integrations, filtering and learned admission reduce stored memory-body tokens by 41.7% before maintenance. In the separate Mem0/Letta comparison, admitted stores and candidate pools are fixed: reranking with speaker and observation-date metadata improves ordinary-question F1 over native ranking by 5.21 and 2.66 percentage points, respectively, under the same 512-token memory budget. These comparisons separately evaluate storage reduction and the use of admitted evidence, while the full-system results expose backend-dependent cost and quality trade-offs.

Our contributions are threefold:

- **Learned admission for selective storage.** We develop and evaluate a lightweight admission classifier that converts candidate text into a retention decision before persistent storage.
- **Explicit token control at retrieval.** We integrate query-conditioned ranking with evidence selection that enforces a hard budget on the rendered memory context.
- **Structured multi-view training for memory admission.** We develop a twelve-view augmentation protocol that varies candidate length, core-text position, and role presentation to train a lightweight memory-admission classifier. Under matched training budgets, this protocol improves admission AUROC over a native-example repetition baseline in an exploratory evaluation using teacher-labelled admission examples derived from LongMemEval.

2 Related Work

Persistent memory architectures and learned management. Persistent-memory systems organize interaction histories for reuse across sessions. Generative Agents combines experience retrieval with higher-level reflection (Park et al., 2023), while A-Mem and MemTree develop interconnected memory notes and hierarchical memory representations, respectively (Xu et al., 2025; Rezazadeh et al., 2025). Beyond memory organization, Memory-R1 learns memory-management operations from downstream task feedback (Yan et al., 2026), and AgeMem unifies long-term memory operations and short-term context management through reinforcement learning (Yu et al., 2026). TokenGate focuses on the storage and recurring prompt-token costs at the write and retrieval interfaces. It formulates write admission as a supervised binary decision, using a separately trained, lightweight classifier to regulate which candidates enter persistent storage.

Retrieval and compression under context budgets. Retrieval and compression determine which external evidence reaches a language model and how much context it consumes. Self-RAG learns on-demand retrieval and reflection over retrieved evidence and generated responses (Asai et al., 2024). RECOMP trains extractive and abstractive compressors to shorten retrieved context and selectively omit unhelpful information (Xu et al., 2024), while LongLLMLingua applies question-aware, coarse-to-fine prompt compression (Jiang et al., 2024). TokenGate performs selection at the candidate-record level and enforces a hard token budget on the final rendered memory evidence, including metadata and separators. This constraint complements relevance-based ranking by explicitly accounting for the variable token costs of retrieved records.

Structured augmentation and multi-view training. Structured augmentation exposes learning algorithms to controlled variations beyond the original training examples. Unsupervised Data Augmentation combines text transformations with consistency training on unlabeled data (Xie et al., 2020), while CoDA combines complementary transformations with consistency and contrastive regularization (Qu et al., 2021). For long-context learning, IN^2 varies the positions of supporting information and balances context lengths (An et al., 2024). TokenGate applies structured input variation to memory admission: its twelve-view protocol combines each native example with controlled variations in candidate length, core-text position, and role presentation, while retaining the original core text and admission label. These views provide supervised training examples for a binary classifier that governs selective storage.

Positioning relative to LightMem. LightMem is particularly close to TokenGate in its objective of reducing the cost of persistent memory. It combines token-level pre-compression, topic-aware memory construction, and offline sleep-time consolidation (Fang et al., 2026). TokenGate places explicit control at the write and retrieval interfaces: learned admission determines which candidate records enter the store, and budgeted selection determines which retrieved records enter the memory context. These two operations act on complete candidates and preserve the text of selected records during admission and retrieval. Our full configuration also incorporates offline maintenance inspired by LightMem. The distinction therefore lies in combining candidate-level admission with an explicit retrieval-token budget, complementing memory consolidation with direct control over retained-memory volume and recurring prompt-token consumption.

3 Methodology

3.1 Overview

TokenGate controls a memory backend's write and retrieval interfaces (Figure 1). Stages A–B filter and score candidate body x_t to authorize writing; source metadata supports provenance rather than supplying recent-dialogue context to A1. Stage C stores admitted records, and stage D selects evidence $S(q) \subseteq R(q)$ under budget B. FULL also uses LightMem-inspired offline maintenance: a local generator proposes ignore/update/delete operations, and checks reject invalid proposals before updating the store and source mappings (Fang et al., 2026). Detailed maintenance and interface semantics appear in Appendix A.

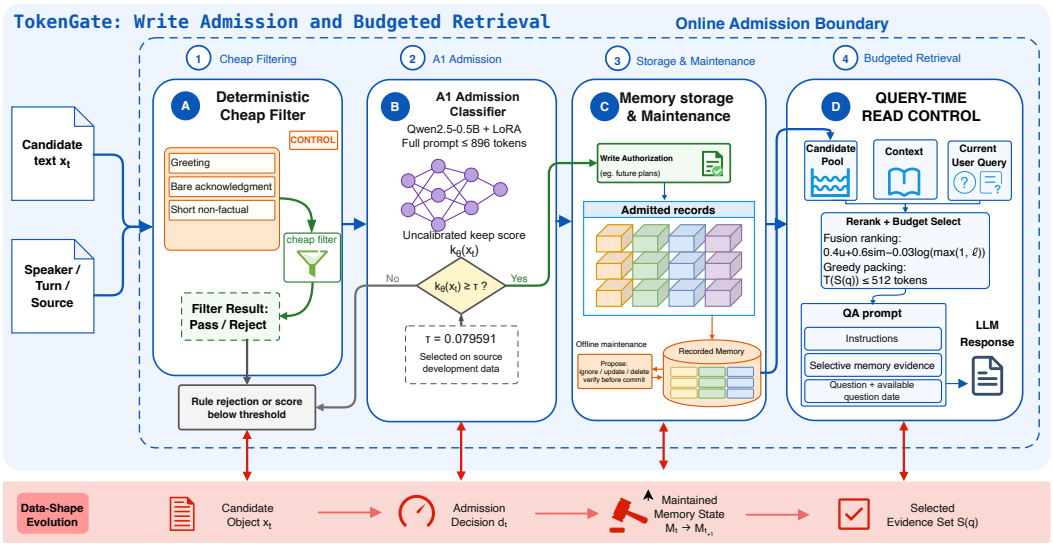

Figure 1: TokenGate workflow: filtering (A), learned write admission (B), storage with offline maintenance in FULL (C), and query-time ranking and budgeted selection (D). The 512-token ceiling applies to rendered memory evidence. Eq. 1 defines write admission; Algorithm 1 specifies training.

3.2 Write Admission

A conservative predicate $g(x_t)$ rejects greetings, bare acknowledgements, and short non-factual turns. Passing candidates enter A1's fixed task frame without recent dialogue. A1 returns an uncalibrated keep score $k_\theta(x_t) \in [0, 1]$ for the admission decision. Let $g(x_t) = 1$ denote passage through the cheap filter. The complete write decision is

$$d_t = \mathbb{I}\{g(x_t) = 1\}\mathbb{I}\{k_\theta(x_t) \geq \tau\}. \tag{1}$$

A1 adapts Qwen2.5-0.5B with LoRA and a classification head (Yang et al., 2024; Hu et al., 2022), training 1.08M parameters (0.22%) with cross-entropy. It takes median view-level logit margins within each seed, averages margins across seeds 42/43/44, then applies a sigmoid. The source-development threshold is frozen at $\tau = 0.079591$. The threshold is selected on source development data and is not derived from a calibrated risk model. The full prompt is capped at 896 tokens; overlength inputs use overlapping windows, distinct from the training augmentation below. All 788 current LoCoMo candidates fit one window. LoRA configuration and window handling are detailed in Appendix A.

3.3 Structured Multi-View Training

A1 is trained on teacher-labelled source admission examples. These training labels are distinct from the separately reviewed 788-candidate LoCoMo admission diagnostic. For source candidates $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, with binary discard/keep labels, we construct

$$\mathcal{D}_{\text{view}} = \{(T_j(x_i), y_i) : 1 \leq i \leq N,\ 1 \leq j \leq 12\}. \tag{2}$$

Each transformation preserves the labelled core verbatim. The twelve views comprise one native view, nine mixtures crossing nominal lengths 240/480/750 with head/middle/tail core placement, and two role-prefix views (`[user]`, `[assistant]`). Mixtures use discard-labelled distractors from other training conversations; original train/dev assignments remain unchanged. Each view is a separate supervised example, without a consistency or contrastive loss. The 4,352 training bases yield 52,224 rows. A matched control repeats native examples twelve times to separate input diversity from training volume. Algorithm 1 specifies construction, optimization, and source-development selection.

Algorithm 1 Structured multi-view construction and A1 training

Require: Source train/dev sets $\mathcal{D}_{\mathrm{tr}}, \mathcal{D}_{\mathrm{dev}}$ with candidate, label, conversation, and ID
Require: Tokenizer $\mathcal{T}$, task frame F, prompt cap $L = 896$, seeds $\mathcal{S} = \{42, 43, 44\}$
Ensure: Three source-selected A1 adapters $\{\theta_s^*\}_{s\in\mathcal{S}}$

$\mathcal{P} \leftarrow \{(x, y, c, h) \in \mathcal{D}_{\mathrm{tr}} : y = 0\}$ ▷ Training-only distractors
for $b \in \{\mathrm{tr}, \mathrm{dev}\}$ **do**
  $\mathcal{D}_b^+ \leftarrow [\,]$
  for all $(x_i, y_i, c_i, h_i) \in \mathcal{D}_b$ **do**
    $\mathcal{V}_i \leftarrow [x_i]$; $\mathcal{P}_i \leftarrow \{z \in \mathcal{P} : c(z) \neq c_i\}$
    for all $(\ell, p) \in \{240, 480, 750\} \times \{\mathrm{head}, \mathrm{middle}, \mathrm{tail}\}$ **do**
      $\mathcal{N} \leftarrow \textsc{DeterministicShuffle}(\mathcal{P}_i, h_i, \ell, p)$; $Z \leftarrow [\,]$
      repeat
        Append the next nonempty candidate from $\mathcal{N}$ to Z
      until $|\mathcal{T}(\textsc{Join}(Z + [x_i]))| \geq \ell$
      $v \leftarrow \textsc{PlaceCore}(x_i, Z, p)$ ▷ Core at head, midpoint, or tail
      Append v to $\mathcal{V}_i$
    end for
    Append `[user]`$\|x_i$ and `[assistant]`$\|x_i$ to $\mathcal{V}_i$
    for all $v \in \mathcal{V}_i$ **do**
      Assert x_i occurs verbatim in v and $|\mathcal{T}(F(v))| \leq L$
      Append $(F(v), y_i, h_i)$ to $\mathcal{D}_b^+$ ▷ Inherit label and base ID
    end for
  end for
end for
for all $s \in \mathcal{S}$ **do**
  Initialize seed s, frozen Qwen2.5-0.5B, LoRA adapters, and classification head
  Train one shuffled epoch on $\mathcal{D}_{\mathrm{tr}}^+$ with cross-entropy and AdamW
  Use learning rate 10^{-4}, microbatch 1, and gradient accumulation 16
  Save adapters $\theta_{s,t}$ at optimizer steps $t \in \{1632, 3264\}$
  for all saved checkpoints t **do**
    $m_i^{s,t} \leftarrow \mathrm{median}_{j=1}^{12}\ \textsc{LogitMargin}(\theta_{s,t}, F(T_j(x_i)))$ for dev bases i
    $a_{s,t} \leftarrow \mathrm{AUROC}(\{(y_i, m_i^{s,t})\}_{i\in\mathcal{D}_{\mathrm{dev}}})$
  end for
  $\theta_s^* \leftarrow \textsc{SelectDevCheckpoint}(\{(\theta_{s,t}, a_{s,t})\}_t, \epsilon = 0.002)$
end for
return $\{\theta_s^*\}_{s\in\mathcal{S}}$ ▷ Freeze before downstream evaluation

3.4 Budgeted Evidence Selection

The backend returns up to 50 native-ranked candidates $R(q)$. Let z_i indicate selection of record m_i, $r_i(q)$ its relevance score, and $T(S)$ the token count of fully rendered, joined evidence, including metadata and separators. The selection objective is

$$\max_{\mathbf{z}\in\{0,1\}^{|R(q)|}} \sum_i z_i r_i(q) \quad \text{s.t.} \quad T(\{m_i : z_i = 1\}) \leq B. \tag{3}$$

The budget covers memory evidence; QA separately adds instructions, the question, and an available question date, without few-shot examples. Relevance is a QA-quality surrogate. The implementation approximates this objective by deterministic greedy packing, skipping non-fitting records and retokenizing each proposed concatenation; it is not an exact optimizer (Jiang et al., 2025). Both ranking studies use $B = 512$ and the same fusion rule:

$$r_i(q) = 0.4u_\phi(m_i, q) + 0.6\,\mathrm{sim}(m_i, q) - 0.03\log(\max(1, \ell_i))\,. \tag{4}$$

Here u_ϕ estimates evidence usefulness, sim is recomputed query–memory cosine similarity, and ℓ_i counts displayed-record tokens. The original utility is TF–IDF logistic regression over question and memory; the separate corrected-input reranking study substitutes pretrained Qwen3-Reranker-0.6B's sigmoid yes/no logit difference without fine-tuning (Zhang et al., 2025). The three-backend module ablation retains the lexical utility.RST supplies unchanged text and captions plus speaker and observation-date fields; observation dates need not be event dates. These fields already appear in QA evidence, so the intervention changes reranker inputs. Cosine scores, coefficients, and length penalties remain fixed; ties follow native rank. Scorer replacement also changes score distributions,

and the additive rule does not model interactions among selected records. Reranker computation is accounted separately from memory-token savings (Appendix D).

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

LoCoMo uses seven training, one development, and two test conversations (Maharana et al., 2024). A1 uses 4,352 source training bases with twelve views per base and 676 development bases. Its weights and admission threshold remain frozen across input versions. Admission prompts are capped at 896 tokens; retrieval packs up to 50 backend candidates into a 512-token memory-evidence budget. QA uses Qwen2.5-7B at temperature zero.

We distinguish three LoCoMo evaluations. The completed corrected-input three-backend rerun provides the storage, cost, and module results in Tables 1 and 4. It restores image captions and source dates and evaluates configurations with and without maintenance. A separate corrected-input reranking study evaluates Mem0 and Letta with maintenance disabled and admitted stores and candidate pools held fixed (Appendix D). Original converted-input results are retained separately as historical evidence. Full training, runtime, and interface settings appear in Appendix A.1 and Table A1.

4.2 BASELINES AND CONTROLLED COMPARISONS

Admission comparisons use TF–IDF logistic regression, the native-repeat-12 control C0, and predecessor P0. C0 matches A1's architecture, training-row count, prompt limit, seeds, and optimization settings, while repeating native examples without view transformations. P0 is the earlier source-trained admission model used in the historical transfer diagnostics.

BASE stores all candidates and uses native ranking under the memory budget. FULL enables filtering, admission, lexical fusion, and maintenance. The module-removal and maintenance-only conditions are defined in Table 4. Throughout this ablation, R denotes lexical fusion, not RST.

The separate reranking study compares native ranking (N), lexical fusion (H), fusion with the utility term removed (D), and Qwen3-based fusion without (R0) or with (RST) speaker and observation-date fields. Within this comparison, admitted stores, candidate pools, answer-generation settings, and token budgets are fixed. H and additional metadata controls are reported in Appendix D. Historical RHELM uses 15 random-storage controls at approximately matched retained-token budgets (Table A2).

4.3 EVALUATION TASKS AND METRICS

The LoCoMo test evaluation contains 788 candidates and 304 questions from conv-26 and conv-30. Ordinary-question F1 and valid-answer rate use 233 questions, evidence coverage uses 231 eligible ordinary questions, adversarial metrics use 71 questions, and format error uses all 304 questions. A valid answer is schema-correct and non-abstaining; validity does not imply semantic correctness. Evidence coverage is the per-question fraction of reference evidence IDs represented in the selected memory.

Corrected-input quality metrics are averaged within each conversation and then equally across the two conversations. In the three-backend rerun, A-MEM is first averaged across three backend write seeds at the question level; Mem0 and Letta each use one backend repeat. The separate Mem0/Letta reranking study has one logical repeat. These backend repeats are distinct from the three-seed A1 ensemble, and cached answers are not independent QA replications.

Both corrected-input studies use the same reference-aware adversarial-scoring protocol: 69 questions require abstention, whereas `conv-26:qa:167` and `conv-26:qa:178` have the reference answer "No" in the source data. Credit requires a schema-correct output with `abstain=true` for the 69 abstention questions, or a schema-correct, non-abstaining output whose normalized answer exactly matches "No" for the two answerable questions. For the completed three-backend rerun, we rescore the saved outputs under this common rule; the original archived evaluation treated all 71 questions as requiring abstention. No questions or model outputs are changed. This score measures

compliance with the specified reference and output protocol, rather than independently assessed semantic correctness.

For the exploratory LongMemEval diagnostic, we use 5,000 derived admission examples with fixed DeepSeek-V4-Flash keep/discard labels, rather than the benchmark's original QA task (Wu et al., 2025). No target examples or labels are used to fit A1 or select its checkpoint. Historical RHELM evaluates rank-budget retention and BM25 QA on 864 conversation-answerable questions (Zhang et al., 2026b). Task-specific counts and controls appear in Table A3.

Storage counts memory-body tokens, whereas retrieval counts fully rendered evidence, including metadata and separators. LLM input and output tokens are accounted separately from admission, embedding, and reranker computation. Original LoCoMo intervals use 10,000 candidate/question resamples conditional on two conversations; historical RHELM uses persona-cluster resampling. Full accounting and admission latency appear in Appendices A–C.

5 RESULTS

5.1 STORAGE EFFICIENCY AND ANSWER RELIABILITY

NO_M reduces records by 55.20% and memory-body tokens by 41.73% (Table 1). FULL adds 46,091 maintenance tokens for two updates (+53 stored tokens) and no deletions. Versus BASE, its F1 changes by +0.92/-1.93/-4.20 percentage points in Mem0/A-MEM/Letta, and total counted LLM tokens increase in Mem0/Letta.

On reviewed original-input LoCoMo, A1 has no established AUROC advantage over TF–IDF and worse Brier error (Table A4).

Table 1: Storage and LLM-token costs at $B = 512$, with and without offline maintenance. Additional answer-quality metrics appear in Table 4.

				LLM input + output tokens (10^3)			
Backend	Setting	F1	Stored tokens	Native	Maint.	QA	Total
Mem0	BASE	.1689	29,447	0.0	0.0	219.0	219.0
	NO_M	.1831	17,159	0.0	0.0	217.7	217.7
	FULL	.1781	17,212	0.0	46.1	217.7	263.8
A-MEM	BASE	.1922	29,447	1,928.4	0.0	218.5	2,146.9
	NO_M	.1769	17,159	953.1	0.0	217.8	1,170.9
	FULL	.1729	17,212	951.6	46.1	217.7	1,215.4
Letta	BASE	.2277	29,447	0.0	0.0	218.3	218.3
	NO_M	.1908	17,159	0.0	0.0	217.9	217.9
	FULL	.1857	17,212	0.0	46.1	217.9	264.0

Stored tokens sum memory bodies over both conversations. LLM costs comprise native memory processing, maintenance proposals/verification, and 304-question QA; cached calls retain logical costs. Admission, embeddings, and non-LLM ranking are excluded. A-MEM resources average three workloads (Appendix B).

5.2 EFFECT OF STRUCTURED MULTI-VIEW TRAINING

On 5,000 teacher-labelled LongMemEval examples, A1 improves AUROC over matched native-repeat-12 control C0 by .012 (95% CI $[.004, .019]$; Table 2). This is an exploratory admission diagnostic; matched controls and label differences from LoCoMo are in Appendix E.1.

5.3 BUDGETED RETRIEVAL WITH METADATA-AWARE RERANKING

With fixed stores, pools, answerer, $B = 512$, and no maintenance, RST improves F1 over native ranking by 5.21/2.66 points in Mem0/Letta, the only backends tested, and coverage by 17.10/7.63 points (Table 3). F1 improves in all four backend–conversation pairs. Speaker/date inputs improve

Table 2: Exploratory LongMemEval AUROC (frozen three-seed ensembles). C0 repeats native inputs; A1 uses twelve views. The historical 256-token truncation flag covers candidate frames with prompt overhead, not body length or A1's 896-token cap. Each length/role partition covers all 5,000 examples.

Evaluation stratum	n	P0	C0: repeat-12	A1: views-12
Overall	5,000	.801	.811	**.823**
Historical candidate frame $>$ 256 tokens	1,649	.554	.586	**.644**
Historical candidate frame $\leq$ 256 tokens	3,351	.845	.856	**.858**
User role	2,505	.838	.847	**.853**
Assistant role	2,495	.620	.628	**.676**

F1 over body-only R0 by 2.22/3.05 points. Other metadata/score controls report coverage only (Table A10).

RST's adversarial scores fall by 3.19/6.43 points versus native ranking, with declines confined to conv-26; semantic interpretation is limited by the protocol and unblinded review (Appendix D.1).

Table 3: Retrieval at $B = 512$: means over two equally weighted conversations, one logical repeat. Coverage: 231 eligible questions; F1/EM/Ord. valid: 233 ordinary; adversarial: 71. Accuracy uses the shared 69-abstention/two-"No" protocol; validity denotes schema-correct, non-abstaining output. Bold marks accuracy/coverage maxima.

Backend	Ranker	Coverage	F1	EM	Ord. valid	Adv. acc	Adv. valid
Mem0	N	.4070	.2001	.0794	.7223	.6622	.3484
	D	.3521	.1692	.0539	.6393	**.7451**	.2762
	R0	.4907	.2300	.0991	.7643	.5771	.4229
	RST	**.5780**	**.2522**	**.1049**	.8186	.6303	.3803
Letta	N	.5159	.2339	.0868	.8026	.6729	.3484
	D	.3636	.1795	.0415	.6837	**.7349**	.2863
	R0	.5081	.2300	.0769	.7676	.6613	.3599
	RST	**.5922**	**.2605**	**.0983**	.8441	.6086	.4127

5.4 Module Ablations

NO_R has the highest mean F1 among the seven configurations (+1.90/2.91/4.73 points versus FULL in Mem0/A-MEM/Letta), with higher ordinary validity and coverage. It is a QA-oriented alternative when maintenance is retained: stores, pools, and 46,091 maintenance LLM tokens per workload match FULL, with nearly unchanged total LLM tokens. Adversarial accuracy changes by -1.06/-5.20/+3.19 points and format error by +0.98/-0.16/+1.43 points. Mem0/Letta F1 falls by 0.49/0.16 points on conv-30. These two-conversation results do not establish global optimality.

NO_A consistently lowers F1 and ordinary validity; NO_M improves mean F1 and coverage but worsens A-MEM validity and format error. All paired trade-offs remain descriptive (Appendix B.1).

6 Discussion and Limitations

The results separate three effects that should be evaluated individually. Admission reduces the amount of stored text; reranking changes how a fixed admitted store is used within the retrieval budget; and maintenance adds processing cost without observed record deletion in these workloads. Their quality effects depend on the backend and scoring protocol. This separation explains why storage savings, ordinary QA, and adversarial-protocol scores need not improve together. The controlled reranking study reuses two previously analyzed LoCoMo test conversations and evaluates Mem0 and Letta with maintenance disabled; cached replays are not independent QA trials. RST improves ordinary-question F1 and evidence coverage but scores below native ranking under the frozen adversarial protocol, with the mean declines concentrated in conv-26 for both backends. For

Table 4: Corrected-input module ablations at a 512-token retrieval budget: ordinary quality, adversarial behaviour, and format reliability.

Condition	F	A	R	M	Backend	F1	Ord. valid	Adv. acc.	Adv. valid	Evidence coverage	Format error
					Mem0	0.1689	0.6302	0.7150	0.2957	0.3666	0.0050
BASE	–	–	–	–	A-MEM	0.1922	0.7204	0.6237	0.3800	0.4545	0.0104
					Letta	0.2277	0.7701	0.6631	0.3582	0.5176	0.0171
					Mem0	0.1781	0.6232	0.6729	0.3378	0.3081	0.0101
FULL	+	+	+	+	A-MEM	0.1729	0.6410	0.6933	0.3138	0.3324	0.0236
					Letta	0.1857	0.6537	0.6410	0.3803	0.3223	0.0101
					Mem0	0.1781	0.6294	0.6729	0.3378	0.3143	0.0101
NO_F	–	+	+	+	A-MEM	0.1729	0.6472	0.6933	0.3138	0.3386	0.0236
					Letta	0.1857	0.6598	0.6410	0.3803	0.3285	0.0101
					Mem0	0.1558	0.5627	0.7252	0.2961	0.3196	0.0000
NO_A	+	–	+	+	A-MEM	0.1604	0.6118	0.7038	0.2998	0.3473	0.0104
					Letta	0.1439	0.5393	0.6729	0.3484	0.3209	0.0073
					Mem0	0.1971	0.7162	0.6622	0.3484	0.4008	0.0198
NO_R	+	+	–	+	A-MEM	0.2020	0.7319	0.6413	0.3660	0.4767	0.0220
					Letta	0.2330	0.8026	0.6729	0.3484	0.5159	0.0243
					Mem0	0.1831	0.6294	0.6729	0.3378	0.3173	0.0101
NO_M	+	+	+	–	A-MEM	0.1769	0.6367	0.7002	0.3033	0.3445	0.0318
					Letta	0.1908	0.6598	0.6410	0.3803	0.3316	0.0101
					Mem0	0.1702	0.6335	0.7150	0.2957	0.3616	0.0050
M_ONLY	–	–	–	+	A-MEM	0.1923	0.7236	0.6447	0.3590	0.4579	0.0122
					Letta	0.2236	0.7701	0.6631	0.3582	0.5083	0.0171

Each metric lists Mem0/A-MEM/Letta. F/A/R/M are filtering/ admission/lexical fusion/maintenance; a dash disables a module. R is lexical fusion, not RST; native ranking keeps $B = 512$. F1/ordinary validity use 233 questions, coverage 231, adversarial metrics 71, and format error all 304. Adversarial accuracy is rescored from saved outputs under Table 3's 69-abstention/two-"No" protocol. Validity is schema-correct, non-abstaining output and can include a correct "No". A-MEM averages three write seeds per question before within-conversation aggregation and equal conversation weighting.

69 of 71 adversarial questions, scoring depends on schema correctness and the abstention flag without independently assessing answer text. Unblinded case review identifies flag–text inconsistencies and potential answerability-label conflicts alongside genuine attribution errors; it does not establish semantic adversarial accuracy or causal mechanisms (Appendix D.1).

LongMemEval provides exploratory admission evidence; historical RHELM uses a separate retention protocol and does not establish full-system transfer. Results from different input versions and evaluation protocols are reported separately. Memory-body savings exclude model and index storage and do not measure total computation. Offline maintenance is accounted for separately: it incurs additional LLM cost and produces no record deletions in the evaluated workloads. Broader evaluation and independent human review of QA and maintenance are needed (Appendix A).

7 Conclusion

TokenGate combines learned admission with metadata-aware evidence selection under a hard token budget. Across three memory backends, admission retains 353 of 788 candidate records and reduces memory-body tokens by 41.7% before offline maintenance. Compared with BASE, FULL with offline maintenance has lower ordinary-question F1 in A-MEM and Letta and higher total counted LLM input and output tokens in Mem0 and Letta. In a separate controlled comparison limited to Mem0 and Letta, with admitted stores and candidate pools fixed, RST improves ordinary-question F1 over native ranking by 5.21 and 2.66 percentage points, respectively, under a 512-token memory budget. Twelve-view training improves admission AUROC over matched native repetition in the exploratory LongMemEval diagnostic. These results support complementary control of memory retention and evidence selection, with backend-dependent quality effects and protocol-dependent adversarial scores.

AI USE STATEMENT

LLMs are components of the studied memory pipelines, including answer generation, maintenance proposals, and auxiliary automatic evaluation. The admission-label ledger records 588 LLM-prelabel/human-review entries and 200 blind-human entries. Historical RHELM reference labels also use LLM annotation, as detailed in Appendix E.3. AI writing and coding assistance supported manuscript revision, statistical checks of existing records, results interpretation, and figure preparation. This manuscript update incorporates the separately completed paired-reranking study; the editing itself did not rerun models. Recorded label review is distinct from independent human QA/maintenance validation, which remains pending. AI assistance does not replace author responsibility for the final text, evidence, and claims.

ETHICS STATEMENT

We use public LoCoMo, LongMemEval, and RHELM benchmark material. The annotation ledger adds binary labels and provenance; third-party content remains subject to its original terms. Persistent agent memory can retain sensitive user information, and selective admission is not a privacy guarantee. Automatic maintenance checks do not establish semantic safety. We disclose input-conversion defects and incomplete independent human validation to limit interpretation of the results.

REPRODUCIBILITY STATEMENT

Appendix A specifies the frozen A1 configuration, data counts, metric denominators, paired intervals, and budget sensitivity. Appendix C defines the microbenchmark boundary; Appendix E.3 describes the historical RHELM protocol. Current results are bound to existing execution, model, input, per-question, and token ledgers. Historical diagnostic materials describe predecessor protocols and do not by themselves reproduce the current FULL system. The dated sensitivity supplement preserves the original statistics, script, and source hashes (Appendix F.3). Appendix D adds the completed reranking protocol and results. The source bundle includes a compact result/provenance record, `reranker_evidence_20260922.json`; full per-question artifacts remain in the research archive. No complete code release or independent human validation is claimed.

REFERENCES

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. Make your LLM fully utilize the context. In *Advances in Neural Information Processing Systems*, volume 37, pp. 62160–62188, 2024. doi: 10.52202/079017-1986. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/71c3451f6cd6a4f82bb822db25cea4fd-Abstract-Conference.html.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, pp. 9112–9141, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/25f7be9694d7b32d5cc670927b8091e1-Abstract-Conference.html.

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*, volume 413 of *Frontiers in Artificial Intelligence and Applications*, pp. 2993–3000. IOS Press, 2025. doi: 10.3233/FAIA251160. URL https://doi.org/10.3233/FAIA251160.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. LightMem: Lightweight and efficient memory-augmented generation. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a05b72653ec5b473732129829ae04195-Abstract-Conference.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Con-*

ference on Learning Representations, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1658–1677. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.91. URL https://aclanthology.org/2024.acl-long.91/.

Yi Jiang, Sendong Zhao, Jianbo Li, Haochun Wang, and Bing Qin. GainRAG: Preference alignment in retrieval-augmented generation through gain signal synthesis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10746–10757. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.527. URL https://aclanthology.org/2025.acl-long.527/.

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. URL https://aclanthology.org/2024.acl-long.747/.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems, 2023. URL https://arxiv.org/abs/2310.08560.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 963–981. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.57. URL https://aclanthology.org/2024.findings-acl.57/.

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

Yanru Qu, Dinghan Shen, Yelong Shen, Sandra Sajeev, Weizhu Chen, and Jiawei Han. CoDA: Contrast-enhanced and diversity-promoting data augmentation for natural language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Ozk9MrX1hvA.

Alireza Rezazadeh, Zichao Li, Wei Wei, and Yujia Bao. From isolated conversations to hierarchical schemas: Dynamic tree memory representation for LLMs. In *International Conference on Learning Representations*, pp. 990–1023, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/0382cb76309820f71c6eacd47b36ce71-Abstract-Conference.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 *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/d813d324dbf0598bbdc9c8e79740ed01-Abstract-Conference.html.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6256–6268, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/44feb0096faa8326192570788b38c1d1-Abstract.html.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. RECOMP: Improving retrieval-augmented LMs with context compression and selective augmentation. In *International Conference on Learning Representations*, pp. 43478–43502, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/bda88ed2892f5e61c9a9bf215c566913-Abstract-Conference.html.

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. 17577–17604. Curran Associates, Inc., 2025. doi: 10.52202/085713-0593. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/19909c36f51abc4856b4560aff3d36d6-Abstract-Conference.html.

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/.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen2.5 Technical Report, 2024. URL https://arxiv.org/abs/2412.15115.

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. URL https://aclanthology.org/2026.acl-long.981/.

Guilin Zhang, Wei Jiang, Xiejiashan Wang, Aisha Behr, Kai Zhao, Jeffrey Friedman, Xu Chu, and Amine Anoun. Adaptive memory admission control for LLM agents. In *ICLR 2026 Workshop on Memory for LLM-Based Agentic Systems (MemAgents)*, 2026a. URL https://openreview.net/forum?id=mmdqUrEY24. Also available as arXiv:2603.04549.

Han Zhang, Zihao Tang, Xin Yu, Xiao Liu, Yeyun Gong, Haizhen Huang, Yan Lu, Weiwei Deng, Feng Sun, Qi Zhang, and Hanfang Yang. Beyond static dialogues: Benchmarking realistic, heterogeneous, and evolving long-term memory. *arXiv preprint arXiv:2605.31086*, 2026b. URL https://arxiv.org/abs/2605.31086.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025. doi: 10.48550/arXiv.2506.05176. URL https://arxiv.org/abs/2506.05176.

A A1 Configuration and Measurement Details

A.1 Experimental Configuration and Comparisons

The corrected-input three-backend rerun supplies Tables 1 and 4; Appendix B specifies its scoring and resource accounting. The separate Mem0/Letta reranking study fixes stores and candidate pools and disables maintenance (Appendix D). Original-input admission and budget diagnostics retain their own input version and aggregation. A1 weights and the admission threshold are frozen across these evaluations. FULL uses lexical fusion and maintenance; RST denotes the separate pretrained pair reranker with speaker/date inputs.

Table A1: Experimental setup. The 896-token admission limit and 512-token retrieval budget apply to different stages. Full runtime settings are in Appendix A.3.

Component	Configuration
Source data	4,352 training bases and 676 development bases; twelve views per base
Admission model	Qwen2.5-0.5B; LoRA rank 8, scale 16, dropout 0.05; 1.08M trainable parameters
A1 optimization	52,224 rows; one epoch; AdamW, learning rate 10^{-4}; microbatch 1; accumulation 16
Model selection	Seeds 42/43/44; source-development base-median AUROC; fixed admission threshold 0.079591
Retrieval	Up to 50 native candidates; $B = 512$ tokens; $\alpha = 0.4$, $\lambda = 0.03$; historical original-input sensitivity at 256/1,024 tokens
Pair reranker	Pretrained Qwen3-Reranker-0.6B without task fine-tuning; batch 1; 2,048-token input cap
QA generation	Qwen2.5-7B via Ollama; temperature 0; 8,192-token context; 256-token answer cap
Runtime	NVIDIA RTX 4050 Laptop, 6GB; PyTorch 2.6.0+cu124; frozen model revisions
Memory backends	Mem0 0.1.118, pinned A-MEM implementation, and Letta 0.9.1

Table A2: Baselines and controls. Results are compared within the indicated task and input version. RST is the pair reranker supplied with speaker and observation-date metadata.

Task	Method	Controlled comparison
LoCoMo admission	TF–IDF logistic gate	Lexical versus learned admission on the same reviewed labels and cheap-filter mask
Training diversity	C0: native-repeat-12	Same model, row count, prompt limit, seeds, and training settings; no view transformations
Transfer diagnostic	P0: predecessor gate	Earlier source-trained admission model evaluated on LongMemEval
Backend integration	BASE / FULL	Store-all plus native ranking versus filtering, admission, fusion, and maintenance
Module ablation	NO_F / NO_A / NO_R / NO_M / M_ONLY	Remove one FULL module, or add maintenance alone to BASE
Corrected retrieval	N / H / D	Native order; original lexical fusion; fusion with the utility term removed
Metadata ablation	R0 / RST	Same pretrained pair scorer, without or with speaker/date inputs
RHELM retention	Random-15	Random storage over seeds 5–19 at approximately matched token retention

Table A3: Evaluation suite. LongMemEval is an exploratory classifier diagnostic; RHELM QA evaluates rank-budget storage with BM25, separate from the unfinished four-module RHELM rerun.

Study	Evaluation data and controls	Measurements
Original-input admission	788 reviewed candidates: 209 keep, 579 discard; shared 15-candidate filter mask	AUROC, AUPRC, Brier
Corrected-input three-backend QA	788 candidates and 304 questions; seven module conditions; $B = 512$	Stored tokens, QA F1/EM, adversarial accuracy
Corrected retrieval	Same two test conversations; 353 admitted records; fixed pools; maintenance disabled	Coverage on 231 eligible questions; F1/EM on 233; adversarial accuracy on 71
LongMemEval	5,000 examples; source-selected P0/C0/A1; no target fitting (Wu et al., 2025)	Admission AUROC and length/role strata
RHELM QA	864 conversation-answerable questions; 25%/50% store-token retention; BM25 at 512 tokens/30 records; 64-token answer cap (Zhang et al., 2026b)	QA F1 against random storage

A.2 Training and Frozen Admission Scoring

Algorithm 1 specifies the twelve-view construction and source-development selection. The settings below apply to the frozen A1 scorer used in the reported evaluations.

Model and training. A1_augmented12 adapts Qwen2.5-0.5B with LoRA rank 8, scale 16, dropout 0.05, and query/key/value/output projections plus the saved classification head. There are 4,352 source training bases (52,224 augmented rows) and 676 development bases (8,112 augmented rows), with twelve views per base. Training uses one epoch, AdamW without a scheduler, learning rate 10^{-4}, cross-entropy, microbatch 1, and gradient accumulation 16. Seeds 42/43/44 select steps 1,632/3,264/1,632 from candidates 1,632 and 3,264 by source-development base-median AUROC (tie tolerance 0.002). Target data are not used for fitting or checkpoint selection.

Loss implementation. The archived trainer passes balanced inverse-frequency weights to `CrossEntropyLoss` with its default mean reduction. Each loss is evaluated on a single example, so the class weight appears in both the numerator and denominator and cancels before gradient accumulation. The effective objective is per-example cross-entropy; this configuration does not provide effective class reweighting.

Frozen scoring. The base revision is `060db6499f32faf8b98477b0a26969ef7d8b9987`. Adapter SHA-256 values are recorded in the model manifest and revision ledger. Each candidate body enters a fixed task frame with a full-prompt cap of 896 tokens; there is no recent-five-turn field. Long candidates use 50%-stride views, median margins within seed, then mean margins across three seeds and sigmoid. The source-development margin threshold is -2.4479166666666665, equivalent to score 0.07959103286809105. Current LoCoMo scoring used BF16 batch 16, right padding, no silent truncation, and one view for each of all 788 candidates, including CF rejects.

Retrieval and maintenance interfaces. The original utility model is a class-balanced logistic regressor over unigram/bigram TF–IDF features of *Query: q; Candidate memory:* m_i (30K vocabulary; 1,134 training and 278 development records). It already reads both question and memory. Replacing it with the pretrained pair scorer changes score distributions as well as the model; both utilities are uncalibrated. The original lexical selector needs no reranker forward pass, whereas RST adds pair-model computation. Packing retokenizes the final concatenation, including numbering, dates, speakers, and separators. Instructions, the question, and an available question date are added outside the memory-evidence budget; current QA prompts contain no few-shot examples.

FULL uses LightMem-inspired offline maintenance (Fang et al., 2026). A fixed local generator proposes ignore, update, or delete operations. Structural checks and a separate precommit semantic check retain the target if a proposal fails; accepted operations update the native store and source mapping before retrieval. These automatic checks and mappings provide traceability but do not establish independent human validation or guarantee preservation of every speaker attribution. The evaluated corrected-input workload accepts two updates and no deletions (Appendix B).

A.3 Runtime and Backend Interfaces

The frozen generator is Ollama `qwen2.5:7b`, served by Ollama 0.33.3 on an NVIDIA RTX 4050 Laptop 6GB profile with Flash Attention enabled, an f16 KV cache, one parallel request, and at most two loaded Ollama models. The generator digest is `845dbda0ea48ed749caafd9e6037047aa19acfcfd82e704d7ca97d631a0b697e`. These are generation-runtime settings; they are separate from the three preloaded A1 seed models in the admission microbenchmark.

QA uses temperature 0, top-p 1.0, top-k 40, repeat penalty 1.0, seed 20260625, an 8,192-token chat context, and a 256-token output cap. The prompt supplies the question date when available, numbered rendered memories, and the question; it restricts answers to supplied evidence and requires JSON fields `answer`, `abstain`, and `citations`. Full chat-template input plus reserved output is checked against the context limit. Maintenance uses the same generator seed, temperature 0, a 32,768-token context, and output caps of 1,024 for proposals and 256 for verification. Native A-MEM writing keeps its native temperature argument (default 0.7), with a 32,768-token context and 1,000-token output cap; its replay seeds are 2026090801/02/03. Shared QA/maintenance seed and cache reuse remain distinct from these backend replay seeds.

Mem0 uses `mem0ai` 0.1.118 with Qdrant client 1.19.0, 384-dimensional MiniLM embeddings, and direct writes with `infer=False`. A-MEM is pinned to upstream commit `0c8039f28fdcc08189a23c07a3437d9d2482f9c2`, with its native analysis/evolution path enabled and evolution threshold 100. Letta is version 0.9.1 at commit `f793a1fd449db8c4407f167119a2bb8f0ec24ba7`, with 1,024-dimensional `mxbai-embed-large` native indexing on CPU. Its archival-memory POST carries a `text` field containing the complete rendered record; atomic identity chunking with chunk size 65,536 preserves exact text. Base tools and agent-driven memory-block editing are disabled. The controlled adapter exposes passage read/update and native-search endpoints to exercise native storage, retrieval, and verified maintenance. These integrations do not imply an unrestricted autonomous-agent evaluation.

Fusion separately recomputes normalized query/body embeddings using `sentence-transformers/all-MiniLM-L6-v2`, revision `1110a243fdf4706b3f48f1d95db1a4f5529b4d41`, in CPU float32, batch 32. Its 256-token encoder limit is independent of A1's 896-token prompt cap and the final 512-token memory budget. The shared cost counter uses the frozen Qwen2.5 tokenizer without added special tokens; complete generation inputs are checked separately after chat-template rendering. The runtime lock records PyTorch 2.6.0+cu124, Transformers 5.13.1, PEFT 0.19.1, Sentence Transformers 5.6.0, scikit-learn 1.9.0, and NumPy 2.4.4. No dependency was upgraded for a new experiment in this revision.

A.4 Reviewed Original-Input Admission

Table A4 evaluates 788 original converted-input candidates with 209 keep and 579 discard labels. The ledger records 200 blind-human labels and 588 LLM-prelabel/human-review entries, with one reviewed binary label per candidate. The final adjudication of `conv-30:D15:7` is discard. Agreement between QA-evidence pseudo-labels and the directly human-labelled audit subset is 73.0% ($\kappa = 0.46$); this motivates the reviewed-label comparison.

Both scorers assign zero to the same 15 cheap-filter rejects. A1 yields AUROC 0.851356, AUPRC 0.611847, and Brier error 0.215497; TF–IDF yields 0.838081, 0.607858, and 0.172125. The paired AUROC difference is 0.013276 (95% CI $[-0.008186, 0.034978]$; 10,000 candidate resamples, seed 20260908), conditional on two conversations. Thus no AUROC advantage is established, and A1 has worse Brier error; Brier is descriptive squared-score error, not evidence of calibration.

Table A4: Reviewed LoCoMo admission. (a) Both scorers assign zero to the same 15 cheap-filter (CF) rejects; Brier is squared-score error. (b) Decisions at the frozen A1 threshold; store-all is the retention reference. Labels contain 209 keep and 579 discard candidates.

(a) Ranking and score error

Pipeline	AUROC ↑	AUPRC ↑	Brier ↓
CF + TF–IDF	.838	.608	**.172**
CF + A1 ensemble	**.851**	**.612**	.215

(b) Storage decisions

Policy	Admitted	False writes	False discards	Keep recall
Store-all	788	579	0	100.00%
CF + A1	357	174	26	87.56%

At the frozen threshold, CF+A1 retains 183 of 209 keep candidates and 174 discard candidates, with 26 false discards (precision .5126, recall .8756, F1 .6466). A1 alone retains 358 candidates; CF+A1 retains 357. These original-input counts differ from the corrected-input 353-record stores. No threshold is retuned on the test labels. Historical budget sensitivity and paired intervals are documented separately in Appendix F.

B Corrected-Input Scoring, Costs, and Module Effects

Tables 1 and 4 use the corrected-input full rerun accepted on 23 September 2026, separately from the historical original-input diagnostics in Appendix F and the no-maintenance ranking experiment in Appendix D. The seven-condition, three-backend subset at 512 tokens contains 10,640 logical QA records and 60 unique final stores. The BASE/FULL subset contains 3,040 QA records and 20 stores. FULL and NO_R share each final store and candidate pool, with only lexical fusion versus native ordering changed. No experiments were added for this revision.

Scoring and denominators. Ordinary F1 and validity use 233 questions, adversarial accuracy and validity use 71, evidence coverage uses 231 eligible ordinary questions, and overall format error uses all 304. The corresponding conv-26 counts are 152/47/150/199; conv-30 counts are 81/24/81/105. Valid output is JSON/schema-correct and non-abstaining, without requiring correctness. The original archived full-rerun evaluation required abstention on all 71 adversarial questions. For the results reported here, saved outputs are rescored using the same 69-abstention/two-"No" protocol as Appendix D, including normalized exact matching and a non-abstaining output for the two "No" references. Adversarial validity remains the rate of schema-correct, non-abstaining outputs and can include a correct "No" answer; adversarial accuracy, validity, and format error therefore no longer form a partition of this set. Frozen LoCoMo F1 assigns zero to refusals and format failures, without dropping questions. Coverage is per-question recall of annotated source IDs in the final selected context, not in storage or the candidate pool. Citation-range errors are separate from JSON/schema failures.

Aggregation and verification. A-MEM averages its three write seeds for each question before averaging eligible questions within each conversation and then weighting the two conversations equally. Mem0 and Letta use one run each. Seeds, conditions, and cache reuses are not independent test samples. Independent re-parsing and scoring verified request/response/context/snapshot bindings, complete question and seed sets, and the 512-token rendered-context constraint. Under the original all-abstention protocol, independent recomputation reproduced the 21 frozen backend–condition summaries within 10^{-12}. For the present reporting, only adversarial accuracy and its paired differences are rescored under the shared reference-aware protocol; the remaining metrics are unchanged, and BASE/FULL ordinary F1 and resource accounting agree with Table 1. Across 10,640 logical rows there are 172 format failures, 4,150 abstentions, two citation-range errors, and zero truncated outputs; all remain in their applicable denominators. These raw totals count seed-level logical records, not independent questions. The archived experiment records retain separate

ordinary/adversarial format-error rates, paired differences, source paths and SHA-256 hashes, and records zero supplemental runs.

Resource boundary. Stored tokens count final post-maintenance memory-body text using the frozen tokenizer, summed across conversations and then averaged across A-MEM write seeds. Model weights, indexes, and database overhead are excluded. BASE has 788 records and 29,447 body tokens; FULL has 353 and 17,212; NO_M has 353 and 17,159; NO_F has 354 and 17,220. For condition c, memory-body savings relative to BASE are $1 - T_c/T_{\mathrm{BASE}}$. On corrected inputs, NO_M saves 41.73% and FULL saves 41.55%; the difference reflects the 53 body tokens added by accepted maintenance updates. LLM input-plus-output tokens cover successful native store construction, maintenance proposals and verification, and all 304 answers for both conversations. Admission and embedding computation are excluded. Bound cached/reused requests retain logical costs; failed builds, duplicate log references, and enclosing stages that would double-count nested calls are excluded. A-MEM costs average three complete workloads. Input/output splits and exact totals remain in the research archive. FULL uses 46,091 maintenance LLM tokens per workload for two accepted updates and no deletions; NO_R retains the same stores, candidate pools, and maintenance cost.

B.1 Conditional Module Effects

Table 4 recomputes all seven configurations on the completed corrected-input rerun, using the same protocol as Table 1. FULL versus BASE lowers ordinary valid-answer rate and evidence coverage in all three backends, while F1 improves only in Mem0. These contrasts describe module effects conditional on the remaining configuration and two test conversations; they do not establish general module superiority or statistical significance.

Removing lexical fusion (NO_R versus FULL) raises mean F1 by 1.90/2.91/4.73 percentage points in Mem0/A-MEM/Letta. Ordinary validity and evidence coverage increase in both conversations for every backend, but Mem0 and Letta F1 decrease slightly on conv-30 (0.49 and 0.16 points); their mean gains are driven by conv-26. Adversarial accuracy falls by 1.06/5.20 points in Mem0/A-MEM as adversarial valid-answer rate rises by 1.06/5.22 points. Letta instead gains 3.19 points of adversarial accuracy and answers fewer adversarial questions, with both changes confined to conv-26. A-MEM's adversarial accuracy improves on conv-26 but declines on conv-30. Overall format error rises by 0.98/1.43 points in Mem0/Letta and falls by 0.16 points in A-MEM. Thus, native ranking's mean F1 gains coexist with backend-dependent adversarial and formatting trade-offs.

Removing admission (NO_A versus FULL) lowers F1 and ordinary validity in both conversations for every backend, while mean adversarial accuracy increases and mean format error decreases. Adversarial valid-answer rate falls overall in each backend. Evidence coverage rises on conv-26 but falls on conv-30 in all three backends; Letta's format error also worsens on conv-30 despite its lower two-conversation mean. Removing filtering (NO_F versus FULL) leaves F1, both adversarial metrics, and format error unchanged, but increases ordinary validity and coverage by 0.62 points in every backend, entirely through conv-30. NO_F retains 354 records and 17,220 body tokens, compared with FULL's 353 and 17,212.

Removing maintenance (NO_M versus FULL) raises mean F1 by 0.50/0.40/0.50 points and increases coverage in all three backends. For Mem0 and Letta, the F1, ordinary-validity, and coverage gains occur only on conv-30, with adversarial metrics and format error unchanged. For A-MEM, ordinary validity falls by 0.44 points and format error rises by 0.82 points, while adversarial accuracy rises by 0.69 points and adversarial valid-answer rate falls by 1.05 points. FULL and NO_M both retain 353 records; maintenance adds 53 body tokens without reducing record count. Adding maintenance alone (M_ONLY versus BASE) likewise leaves 788 records. It slightly raises Mem0 F1 and ordinary validity on conv-26 while lowering coverage, and lowers Letta F1 with no change in answer-validity, adversarial, or formatting rates. A-MEM's F1 changes nearly cancel across conversations; validity, coverage, and adversarial accuracy increase overall, but so does format error. Maintenance therefore has mixed quality effects in these fixed comparisons, without observed record compression.

Table A5: Mem0 paired module differences (percentage points). Each contrast is treatment minus control on matched questions and seeds. Mean weights conversations equally; conversation rows retain seed averaging. Adv. valid measures answering behaviour, not accuracy.

Contrast	Scope	F1	Ord. valid	Adv. acc.	Adv. valid	Coverage	Format error
FULL−BASE	Mean	+0.92	-0.70	-4.21	+4.21	-5.85	+0.50
	conv-26	-1.28	-2.63	-4.26	+4.26	-4.61	+1.01
	conv-30	+3.12	+1.23	-4.17	+4.17	-7.10	0.00
NO_F−FULL	Mean	0.00	+0.62	0.00	0.00	+0.62	0.00
	conv-26	0.00	0.00	0.00	0.00	0.00	0.00
	conv-30	0.00	+1.23	0.00	0.00	+1.23	0.00
NO_A−FULL	Mean	-2.23	-6.05	+5.23	-4.17	+1.15	-1.01
	conv-26	-2.16	-5.92	+2.13	0.00	+2.61	-2.01
	conv-30	-2.30	-6.17	+8.33	-8.33	-0.31	0.00
NO_R−FULL	Mean	+1.90	+9.30	-1.06	+1.06	+9.27	+0.98
	conv-26	+4.29	+11.18	-2.13	+2.13	+9.56	+1.01
	conv-30	-0.49	+7.41	0.00	0.00	+8.99	+0.95
NO_M−FULL	Mean	+0.50	+0.62	0.00	0.00	+0.93	0.00
	conv-26	0.00	0.00	0.00	0.00	0.00	0.00
	conv-30	+1.00	+1.23	0.00	0.00	+1.85	0.00
M_ONLY−BASE	Mean	+0.13	+0.33	0.00	0.00	-0.50	0.00
	conv-26	+0.25	+0.66	0.00	0.00	-1.00	0.00
	conv-30	0.00	0.00	0.00	0.00	0.00	0.00

Table A6: A-MEM paired module differences (percentage points). Each contrast is treatment minus control on matched questions and seeds. Mean weights conversations equally; conversation rows retain seed averaging. Adv. valid measures answering behaviour, not accuracy.

Contrast	Scope	F1	Ord. valid	Adv. acc.	Adv. valid	Coverage	Format error
FULL−BASE	Mean	-1.93	-7.94	+6.96	-6.62	-12.21	+1.31
	conv-26	-2.29	-10.53	+1.42	-2.13	-11.96	+1.68
	conv-30	-1.57	-5.35	+12.50	-11.11	-12.46	+0.95
NO_F−FULL	Mean	0.00	+0.62	0.00	0.00	+0.62	0.00
	conv-26	0.00	0.00	0.00	0.00	0.00	0.00
	conv-30	0.00	+1.23	0.00	0.00	+1.23	0.00
NO_A−FULL	Mean	-1.25	-2.92	+1.05	-1.40	+1.49	-1.31
	conv-26	-1.60	-4.61	+0.71	-1.42	+3.31	-1.68
	conv-30	-0.89	-1.23	+1.39	-1.39	-0.34	-0.95
NO_R−FULL	Mean	+2.91	+9.09	-5.20	+5.22	+14.43	-0.16
	conv-26	+4.66	+14.47	+0.71	+0.71	+16.11	0.00
	conv-30	+1.17	+3.70	-11.11	+9.72	+12.74	-0.32
NO_M−FULL	Mean	+0.40	-0.44	+0.69	-1.05	+1.20	+0.82
	conv-26	-0.05	-0.88	0.00	-0.71	+0.56	+1.01
	conv-30	+0.85	0.00	+1.39	-1.39	+1.85	+0.63
M_ONLY−BASE	Mean	+0.01	+0.32	+2.10	-2.10	+0.34	+0.18
	conv-26	+0.28	+0.22	+1.42	-1.42	+0.41	+0.67
	conv-30	-0.27	+0.41	+2.78	-2.78	+0.27	-0.32

Table A7: Letta paired module differences (percentage points). Each contrast is treatment minus control on matched questions and seeds. Mean weights conversations equally; conversation rows retain seed averaging. Adv. valid measures answering behaviour, not accuracy.

Contrast	Scope	F1	Ord. valid	Adv. acc.	Adv. valid	Coverage	Format error
FULL−BASE	Mean	-4.20	-11.64	-2.22	+2.22	-19.53	-0.70
	conv-26	-6.36	-17.11	-12.77	+12.77	-20.39	+0.50
	conv-30	-2.03	-6.17	+8.33	-8.33	-18.66	-1.90
NO_F−FULL	Mean	0.00	+0.62	0.00	0.00	+0.62	0.00
	conv-26	0.00	0.00	0.00	0.00	0.00	0.00
	conv-30	0.00	+1.23	0.00	0.00	+1.23	0.00
NO_A−FULL	Mean	-4.19	-11.44	+3.19	-3.19	-0.14	-0.28
	conv-26	-1.01	-0.66	+6.38	-6.38	+3.11	-1.51
	conv-30	-7.36	-22.22	0.00	0.00	-3.40	+0.95
NO_R−FULL	Mean	+4.73	+14.89	+3.19	-3.19	+19.36	+1.43
	conv-26	+9.62	+22.37	+6.38	-6.38	+22.11	0.00
	conv-30	-0.16	+7.41	0.00	0.00	+16.60	+2.86
NO_M−FULL	Mean	+0.50	+0.62	0.00	0.00	+0.93	0.00
	conv-26	0.00	0.00	0.00	0.00	0.00	0.00
	conv-30	+1.00	+1.23	0.00	0.00	+1.85	0.00
M_ONLY−BASE	Mean	-0.41	0.00	0.00	0.00	-0.93	0.00
	conv-26	-0.04	0.00	0.00	0.00	0.00	0.00
	conv-30	-0.79	0.00	0.00	0.00	-1.85	0.00

C A1 Write-Admission Latency Protocol

The audited microbenchmark uses the same A1_augmented12 weights (seeds 42/43/44), Qwen2.5-0.5B base revision, and 896-token full-prompt window as the current scorer. All three independent seed models were resident before timing on an NVIDIA RTX 4050 Laptop GPU, with BF16 and batch size 1 (PyTorch 2.6.0+cu124, CUDA 12.4). Fifty warm-up candidates preceded five randomized passes over the historical 775 model-invoked candidates ($N = 3{,}875$; shuffle seed 43). This population predates the current 15-reject CF mask. No inference was rerun for this paper revision.

Table A8: Audited A1 microbenchmark latency in milliseconds, batch 1, $N = 3{,}875$. These are classifier-stage measurements, not full-system latency.

Boundary	Mean	p50	p95
Model and aggregation	161.783	161.571	162.872
Candidate to decision	162.855	162.638	164.011

Candidate-to-decision timing includes task/view construction, tokenization, host-to-device transfer, all three forward passes, median-view and mean-seed margin aggregation, sigmoid, and thresholding, with synchronized GPU boundaries. Model/aggregation timing begins with pretokenized CPU tensors. Model loading took 2,309.3 ms and is excluded. No backend, retrieval, 7B generation, maintenance, or judge time is included; the benchmark does not measure A1/7B co-residency or concurrency. A derived zero-bypass average is not reported as measured full-pipeline latency.

The canonical score trace used BF16 batch 16, whereas this microbenchmark used batch 1. Maximum score drift was 0.08856, with two decision changes at the old 0.45 threshold and none at the current deployment threshold. Repeated batch-1 scores were identical. This numeric distinction is preserved rather than claiming all traces match exactly.

D CORRECTED-INPUT PAIRED RERANKING

This study isolates reranking on corrected LoCoMo inputs with frozen A1 and no maintenance. It is separate from the seven-condition full rerun in Appendix B.

Inputs and fixed controls. Corrected conv-26/30 contain 788 candidates, including 188 restored image captions and official source dates. Frozen CF+A1 retains 189 and 164 records, respectively (353 total; 17,159 body tokens versus 29,447 for store-all). Across conditions, the admitted store, up-to-50 candidate pool, MiniLM cosine scores, displayed-record lengths, and 512-token budget remain fixed within each backend. Fusion conditions use the coefficients specified below; all conditions use skip-if-too-long packing. Mem0 and Letta are evaluated; this study supplies no A-MEM result. Qwen2.5-7B answering and its frozen scoring protocol remain unchanged.

Model and metadata. Qwen3-Reranker-0.6B (Zhang et al., 2025) is used without LoCoMo fine-tuning, at revision `e61197ed45024b0ed8a2d74b80b4d909f1255473`. The model jointly reads a question and one candidate memory; this is question–memory pair scoring, not a newly trained pairwise loss over two candidates. The relevance score is the sigmoid of the final yes/no logit difference. Four document views retain the same body/captions and expose neither field (R0), speaker only (RS), observation date only (RT), or both (RST). Missing fields are explicitly unavailable; no event time, timezone, or question date is invented. Observation time is the recording time, not necessarily the event time. The QA context already contains metadata, so the intervention changes reranker input only. Old utility also reads the question and memory, through lexical features.

Numerics and development rule. Inference uses BF16 with final-logit arithmetic in FP32, batch 1, left padding, a 2,048-token input ceiling, and no silent truncation. A mixed-length batch-4 preflight exceeded the prespecified score-difference tolerance (0.05188 versus 0.02); batch 1 was fixed before viewing scientific metrics, and single-input repeat differences were zero. The corrected development conversation conv-43 has 680 candidates, 286 admitted records, and 242 questions per backend. Main development coverage uses 178 ordinary questions with nonempty, fully resolvable evidence. A prespecified resource gate permits test QA if R0 or RST exceeds N and D in backend-mean coverage and is no worse in either backend. Both pass. This rule is not a significance test. No test-set search over prompts, coefficients, temperatures, or budgets is performed.

Conditions and measurement. N uses native order; H uses $0.4u_{\text{old}} + 0.6c - 0.03\ln(\max(1, L))$; D removes the utility term. R0/RS/RT/RST substitute the corresponding pair-model score while keeping the other terms. PURE-TEXT and PURE-BOTH rank directly by the new score as planned offline auxiliaries. H uses corrected inputs without maintenance and is distinct from FULL.

Test metrics average conversations equally. Coverage uses 150/81 eligible ordinary questions in conv-26/30, ordinary F1/EM/validity use 152/81, and adversarial metrics use 47/24. The shared reference-aware protocol in Appendix B expects 69 abstentions and two non-abstaining `No` answers. Validity measures schema-correct, non-abstaining output. Table A9 supplies H; Table A10 reports development and test coverage. RS/RT and pure-score auxiliaries have no separate QA evaluation.

Table A9: Corrected-input lexical-fusion control H, supplementing the N/D/R0/RST results in Table 3. Two-conversation equal means; one logical repeat.

Backend	Ranking	F1	EM	Ord. valid	Adv. acc.
Mem0	H	.1831	.0761	.6294	.6729
Letta	H	.1908	.0699	.6598	.6410

Quality trade-offs. RST improves F1 over N in all four backend–conversation pairs, but Letta/conv-26 EM decreases. Removing old utility (D) raises coverage over H while lowering F1 in both backends. R0 improves Mem0 F1 over N but is slightly worse in Letta. Adding speaker to R0 gives a backend-equal test-coverage gain of 8.35 points; date alone gives 0.97, and date after speaker gives 0.22. These coverage controls do not establish separate F1 gains. High-frequency generic-memory selection persists.

Table A10: Evidence-ID coverage by condition. Development uses conv-43; test is the equal mean of conv-26 and conv-30. These are provenance metrics, not semantic correctness scores.

	Development		Test	
Condition	Mem0	Letta	Mem0	Letta
N	.5266	.6150	.4070	.5159
H	.5315	.5401	.3173	.3316
D	.5539	.5689	.3521	.3636
R0	.6556	.6782	.4907	.5081
RS	.6796	.6972	.5758	.5900
RT	.6604	.6801	.5035	.5147
RST	.6951	.7121	.5780	.5922
PURE-TEXT	.6645	.6773	.5263	.5297
PURE-BOTH	.6937	.7301	.6330	.6254

Execution and cost. The 1,824 D/R0/RST QA tasks comprise 1,401 actual generations and 423 exact-request cache reuses; 1,216 N/H controls reuse historical answers after request/context verification. This is one logical repeat. Development and test across four views require 149,332 distinct pair-model requests and approximately 110.54 minutes of forward execution plus score writing. These are aggregate research costs, not per-query deployment latency. Model files total 1.21 GB; new QA generation uses 943,806 input and 58,359 output tokens, with 29.27 minutes of summed generation wall time. Reranking computation and model storage are separate from memory-body savings. Checks confirm fixed pools, budget compliance, and model/implementation hashes; answer errors remain in denominators.

D.1 Conversation-Level Adversarial Analysis

Table A11 localizes the adversarial-score declines. On Letta/conv-26, RST changes accuracy from .5957 to .4255 (28 to 20 of 47 correct), while conv-30 improves. Mem0 also declines only on conv-26. For Letta/conv-26, ten regressions change schema-correct outputs from `abstain=true` to `abstain=false`, partly offset by two opposite transitions.

Table A11: Adversarial split by conversation for native ranking (N) and RST, over the 47 (conv-26) and 24 (conv-30) adversarial questions only. Valid is the non-abstaining, format-correct answer rate, not correctness. The frozen protocol requires abstention on 45/47 conv-26 questions and all 24 conv-30 questions; two conv-26 questions require "No". Δ is RST minus N in percentage points.

Backend	Conv.	n	N acc	RST acc	Δ (pp)	N valid	RST valid
Mem0	conv-26	47	.5745	.5106	-6.38	.4468	.5106
	conv-30	24	.7500	.7500	0.00	.2500	.2500
Letta	conv-26	47	.5957	.4255	-17.02	.4468	.6170
	conv-30	24	.7500	.7917	$+4.17$	.2500	.2083

For gold-abstention questions, correctness checks the schema and abstention flag without independently judging answer text. Unblinded inspection found attribution errors, flag–text inconsistencies, and possible answerability-label conflicts. Letta/RST corrects the speaker premise in `conv-30:qa:86` but receives no credit because its flag is false. Mem0/D and Mem0/RST both answer Bach and Mozart to `conv-26:qa:185`; only D receives credit because its flag is true, although the source attributes the preference to the other speaker. Source inspection also raises answerability questions for `conv-26:qa:161` and `conv-26:qa:184`.

Among the ten Letta/conv-26 regressions, the inspected anchor is already selected under native ranking in six cases and newly selected under RST in four. These observations do not identify causal mechanisms or constitute independent adjudication; labels and scores remain unchanged. Ordinary-question coverage concerns a different population. Development uses one conversation and testing

reuses two previously analyzed conversations, so this study is not a new blind test and does not establish population significance, non-inferiority, or third-party pretraining decontamination.

E Cross-Distribution Admission and Retention Diagnostics

These diagnostics evaluate frozen source-trained scorers under target-distribution shift. LongMemEval measures admission-label agreement; historical RHELM evaluates admission and a separate rank-budget/BM25 QA protocol. Neither evaluates the completed corrected-input four-module system outside LoCoMo.

E.1 LongMemEval Evaluation

The 5,000 LongMemEval-derived admission examples contain 1,413 KEEP and 3,587 DISCARD labels, all produced by the fixed DeepSeek-V4-Flash annotator (Wu et al., 2025). AUROC measures agreement with this protocol, rather than downstream QA or independently human-annotated target labels. Candidates have median/P90 lengths of 206/514 tokens and explicit user/assistant roles.

C0 and A1 match architecture, 52,224 training rows, one epoch, the 896-token prompt cap, three seeds, and source-only checkpoint selection. C0 repeats each native source example twelve times. A1 uses the twelve structured views in Section 3.3, retaining each core and its source admission label. Neither condition fits to target labels.

Table 2 reports the complete overall and stratified AUROCs. A1 exceeds P0 by .022 (95% CI $[.011, .032]$) and C0 by .012 ($[.004, .019]$), using paired bootstrap resampling over 4,352 groups defined by the stored `session_digest`. These exploratory target estimates do not establish source-domain non-regression or current LoCoMo QA preservation.

The historical trunc256 flag tests whether the complete candidate-only frame, including prompt overhead, exceeds 256 tokens; it is independent of A1's 896-token cap. In that stratum, A1–P0 is .089 ($[.044, .132]$); the assistant-role difference is .055 ($[.023, .088]$). Each displayed role/truncation partition covers all 5,000 examples. Three earlier input-length rows covered only 4,994 examples and remain withheld, along with their derived conclusions, pending reconciliation.

E.2 Historical Inference-Only Controls

Earlier approaches did not improve overall LongMemEval AUROC over P0 (.801): 512-token training reached .785, frozen raw-max aggregation .767, and a normalized hybrid .794. The inference-only variants use P0-256 adapters, overlapping 128-token payloads with 25% overlap, and complete candidate coverage; these are separate from current A1 inference. In the trunc256 stratum, raw-max and hybrid gains over P0 are .024 ($[-.017, .065]$) and .031 ($[-.001, .062]$), while the hybrid's overall difference is $-.007$ ($[-.012, -.002]$). Neither localized interval excludes zero. The hybrid hypothesis followed inspection of earlier target results, so it remains exploratory despite source-fitted normalization. Detailed predecessor development tables are omitted from this manuscript.

E.3 RHELM Sealed External Evaluation

Protocol and reference labels. RHELM comprises 629 sessions across ten persona trajectories (Zhang et al., 2026b); the converted admission set has 23,548 candidates (11,774 user and 11,774 assistant). Nine adapters (P0/C0/A1, three seeds each) were frozen and SHA-256-verified on 22 July 2026. Predictions were produced on 23 July at 17:11 UTC and sealed in a read-only manifest before label generation.

DeepSeek V4 Flash produced labels under a structured 19-category rubric (Pass A). A second pass with reordered prompts used the same model; agreement was $\kappa = .52$, not independent human validation. The final references use Pass A with UNCERTAIN excluded: 7,383 KEEP and 16,165 DISCARD. KEEP prevalence is 60.9% for user messages and 1.8% for assistant messages. These labels support descriptive comparisons rather than confirmatory generalization to a human gold standard.

Admission findings. Table A12 shows A1 below P0/C0 on overall and macro-persona AUROC. A1 leads within user messages, while C0 leads within assistant messages. Approximately 69.5% of positive–negative pairs combine a user-positive with an assistant-negative, so overall AUROC is strongly affected by role separation.

Table A12: RHELM admission AUROC by role and macro-persona mean. All values are ensemble AUROC over three seeds.

Method	Overall	User	Assistant	Macro persona
P0	0.866	0.707	0.584	0.863
C0	**0.868**	0.703	**0.877**	**0.865**
A1	0.859	**0.725**	0.850	0.857

Table A13: RHELM admission AUROC stratified by candidate token length (label-free bins fixed before evaluation). Best value in each row in bold.

Token range	n	P0	C0	A1
≤ 64	8,859	0.730	0.722	**0.743**
65–128	3,995	**0.843**	0.837	0.839
129–256	4,270	0.825	**0.958**	0.939
257–512	4,261	0.651	**0.920**	0.889
≥ 513	2,163	0.553	**0.652**	0.584

Table A13 uses label-free length bins fixed before evaluation. A1 leads only in the shortest bin. C0's high AUROCs in the 129–256 and 257–512 bins do not establish equally strong within-role ranking because role and label prevalence are associated with length.

Historical rank-budget QA. The verified QA set contains 864 conversation-only questions, excluding 441 that require unavailable email or attachment evidence. A1 ranks candidates into 25% or 50% store-token budgets; BM25 retrieves at most 30 records under 512 tokens. Qwen2.5-7B answers and judges with 64-token output caps, temperature 0, and seed 20260625. This differs from current threshold admission, fusion, and maintenance. Reconstruction with current A1 matches retained IDs, text, and tokens, but does not restore every original scoring-process log.

Table A14: Historical RHELM QA on 864 questions. Random F1 is mean $\pm$ SD over seeds 5–19; random EM is the seed mean. Random stores use 1.96–2.00% more tokens than A1. F1 intervals resample ten persona clusters.

Retention	Random F1	A1 F1	Δ F1	95% CI	Random EM	A1 EM
25%	$.0913 \pm .0077$	**.1300**	$+.0387$	$[.0175, .0543]$	.0405	**.0602**
50%	$.1100 \pm .0053$	**.1338**	$+.0238$	$[.0087, .0362]$	.0481	**.0602**

At 25%/50% retention, A1 minus mean random-15 F1 is .038664/.023848, with intervals $[.017509, .054270]$/$[.008708, .036220]$. EM differences are .019676/.012037, with intervals $[.003862, .031172]$/$[.002408, .020383]$. Intervals use 10,000 resamples of ten persona clusters (seed 20260908), with QA-count weighting inside each draw. They are unadjusted estimation intervals, not familywise tests or evidence for a universally best gate. Random seeds 5–19 use 1.96–2.00% more tokens than A1.

The retained A1 bridge and repaired random seeds 5–19 have matching endpoint audits. Earlier seeds 0–4, pooled twenty-seed estimates, and the previous store-all line are excluded from the common verified comparison because the legacy component lacks endpoint-digest verification. The separate four-module RHELM run has no completed QA outputs (0 of 10,368 planned logical answers). Its stores/retrievals are not a finished system result. The reported rank-budget QA gains do not establish a uniform A1 classifier advantage over C0.

F HISTORICAL INPUT VERSION, SENSITIVITY, AND ARTIFACT SCOPE

F.1 INPUT VERSION AND REPLAY BOUNDARY

The original LoCoMo conversion omitted 188 image captions and mismatched 770 of 788 candidate dates across 37 of 38 sessions. The current corrected-input studies repair these fields; the historical diagnostics below retain their original inputs. Their 357-record CF+A1 stores and question-weighted averages must not be substituted for the current 353-record, conversation-equal results.

The historical backend evaluation contains 39 backend/condition/budget cells, three recorded replays per cell, and 35,568 logical QA rows over 304 identities. It records 12,765 successful generations and 22,803 cache hits. For Mem0 and Letta, all 3,952 answer/context pairs in each of replays 2 and 3 match replay 1, so these are not independent answering trials. A-MEM write seeds can change stores, but deterministic QA and cache reuse also limit replication claims. Replays are averaged within question before question-weighted aggregation. Ordinary F1 uses 233 questions; refusals and format failures remain in denominators.

F.2 HISTORICAL BUDGET SENSITIVITY

Table A15 preserves the 256/512/1,024-token comparison and its original paired-question intervals: 10,000 resamples of question identities, seed 20260908, conditional on two conversations. These intervals do not model variation across conversation histories. This historical budget sweep does not establish budget robustness for the corrected-input full rerun. The old 120-token component/backend/latency studies and predecessor pressure-test tables are omitted because they use different controllers and protocols.

Table A15: Original converted-input budget sensitivity with question-weighted means and original question-bootstrap intervals. A1's 896-token input cap is independent of retrieval memory budgets.

Backend	Budget	BASE F1	FULL F1	Δ 95% CI
Mem0	256	0.1520	0.1313	$[-0.0511, 0.0092]$
Mem0	512	0.1790	0.1518	$[-0.0579, 0.0033]$
Mem0	1024	0.1858	0.1936	$[-0.0256, 0.0422]$
A-MEM	256	0.1531	0.1425	$[-0.0385, 0.0175]$
A-MEM	512	0.1846	0.1615	$[-0.0512, 0.0053]$
A-MEM	1024	0.2017	0.1972	$[-0.0303, 0.0216]$
Letta	256	0.1914	0.1416	$[-0.0827, -0.0179]$
Letta	512	0.2166	0.1789	$[-0.0684, -0.0071]$
Letta	1024	0.2368	0.1827	$[-0.0869, -0.0220]$

F.3 CONVERSATION-CLUSTER SENSITIVITY

The dated supplement `conversation_sensitivity_20260918_v1` uses the same frozen original-input outcomes. After averaging replay values within each question, it resamples two whole conversations with replacement 10,000 times (seed 20260908), pooling questions within each draw. It retains the original QA weighting: 152 ordinary questions in conv-26 and 81 in conv-30.

With two clusters there are four ordered resamples and at most three distinct means. Either conversation alone has probability one quarter and the two together one half; the 2.5th/97.5th percentile limits therefore equal the conversation-specific extremes. Table A16 reports a sensitivity diagnostic, not calibrated population uncertainty. Two clusters cannot establish population significance, robust non-inferiority, or repair input defects.

Under this original-input protocol, FULL has lower F1 than BASE in both conversations for every backend. NO_R has positive pooled differences but negative differences on conv-30 in all three backends. It retains maintenance; these results do not evaluate admission without maintenance or establish quality preservation.

Table A16: Original converted-input, post-hoc conversation sensitivity at $B = 512$. All values are ordinary-F1 differences in percentage points. The overall column preserves the original QA-weighted point estimate. Percentile limits describe the discrete two-cluster resampling distribution, not reliable population uncertainty.

Backend	Contrast	Overall	conv-26	conv-30	Percentile limits
Mem0	FULL−BASE	-2.722	-1.658	-4.721	$[-4.721, -1.658]$
Mem0	NO_R−BASE	$+0.048$	$+0.478$	-0.759	$[-0.759, +0.478]$
A-MEM	FULL−BASE	-2.310	-2.108	-2.688	$[-2.688, -2.108]$
A-MEM	NO_R−BASE	$+2.148$	$+4.195$	-1.693	$[-1.693, +4.195]$
Letta	FULL−BASE	-3.774	-3.704	-3.906	$[-3.906, -3.704]$
Letta	NO_R−BASE	$+0.309$	$+1.260$	-1.476	$[-1.476, +1.260]$

Archived records and availability. The dated research archive preserves `paired_effects.json`, `per_question_metrics.jsonl`, `statistics_report.py`, their SHA-256 hashes, and `evidence_revision_20260918.json`. The frozen function reproduces all 180 original paired comparisons within 10^{-12}; the supplemental script also enumerates the exact two-cluster distribution. These historical statistics remain distinct from the current reference-aware rescoring in Appendix B.

The source bundle's compact `reranker_evidence_20260922.json` records the separate paired-reranking study's revision, aggregates, counts, limitations, and source hashes. It does not by itself reproduce the full rerun or its reference-aware rescoring. Full per-question scores, candidate traces, answers, token ledgers, frozen inputs, and adapters remain in the research archive; the manuscript source ZIP is not a complete executable experiment release. No additional experiments were run for this revision. Independent human QA and maintenance validation remains pending, separate from the recorded admission-label review.