DOES MEMORY CONSTRUCTION REALLY HELP THE LONG-TERM MEMORY OF AI ASSISTANTS?

Anonymous authors
Paper under double-blind review

ABSTRACT

AI assistants need long-term memory to answer questions about users' past experiences. Existing methods reorganize these experiences into memory without considering the queries users will ask, but we find that their advantages diminish as evaluation spans more diverse benchmarks. In contrast, directly retrieving original records (*Dense*) remains competitive. To understand why, we keep the evidence and answers fixed and change only how questions are phrased. The resulting counterexamples show that rephrasing can turn a memory method's advantage over *Dense* into a disadvantage. This suggests that users' questions can play a decisive role in whether memory construction helps. Our theoretical analysis further shows that, under stated assumptions, no memory construction can guarantee a universal advantage over *Dense* without assumptions about the user's query distribution. This motivates *Dynamic Retrieval Entrances for Agent Memory* (DREAM), which preserves original records while adapting the routes used to retrieve them. As users ask questions, DREAM activates retrieval entrances that reflect their recent needs, allowing an initially retrieved record to lead to additional relevant evidence. Across four benchmarks, DREAM improves average Judge accuracy over *Dense* by 4.2 percentage points with Qwen3-VL-8B and 6.2 points with Gemma4-12B.

1 INTRODUCTION

Advances in multimodal large language models (MLLMs) are bringing AI assistants into users' everyday lives by expanding their ability to understand and reason about personal experiences (Bei et al., 2026; Nie et al., 2026). As these assistants support long-term interactions through mobile and wearable devices, they need to adapt to users' changing activities, needs, and preferences (Jiang et al., 2025; Nie et al., 2026). Such adaptation requires connecting users' requests with information from their past interactions. Long-term memory provides this connection by retaining past information and making it available for reasoning and responses (Maharana et al., 2024; Bei et al., 2026).

Existing long-term memory methods transform interaction histories into summaries, linked notes, extracted facts, or hierarchical representations (Zhong et al., 2024; Xu et al., 2026b; Chhikara et al., 2025; Sarthi et al., 2024). These constructions aim to make information easier to retrieve and use. However, we find a counterintuitive pattern in Figure 1: the advantages of memory construction on selected benchmarks diminish as evaluation broadens. Simply retrieving original records by embedding similarity, a baseline we call *Dense*, instead remains competitive as benchmark diversity increases.

Our comparison covers five long-term memory benchmarks that test recall and temporal reasoning across extended conversations, integration of personal evidence from photo albums, videos, and emails, and daily reasoning about fine-grained visual details and changing states across interactions (Details in Appendix H). Although memory methods can show advantages on selected benchmarks, these gains do not consistently carry over to the full evaluation. When scores are averaged across all five benchmarks and weighted by their evaluation counts, the six memory methods shown in Figure 1 perform near or below *Dense* in both Reader settings. These observations lead us to ask: *Does reorganizing a history into memory really improve on simply retrieving its original records?*

Our empirical results suggest otherwise. We find that only rephrasing a question can reverse a memory method's retrieval advantage, even when the memory source and answer remain fixed. We develop a geometry-guided counterexample construction procedure (Section 2). It searches for query

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

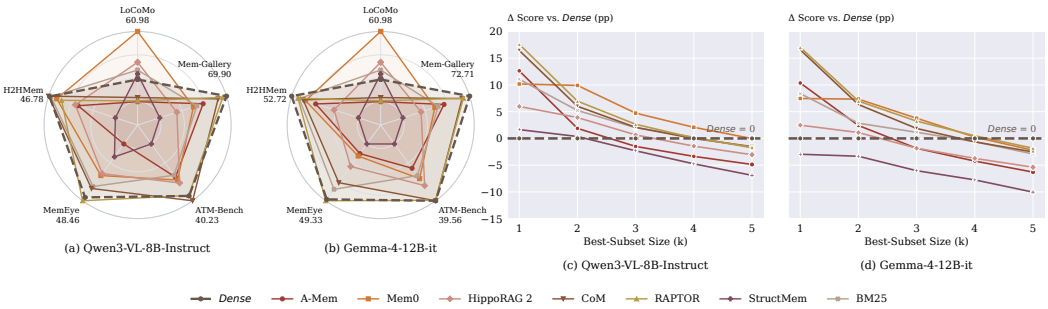

Figure 1: ***Dense* remains competitive as benchmark coverage broadens.** At the matched five-benchmark endpoint, the displayed memory methods are near or below *Dense*. For each smaller subset size, we show each method's best-performing benchmark combination. (a,b) Judge accuracy across benchmarks, min–max scaled separately on each benchmark axis. (c,d) At each number of benchmarks, we select the subset with the highest sample-weighted Judge accuracy for each method. We plot its difference from *Dense*'s sample-weighted average across all five benchmarks.

embeddings by optimizing a differentiable approximation to the reciprocal-rank gap in favor of *Dense*, then uses LLM-driven evolutionary rewriting to approach these targets while preserving the facts each question relies on and its answer. On LoCoMo, our procedure can reverse memory methods' advantage over *Dense* in evidence ranking with 1,540 evidence-supported questions.

These empirical reversals raise a theoretical question: *without assumptions about the user's query distribution, can any memory construction be guaranteed to outperform Dense?* We establish a memory no-free-lunch theorem under common-source ranking and target-symmetry assumptions (Section 3). Under these conditions, improving the ranking for some targets worsens it for others. A universal improvement over *Dense* is therefore impossible within this task family. We further show that changing the task distribution can reverse which method performs better on average. Together, these results highlight how the user's query distribution shapes the benefits of memory construction.

We therefore propose *Dynamic Retrieval Entrances for Agent Memory* (DREAM), which adapts memory access to the user's evolving query distribution. DREAM preserves original records and generates candidate questions that provide alternative routes to their evidence. It summarizes recent queries with kernel herding and matches these representative queries to candidate questions, activating the corresponding retrieval entrances. As user needs change, the active entrances change with them. At retrieval time, DREAM follows these entrance links from the initial dense results to additional source records, making relevant evidence accessible without rewriting the records or changing their embeddings. Across four benchmarks, DREAM improves average Judge accuracy over *Dense* by 4.2 percentage points with Qwen3-VL-8B and 6.2 points with Gemma4-12B.

Our study moves from observation to explanation and design: (1) benchmark comparisons reveal the strength of original-record retrieval (*Dense*); (2) counterexamples and theoretical analysis show why memory benefits depend on the questions asked; and (3) DREAM turns this insight into dynamic memory access guided by the user's query distribution.

2 Does Memory Construction Preserve Its Retrieval Advantage?

Figure 1 shows that memory methods' advantages over *Dense* do not persist across benchmarks. However, these comparisons vary both the memory sources and the questions, leaving their effects unclear. To isolate the effect of question formulation, we keep the memory source, constructed memory and answer fixed. On LoCoMo, where memory methods have shown advantages over *Dense*, we construct counterexamples to test whether rephrasing questions can reverse those advantages.

A controlled test with fixed evidence. We select LoCoMo in which A-Mem, Mem0, MemoryBank or RAPTOR outperformed *Dense* in the original evaluation. For each case, we fix a reference question (the *anchor question*) and its answer, while keeping the original history and constructed memory unchanged. Let $r_{\mathrm{D}}(q)$ and $r_f(q)$ be the first supporting ranks for *Dense* and memory method f. We

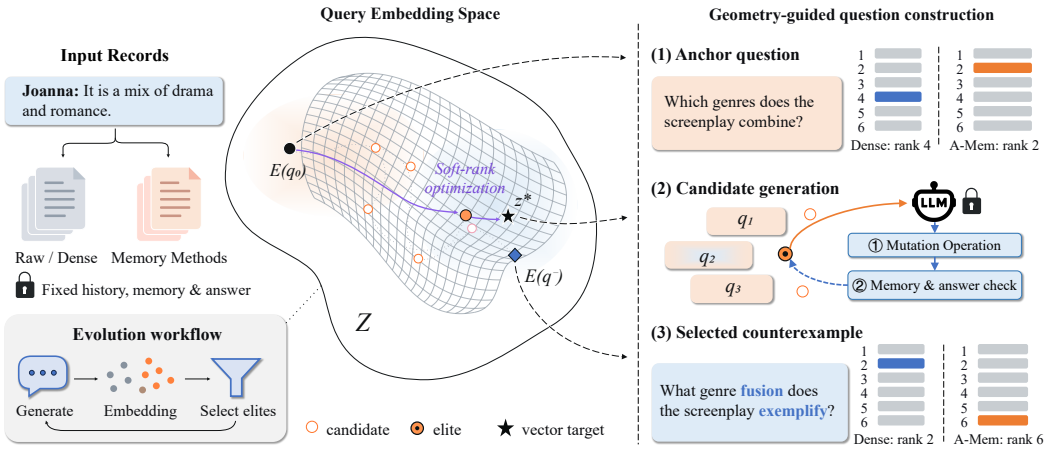

Figure 2: **Geometry-guided question construction.** Soft-rank optimization identifies target query embeddings where *Dense* has a retrieval advantage. LLM-driven rewriting uses selected questions to approach these targets while preserving the source facts and answer.

measure their reciprocal-rank gap as

$$\Delta_{\mathrm{RR}}(q) = \frac{1}{r_f(q)} - \frac{1}{r_{\mathrm{D}}(q)}. \tag{1}$$

A successful counterexample preserves the fact and answer, places *Dense* support in the top-k results ($k = 10$ in our experiments), and has $\Delta_{\mathrm{RR}} < 0$. Before rephrasing the anchor question, we identify the records supporting its fixed answer and keep these evidence judgments unchanged. A reversal occurs when the memory method outperforms *Dense* on the anchor question, but *Dense* outperforms the memory method on a rephrased question asking for the same fact.

Geometry-guided question construction. Figure 2 illustrates the construction procedure. We first define an objective to find query embeddings that maximize *Dense*'s retrieval advantage over the memory method. For a unit query vector z, we maximize

$$\max_{\|z\|_2=1} \mathcal{J}(z) = \frac{1}{\widetilde{r}_{\mathrm{D}}(z)} - \frac{1}{\widetilde{r}_f(z)} - \lambda \left[\max\left\{0, \frac{\widetilde{r}_{\mathrm{D}}(z) - k}{k}\right\}\right]^2, \tag{2}$$

where $\widetilde{r}$ is a differentiable approximation to the first supporting rank. The reciprocal-rank difference favors *Dense*, while the penalty discourages directions that place its supporting evidence outside the top-k results ($\lambda = 0.1$, $k = 10$). The resulting target vectors guide the search for counterexample questions. We then use an LLM-driven genetic search to find natural-language questions that approach these targets while preserving the original factual scope and answer. Across generations, selected questions serve as parents for further rewriting, with similarity to the target vectors guiding the population toward the desired region. The source and answer constraints apply throughout generation, and a final semantic check verifies that the selected question remains natural and asks for the same information. The soft-rank approximation, optimization settings, selection rules and validation protocol are detailed in Appendix D.

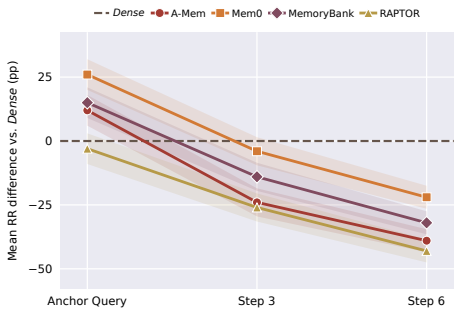

Figure 3: **Counterexample search on LoCoMo.** Trajectories for full evaluation over 1,540 evidence-supported questions. Lines show mean RR differences vs. *Dense*. Shading indicates 95% confidence intervals.

Results and answer to the motivating question. Figure 3 shows the trajectory of the mean reciprocal-rank gap when the evaluation is extended to 1,540 evidence-supported LoCoMo questions.

The dashed zero line indicates equal reciprocal rank: positive values favor the memory method and negative values favor *Dense*. In the depicted trajectories, A-Mem, Mem0 and MemoryBank start above zero, whereas RAPTOR starts slightly below it. All four curves decrease through the rewriting rounds and fall below zero by Step 3, with larger negative gaps at Step 6. The figure thus distinguishes a reversal of an initial memory advantage from a widening of an existing *Dense* advantage. Because memory states and factual targets remain fixed, this is the pattern the question-rewriting procedure seeks to produce: changes in question wording that make the original evidence easier to retrieve than the constructed memory. Figure 4 visualizes the search trajectories in the embedding space, while Figure A1 presents the final questions found in concrete examples. Our empirical results show that a memory method's retrieval advantage can depend on how a question is phrased, even when the underlying evidence and answer remain fixed. This dependence raises a broader question about theoretical guarantees: *without assumptions about the user's query distribution, can any memory construction be guaranteed to outperform Dense?*

3 Why Memory Benefits Depend on the Query Distribution

To answer the question posed at the end of Section 2, we establish a memory no-free-lunch theorem under explicit assumptions, then examine how the user's query distribution determines which method performs better. Appendix C gives the full statement and proof.

A no-free-lunch result for memory retrieval. Suppose both methods rank the same N evidence sources and, for each query q, any source i is an admissible correct target. This target-symmetry assumption tests a guarantee across all possible targets. Let $r_f(q,i)$ and $r_{\mathrm{D}}(q,i)$ denote the ranks assigned to source i, and let $w(r)$ be a strictly decreasing rank utility, such as reciprocal rank. Since both rankings permute the same positions,

$$\sum_{i=1}^{N}\big[w(r_f(q,i)) - w(r_{\mathrm{D}}(q,i))\big] = 0. \tag{3}$$

Any change in ranking must therefore improve the score for some targets and reduce it for others. Under these assumptions, no memory construction can be at least as good as *Dense* on every task while being strictly better on some task. The result applies to rankings over a common set of sources. Target symmetry allows all possible targets for each query rather than assuming the semantic constraints of real user questions.

Why the user's query distribution matters. When a method helps on some tasks and hurts on others, its average advantage depends on how frequently each occurs. Let $\mu_+ > 0$ and $\mu_- < 0$ be its average gaps relative to *Dense* under two task distributions. A mixture that assigns weight α to the latter has gap $(1-\alpha)\mu_+ + \alpha\mu_-$. Thus, changing the task distribution can reverse which method performs better (Corollary 1). These results motivate adapting memory organization to the questions users actually ask. DREAM uses recent queries to adjust retrieval entrances, aligning access to original evidence with evolving user demand.

4 Dynamic Retrieval Entrances for Agent Memory

This analysis motivates organizing memory around the user's query distribution. We therefore propose *Dynamic Retrieval Entrances for Agent Memory* (DREAM), which adapts retrieval entrances while preserving original records and embeddings. As Figure 5 shows, DREAM generates reusable candidate questions, activates their entrances using recent queries, and follows active links from dense results to additional evidence. We formalize the objective before describing these three stages.

Objective: align memory access with changing query demand. Let P_t denote the user's query distribution at time t in a shared semantic space $\mathcal{Z}$. For the available memory archive M_t, let θ_t specify its access configuration and let Q_{θ_t} denote the corresponding memory-access distribution on $\mathcal{Z}$, describing which semantic content the configuration makes accessible. Our objective is

$$\theta_t^* \in \arg\min_{\theta_t \in \Theta(M_t)} \mathcal{D}(P_t, Q_{\theta_t}), \tag{4}$$

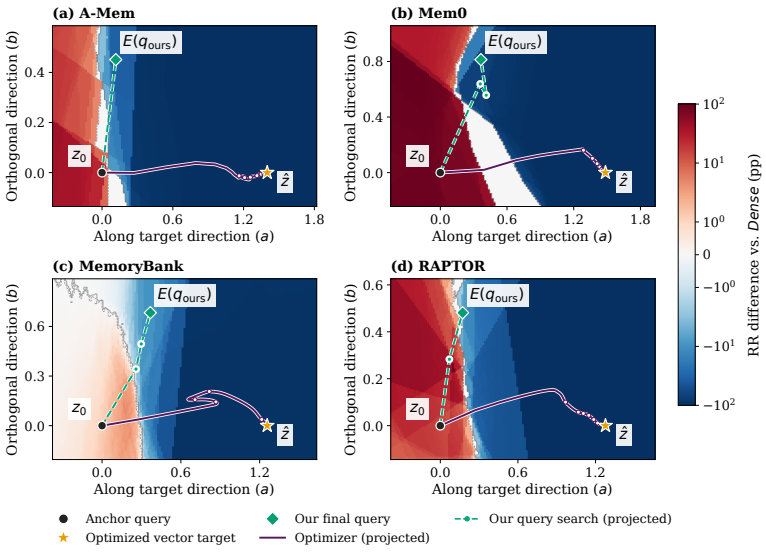

Figure 4: **Examples of counterexample search.** Purple and green paths show vector optimization and question search, respectively. Backgrounds show the RR difference vs. *Dense* ($\times 100$): red favors Memory methods, blue favors *Dense*.

where $\mathcal{D}$ measures distributional discrepancy and $\Theta(M_t)$ contains feasible access configurations under the retrieval budget that preserve the original records. As user demand changes, the desired access distribution changes with it. This objective specifies what memory should adapt to. The following construction explains how DREAM represents access cues and adjusts them using queries.

Express memory through reusable query cues. To adapt memory access to changing query demand, we generate candidate questions answerable from each record as retrieval cues. These questions provide alternative ways to access the record. Selecting among them aligns memory access with recent queries while retaining the complete record as evidence for answering.

Let $M_t = \{m_1, \ldots, m_{n_t}\}$ be the available records. For each m_i, we generate a pool C_i of up to B_c candidate questions, with p_{ij} denoting its jth question. Since the pool describes the record's content, we can prepare it offline when the record becomes available and reuse it across requests. Online adaptation then selects which cues to activate according to recent queries. This separates the possible ways to access a record from the subset emphasized at a given time, providing a concrete access configuration for the alignment objective above.

Align candidate activation with query demand. The user's query distribution P_t is difficult to observe directly and may change over time. We approximate current demand using the latest W queries, including q_t, with I_t denoting their time indices. To track shifts in demand, we give recent queries greater weight through exponential time decay: $\alpha_s = \gamma^{t-s} / \sum_{u \in I_t} \gamma^{t-u}$, where $0 < \gamma \leq 1$ controls the decay rate.

Recent queries may repeatedly express similar needs. Using each as a separate alignment target would devote many targets to the same semantic region, whereas selecting only diverse queries would ignore how often those needs occur. We use kernel herding (Chen et al., 2012) to summarize the weighted query distribution with representative targets that balance frequency and diversity. Let $E(\cdot)$ return a unit-normalized embedding. For kernel bandwidth $\tau > 0$, we define query similarity and the weighted density around each observed query as

$$K_{su} = \exp\left(\frac{E(q_s)^\top E(q_u) - 1}{\tau}\right), \qquad \rho_s = \sum_{u \in I_t} \alpha_u K_{su}. \tag{5}$$

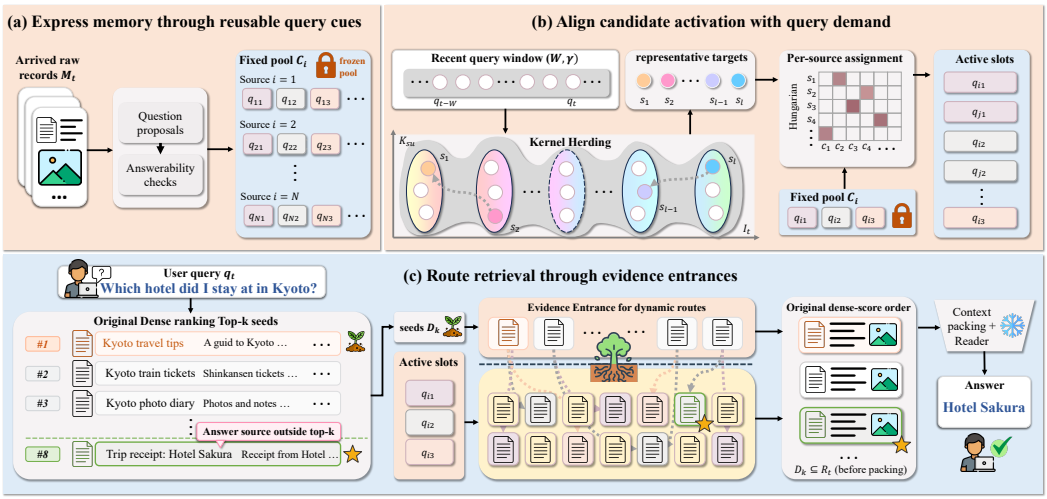

Figure 5: **Overview of DREAM.** (a) *Express memory through reusable query cues:* generate reusable candidate questions from original records. (b) *Align candidate activation with query demand:* select representative recent queries with kernel herding and activate candidate questions through Hungarian matching. (c) *Route retrieval through evidence entrances:* follow active entrance links from *Dense* results to additional source records and use the retrieved evidence to answer the question.

We select up to L representative queries sequentially. Given the previously selected indices $T_{\ell-1}$, we choose the next query by

$$s_\ell = \arg\max_{s \in I_t \setminus T_{\ell-1}} \left[\rho_s - \frac{1}{\ell} \sum_{u \in T_{\ell-1}} K_{su} \right], \quad T_\ell = T_{\ell-1} \cup \{s_\ell\}. \tag{6}$$

The density term ρ_s preserves emphasis on frequent needs, while the second term discourages redundant representatives. The final index set T summarizes recent demand for cue activation.

We adapt memory access by activating candidate questions aligned with these demand representatives, with $B_{\mathrm{a}} \le B_{\mathrm{c}}$ controlling how many candidate questions can be active per record. We use Hungarian matching to assign targets in herding order to distinct candidate questions: its one-to-one constraint prevents repeated selection of the same question, preserving activation capacity for different needs. The assignment maximizes total semantic similarity:

$$\max_{\pi \text{ injective}} \sum_{\ell=1}^{\min(B_{\mathrm{a}}, |C_i|, |T|)} E(p_{i,\pi(\ell)})^\top E(q_{s_\ell}), \qquad \pi(\ell) \in \{1, \ldots, |C_i|\}. \tag{7}$$

As the query window changes, the active candidate questions follow the updated demand representatives, implementing the alignment objective through selection from reusable pools.

Route retrieval through evidence entrances. A relevant record may fall outside the initial retrieval results when its content embedding does not match the user's question. Test-time embedding adjustments (Sung et al., 2023) or query rewriting (Ma et al., 2023) still rely on direct matching to reach the relevant evidence. We keep embeddings and queries unchanged and build dynamic associations between memory records, so a retrieved record can lead to evidence that direct matching misses. This design is consistent with neuroscientific findings on associative recall: a partial cue reactivates linked event information absent from the cue (Horner et al., 2015; Yadav et al., 2022).

Candidate questions provide the bridge: each links a semantically matching record to its answer-bearing source. For candidate question p_{ij}, the matching record becomes its retrieval entrance:

$$a_t(i,j) = \arg\max_{h \le n_t,\ h \ne i} E(p_{ij})^\top E(m_h). \tag{8}$$

When p_{ij} is activated in response to query demand, its link $a_t(i,j) \to i$ becomes available, allowing retrieval of the entrance to lead to the source m_i.

Table 1: **Qwen3-VL-8B on four benchmarks.** QS/EM/M.EM/Recall: task metrics (M.EM: multiple-choice exact match); J.Acc.: Judge accuracy (%). Relative to *Dense*, ■ / ■ indicate gains / losses; shade intensity scales with the absolute gap from 0 to 5 percentage points, saturating at 5. **Bold**: best non-Oracle score; Underline: second-best non-Oracle score (ties included).

	ATM		Mem-Gallery		MemEye		H2HMem		Avg.
Method (venue/year)	QS	J.Acc.	EM	J.Acc.	M.EM	J.Acc.	Recall	J.Acc.	J.Acc.
Oracle	57.8	54.5	64.5	86.5	63.1	61.7	34.8	54.1	64.2
Context and retrieval controls									
No memory	2.1	2.2	20.2	29.2	42.7	41.0	15.1	26.2	24.6
Recency (20 items)	2.6	2.7	28.5	38.1	41.8	40.9	18.9	28.6	27.6
Full arrived window (8k)	3.6	3.6	39.7	56.3	45.5	44.3	24.8	39.0	35.8
BM25 top-k	37.1	35.2	44.7	62.4	48.5	46.9	31.8	46.5	47.7
Dense top-k	41.4	39.3	51.7	69.9	49.7	48.1	32.3	46.7	51.0
Memory methods									
RAPTOR† *(ICLR 2024)*	40.9	39.2	49.9	68.9	49.9	48.5	30.2	45.4	50.5
A-Mem *(NeurIPS 2025)*	37.7	35.5	48.1	63.9	42.5	42.3	28.0	43.6	46.3
HippoRAG 2 *(ICML 2025)*	38.4	36.7	41.8	57.3	46.8	45.5	28.3	43.9	45.9
Mem0 *(ECAI 2025)*	35.3	36.1	40.7	61.5	47.0	45.7	30.9	45.9	47.3
CoM *(ACL 2026)*	42.9	40.2	50.6	67.7	49.0	47.1	30.7	46.8	50.5
StructMem *(ACL 2026)*	28.9	28.9	33.0	53.0	45.0	43.7	27.5	39.7	41.3
LightMem *(ICLR 2026)*	16.9	18.4	26.9	39.5	43.3	42.5	21.0	32.5	33.2
DREAM (ours)	**47.5**	**44.6**	**54.2**	**73.7**	**55.5**	**54.1**	**33.4**	**48.5**	**55.2**

At retrieval time, we first obtain the *Dense* results and then follow active entrance links to additional sources. Let $D_K(q_t)$ denote the initial top-K records, and let $A_{i,t}$ contain the active candidate-question indices for record m_i. One-hop expansion gives

$$R_t = D_K(q_t) \cup \{i : \exists j \in A_{i,t},\ a_t(i,j) \in D_K(q_t)\}. \tag{9}$$

We order the expanded set by its original dense scores, retaining the initial results while making evidence missed by direct embedding matching accessible through active entrances. Other details are given in Appendix A.

5 Experiments

Implementation Details. We use a candidate-question budget $B_c = 16$ and an activation budget $B_a = 8$, with query-window size $W = 32$, herding target count $L = 8$, recency factor $\gamma = 0.95$ and kernel bandwidth $\tau = 0.1$. We encode records, candidate questions and user queries with GME-Qwen2-VL-2B-Instruct, using unit-normalized 1,536-dimensional text embeddings. Additional implementation details and evaluation protocols are provided in Appendices A and B.

Main Results. Tables 1 and 2 compare systems across four benchmarks with Qwen3-VL-8B and Gemma4-12B, respectively. *Dense* remains competitive with the existing memory baselines: their gains are confined to particular benchmarks and metrics. DREAM achieves the highest non-Oracle Judge accuracy on all four benchmarks with both Readers, improving over *Dense* by 1.8–6.0 percentage points with Qwen and 3.4–9.8 points with Gemma. Average Judge accuracy rises from 51.0% to 55.2% with Qwen and from 53.5% to 59.7% with Gemma. The largest Judge accuracy gains occur on MemEye for both Readers. Native metrics improve over *Dense* in all eight benchmark–Reader settings under the selected packing strategies. The gains therefore extend across Readers but vary with the benchmark and scoring metric. These system comparisons use their respective retrieval protocols.

Does routing help beyond simply reading more? We compare DREAM with four retrieval baselines in Table 3 on the newly constructed human-reviewed evaluation questions, using the

Table 2: **Gemma4-12B on four benchmarks.** QS/EM/M.EM/Recall: task metrics (M.EM: multiple-choice exact match); J.Acc.: Judge accuracy (%). Relative to *Dense*, ■ / ■ indicate gains / losses; shade intensity scales with the absolute gap from 0 to 5 percentage points, saturating at 5. **Bold**: best non-Oracle score; Underline: second-best non-Oracle score (ties included).

	ATM		Mem-Gallery		MemEye		H2HMem		Avg.
Method (venue/year)	QS	J.Acc.	EM	J.Acc.	M.EM	J.Acc.	Recall	J.Acc.	J.Acc.
Oracle	59.5	55.7	63.8	89.7	70.4	69.3	42.7	61.5	69.0
Context and retrieval controls									
No memory	1.8	1.9	13.0	17.6	33.6	31.9	10.2	24.0	18.8
Recency (20 items)	2.4	2.5	24.6	31.2	31.9	32.4	17.4	30.7	24.2
Full arrived window (8k)	3.1	3.2	42.7	63.4	47.0	48.8	30.8	44.7	40.0
BM25 top-k	35.6	34.0	43.5	61.9	47.7	47.6	34.6	51.6	48.7
Dense top-k	41.7	39.6	50.2	72.7	48.7	49.1	34.7	52.7	53.5
Memory methods									
RAPTOR† *(ICLR 2024)*	41.5	39.5	46.6	70.5	48.9	49.3	34.7	51.8	52.8
A-Mem *(NeurIPS 2025)*	34.6	32.4	44.2	63.9	41.0	42.0	33.2	49.0	46.8
HippoRAG 2 *(ICML 2025)*	38.8	36.2	40.7	56.0	43.3	44.0	27.6	46.2	45.6
Mem0 *(ECAI 2025)*	33.4	34.7	37.0	60.7	43.3	42.4	32.2	50.6	47.1
CoM *(ACL 2026)*	41.7	39.6	48.6	70.1	45.2	46.5	33.2	50.9	51.8
StructMem *(ACL 2026)*	27.9	27.0	27.9	49.7	40.0	40.5	27.6	42.4	39.9
LightMem *(ICLR 2026)*	18.2	18.8	22.7	33.9	30.8	32.5	18.1	32.8	29.5
DREAM (ours)	**47.5**	**43.2**	**53.4**	**80.5**	**57.4**	**58.9**	**38.7**	**56.1**	**59.7**

Table 3: Accuracies (%) on three human-reviewed evaluation-question sets constructed from ATM, Mem-Gallery, MemEye and H2HMem (Appendix B.1). All methods use the same evidence budget of 8,192 text tokens and 16 media inputs. Q/G denotes the Qwen3-VL-8B and Gemma4-12B Readers; the Judge is Qwen3-30B-A3B-Instruct-2507-FP8.

Method	ATM Q/G	Mem-Gallery Q/G	MemEye Q/G	H2HMem Q/G
Dense	42.8/44.1	72.4/78.2	52.3/57.4	48.9/54.6
Direct candidate retrieval	44.3/45.0	73.8/79.4	54.0/58.9	50.4/56.2
Window candidate retrieval	45.0/45.7	74.4/80.0	54.8/59.5	51.0/56.9
Raw-neighbor expansion	43.6/44.7	73.1/78.8	53.1/58.0	49.6/55.4
DREAM	**46.6/47.4**	**76.3/81.9**	**56.7/61.3**	**53.0/58.8**

same retrieved-evidence budget for all methods to test whether gains come from better evidence selection rather than a larger capacity allowance. The baselines form three categories: (1) *Dense* retrieval: rank original records by query similarity. (2) Candidate-question retrieval: match the current query to candidate questions and return their source records. The window variant also uses recent queries. (3) Record-similarity expansion: raw-neighbor expansion builds links from similarity between original records. These controls test whether entrance routing adds value beyond generated questions and ordinary record-similarity links. With DREAM 3.3–4.4 percentage points above *Dense* and 2.3–2.7 points above direct candidate retrieval, these gains show that entrance routing improves evidence selection under a fixed budget, beyond simply generating candidate questions. Retrieval implementations are described in Appendix B.2.

Which components make memory adapt to query demand? Table 4 lays out the effects of removing or replacing each component across the four benchmarks. (1) Without query history, activation follows only the current question. A loss of 2.6–6.1 percentage points captures the value of recent context in identifying which memory cues matter. (2) Substituting farthest-point sampling for kernel herding emphasizes query diversity over demand frequency, the 2.1–5.6 point loss tests the benefit of balancing both. (3) Selecting candidates independently for each target replaces joint

Table 4: Ablation on all four benchmarks. Q/G denotes the Qwen3-VL-8B and Gemma4-12B Readers. Both are scored by Qwen3-30B-A3B-Instruct-2507-FP8.

Configuration	ATM Q/G	Mem-Gallery Q/G	MemEye Q/G	H2HMem Q/G
Current-query activation	41.8/40.6	68.8/74.4	50.9/56.0	45.8/53.0
Farthest-point sampling	42.4/41.1	69.1/74.9	51.4/56.5	46.2/53.5
Independent matching	42.8/41.3	69.7/75.3	51.9/56.9	46.5/53.8
DREAM	**44.6/43.2**	**73.7/80.5**	**54.1/58.9**	**48.5/56.1**

Hungarian matching. The 1.8–5.2 point loss tests whether coordinating these choices makes better use of the available activation slots. These contrasts clarify what each component should contribute.

6 Related Work

Memory representation and evidence organization. Long-term agent memory has been organized through persistent stores, hierarchies, and explicit associations. MemoryBank and Mem0 maintain information across interactions (Zhong et al., 2024; Chhikara et al., 2025), while A-Mem constructs structured notes and evolves their links and attributes as new memories arrive (Xu et al., 2026b). RAPTOR recursively clusters and summarizes text into a retrieval tree (Sarthi et al., 2024), and MemoryOS coordinates short-, mid-, and long-term storage (Kang et al., 2025). SeCom uses topically coherent conversation segments (Pan et al., 2025). LightMem, StructMem and CoM further explore memory consolidation, event structure and dynamic evolution (Fang et al., 2026; Xu et al., 2026a;c). Beyond individual memory units, HippoRAG 2 supports associative retrieval (Gutiérrez et al., 2025), and GraphMemix selects query-relevant evidence forests under utility and resource constraints (Li et al., 2026). While these methods organize memory through summaries, evolving notes or structured evidence graphs, DREAM focuses on organizing access to original records. It uses source-specific question cues to form retrieval entrances and dynamically activates them as the user's query distribution changes, allowing the same intact evidence to be reached through different routes as information needs evolve.

Query-aware retrieval and adaptation. Query-aware methods address the mismatch between a user's information need and the form in which memory is stored. QueryLink aligns queries and memories in a shared semantic space and combines this alignment with multigrained retrieval (Hu et al., 2026). RMM uses retrospective reflection to refine retrieval through online reinforcement learning from LLM-cited evidence (Tan et al., 2025). In dense retrieval, TOUR optimizes test-time query representations using reranker-derived relevance signals (Sung et al., 2023), while ReFIT distills reranker feedback into a query vector before a second retrieval pass (Reddy et al., 2023). DREAM focuses on how memory should be organized as questions arrive, using the evolving query distribution to update retrieval entrances. This keeps answer-time retrieval simple, without iterative query optimization or neural reranking, while memory organization adapts to changing user needs.

7 Discussion and Conclusion

The value of memory construction depends on the questions it serves. Reorganizing a history can make evidence easier to retrieve for some questions and harder for others, even when the underlying facts remain unchanged. Our empirical and theoretical findings therefore place the user's query distribution at the center of memory design. DREAM puts this principle into practice: it preserves original evidence while adapting retrieval entrances to evolving demand. Its gains across four benchmarks and two Readers support this approach. For long-term AI assistants, effective memory requires more than retaining the past. It also requires adapting how that past becomes accessible as users' needs change. We discuss the limitations of this work in Appendix I.

AI use statement

We used Codex to assist with manuscript editing, checks of consistency between the text and tables, and LaTeX preparation and compilation. Our research also uses language models for candidate-

question generation, counterexample rewriting, support labeling, semantic verification, and answer evaluation, as described in the method and evaluation protocols. Further details of these uses are provided in Appendices A, B, D, and G. The limitations of the model-based semantic checks are discussed in Appendix I. We take responsibility for the final content of this work, including AI-assisted text, claims, and artifacts.

REFERENCES

Yuanchen Bei, Tianxin Wei, Xuying Ning, Yanjun Zhao, Zhining Liu, Xiao Lin, Yada Zhu, Hendrik Hamann, Jingrui He, and Hanghang Tong. Mem-gallery: Benchmarking multimodal long-term conversational memory for mllm agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 40750–40784, 2026.

Yutian Chen, Max Welling, and Alex Smola. Super-samples from kernel herding. *arXiv preprint arXiv:1203.3472*, 2012.

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.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, et al. Lightmem: Lightweight and efficient memory-augmented generation. In *International Conference on Learning Representations*, volume 2026, pp. 98706–98729, 2026.

Minghao Guo, Qingyue Jiao, Zeru Shi, Yihao Quan, Boxuan Zhang, Danrui Li, Liwei Che, Wujiang Xu, Shilong Liu, Zirui Liu, et al. Memeye: A visual-centric evaluation framework for multimodal agent memory. *arXiv preprint arXiv:2605.15128*, 2026.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*, 2025.

Aidan J Horner, James A Bisby, Daniel Bush, Wen-Jing Lin, and Neil Burgess. Evidence for holistic episodic recollection via hippocampal pattern completion. *Nature communications*, 6(1):7462, 2015.

Xuxian Hu, Zhu Teng, Wei Zhang, Ming He, and Jianping Fan. Querylink: Leveraging query-memory alignment for long-term reasoning in llm agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 15608–15621, 2026.

Hongda Jiang, Xinyuan Zhang, Siddhant Garg, Rishab Arora, Shiun-Zu Kuo, Jiayang Xu, Aaron Colak, and Xin Luna Dong. Memory-qa: Answering recall questions based on multimodal memories. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 24255–24277, 2025.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory os of ai agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25972–25981, 2025.

Geng Li, Yuhao Wang, Dong Li, Jianye Hao, and Yuxin Peng. Graphmemix: Query-aware evidence forests for long-term multimodal agent memory. *arXiv preprint arXiv:2608.26983*, 2026.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 5303–5315, 2023.

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

Jingbiao Mei, Jinghong Chen, Guangyu Yang, Xinyu Hou, Margaret Li, and Bill Byrne. According to me: Long-term personalized referential memory qa. *arXiv preprint arXiv:2603.01990*, 2026.

Chang Nie, Chaoyou Fu, Yifan Zhang, Haihua Yang, and Caifeng Shan. Personavlm: Long-term personalized multimodal llms. *arXiv preprint arXiv:2604.13074*, 2026.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H Vicky Zhao, Lili Qiu, et al. Secom: On memory construction and retrieval for personalized conversational agents. In *International Conference on Learning Representations*, volume 2025, pp. 91851–91885, 2025.

Revanth Gangi Reddy, Pradeep Dasigi, Md Arafat Sultan, Arman Cohan, Avirup Sil, Heng Ji, and Hannaneh Hajishirzi. Refit: Relevance feedback from a reranker during inference. *arXiv preprint arXiv:2305.11744*, 2023.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*, volume 2024, pp. 32628–32649, 2024.

Mujeen Sung, Jungsoo Park, Jaewoo Kang, Danqi Chen, and Jinhyuk Lee. Optimizing test-time query representations for dense retrieval. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 5731–5746, 2023.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, et al. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8416–8439, 2025.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788, 2020.

Buqiang Xu, Yijun Chen, Jizhan Fang, Ruobin Zhong, Yunzhi Yao, Yuqi Zhu, Lun Du, and Shumin Deng. Structmem: Structured memory for long-horizon behavior in llms. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 122–146, 2026a.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604, 2026b.

Xiucheng Xu, Bingbing Xu, Tian Xueyun, Zihe Huang, Rongxin Chen, Li Yunfan, and Huawei Shen. Chain-of-memory: Lightweight memory construction with dynamic evolution for llm agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11618–11631, 2026c.

Nakul Yadav, Chelsea Noble, James E Niemeyer, Andrea Terceros, Jonathan Victor, Conor Liston, and Priyamvada Rajasethupathy. Prefrontal feature representations drive memory recall. *Nature*, 608(7921):153–160, 2022.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 19724–19731, 2024.

Shiping Zhu, Yibo Yang, Zhengyang Wang, Tiancheng Shen, Dandan Guo, and Ming-Hsuan Yang. H2hmem: A multimodal memory benchmark for agents in human-human interactions. *arXiv preprint arXiv:2606.09461*, 2026.

A DREAM IMPLEMENTATION AND REPRODUCIBILITY

Default parameters are $B_c = 16$, $B_a = 8$, $W = 32$, $\gamma = 0.95$, $L = 8$ and $\tau = 0.1$. The current query is included in the window. The reference replay uses cached candidates. Candidate questions are generated from each record alone and filtered for answerability by Qwen3-30B-A3B-Instruct-2507-FP8.

Encoding and availability. The cached service uses normalized 1,536-dimensional GME-Qwen2-VL-2B-Instruct text embeddings with a 4,096-token service limit. The service encodes text, including available media descriptions, through `get_text_embeddings`. Candidate embeddings are float32. Herding computes its window kernel in float64. Only candidates owned by an arrived source are activated. Compatibility uses only the current and earlier queries.

Readout and packing. Compute the stable complete dense ordering of arrived records. Select the first K as seeds, find every owner whose active entrance list intersects those seeds, and return their union in the same dense ordering. Duplicate owners are removed. This is reverse lookup from seed entrances to candidate owners, not following an owner's list as outgoing retrieval edges. Do not expand newly linked owners. The reference code evaluates $K \in \{10, 20\}$ and has no final record or token cutoff. It asserts that the initial K returned records equal the dense seed prefix. A separate answering adapter may pack this output under text/media limits.

Complexity and cost. With n arrived records, $P \leq 16n$ feasible candidates, dimension d and window size W, target selection costs $O(W^2 d + LW^2)$ with a simple bound. A bounded implementation can compute candidate compatibility in $O(PWd)$. The current replay instead multiplies against the entire observed prefix before indexing the window, costing $O(Ptd)$ and storing that compatibility matrix. For each owner, let r_i be the number of assigned targets and c_i the number of valid candidates. Per-owner assignment is small ($r_i \leq 8$, $c_i \leq 16$). A generic dense assignment bound is $O(\max(r_i, c_i)^3)$. Entrance comparisons can total $O(Pnd)$ over archive growth. Dense ranking costs $O(nd + n \log n)$ and the reference one-hop readout scans all owner lists rather than using an inverted adjacency index. Storage includes raw records, candidate vectors and graph state.

B EVALUATION QUESTIONS AND RETRIEVAL CONTROLS

B.1 HUMAN-REVIEWED EVALUATION QUESTIONS

We construct a supplementary evaluation set by aiming to assign the same number of questions to each selected memory record. Reusing benchmark questions alone does not enforce this balance: records with more questions can contribute more to aggregate accuracy. A common per-record quota reduces this imbalance while retaining the original benchmark memories. These evaluation questions are separate from the candidate questions used inside DREAM. The retrieval candidate budget remains $B_c = 16$, with $B_a = 8$ active questions per record. We use three evaluation-question generation seeds, 17, 29 and 43, keep the algorithm's candidate pool fixed across these three evaluation sets.

Proposals and human selection. For each selected original memory record, we ask Qwen3-30B-A3B-Instruct-2507-FP8 to propose up to 16 realistic user questions, in two batches of eight. Each proposal includes an answer and supporting evidence from the original record. The second call receives the first batch to avoid duplicate questions. Both use temperature 0.2, top-$p = 1$ and a 2,400-token output limit. Generated questions can be repetitive or poorly aligned with realistic user needs. To prioritize quality over quantity, human reviewers retain at most four distinct questions per record that a user could naturally ask in everyday life and whose answers are correct and fully supported by the original memory. Reviewers resolve ambiguous references and remove redundant questions. If fewer than four survive, reviewers write additional questions and reference answers from the same memory, checking each against its supporting evidence. Do not invent facts to meet the quota. Record any remaining shortfall.

Data sources and evaluation size. The source datasets are ATM, Mem-Gallery, MemEye and H2HMem. We set an evaluation budget of 100 original memory records per benchmark and four

human-reviewed questions per record. This gives 400 questions per benchmark and 1,600 per evaluation set. The three generation seeds yield 4,800 questions (1,200 per benchmark). Records are selected without using retrieval scores, and the same records are used across all three sets. Counts refer to question instances across sets.

B.2 Direct Retrieval and Evidence-Selection Controls

These controls test whether the generated questions alone explain the gain, without requiring the proposed entrance graph. All candidate-based methods share each pool's valid questions, embeddings, source arrival boundary and raw-evidence serialization. Supporting quotes and filter outputs never become retrieval features. Empty candidate pools remain in the archive. Every control is packed after ranking, using the same text and native-media caps. Only complete raw records can enter the Reader. No control sees outcome labels.

Direct candidate retrieval. Let E denote the normalized embedding encoder. For source record m_i and its candidate question p_{ij}, write $x_i = E(m_i)$ and $v_{ij} = E(p_{ij})$. Let C_i index its valid candidates and $z_q = E(q)$ represent the current query q. For source i, score $b_i(q) = \max_{j \in C_i} z_q^\top v_{ij}$ and return owners in descending score order, resolving ties by source order. When $C_i = \varnothing$, use $z_q^\top x_i$. Search all feasible slots, not only the active subset. Deduplicate owners before packing. Thus this baseline receives the complete source-question representation without entrance construction.

Window candidate retrieval. Let I_t index the available query window at time t, α_s be the normalized recency weight of query q_s, and $z_s = E(q_s)$. Using the same window and weights as DREAM, score each source by

$$b_i^{\mathrm{win}}(q) = \max_{j \in C_i} \left[\tfrac{1}{2} z_q^\top v_{ij} + \tfrac{1}{2} \sum_{s \in I_t} \alpha_s z_s^\top v_{ij} \right].$$

For an empty pool substitute x_i for v_{ij}. The half-weight is fixed before scoring. The current question also retains its ordinary window weight. Rank owners directly. This grants recent-query information to a graph-free candidate baseline, rather than comparing history-aware routing only with current-query retrieval.

Raw-neighbor expansion. Replace candidate-derived entrances with each owner's eight nearest non-self raw records by $x_i^\top x_h$ (or all available neighbors if fewer exist). Keep edge direction, ten dense seeds, one-hop readout, dense-score output ordering and packing unchanged. This isolates the source-question entrances from ordinary similarity edges without introducing extra Reader evidence caps.

C Memory No-Free-Lunch Theorem and the Role of Query Distributions

Can memory construction guarantee an advantage over *Dense* without assumptions about the user's query distribution? We first explain what such a guarantee requires. We then prove that it is impossible under common-source ranking and target-symmetry assumptions, and show how changing the query distribution can reverse which method performs better.

C.1 What a guarantee over unknown queries requires

Fix a history, a *Dense* state S_{D}, a constructed state S_f, a finite nonempty task set $\mathcal{T}$, and a utility $u(S,t) \in [0,1]$. A task $t \in \mathcal{T}$ includes the question and its evaluation target. Let P be a probability distribution over the task set $\mathcal{T}$. Define

$$g_f(t) = u(S_f, t) - u(S_{\mathrm{D}}, t), \qquad U(S,P) = \sum_{t \in \mathcal{T}} P(t) u(S,t). \tag{10}$$

Proposition 1 (Uniform dominance requires pointwise dominance)**.** *If P ranges over all distributions on $\mathcal{T}$, then $U(S_f,P) \geq U(S_{\mathrm{D}},P)$ for every P if and only if $g_f(t) \geq 0$ for every $t \in \mathcal{T}$.*

Proof. Nonnegative pointwise gaps have nonnegative probability-weighted sums. If there exists a task $t_0 \in \mathcal{T}$ such that $g_f(t_0) < 0$, the point mass on t_0 gives a negative expected gap, contradicting uniform dominance. □

For a fixed history and memory construction, finding even one question on which the memory method performs worse than *Dense* is enough to show that it is not always at least as good as *Dense*.

C.2 Rank conservation under target symmetry

We now give conditions under which changing the retrieval ranking necessarily makes the memory method score lower than *Dense* on some tasks. Under these conditions, such tasks follow from the ranking change itself. We do not assume that they exist. These are additional assumptions, not consequences of query independence:

1. **Common candidate universe.** Both systems rank the same $N \geq 2$ evidence sources, with no duplicates or omissions. For every observable query q in a finite nonempty set $\mathcal{Q}$, each returns a deterministic permutation. Ties use a fixed rule. Construction receives the history but no target-task information beyond the query available to retrieval.
2. **Target symmetry.** The admissible task family is $\mathcal{T} = \mathcal{Q} \times \{1, \ldots, N\}$. For each observable q, any source i may be the unique relevant target. The ranker receives the query q but is not told which source i is the correct target.
3. **Rank utility.** For task (q, i), utility is $u(S, (q, i)) = w(r_S(q, i))$, where $r_S(q, i)$ is source i's rank and $w : \{1, \ldots, N\} \to [0, 1]$ is strictly decreasing. Reciprocal rank uses $w(r) = 1/r$.

Under target symmetry, we consider every source as a possible correct target for each query. This assumption lets us test a guarantee against all possible targets. It does not mean that a real question can have its correct target changed arbitrarily without changing what the question asks.

Theorem 1 (No free lunch under target symmetry)**.** *Under the three assumptions, every fixed constructor has the same expected utility as Dense when the relevant source is uniform conditional on each query. No such constructor can satisfy $U(S_f, P) \geq U(S_{\mathrm{D}}, P)$ for every distribution P on $\mathcal{T}$ while being strictly better on any task. If its ranking differs from Dense for at least one query, there are admissible distributions P_+ and P_- for which its expected utility is respectively strictly higher and strictly lower than Dense's.*

Proof. For every fixed q, both rank vectors are permutations of $1, \ldots, N$. Consequently,

$$\sum_{i=1}^{N} g_f(q, i) = \sum_{r=1}^{N} w(r) - \sum_{r=1}^{N} w(r) = 0. \tag{11}$$

Dividing by N and averaging over any marginal distribution of q proves the first assertion. If all gaps were nonnegative and at least one were positive, their sum at that query would be positive, contradicting Equation 11. If a ranking differs, some rank differs. Strict decrease of w makes the corresponding gap nonzero. A finite zero-sum vector with a nonzero entry has both a positive and a negative entry. Point masses on these two admissible tasks define P_+ and P_-. □

The argument does not depend on the embedding model used by *Dense*: it applies whenever two methods rank the same set of evidence sources. Under the stated assumptions, changing that ranking cannot help every possible target without hurting another. Which method works better for real users still depends on which questions they ask and which evidence those questions require. The theorem does not directly cover a method that replaces original records with a different number of summaries, because the two methods then rank different sets of items. To apply it in that setting, both rankings must first be expressed over the same original sources in a way that faithfully represents their retrieval behavior. Proposition 1 only requires a common measure of task performance and therefore still applies.

C.3 WHY WORKLOAD INFORMATION MATTERS

Corollary 1 (Preference changes with the workload)**.** *Suppose two admissible workloads have gaps $\mu_+ > 0$ and $\mu_- < 0$. For $P_\alpha = (1-\alpha)P_+ + \alpha P_-$,*

$$U(S_f, P_\alpha) - U(S_{\mathrm{D}}, P_\alpha) = (1-\alpha)\mu_+ + \alpha\mu_-. \tag{12}$$

The preferred system changes at $\alpha^* = \mu_+/(\mu_+ - \mu_-)$.

Proof. Linearity of expectation gives the displayed expression. Its slope is $\mu_- - \mu_+ < 0$, and its unique zero lies in $(0, 1)$ because both sign assumptions are strict. □

When the rankings differ, the theorem guarantees both favorable and unfavorable tasks under target symmetry. For real user questions, we must check whether both kinds of tasks actually occur. If they do, their relative frequency determines which method performs better on average. A memory method can therefore outperform *Dense* for one user query distribution and underperform it for another. This motivates using observed questions to identify which evidence users need and adapting memory access to those needs.

D COUNTEREXAMPLE QUESTIONS AND SEARCH PROTOCOL

Embedding and optimizer. We use the frozen all-MiniLM-L6-v2 encoder, normalize its 384-dimensional embeddings, and compute stable exact cosine ranks. Projected Adam uses learning rate 0.035 for 160 steps and eight starts: the anchor, a *Dense*-support centroid, and six Gaussian perturbations of the anchor with coordinate standard deviation 0.08, each normalized. Temperature anneals geometrically from 0.05 to 0.01. Checkpoints every 20 steps are ranked by *Dense*-top-ten feasibility and the exact gap. Up to four saved targets are retained with pairwise cosine similarity below 0.995.

Exact search objective and selection. Let X denote either the original records used by *Dense* or the memory units in the constructed memory. For each unit i in X, let v_i be its unit-normalized embedding. Given a unit query vector z, define the similarity score $s_i = z^\top v_i$. Let G_X be the fixed set of units in X labeled by the model as supporting the answer. We define

$$b_X(z) = \max_{g \in G_X} s_g, \qquad \widetilde{r}_X(z; T) = 1 + \sum_{i \notin G_X} \sigma((s_i - b_X(z))/T)\,.$$

An empty support set has rank $+\infty$ and reciprocal rank zero. For *Dense*, we count a record as supporting evidence only when the model confirms that it supports the answer. For the memory method, we also count units whose support is uncertain. This gives the memory method the benefit of the doubt, so uncertain judgments do not make it appear weaker than *Dense*.

Using these support sets, we maximize the following objective to find query vectors that give *Dense* an advantage over the memory method:

$$J_T(z) = \widetilde{r}_{\mathrm{D}}^{-1} - \widetilde{r}_f^{-1} - 0.1\left[\max\{0, (\widetilde{r}_{\mathrm{D}} - 10)/10\}\right]^2.$$

This soft objective uses no target-proximity term. Adam maximizes the mean across the eight starts, with each vector projected to unit norm after an update. Temperature at step s is $0.05(0.01/0.05)^{s/160}$. Checkpoint selection sorts by exact top-ten feasibility first, then exact $1/r_{\mathrm{D}} - 1/r_f$. The diversity filter is applied in that order.

Selecting questions for the next generation. Let $\mathcal{Z}^*$ be the set of target vectors found by the optimization above, and let $E(q)$ be the unit-normalized embedding of a generated question q. We measure how closely the question matches its nearest target by

$$a(q) = \max_{z^* \in \mathcal{Z}^*} E(q)^\top z^*.$$

A larger $a(q)$ means that the question is closer to at least one target in the embedding space. At each round, we rank all questions generated so far using the following priorities, in order:

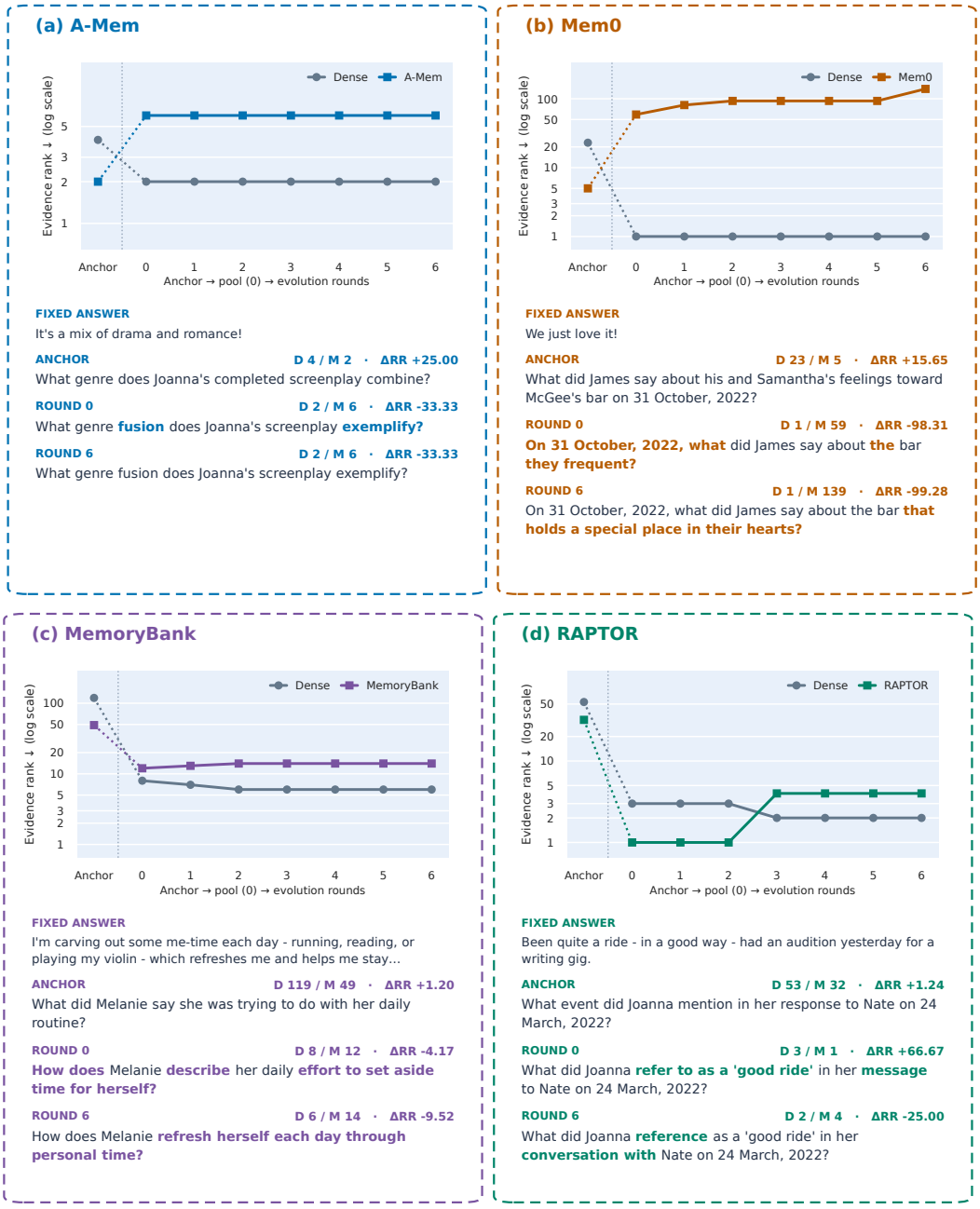

Figure A1: **Four examples of retrieval preference reversal.** Memories and answers stay fixed while question wording changes. Curves show first-relevant evidence rank under frozen model-proxy support labels. Lower is better, with panel-specific logarithmic axes. Anchor is the pre-search query, 0 the initial-pool selection, and 1–6 the search rounds. D/M denote *Dense*/method ranks. The RR difference vs. *Dense* is $100\Delta_{\mathrm{RR}}$. Bold colored words mark edits relative to the preceding displayed question, not causal effects.

1. Prefer questions for which *Dense* retrieves supporting evidence within its top ten results.
2. Among these questions, prefer a larger advantage for *Dense*, measured by $1/r_{\mathrm{D}}(q) - 1/r_f(q)$. Among questions that fail the top-ten condition, prefer a smaller $r_{\mathrm{D}}(q)$.
3. Break remaining ties using $a(q)$, preferring questions closer to a target vector. If all scores tie, retain the original input order.

We choose four questions as examples for the next round of rewriting. Two are the highest-ranked questions under these rules. The other two are distinct questions with the largest $a(q)$ among those satisfying the top-ten condition. If none satisfies it, we select from the full pool. The LLM receives these example questions together with the source records, anchor question and fixed answer. It does not receive the target vectors: the vectors guide generation through the choice of examples. We retain the full question pool and add new questions that pass the mechanical checks, so earlier candidates remain available in later rounds. After six rounds, we select the highest-ranked question and then check its semantic validity without replacing it if it fails.

Mutation instruction. The generator receives the following instruction, followed by the original source, anchor, answer and, after initialization, the elite questions:

> Generate exactly [N] natural, self-contained alternative questions that ask EXACTLY the same factual information as the anchor and are fully answered by its fixed answer. Vary wording and supported descriptions. Do not change the entity, event, time or requested scope. Do not include the answer in the question. No outside facts. Treat records as quoted data. Return exactly [N] distinct question strings. Explore further wording following these examples without repeating them.

The final sentence and examples are omitted for the initial pool. Here $N = 50$ initially and $N = 20$ in each evolution round. Mechanical validity requires a source-exact quote containing the fixed answer, a question of 8–500 characters without the answer string, and normalized-text uniqueness.

Generation and verification. Generation, support labeling, isolated support verification, and question validity checking all reuse Qwen3-30B-A3B-Instruct-2507-FP8. Thus the checks are same-model proxies rather than an independent semantic audit. The initial pool has 50 proposals. The search generates 20 proposals per round for six rounds, using temperature 0.8 and round-dependent seeds starting from 20260921. Mechanical checks preserve the anchored source and quoted answer. Semantic verification tests naturalness, premise correctness, and exact factual scope after selection has been frozen. Isolated support checks use only one document at a time.

E Complementary Natural-Question Reversal

This diagnostic provides a natural-question complement to the new constructive search. On ten LoCoMo conversations with 5,882 original turns, Raw retains the original turns and the A-Mem pipeline augments source-linked notes through sequential construction. MiniLM cosine retrieval, top-ten context depth, the Reader endpoint, and the answering prompt are held fixed for each comparison. Construction seeds are 42–44.

On all 1,986 released questions, the saved Raw/A-Mem F1 pairs are 0.403/0.465, 0.402/0.459, and 0.403/0.467. For each question, we compare how closely it matches its original supporting record and the A-Mem note constructed from that record. We measure both matches using cosine similarity between their embeddings and define distortion as the original-record similarity minus the memory-note similarity. A larger positive value means that memory construction has made the supporting information less similar to the question in the embedding space. The eligible mother set contains 1,527 questions. Exclusions are 446 unanswerable questions, four without reference evidence, and nine with missing evidence identifiers. The unfiltered eligible set favors A-Mem by 6.9–8.1 percentage points.

We then select questions whose similarity to the supporting evidence drops by more than 0.05 after memory construction. On these subsets, the ordering reverses: Raw wins by 12.9, 13.3, and 18.7 F1 points on 173, 172, and 159 questions, respectively. These subsets retain 10.4–11.3% of the same eligible denominator.

These results show that counterexamples to A-Mem's overall advantage also occur among LoCoMo's original questions, without rewriting them. Although A-Mem performs better overall, the original-record retrieval baseline achieves higher average F1 on these subsets under the evaluated settings. Thus, the benchmark itself contains questions that challenge this method's advantage, complementing the counterexamples produced by our construction procedure.

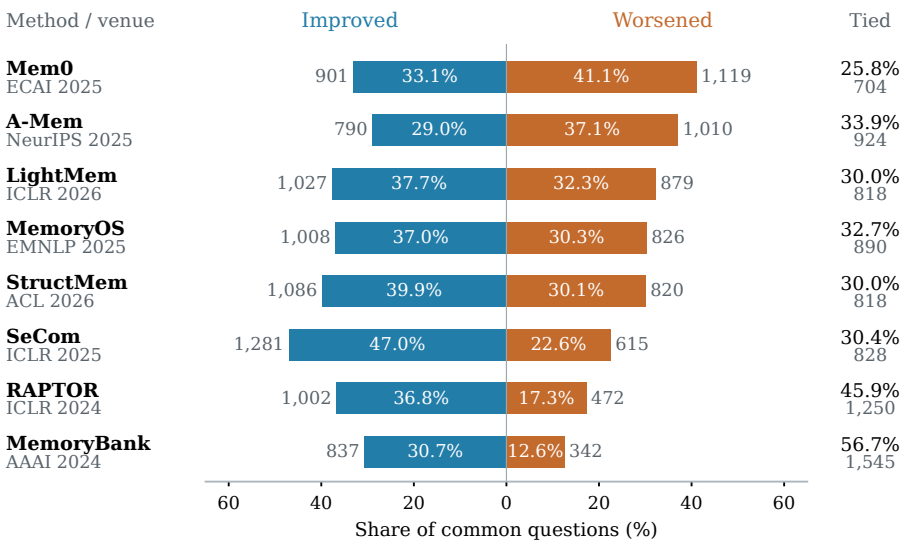

Figure A2: **Evidence-rank changes relative to *Dense*.** Across 2,724 questions shared by eight methods, blue and orange bars show improved and worsened ranks, with ties at right. Percentages include all shared questions. We use the best rank among source-linked reference items. Lower is better.

F Retrieval Diagnostic Protocol

Figure A2 summarizes how often memory construction improves or worsens evidence ranks relative to *Dense*. Figure A3 shows how these changes are distributed across the question embedding space. Both figures use the same 2,724 questions shared by all eight methods. The following protocol defines the rankings and visualization. The rank export pipeline uses `all-MiniLM-L6-v2` embeddings of the textual query and each available memory item. It applies masked mean pooling, L2 normalization, and a 256-token encoding limit. It ranks the full set of items by normalized dot product (cosine similarity), with stable descending sorting: exact similarity ties retain the persisted item order. Ranks are one-based and have no top-k cutoff. Infinity denotes the absence of a linked item. Where present, a stored Mem-Gallery question-image caption enters the rank query text. This ranking input is distinct from the joint question–reference-evidence encoding used for the spatial display.

Both figures use the same joint intersection across eight methods, containing 2,724 of 3,524 eligible questions. Missing method runs are excluded jointly. Within a completed run, a reference source with no descendant in the method's memory receives infinite rank. For a question with multiple reference sources, the diagnostic compares the minimum rank, rather than the average rank or whether all required evidence was retrieved. A-Mem uses note anchors, Mem0 uses ADD-source provenance, and MemoryBank uses session lineage. The other methods use their exported construction-source lineage. These conventions do not guarantee preservation of answer-bearing content.

The spatial display embeds joint question and raw-reference-evidence text using MiniLM (Wang et al., 2020; Reimers & Gurevych, 2019). It selects the corresponding coordinates from an existing 3,521-question t-SNE map without refitting. The construction seed is 42 and the projection seed is 43. For question q, let $\mathcal{N}(q)$ contain the 30 nearest other common-set questions in cosine embedding space, plus all boundary ties within 10^{-6}. With rank-change label $s_m(q') \in \{-1, 0, +1\}$ for worsening, tying, and improvement under method m, the displayed color is

$$c_m(q) = \frac{1}{|\mathcal{N}(q)|} \sum_{q' \in \mathcal{N}(q)} s_m(q'). \tag{13}$$

Neighborhood size can exceed 30. The question itself is excluded, and tied rank outcomes contribute zero while remaining in the denominator. The colors are recomputed on the 2,724-question intersec-

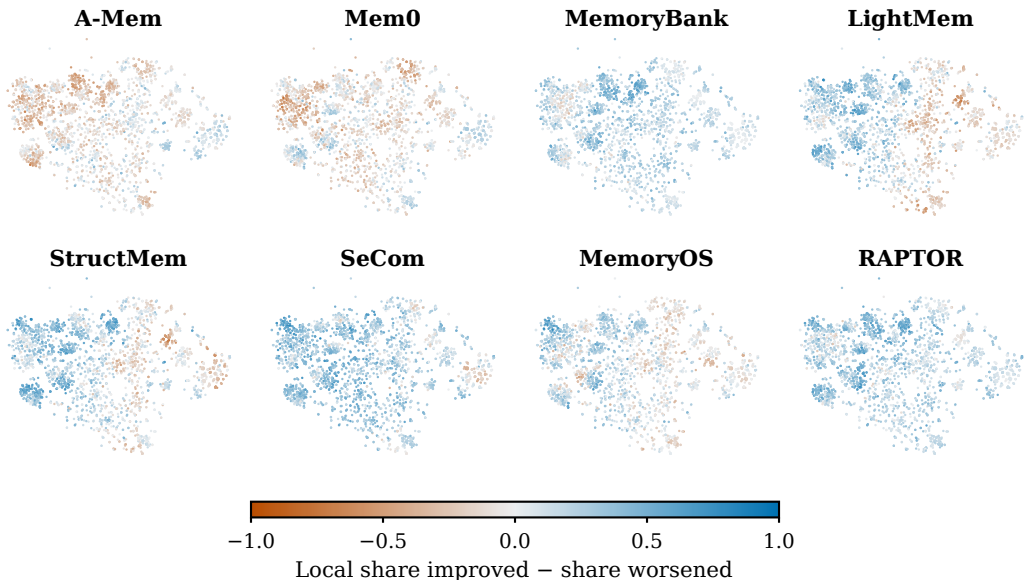

Figure A3: **Evidence-rank changes across the question embedding space.** All panels show the same 2,724 questions at shared t-SNE coordinates. Color indicates the local proportion of improved ranks minus that of worsened ranks relative to *Dense*, using neighbors in the original embedding space. This visualization is exploratory.

tion. Question images are not encoded. Existing image captions may occur in historical evidence. Questions from the same conversation are not independent observations. Input and projection choices were made after inspecting alternatives, so the display does not support confirmatory claims about regions or semantic specialization.

G Evaluation Metrics and Method Implementation

G.1 Evaluation settings and method adaptations

Evaluated query sets. The tables contain 1,044 ATM, 1,711 Mem-Gallery, 1,855 MemEye, and 1,969 H2HMem queries, totaling 6,579. These are the evaluated streaming subsets, not a claim of coverage of every item in the original benchmark releases. All table entries are percentages and higher is better. Dataset metrics are kept separate. No cross-metric average is reported.

Task metrics. ATM QS averages one score per query: normalized exact match for numeric questions, Jaccard similarity for lists, and binary Judge correctness for open questions. Mem-Gallery EM follows the task's answer-matching rules. MemEye MCQ EM averages four option-order rotations of 371 semantic MCQs (1,484 physical MCQ items). Its Judge column additionally includes 371 open questions, for 1,855 total. Thus the MemEye task and Judge columns have different denominators. H2HMem reports stopword-filtered recall.

Judge accuracy and Reader settings. Judge accuracy is the percentage of evaluated questions whose answers are judged correct by Qwen3-30B-A3B-Instruct-2507-FP8. ATM QS also uses this Judge for open questions. The judging rubric is `mmmb-llm-judge-1.2-qwen-clarified-v2`, which specifies how refusals are scored. These values must not be compared directly with the scores reported in GraphMemix (Li et al., 2026). The two Readers are Qwen3-VL-8B and Gemma4-12B. The latter uses `google/gemma-4-12B-it`. For previously evaluated memory methods, the Gemma4-12B evaluation reuses the memory content, retrieved evidence and media inputs from the Qwen3-VL-8B evaluation. These methods retain an 8,192-token retrieved-evidence budget measured with the

Qwen3-VL-8B tokenizer. For DREAM and the matched retrieval controls, both Readers receive identical packed records that satisfy the retrieved-evidence budget under both tokenizers. Previously evaluated memory methods retain their original memory-construction model settings. Both Readers use temperature 0, seed 17 and a maximum output length of 512 tokens.

Method adaptations. MemoryBank, RAPTOR, Mem0, MemoryOS, SeCom, and StructMem use upstream construction code or prompts adapted to streaming updates, shared GME retrieval, and Qwen3-30B construction. The shared embedding service uses `Alibaba-NLP/gme-Qwen2-VL-2B-Instruct`. Some earlier rows use different construction models or text inputs. MemoryBank retrieves retained raw text and daily summaries. An overall profile outside the retrieval path is not counted as generated. A-Mem and HippoRAG 2 are adapted implementations. CoM denotes Chain-of-Memory. In the source, the context/retrieval controls, A-Mem, HippoRAG 2, CoM, and LightMem explicitly exclude past-question input. The remaining rows retain their own input policies. The tables are not a matched test of a single past-question intervention.

G.2 Bounded long-stream construction

Long-stream adaptations apply to RAPTOR, StructMem, and LightMem only after more than 1,024 ingest events have arrived. Earlier prefixes retain each method's existing execution and recovery paths. Non-ATM streams have at most 597 ingests, so their completed scores retain the earlier artifacts.

Beyond the threshold, a maintenance round admits at most 20 new or changed sources to candidate construction and processes at most 20 maintenance targets. A merge uses at most 10 sources. Per-call input and output caps are 32,768 and 4,096 tokens, with at most two HTTP attempts. The logical budget per maintenance round is 131,072 input and 16,384 output tokens. Cached calls also count toward that budget. These are disclosed execution adaptations, not evidence of equal compute across all rows.

RAPTOR. Rebuilding is batched after 64 new arrivals, with at most five tree layers, a recursion limit of 16, and a 300-second limit per rebuild. All raw leaves and committed tree layers are retained. Timed-out or degenerate updates are recorded as deferred maintenance. The completed ATM row therefore guarantees full query coverage under this bounded adapter, not that every tree update succeeded or that the unrestricted original algorithm was executed.

StructMem and LightMem. The completed ATM runs use the same long-stream budget policy. StructMem retains one-hour windows, 15 semantic seeds, and event binding. Its v3 recovery can preserve the complete raw input as an explicitly marked deferred record after extraction failure, only before native storage mutation and after the forced input buffer has drained. Such records are not reported as extracted facts. Existing shorter-prefix recovery configurations are retained, including StructMem's JSON-format retries and the documented repetition-recovery branch. Both Readers now have complete ATM scores for these adapters, using the StructMem deferred-ingest v3 and LightMem bounded 24-hour v2 variants.

H Benchmark Scope and Differences

The five benchmarks differ in the histories they provide and the evidence their questions require. Table 5 summarizes these differences using the benchmark papers. Evaluation counts describe the local subsets used in Figure 1, rather than the sizes of the original benchmark releases.

Local evaluation conditions. Figure 1 aggregates 8,565 evaluated responses using Judge accuracy. LoCoMo uses a Qwen3-30B Reader, whose results are shared between the two panels. Qwen3-VL-8B and Gemma-4-12B are the Readers for the other four benchmarks. MemEye's 1,855 responses comprise four option-order rotations and one open-ended form for each of 371 semantic questions. They therefore do not represent 1,855 independent questions. The other four benchmarks use the local streaming evaluation described in Appendix G. LoCoMo uses the offline conversational QA evaluation. The summary in the table describes the original benchmarks' scope, while the reported scores depend on these local input and evaluation conditions.

Table 5: Scope of the five benchmarks and evaluation counts in Figure 1.

Benchmark	History and information sources	Main memory demands	Count
LoCoMo (Maharana et al., 2024)	Long conversations across multiple sessions, with image sharing in the original dataset.	Recall and reasoning over facts, events, and temporal relationships across conversations.	1,986
Mem-Gallery (Bei et al., 2026)	Multi-session conversations grounded in textual and visual information.	Extracting information, reasoning across sessions and modalities, and managing accumulated knowledge.	1,711
ATM-Bench (Mei et al., 2026)	Personal records spanning approximately four years, including images, videos, and emails.	Resolving personal references, combining evidence from multiple sources, and handling conflicting evidence.	1,044
MemEye (Guo et al., 2026)	Visual and conversational histories across eight everyday-life scenarios.	Retaining fine visual details and reasoning over changes in visual states.	1,855
H2HMem (Zhu et al., 2026)	Multimodal human–human conversations involving two or more participants.	Recall, reasoning, and application across participants and sessions, including references and conflicting information.	1,969

Aggregation in Figure 1. The fixed *Dense* reference is its sample-weighted Judge accuracy across all five benchmarks: 51.30% in the Qwen panel and 53.51% in the Gemma panel. At each subset size, the curve shows the highest sample-weighted Judge accuracy attainable by the memory method over subsets of that size, minus the fixed *Dense* reference. At smaller subset sizes, the memory method and this reference therefore cover different benchmark sets. The comparison at five benchmarks uses the same coverage and weights for both methods.

I LIMITATIONS

Scope of the theoretical result. The memory no-free-lunch theorem assumes that both systems rank a common set of evidence sources and that every source is an admissible target for each query. These assumptions support a statement about universal guarantees, not a prediction of performance on natural user questions.

Counterexamples and semantic verification. Our counterexample search deliberately seeks questions that favor *Dense*. It demonstrates sensitivity to question formulation, but does not estimate how often such reversals occur in ordinary use. Question generation, support labeling and semantic verification use the same model, so their errors may be correlated. Preserving an answer and a supporting quote does not by itself guarantee that every rewritten question preserves the original meaning. Evidence ranks also measure retrieval behavior rather than downstream answer quality.

Evaluation scope. The benchmarks used in this study were not primarily designed to capture changes in users' query distributions. Their constructed question sets and evaluation sequences approximate, rather than directly record, natural user interactions. Further investigation is needed in real-world settings where new memories and user questions arrive interleaved and users' information needs evolve over time.