MEMORYLIFEBENCH: EVALUATING SURVIVAL-BASED AND UTILITY-RANKED MEMORY RETENTION IN LLM ASSISTANTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-running LLM assistants must decide which memories to retain under a storage budget. Predicting when a memory-related event occurs is a plausible intermediate objective, but does accurate prediction lead to effective retention? MemoryLifeBench examines this question using 10,152 right-censored records: synthetic records measure time to factual invalidation, whereas LoCoMo and LongMemEval records measure time to observed future evidence use. A Cox-trained predictor with 213,889 trainable parameters achieves a test C-index of 0.7218, exceeding five prompted and heuristic baselines. Nevertheless, MemoryLifeBench-TTL, which converts survival predictions into expiry thresholds, underperforms FIFO and reference-recency LRU on downstream question answering. MemoryLifeBench-Utility instead retains the top-N memories by a separately supervised future-use score, increasing LoCoMo evidence retention from 0.6687 to 0.8765 at matched capacity. Its downstream improvement over FIFO and the TTL policy is exploratory: 80% of the 120-question LoCoMo sample comes from training conversations, and the 24-question validation/test subset does not establish superiority. An evidence oracle also outperforms retaining all memories with ordinary retrieval. These results separate event-time discrimination, budgeted retention, and retrieval quality, and motivate evaluating the memory decision directly rather than relying on a temporal prediction metric. https://github.com/researchwork912/anonymous-memory-benchmark

1 INTRODUCTION

Personal LLM assistants accumulate facts, preferences, plans, and events across sessions (Park et al., 2023; Zhong et al., 2024). A useful memory system must preserve information needed for later questions while controlling storage and retrieval costs (Packer et al., 2023; Chhikara et al., 2025). Retaining everything avoids eviction errors but can still leave relevant evidence outside the retrieved context. Conversely, a compact store can fail because the required evidence has been deleted. These are different failure modes, even when both appear as an incorrect answer.

Temporal prediction offers a natural approach to retention. Facts can expire, plans can change, and some information is needed only briefly (Rasmussen et al., 2025). Survival analysis models event times while accommodating observations whose event has not yet occurred (Cox, 1972; Kvamme et al., 2019). However, a survival model's prediction target and an assistant's retention objective need not coincide. A model may correctly order event times without estimating an appropriate expiry threshold or identifying the most useful subset at a fixed capacity.

MemoryLifeBench studies this prediction-to-decision gap. The benchmark combines synthetic time-to-invalidation records with time-to-future-use records derived from LoCoMo (Maharana et al., 2024) and LongMemEval (Wu et al., 2025). The distinction matters: an early synthetic event means that a fact became invalid soon, whereas an early conversation-derived event means that the information was needed soon. Pooling the records provides a common statistical task, but does not make the two events interchangeable. Throughout the paper, *event time* denotes this source-dependent target; *invalidation lifetime* and *future-use time* retain their separate meanings.

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.

The empirical result has two parts. First, a compact Cox-trained predictor ranks benchmark event times better than the evaluated prompted LLMs and heuristics. Second, the associated TTL policy performs worse than FIFO and reference-recency LRU on downstream QA. MemoryLifeBench-Utility, a policy that ranks memories by observed-future-use probability, recovers much of the lost evidence. This alternative changes both the score and the selection rule, so its improvement does not isolate either change or establish that survival supervision is necessary.

The paper makes three contributions. **(1) A benchmark formulation:** MemoryLifeBench provides 10,152 records with explicit source-dependent event and censoring definitions. **(2) A prediction-to-policy evaluation:** event-time discrimination, evidence retention, and downstream QA are evaluated separately, revealing that a strong temporal ranking can coexist with a weak TTL retention policy. **(3) A bounded utility-policy analysis:** matched-capacity selection, an additive-evidence argument, and retrieval-oracle comparisons clarify what utility ranking addresses and what remains unverified. The central empirical claim concerns the evaluated policies and samples; it does not establish a general advantage over LRU or conversation-level generalization on LoCoMo.

2 Related Work

Memory management for LLM agents. Generative Agents combine recency, importance, and relevance to retrieve experiences (Park et al., 2023); MemGPT manages context through hierarchical memory and paging (Packer et al., 2023). MemoryBank uses an Ebbinghaus-inspired update and forgetting mechanism (Zhong et al., 2024), while Reflective Memory Management adapts memory construction and retrieval through reflection (Tan et al., 2025). Mem0 extracts, consolidates, and retrieves conversational information (Chhikara et al., 2025). These systems motivate retention and consolidation, but their overall architectures address a broader problem than the controlled policy comparison here. The evaluated Mem0 configuration is therefore reported separately, with its extraction-model confounds.

Structured retrieval and episodic reasoning. HippoRAG and HippoRAG 2 use structured retrieval for long-term memory (Gutierrez et al., 2024; 2025). GraphRAG uses graph-based summarization (Edge et al., 2024); Zep organizes temporally annotated knowledge (Rasmussen et al., 2025); A-Mem builds linked, evolving memory notes (Xu et al., 2025). REMem combines time-aware episodic representations with reasoning over a hybrid memory graph (Shu et al., 2026). MemoryLifeBench complements these approaches by separating which evidence survives a retention policy from which surviving evidence reaches the answering model. Refusal and error analyses follow the evaluation approach credited to REMem.

Benchmarks and temporal prediction. LoCoMo (Maharana et al., 2024) and LongMemEval (Wu et al., 2025) evaluate long-term conversational memory through QA. Their evidence annotations support a future-use target, but do not directly identify factual invalidation. MemoryLifeBench uses the Cox proportional-hazards formulation (Cox, 1972) and neural time-to-event implementation tools (Kvamme et al., 2019). The contribution is the benchmark and decision analysis, rather than a new survival objective or a claim of state-of-the-art survival modeling.

3 MemoryLifeBench: Data and Prediction Targets

3.1 Source-dependent events and censoring

Each memory record contains a statement, a duration T_i, and an event indicator $\delta_i \in \{0, 1\}$. For the synthetic subset, $\delta_i = 1$ denotes an injected contradiction, update, or scheduled expiry, and T_i measures time from the statement to invalidation. For LoCoMo and LongMemEval, the event is the next observed use of the memory as QA evidence, linked through the extracted evidence identifiers. These sources contribute 36.6% and 63.4% of the records, respectively. Their aggregate C-index measures discrimination over a mixture of event processes, not a universal physical lifetime.

When an event is observed, duration is the event timestamp minus the statement timestamp. Otherwise, a synthetic record is right-censored at its last scheduled probe. A conversation-derived record is censored at the latest available timestamp in its conversation. The latter is an administra-

Table 1: MemoryLifeBench records by source and split. Synthetic events are invalidations; LoCoMo and LongMemEval events are observed future uses.

Source	Train	Validation	Test	Total
Synthetic	3,199	256	265	3,720
LongMemEval	3,162	359	375	3,896
LoCoMo	1,936	324	276	2,536
Total	8,297	939	916	10,152

tive observation-window proxy: absence of a recorded use does not imply permanent irrelevance. Conversation-derived records have a reported median of eight memories per conversation and a median span of 17.5 days, with spans up to 298 days. Sensitivity to alternative censoring conventions has not been evaluated.

The Future-Utility head uses a related but different target: whether a memory is observed as evidence at any later point before its conversation ends. It is therefore trained on a variable observation window, not a common fixed horizon. On conversation-derived records, survival and utility supervision both originate in future-use annotations. Their comparison concerns different objectives over partly shared supervision, rather than independent notions of validity and usefulness.

3.2 Composition, splits, and label audit

Table 1 summarizes the record splits. The extraction groups are split into 568/71/71 training/validation/test conversations, with disjoint conversation identifiers in the record-level benchmark. Observed-event rates are 37.2%, 35.5%, and 36.6%, respectively. Training rates differ by source: 48.8% for LoCoMo, 43.1% for synthetic records, and 24.1% for LongMemEval. A low observed-event rate describes the available annotations and observation window; it does not establish that the remaining memories are permanent or useless.

Label validation remains incomplete. A duration audit identified 470 discrepancies among 10,152 records (4.63%), all in LongMemEval: 449 floor-clamping or within-one-day artifacts and 21 larger gaps, up to 53.2 days. A prepared 45-record human review has not been rated, and the original extraction pipeline is not released. The reported experiments have not been rerun on an audited correction of these records.

4 MemoryLifeBench Models and Retention Policies

The predictor variants are **MemoryLifeBench-Survival** (MLB-Survival) and **MemoryLifeBench-Joint** (MLB-Joint). Their retention variants are **MemoryLifeBench-TTL** (MLB-TTL), **MemoryLifeBench-Utility** (MLB-Utility), and **MemoryLifeBench-Combo** (MLB-Combo). These names distinguish the benchmark, the learned predictors, and the decisions evaluated downstream.

4.1 Event-time and joint predictors

A frozen 768-dimensional BGE-base embedding feeds a 213,889-parameter MLP in MLB-Survival. For representation z_i, the scalar risk score $f_\theta(z_i)$ defines

$$h(t \mid z_i) = h_0(t)e^{f_\theta(z_i)}, \qquad S(t \mid z_i) = S_0(t)^{e^{f_\theta(z_i)}}. \tag{1}$$

The MLP minimizes negative Cox partial log-likelihood on (T_i, δ_i). With observed-event set $D = \{i : \delta_i = 1\}$ and risk set $R(T_i) = \{j : T_j \geq T_i\}$, the objective is

$$\mathcal{L}_{\text{Cox}} = -\sum_{i \in D} \left[f_\theta(z_i) - \log \sum_{j \in R(T_i)} e^{f_\theta(z_j)} \right]. \tag{2}$$

Censored records contribute to risk sets while they remain observed, even though they do not supply an observed-event term. The baseline survival is fitted from training data after learning the risk

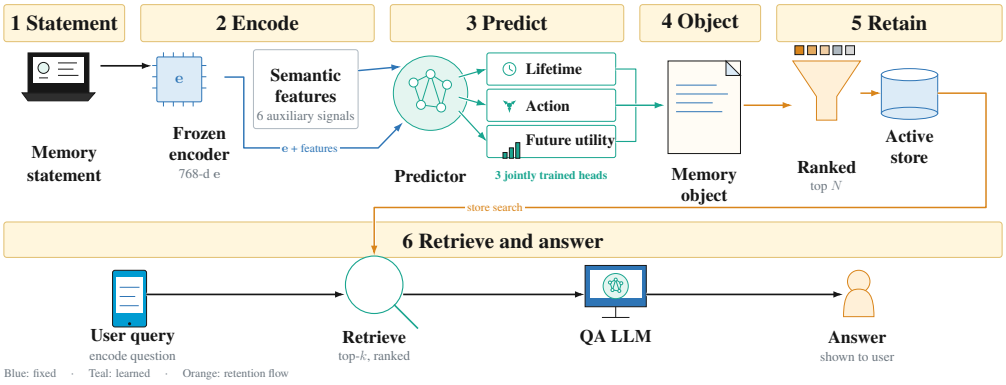


Figure 1: MemoryLifeBench lifecycle overview. Blue: fixed/frozen components; teal: learned MemoryLifeBench modules; orange: lifecycle/retention MemoryLifeBench machinery. The six auxiliary feature extractors, the memory object's fields, and the eviction policy are named in full in Section 4; this figure gives the overall flow.

score. Higher predicted risk corresponds to an earlier event; it does not have a source-independent interpretation as lower memory usefulness. In the synthetic subset, an earlier event indicates earlier invalidation. In the conversation-derived subset, it indicates earlier evidence use. The same risk ordering therefore supports different semantic interpretations before any policy is applied. The proportional-hazards assumption has not been tested. Pooling invalidation and future-use events introduces a separate modeling limitation: a common likelihood does not demonstrate a common event-generating mechanism. Full definitions appear in Appendix A.1.

MLB-JOINT combines the BGE embedding with six frozen feature extractors: intent, entities/NER, temporal cues, emotion/preferences, novelty, and contradiction. The extractors contain approximately 433M frozen parameters in total and use forward inference without autoregressive generation. Concatenation or learned gating produces a shared representation for three heads. The concatenation model has 425,734 trainable parameters; this count excludes the frozen encoders.

The **Lifetime head** uses the Cox objective above. The **Action head** predicts store, update, merge, or forget with inverse-frequency class weighting. Labels map lifecycle update, contradiction, and natural expiry to update, merge, and forget; all other labels map to store. Consequently, every non-store training label originates in the synthetic subset. The **Future-Utility head** predicts observed future evidence use and is trained only on LoCoMo/LongMemEval usage labels. A fourth importance score is a hand-written heuristic, not a learned head. A custom training loop combines the survival, action, and utility losses with weights $1.0 : 0.5 : 0.5$.

4.2 FROM PREDICTIONS TO RETAINED SUBSETS

Let O be a conversation's extracted memory objects and t its final timestamp. An object stores text, embedding, importance, predicted TTL, action, utility probability, and provenance. For survival-probability cutoff q, the TTL is

$$\tau_q(z_i) = \sup\{s : S(s \mid z_i) \geq q\}, \qquad A_{\text{TTL}} = \{i : a_i \neq \text{forget},\ t - t_i \leq \tau_q(z_i)\}. \tag{3}$$

Here t_i is creation time and a_i the predicted action. TTL is capped at 3,650 days when the curve does not cross the cutoff; $q = 0.5$ is the original setting. Lowering q extends TTL and usually increases retained storage.

The ranked policies retain $N = |A_{\text{TTL}}|$ objects. MLB-UTILITY selects the largest Future-Utility scores $\hat{u}_i$. MLB-COMBO uses an equal-weight blend of $\hat{u}_i$ and the clipped remaining-life fraction $\text{clip}(1 - (t - t_i)/\tau_q(z_i), 0, 1)$. FIFO ranks by creation time. LRU ranks by the latest reference anywhere in the conversation; it is a *reference-recency snapshot baseline*, not causal online LRU. No-Forget retains every object and is therefore an unmatched-capacity reference. It is not an accuracy ceiling because retrieval and answering can still fail.

All QA pairs from a conversation use the same final active set. The evaluation consequently measures retrospective subset-selection quality. It does not test whether an irreversible eviction decision, made before later references are available, would preserve the same evidence. The capacity is also induced by MLB-TTL, rather than selected independently; each quantile comparison uses its corresponding induced budget.

Why utility ranking is relevant. If Y_i indicates future evidence use and $u_i = P(Y_i = 1 \mid \mathcal{I})$ is its true conditional probability given available information, the additive objective is

$$\max_{S:\,|S|\le N} \mathbb{E}\left[\sum_{i\in S} Y_i \mid \mathcal{I}\right] = \max_{S:\,|S|\le N} \sum_{i\in S} u_i. \tag{4}$$

Keeping the top-N probabilities is optimal for this objective by an exchange argument (Appendix A.4); independence among memories is unnecessary. The result does not establish optimality for a learned $\hat{u}_i$ or for QA requiring complementary evidence. Nor does it imply that all threshold rules are inferior: the empirical question is whether the evaluated score and cutoff choose a useful subset.

System costs. The implementation uses a brute-force NumPy vector store at approximately 10K-memory scale. Novelty and contradiction features scan previous memories, giving $O(cd)$ work per write for c prior memories of dimension d, in addition to encoder inference. Ranked selection by sorting costs $O(M \log M)$ for M candidates; retrieval requires $O(Md)$ similarity computation before top-k selection. Embedding storage is $O(Md)$. These costs exclude QA generation; a small trainable parameter count does not imply low total inference cost. Appendix A.3 separates the evaluated policy from additional lifecycle operations such as compaction and decay.

5 Experimental Protocol

Training and metrics. Training uses Adam with learning rate 10^{-3}, weight decay 10^{-4}, no schedule, and at most 200 epochs. Early stopping restores the best checkpoint with patience 15 for survival and 20 for the joint model. Batch size is 256 for survival and full-batch for joint training. Runs use one H200 on a shared eight-GPU node, PyTorch 2.6, pycox 0.3, and torchtuples 0.2. Survival experiments use five seeds (13, 42, 1337, 2024, 7), and joint experiments use three (13, 42, 1337); multi-seed summaries report mean and standard deviation.

Event-time evaluation uses C-index, time-dependent AUC, and integrated Brier score (IBS). The first two measure discrimination; IBS measures probabilistic prediction error. Evidence retention checks whether an extracted gold-evidence memory survives eviction. GPT-4o answers downstream questions from the top-5 retrieved memories, scored with exact match (EM) and token F1. BLEU-1, LLM-judge scores, refusal analysis, and evidence-oracle references provide additional diagnostics. LoCoMo category-5 unanswerable questions are excluded from ordinary answerable-QA results and evaluated separately for refusal.

Temporal baselines. The test ($n = 916$) and validation ($n = 939$) records are shared across a recency-frequency heuristic, a day/week/permanent classifier, and zero-shot TTL prompting of Qwen2.5-7B-Instruct, GPT-4o, and Gemini 2.5 Pro. The LLM prompt asks how long a statement remains "true and useful," which differs from the conversation-derived future-use endpoint. These are references for prompted temporal judgment rather than endpoint-matched survival estimators. All LLM calls use temperature 0.0. Exact prompts are provided in Appendix D.

Samples and overlap. The initial LoCoMo policy comparison uses 1,542 answerable questions; the evidence diagnostic uses 1,304 questions with extracted evidence. The subsequent utility-policy comparison uses a matched sample of 120 LoCoMo questions, 12 from each of ten conversations, and 25 LongMemEval questions. Eight LoCoMo conversations train the joint predictor, so 96 of the 120 downstream questions reuse training conversations. The other 24 questions come from one validation and one test conversation. This combined subset is unseen during parameter fitting, but includes validation data and is not a test-only generalization estimate. All 25 LongMemEval downstream conversations come from held-out splits.

Table 2: Evaluation units and scope. Question counts differ because the analyses require different annotations or model calls. A memory record is not an independent conversation.

Analysis	Count	Unit and scope
Temporal test	916	Records from test-split conversations
Initial LoCoMo QA	1,542	Initial answerable-QA questions
LoCoMo retention	1,304	Questions with extracted gold evidence
Matched LoCoMo QA	120	Questions; 96 overlap training conversations
LoCoMo validation/test	24	Questions from two conversations
Matched LongMemEval QA	25	Questions from held-out conversations

Table 3: Reported MemoryLifeBench test C-index ($n = 916$). This is the headline run; multi-seed summaries are reported separately in the text.

Predictor	C-index ↑
MLB-SURVIVAL	**0.7218**
Day/week/permanent classifier	0.6298
GPT-4o prompted TTL	0.5411
Qwen2.5-7B prompted TTL	0.5207
Gemini 2.5 Pro prompted TTL	0.4806
Recency-frequency heuristic	0.4753

Table 2 separates the experimental units. The initial 1,542-question comparison and later 120-question evaluation must not be read as one continuous leaderboard: their question sets differ, and only the later sample includes the utility policy's matched QA comparison. The 1,304-question diagnostic additionally conditions on successful evidence extraction. Retention of extracted evidence does not measure whether the extraction stage captured every fact required by the original question.

Uncertainty and scope. The reported temporal comparisons use 1,000 record-level bootstrap resamples; downstream comparisons use 10,000 paired question-level resamples. Both ignore dependence within conversations. Downstream p-values are exploratory and one-sided for the stated direction of improvement. Failure to reject such a test is neither evidence of equivalence nor a two-sided assessment of inferiority. Conversation-clustered uncertainty has not been estimated.

6 RESULTS: PREDICTION, RETENTION, AND RETRIEVAL

6.1 EVENT-TIME RANKING IMPROVES OVER THE EVALUATED REFERENCES

MLB-SURVIVAL achieves the highest headline test C-index (Table 3), with the same ordering on validation (0.7134). The reported one-sided record bootstrap gives $p < 0.001$ against every baseline. Against GPT-4o, the test difference is $+0.1801$, with 95% interval $[+0.1340, +0.2235]$. These results support discrimination on the pooled benchmark; they do not establish source-specific performance or a conversation-clustered population estimate.

Test time-dependent AUC/IBS are 0.7895/0.2463 for MLB-SURVIVAL and 0.5339/0.4882 for GPT-4o. Baseline survival curves are step functions constructed from scalar TTL predictions, whereas MLB-SURVIVAL fits a Cox survival curve. The IBS comparison therefore changes the probabilistic representation as well as the predictor and cannot isolate a calibration advantage. Rephrasing the prompt instruction in three ways, while holding memory text fixed, gives mean TTL coefficients of variation of 0.3845 for GPT-4o and 0.7456 for Qwen2.5-7B. The non-prompted predictor is invariant to this specific intervention by construction; memory-text paraphrases and repeated-run nondeterminism were not isolated.

Across seeds, test C-index is 0.7312 ± 0.0131 for BGE-base MLB-SURVIVAL, 0.7382 ± 0.0048 for BGE-large, 0.7553 ± 0.0045 for concatenation-based MLB-JOINT, and 0.7304 ± 0.0082 for gated fusion. These are descriptive comparisons. Action accuracy is 86.4%, with recall 1.000 and precision

Table 4: Initial LoCoMo downstream comparison ($n = 1{,}542$ questions) and evidence diagnostic ($n = 1{,}304$). Bounded policies use the capacity induced by MLB-TTL at $q = 0.5$; No-Forget retains all memories.

Policy	Mean EM	Mean F1	Evidence retention
No-Forget	0.1706	0.3007	1.0000
FIFO	0.1316	0.2347	0.7247
LRU (reference recency)	0.1245	0.2319	0.7676
MLB-TTL	0.1102	0.2054	0.6687

Table 5: LoCoMo evidence retention at budgets induced by each MLB-TTL cutoff. Every row matches capacity across the four policies; capacities differ across rows.

Cutoff q	FIFO	LRU	MLB-TTL	MLB-UTILITY
0.5	0.7247	0.7676	0.6687	**0.8765**
0.2	0.9225	0.9517	0.9080	**0.9663**
0.1	0.9670	0.9709	0.9502	**0.9877**

0.41–0.69 for the synthetic minority classes; Future-Utility AUC is 0.71–0.77. No matched-input utility-only ablation identifies whether joint survival training improves the utility head.

6.2 THE TTL POLICY LOSES EVIDENCE DESPITE STRONG DISCRIMINATION

MLB-TTL performs worse than both bounded baselines on LoCoMo (Table 4); the reported LongMemEval evaluation has the same ordering with a smaller gap. The evidence diagnostic confirms that the difference is already present before retrieval and answer generation: MLB-TTL retains 0.6687 of the linked evidence, compared with 0.7247 for FIFO and 0.7676 for LRU.

The traced evicted-evidence losses under MLB-TTL are attributed entirely to TTL expiry (100%), with none attributed to the Action head's forget prediction (0%). Among prematurely evicted evidence memories, mean predicted TTL is 114.4 days, compared with a mean required age of 164.1 days. This selected-subset statistic illustrates early removal but is not a population calibration estimate. In particular, C-index is invariant to monotone changes in score and does not validate the absolute cutoff in Equation 3.

Lowering the survival-probability cutoff increases retention, but also increases storage. The full sweep raises MLB-TTL evidence retention from 0.6687 at $q = 0.5$ to 0.9716 at $q = 0.05$, while the retained storage fraction rises from 70.9% to 97.1% (Appendix B.3). FIFO and LRU still retain more evidence at their corresponding matched capacities. This supports a problem with the evaluated TTL-derived selection; it does not by itself identify thresholding, endpoint semantics, or calibration as the unique cause.

6.3 UTILITY RANKING IMPROVES THE DIAGNOSTIC, WITH LIMITED QA EVIDENCE

At the original capacity, MLB-UTILITY raises evidence retention by 0.2078, from 0.6687 to 0.8765 (Table 5). It has the highest diagnostic retention at each listed cutoff. The blended MLB-COMBO improves on MLB-TTL but underperforms pure utility ranking in the reported diagnostic. The comparison changes both the predictive signal and the decision rule; an equivalent-input utility-only model and a TTL-based top-N control would be needed to separate these effects.

Evidence retention is a proxy, so Table 6 reports downstream QA separately. On the 120-question sample, the reported paired bootstrap favors MLB-UTILITY over FIFO (EM $p = 0.002$) and MLB-TTL (EM $p = 0.006$); both comparisons also have reported F1 $p < 0.01$. It does not establish an advantage over LRU (EM $p = 0.367$, F1 $p = 0.648$). Mean EM equals No-Forget on this sample, while F1 is slightly lower. These statements concern the reported question-level tests, without correction for conversation dependence or training overlap. Full intervals appear in Appendix B.1.

Table 6: Matched LoCoMo QA comparison. The 120-question sample includes 96 questions from training conversations; the 24-question subset combines validation and test conversations. Results are diagnostic, not evidence of general superiority.

	All 120 questions		Validation/test, $n = 24$	
Policy	EM	F1	EM	F1
FIFO	0.0417	0.1294	0.0000	0.1223
MLB-TTL	0.0500	0.1575	0.0000	0.0891
MLB-UTILITY	0.0917	0.1949	0.0833	0.1592
LRU	0.0833	0.1998	0.0833	0.1877
No-Forget	0.0917	0.1957	0.0833	0.2016

On the 24 questions from validation/test conversations, MLB-UTILITY has F1 0.1592, below LRU's 0.1877 and No-Forget's 0.2016. Its EM advantage over FIFO and MLB-TTL does not reach the reported one-sided significance threshold ($p = 0.119$). This small, two-conversation subset does not establish superiority; failure to reject the one-sided improvement tests must not be interpreted as equivalence. Larger test-only evaluations are needed before attributing the full-sample gains to generalization.

Association with evidence use. Across 2,536 LoCoMo memories, the pooled AUC against the evidence label is 0.6709 for the utility score and 0.2852 for median predicted TTL. Utility AUC ranges from 0.61 to 0.76 across the ten conversations. This association is consistent with better evidence selection, but it is measured on the same pooled data and uses a directly supervised evidence-use target. It is not an independent causal explanation. LongMemEval's 92 linked-evidence cases are near ceiling: retention is 0.9783 for MLB-TTL, 0.9891 for MLB-UTILITY, and 1.0000 for the other policies.

6.4 ANSWER SCORING, RETRIEVAL, AND AN INDEPENDENT SYSTEM BASELINE

Scoring and refusal. GPT-4o judge rescoring marks 160 of 725 evaluated responses (22.1%) as substantively correct despite an EM mismatch. Scores increase by a reported factor of 3.6–4.4 without establishing an advantage for MLB-UTILITY over LRU or No-Forget. At least one judge decision is flagged as potentially erroneous (Appendix B.2). On a balanced sample of 120 answerable and 120 unanswerable LoCoMo questions, refusal recall is similar across policies (0.883–0.892), while precision ranges from 0.639 for FIFO to 0.726 for MLB-UTILITY. A manually categorized 100-error sample contains 54% false refusals. These results identify missing usable evidence as a recurring failure mode; refusal alone cannot distinguish eviction from retrieval failure. Full metrics and question-type analyses appear in Appendix B.

Retrieval remains a separate bottleneck. An oracle provides the gold-evidence memories directly, bypassing ordinary retrieval. On 107 matched LoCoMo questions it exceeds No-Forget by 0.0654 EM ($p = 0.0073$) and 0.1416 F1 ($p < 0.0001$). Thus, preserving every memory does not ensure that top-5 retrieval supplies the right evidence. The oracle is a privileged reference, not a theoretical ceiling on answer quality, because generation can still fail. A smaller full-context comparison and LongMemEval results are reported in Appendix B.7.

Mem0 comparison. Mem0 indexes all ten LoCoMo conversations and answers the same 120-question sample. The evaluated configuration substitutes local Qwen2.5-7B extraction for GPT-4o to reduce API costs. Timestamp handling differs from the intended setup, and malformed JSON affects 195 of 5,882 indexing calls (3.3%). Mem0 achieves EM 0.0833 and F1 0.2275, compared with 0.0917 and 0.1949 for MLB-UTILITY. The reported bootstrap does not establish an F1 difference from MLB-UTILITY, LRU, or No-Forget ($p > 0.13$), although Mem0 exceeds FIFO and MLB-TTL in the reported comparisons. These results neither establish equivalence nor a MemoryLifeBench advantage. Extraction and timestamp confounds preclude a controlled system-level ranking (Appendix B.8).

7 Discussion and Limitations

The strongest supported finding is specific: the evaluated temporal predictor discriminates benchmark event times well, while its TTL-based retention policy performs poorly. A rank statistic does not validate an absolute expiry decision. Moreover, the real-conversation-derived endpoint is future evidence use, whose interpretation differs from factual expiry. Both issues matter when translating a survival curve into a deletion rule. The experiments do not establish that thresholding is intrinsically unsuitable or that survival information cannot support a stronger policy.

Event semantics are part of the decision. A future-use timestamp describes when a memory was needed, not when it became false. For example, a travel date can be used to answer a later question even after the trip has ended. Deleting the statement because its predicted event has passed need not preserve retrospective QA. Conversely, retaining a fact that remains true need not help any question in the evaluated sample. The benchmark exposes this distinction, but its pooled results cannot determine whether separate invalidation and future-use models would resolve the observed losses. Source-conditioned endpoints and a common, explicitly specified prediction horizon would make that comparison more interpretable.

What the utility result establishes. MLB-Utility aligns subset selection with observed future evidence use and improves the pooled retention diagnostic. Its QA results motivate further study, but the full LoCoMo sample overlaps training and the validation/test subset is small. The comparison with MLB-TTL simultaneously changes inputs to the decision and the selection rule. Without a matched-input utility-only ablation, the downstream contribution of survival supervision remains unknown. The additive-evidence proposition also omits complementary memories, repeated uses, and variable storage costs; it cannot guarantee QA optimality.

Evaluation boundaries. The final-snapshot protocol gives the LRU baseline access to reference times unavailable to an earlier online decision. Capacity is chosen from MLB-TTL rather than an independent budget grid. Question- and record-level bootstraps omit conversation clustering, and one-sided non-significance does not demonstrate that policies are tied. These limitations restrict the conclusions to the evaluated retrospective samples and budgets. A chronological test with irreversible eviction, independently specified capacities, and more test conversations would answer a different and more deployment-relevant question.

Data, modeling, and reproducibility. The mixed endpoints, unresolved duration discrepancies, untested administrative censoring, and incomplete human validation limit interpretation of absolute event times. Source-specific evaluation and broader classical/neural survival baselines are absent, and proportional hazards has not been checked. Non-store action labels are synthetic only. Exact replication also requires the six extractor identities, exact model/API versions, checkpoint selection criteria, and bootstrap implementation. The implementation includes 51 reported tests, but the unreleased extraction pipeline and incomplete label validation remain reproducibility limitations.

8 Conclusion

MemoryLifeBench evaluates the connection between event-time prediction and memory retention for LLM assistants. A compact Cox-trained model achieves strong discrimination on its mixed-endpoint benchmark, yet MemoryLifeBench-TTL loses more evidence and yields lower QA scores than the evaluated recency baselines. MemoryLifeBench-Utility improves matched-capacity evidence retention, while its downstream generalization and contribution from joint survival training remain unresolved. An evidence oracle further identifies a retrieval gap even when no memory is deleted. Together, the results motivate evaluating temporal prediction, retained evidence, and retrieved-context QA as separate stages. The next empirical priorities are source-specific targets, matched-input ablations, and conversation-disjoint online evaluation.

Ethics Statement

This work reuses two published, already-released long-term-memory benchmarks (LoCoMo, LongMemEval) under their original terms, plus a synthetically generated dialogue set with no real user data; no new human subjects data was collected. The LLM-judge and QA baselines process only this benchmark text. We are not aware of a direct harm pathway from a memory-lifetime ranking model; the known limitations of the evaluation (label noise, sample size, retrieval-quality confound) are disclosed in Section 7 rather than in a separate risk analysis.

Reproducibility Statement

Every reported number's data source, split, and metric is specified at first use (Sections 3.1–3.2 for the benchmark construction, Section 5 for evaluation protocol). Model architecture, loss, and training-loop details for the survival, joint, and downstream-policy models are given in Sections 4–5 and Appendix A; formal policy/complexity definitions and the full algorithm in Appendices A.2–A.3. All LLM prompts used anywhere in this work (baselines, QA answering, judge rescoring) are reproduced verbatim in Appendix D rather than only referenced. Multi-seed results are reported where applicable (Section 6.1); known sources of irreducible noise—the $\sim$4.6% `duration_days` label-mismatch rate and the incomplete 45-record human validation sample—are disclosed in Section 7 rather than hidden. Code and the full benchmark construction pipeline will be released publicly upon acceptance; an anonymized link could not be included here without risking inadvertent deanonymization through commit history or hosting metadata.

References

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.

David R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society, Series B*, 34(2):187–202, 1972.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

Bernal Jiménez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Bernal Jiménez Gutierrez, 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.

Havard Kvamme, Ørnulf Borgan, and Ida Scheel. Time-to-event prediction with neural networks and Cox regression. *Journal of Machine Learning Research*, 20(129):1–30, 2019.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. *arXiv preprint arXiv:2402.17753*, 2024.

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

Joon Sung Park, Joseph O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*, 2023.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956*, 2025.

Yiheng Shu, Saisri Padmaja Jonnalagedda, Xiang Gao, Bernal Jiménez Gutiérrez, Weijian Qi, Kamalika Das, Huan Sun, and Yu Su. REMem: Reasoning with episodic memory in language agent. *International Conference on Learning Representations (ICLR), arXiv:2602.13530*, 2026.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long T. Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, Anand Iyer, Tianlong Chen, Huan Liu, Chen-Yu Lee, and Tomas Pfister. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. *arXiv preprint arXiv:2503.08026*, 2025.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. *International Conference on Learning Representations (ICLR), arXiv:2410.10813*, 2025.

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

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory. In *AAAI Conference on Artificial Intelligence*, 2024.

A Formal Definitions and Implementation Details

A.1 Survival objective and evaluation

For record i, let z_i denote its frozen embedding for MLB-Survival, or its fused representation for MLB-Joint. The predictor produces log-risk $f_\theta(z_i)$ and the Cox hazard and survival functions in Equation 1. The baseline functions h_0 and S_0 are fitted non-parametrically from the training data, rather than learned by the MLP. With observed-event set $D = \{i : \delta_i = 1\}$ and risk set $R(t) = \{j : T_j \geq t\}$, the negative partial log-likelihood is

$$\mathcal{L}_{\text{Cox}}(\theta) = -\sum_{i \in D} \left[f_\theta(z_i) - \log \sum_{j \in R(T_i)} e^{f_\theta(z_j)} \right]. \tag{5}$$

The single-task implementation uses pycox/torchtuples; the joint model uses a custom loop so gradients can update the shared representation. Exact replication requires documentation of event-time ties and how risk sets are formed in mini-batches.

For comparable pairs $E = \{(i, j) : T_i < T_j, \delta_i = 1\}$ and in the absence of tied predictions, concordance is

$$C = \frac{1}{|E|} \sum_{(i,j) \in E} \mathbb{1}\{f_\theta(z_i) > f_\theta(z_j)\}. \tag{6}$$

This definition orders higher risk before lower risk. It depends on score order rather than scale and does not validate an absolute TTL. The implementation's tie convention should accompany the released metrics. The survival-probability cutoff q in Equation 3 is not the event-time CDF quantile: decreasing q extends the retained duration. Curves that do not cross the cutoff are capped at 3,650 days.

A.2 Snapshot policy definitions

For each conversation, all policies select once from the complete extracted object set O, at the latest conversation timestamp t. Every QA question uses that same active set. MLB-TTL retains objects whose action is not forget and whose age does not exceed predicted TTL; its final count determines N. Ranked policies select the N largest values of the following scores:

$$s_{\text{FIFO}}(i) = t_i, \tag{7}$$

$$s_{\text{LRU}}(i) = \text{latest reference time of memory } i, \tag{8}$$

$$s_{\text{Utility}}(i) = \hat{u}_i, \tag{9}$$

$$s_{\text{Combo}}(i) = 0.5\hat{u}_i + 0.5 \,\text{clip}\left(1 - \frac{t - t_i}{\tau_q(z_i)}, 0, 1\right). \tag{10}$$

Never-referenced objects receive LRU score $-\infty$. No-Forget retains O without imposing N. The LRU score includes references anywhere in the transcript, including references that an earlier online decision could not have observed. The evaluation therefore concerns final subset selection and does not model irreversible, chronologically enforced capacity.

A.3 Lifecycle operations and resource accounting

The implementation separates three operations. **Write** computes a frozen embedding, auxiliary features, the three learned-head outputs, and heuristic importance; it stores the resulting memory object with provenance and adds an audit-log entry. **Select** applies the policy definitions above to determine the retained subset. **Retrieve** embeds the question, selects similarity-matched candidates, reranks by importance/utility, and calls the grounded-QA model. The implementation also contains compaction, reflection through utility decay beyond TTL, and an append-only audit log. Their separate effects are not isolated in the reported policy experiments; the named utility policy is defined by ranking its stored utility probabilities.

With embedding dimension d and c previous memories in a conversation, novelty and contradiction scans contribute $O(cd)$ work to a write. Across a conversation this becomes $O(c^2 d)$ under naive

scanning, in addition to feature-model inference. Sorting M candidate scores costs $O(M \log M)$; this bound applies to ranked selection, not an unspecified duplicate-merging routine. Brute-force retrieval computes M similarities in $O(Md)$, followed by top-k selection and reranking. Approximate indexing could change these costs, but no exact logarithmic guarantee or measured speedup is established here.

The store uses $O(Md)$ embedding space plus text, provenance, and scalar metadata. At $M = 10{,}000$ and $d = 768$, float32 embeddings alone require 30.72 MB (decimal units), excluding text and software overhead. A materialized similarity vector uses $O(M)$ auxiliary space; a streaming bounded heap can retain $O(k)$ scores. The 213,889 and 425,734 parameter counts refer to trainable predictors, excluding the BGE encoder and approximately 433M frozen auxiliary parameters. Both predictors contain fewer than 500K trainable parameters. No autoregressive generation is used by these predictors, but frozen forward inference still consumes compute.

The recorded usage is 258,785 tokens for the GPT-4o TTL baseline and 210,072 for Gemini. These counts are not end-to-end cost comparisons against the complete memory system, which also includes extraction and QA. The implementation has 51 reported tests covering bootstrap arithmetic, TTL behavior, six policies, and evidence-retention arithmetic. Code is MIT-licensed; the MemoryLifeBench data release is described as CC BY-NC 4.0, with LoCoMo/LongMemEval attribution required.

A.4 Optimality for an additive evidence objective

Let $\mathcal{I}$ be the information available at selection time, $Y_i \in \{0, 1\}$ indicate future evidence use, and $u_i = P(Y_i = 1 \mid \mathcal{I})$. Each memory has equal storage cost and the budget permits at most N memories. For this additive objective, conditional linearity of expectation gives Equation 4, regardless of dependence among the Y_i.

Proposition 1. *For known $u_i \geq 0$, a subset containing the $\min(N, |O|)$ largest values of u_i maximizes the conditional expected number of retained evidence memories.*

Proof. Nonnegative scores allow any smaller feasible subset to be extended without decreasing its objective. If a full feasible subset omits a higher-scored memory in favor of a lower-scored one, exchanging the two increases the objective by their score difference. Repeated exchanges produce a top-N subset. Exchanges between tied scores leave the objective unchanged, so the optimum need not be unique. □

The proposition does not require independent memory uses, but it does require the true conditional probabilities, equal storage costs, and an additive objective. A question requiring a particular combination of memories need not be served by the subset with the largest sum of marginal utilities. Likewise, an estimated utility score need not improve on recency ranking. A strong C-index ranks the source-dependent event times; it does not establish either accurate utility probabilities or optimal retention for QA.

B Extended Downstream Results

The original plots are reproduced without changing their artwork or legend labels. Their legacy labels `ours`, `ours_utility`, and `ours_combo` refer to MLB-TTL, MLB-Utility, and MLB-Combo, respectively.

B.1 Reported bootstrap comparisons and the validation/test subset

Table 7 preserves the reported EM bootstrap differences and intervals on the matched 120-question LoCoMo sample. Eight conversations contribute 96 training-overlapping questions. The remaining 24 questions come from the validation and test conversations, 12 each; their policy scores are included in Table 6 in the main text.

On the 24-question validation/test subset, the reported one-sided EM tests for MLB-Utility improving on FIFO or MLB-TTL give $p = 0.119$. For improvement in F1, the reported values are

Table 7: Reported question-level bootstrap comparisons of MLB-UTILITY on LoCoMo ($n = 120$). Differences and intervals are retained at their reported precision. These tests do not account for conversation clustering.

Comparator	Reported EM difference, 95% interval	One-sided p
FIFO	$+0.0499$ $[+0.0167, +0.0917]$	0.002
MLB-TTL	$+0.0415$ $[+0.0083, +0.0833]$	0.006
No-Forget	0.0000 $[0.0000, 0.0000]$	1.000
LRU	$+0.0083$ $[0.0000, +0.0250]$	0.367

$p = 0.798$ against LRU and $p = 1.000$ against No-Forget. These directional tests do not establish superiority. They also do not justify the stronger statement that neither direction is significant: the corresponding two-sided or reverse-direction tests were not reported. The subset includes validation data, so it should not be described as a pure held-out test evaluation.

B.2 QUALITATIVE TRACES AND SCORING DISCREPANCIES

The reported trace analysis identifies eight questions for which MLB-TTL removed a gold-evidence memory and answered incorrectly, while MLB-UTILITY retained that memory and answered correctly. One example asks when Jon lost his job as a banker (reference: 19 January 2023). The evidence states that Jon lost the job the day before the conversation. After its removal under MLB-TTL, the answer is "I don't have that information yet"; with the evidence retained under MLB-UTILITY, the answer is "2023-01-19." The store trace supports an eviction explanation in this example. The refusal text alone would not prove that the evidence had been deleted.

One judge discrepancy remains flagged: a reference answer of "three years" was treated as equivalent to "2019," although the judge prompt did not include the conversation date needed to justify that conversion. This limits reliance on the judge as an error-free reference. Exact match also misses correct answers expressed in a different format; judge rescoring and token F1 are therefore complementary diagnostics, not replacements for trace inspection.

B.3 SURVIVAL-PROBABILITY CUTOFF SWEEP

Table 8: MLB-TTL storage and evidence retention on LoCoMo. The cutoff is a survival probability; smaller values extend TTL.

Cutoff q	Storage retained	Evidence retention
0.5	70.9%	0.6687
0.2	91.3%	0.9080
0.1	95.2%	0.9502
0.05	97.1%	0.9716

B.4 CROSS-METRIC RESULTS

Table 9: LoCoMo matched-sample metrics ($n = 120$). No column implies that its numerically highest policy is significantly superior.

Policy	EM	Token F1	BLEU-1	Judge score
FIFO	0.0417	0.1294	0.0915	0.1833
LRU	0.0833	0.1998	0.1586	0.3417
No-Forget	0.0917	0.1957	0.1562	0.3333
MLB-TTL	0.0500	0.1575	0.1116	0.2000
MLB-UTILITY	0.0917	0.1949	0.1549	0.3500

The reported judge-based tests favor MLB-UTILITY over FIFO and MLB-TTL with $p < 0.001$, but no significant improvement over LRU or No-Forget. The overlap and resampling limitations apply to every metric. BLEU-1 preserves the qualitative separation between the weaker TTL/FIFO policies and the higher-scoring alternatives, but does not identify MLB-UTILITY as the best policy.

B.5 REFUSAL EVALUATION

LoCoMo includes approximately 446 category-5 questions with no supported answer and a plausible decoy. Following the refusal evaluation approach credited to Shu et al. (2026), the reported analysis uses 120 unanswerable and 120 answerable questions. Precision measures the fraction of refusals that are appropriate; recall measures the fraction of unanswerable questions refused. A precision contrast was reported at $p < 0.001$, without a complete specification of the pairwise tests and their uncertainty estimates. The descriptive values are therefore more directly interpretable than a blanket significance claim. Similar recall alongside lower precision is consistent with more false refusals on answerable questions.

Table 10: Refusal precision, recall, and F1 on the balanced 240-question sample.

Policy	Precision	Recall	F1
No-Forget	0.718	0.892	0.796
FIFO	0.639	0.883	0.741
LRU	0.709	0.892	0.790
MLB-TTL	0.648	0.892	0.751
MLB-UTILITY	0.726	0.883	0.797

B.6 ERROR TAXONOMY AND QUESTION TYPES

A reproducible sample of 100 judge-negative responses, pooled across five policies and both benchmarks, was categorized by a single manual read-through. Table 11 retains the reported categories. No second rater or inter-rater agreement estimate is available. These are descriptive labels over selected wrong predictions, not population-level causal attributions.

Table 11: Reported taxonomy of 100 sampled errors.

Category	Description	Share
False refusal	Refusal despite an answerable question	54%
Wrong value	Incorrect specific fact or unsupported detail	21%
Incomplete	Only part of a multi-item reference answer	15%
Date conversion	Date or relative-time conversion off by about one day/unit	5%
Reasoning error	Missed comparative or causal relationship	2%
Possible judge error	Potentially correct answer scored incorrectly	3%

False refusal is the largest category, but does not alone demonstrate that eviction caused every such error: retrieval failure can also withhold evidence from the answer model. The trace examples provide more specific support for eviction in individual cases.

Effects also vary across LoCoMo question types. On 54 single-hop questions, judge scores are 0.537 for MLB-UTILITY, 0.204 for FIFO, and 0.259 for MLB-TTL. On 48 multi-hop questions, EM is 0.083 for MLB-UTILITY and No-Forget versus 0.021 for FIFO and MLB-TTL. On 16 temporal questions, all five policies have EM 0.1875. The two open-domain questions and most LongMemEval categories (one to eight questions each) are too small for stable category-level conclusions. The aggregate improvement therefore should not be interpreted as uniform across question types.

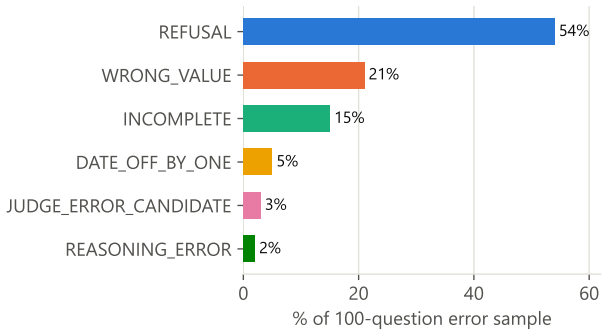


Figure 2: Error-mode taxonomy over the 100-question sample.

B.7 Oracle and full-context references

Oracle supplies only a question's gold-evidence memories. Full-Context retains all memories, as No-Forget does, but removes the top-5 retrieval cap. Oracle has privileged evidence access; it remains subject to extraction and answer-generation limitations, and is not a mathematical accuracy ceiling. Table 12 lists descriptive means over unequal samples; inferential comparisons require the matched subsets reported below.

Table 12: Oracle, Full-Context, and No-Forget results. Sample sizes differ; rows are not all paired comparisons.

Benchmark	Policy	Questions	EM	F1
LoCoMo	Oracle	107	0.1589	0.3473
LoCoMo	Full-Context	30	0.1333	0.2980
LoCoMo	No-Forget	120	0.0917	0.1957
LongMemEval	Oracle	22	0.1364	0.1814
LongMemEval	Full-Context	25	0.0800	0.1525
LongMemEval	No-Forget	25	0.0800	0.1525

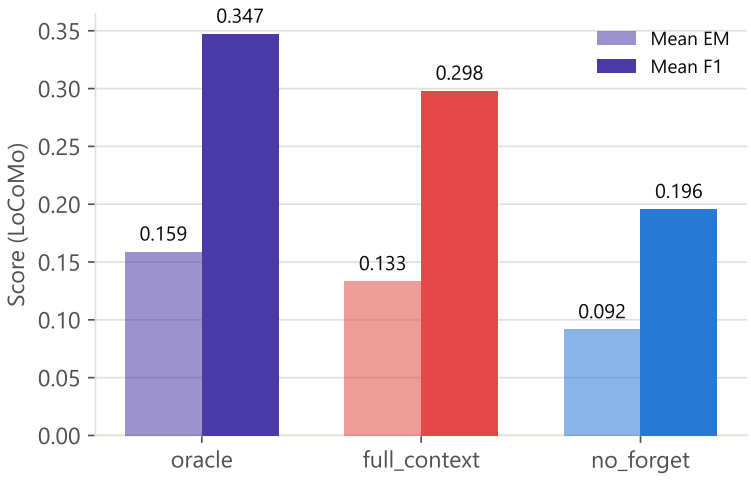


Figure 3: EM/F1 for oracle, full-context, and No-Forget references (LoCoMo).

On the 22 LoCoMo questions matched across all three policies, EM is 0.227 for Oracle, 0.136 for Full-Context, and 0.182 for No-Forget. F1 is 0.456, 0.350, and 0.297, respectively. Full-Context therefore lies between the other references on F1, but not on EM. The reported Full-Context/No-Forget F1 comparison gives $p = 0.041$; the other two reported pairwise tests give $p = 0.061$ and $p = 0.38$. The available aggregate report does not map those latter values unambiguously to individual pairs. The larger, 107-question Oracle/No-Forget comparison is reported separately in the main text.

B.8 MEM0 CONFIGURATION AND COMPARISON

An indexing-cost calibration used 60 turns and measured $0.01063 per turn for GPT-4o through OpenRouter, implying $62.55 for 5,882 turns. The reported run instead used Mem0's provider interface with a local Qwen2.5-7B-Instruct server. This removed the corresponding metered API charge but did not eliminate hardware cost. Indexing varied from approximately 4.4 seconds per turn (369 turns in 27.3 minutes) to more than 30 seconds per turn under shared-node contention; total indexing took several hours. These measurements describe that run, not a controlled latency benchmark.

The evaluated open-source path lacked working timestamp support, so dates were inserted as text prefixes. One observed relative-date conversion used 2026 rather than the conversation's 2023. In addition, 195 of 5,882 indexing calls produced malformed JSON (3.3%), contributing no extracted memories for those turns. These are confounds of the evaluated configuration and extraction substitution, rather than general properties of Mem0.

Table 13: Mem0 and the MemoryLifeBench policy pipeline on the same 120 LoCoMo questions. Extraction models and timestamp handling differ.

Policy/system	EM	F1	BLEU-1
Mem0	0.0833	0.2275	0.1808
No-Forget	0.0917	0.1957	0.1562
MLB-UTILITY	0.0917	0.1949	0.1549
LRU	0.0833	0.1998	0.1586
MLB-TTL	0.0500	0.1575	0.1116
FIFO	0.0417	0.1294	0.0915

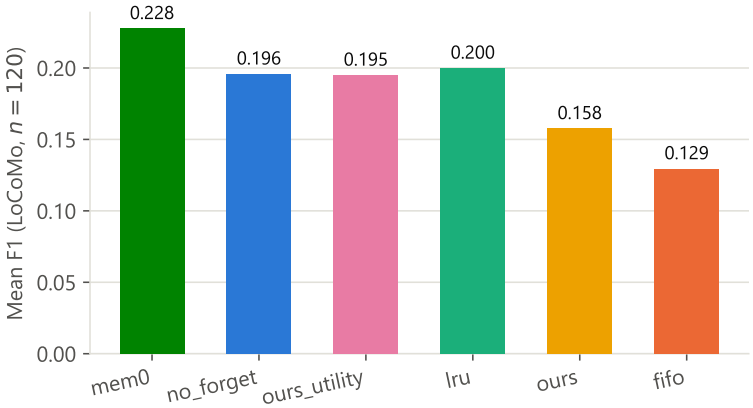


Figure 4: Mem0 vs. MemoryLifeBench retention policies (F1, LoCoMo, $n = 120$).

Mem0 has the highest descriptive F1 and BLEU-1. The reported F1 comparisons favor Mem0 over FIFO ($p < 0.001$) and MLB-TTL ($p = 0.005$), while comparisons with No-Forget, MLB-UTILITY, and LRU have $p > 0.13$. These tests do not establish equivalence and do not remove the system-level confounds.

C USE OF LARGE LANGUAGE MODELS

Experimental components. GPT-4o, Gemini 2.5 Pro, and local Qwen2.5-7B are prompted TTL baselines. GPT-4o also supplies grounded QA, judge rescoring, and synthetic-dialogue generation. Local Qwen2.5-7B performs extraction in the evaluated Mem0 configuration. These roles are part of the experimental pipeline.

Research and writing assistance. The project was developed through substantial collaboration with an AI coding assistant (Claude, Anthropic), under the author's direction and review. The assistant contributed implementation code, experiment execution and analysis, and substantial manuscript drafting, including corrections to earlier evidence-linkage and policy-superiority claims. An OpenAI assistant subsequently assisted with manuscript organization, terminology, mathematical exposition, bibliography reconstruction and LaTeX layout. That editorial pass did not rerun the experiments. The authors remain responsible for verifying the reported measurements, references, and claims.

D PROMPTS

For reproducibility, every prompt used in this work is reproduced verbatim below, rather than only referenced by file path.

D.1 TTL-PREDICTION BASELINE PROMPT (SECTION 6.1)

Used identically for all three LLM-prompted-TTL baselines (GPT-4o, Gemini 2.5 Pro, local Qwen2.5-7B) – the same prompt on purpose, so "which model" is the only variable being compared. The prompt asks for how long a fact remains true and useful; as noted in Section 3.1, this does not exactly match the real-data endpoint of time to future evidence use, so these LLM results are reference baselines rather than a clean test of validity-lifetime estimation.

You are estimating how long a piece of memory stays useful for a personal AI assistant.

Memory statement: "`{text}`"

Question: starting from when this was said, how many days from now is this fact likely to remain true and useful to remember, before it becomes outdated or irrelevant? Some facts are permanent (use a large number like 3650 for "essentially forever"), some last weeks or months, and some are only relevant for a day or two.

Answer with ONLY a single integer number of days. No words, no explanation.

D.2 GROUNDED-QA PROMPT (SECTION 6.2, ALL DOWNSTREAM POLICY COMPARISONS)

The only prompt that answers a question against retrieved memories – every EM/F1/BLEU-1/judge/refusal number in Section 6.2 traces back to a call using this template.

You are a personal AI assistant answering a question using only memories retrieved from what the user has told you before. Each memory is dated. For "when" questions, do the arithmetic: if a memory dated 2023-05-08 says something happened "yesterday", the answer is 2023-05-07, not "recently" or "yesterday" – always convert relative time words (yesterday, last year, last week, etc.) into an absolute date/year using the memory's date, and give that absolute answer.

Retrieved memories (most relevant first, each dated, each with how confident the system is that it's still true/relevant): `{retrieved_memories_block}`

User's question: "`{query}`"

Answer using ONLY the retrieved memories above. If none of them actually answer the question, say you don't have that information yet – do not guess or use outside knowledge. Give the SHORTEST correct answer: a bare date, name, number, or short phrase if that's all the question asks for – not a full sentence restating the question. Only use a full sentence if the question genuinely requires one.

Note the explicit refusal instruction ("say you don't have that information yet") – this is the exact phrase the refusal classifier in Section 6.4 was calibrated to detect.

D.3 LLM-JUDGE PROMPT (SECTION 6.4)

You are grading whether a candidate answer to a question is substantively correct, given a known reference answer. Judge by MEANING, not exact wording – paraphrases, different units/formats for the same fact, and answers that include the correct fact plus extra correct context all count as correct. A vague, wrong, contradictory, or "I don't know" answer when the reference has a real answer counts as incorrect.

Question: "`{question}`"
Reference answer: "`{reference}`"
Candidate answer: "`{prediction}`"

Respond with exactly one word: CORRECT or INCORRECT. No explanation.

D.4 MEMORY-EXTRACTION PROMPT (DOCUMENTED, NOT EXERCISED BY THIS PAPER'S RESULTS)

Exists for a future live-deployment path (raw dialogue turns to extracted memory statements) – MemoryLifeBench's own records are already extracted, pre-labeled statements (Section 3), so this prompt is not called anywhere in producing any number reported in this paper. Included for completeness, not because it was used.

You extract durable, memory-worthy statements from a single conversational turn, for a personal AI assistant's long-term memory store.

Conversation turn (speaker: `{speaker}`): "`{turn_text}`"

If this turn states a fact, preference, plan, or event worth remembering for future conversations, output it as a short, self-contained third-person statement (e.g. "User lives in Denver.", "User has a flight on March 3rd."). If it's small talk, a question with no lasting content, or nothing worth remembering, output exactly: NONE

Output ONLY the extracted statement (or NONE). No explanation.