Life of Research · Published 3 November 2026 · DOI 10.5281/zenodo.23899944

A Recall Score Is Not a Use Forecast: What Factual-Memory Benchmarks Predict About Whether a Conversational Memory System Uses What It Stores, Why the Published Transfer Failure Runs Through the Harness and the Write Policy, and the Within-Harness Test That Would Settle It

Pranay Mahendrakar
SONYTECH · ORCID 0009-0003-7224-029X
pranaymahendrakar@sonytech.in · mahendrakarpranay@gmail.com · +91 6361723454

Abstract. Long-term memory systems for conversational assistants are compared almost entirely on question-answering benchmarks that ask for a fact stated somewhere in a long history. A 2026 benchmark for what its authors call memory utilization reports that strength on those benchmarks does not reliably carry over to tasks that require combining scattered, implicit and distractor-laden evidence into a long answer. In this analysis the author reviews and synthesises published results to show that this conclusion bundles three claims that the evidence supports to very different degrees. The first, that retrieving the right evidence is necessary for good use but not sufficient, is well supported and is not new: it was already visible inside the recall benchmarks, which report correct retrieval followed by a wrong answer on 15 to 19 percent of questions, summaries retrieved with high recall that did not help, and models that recall a stated preference but fail to apply it unprompted. The second, that a system's rank on a recall benchmark does not predict its rank on use, rests on comparisons with scores the systems' developers reported for themselves. Against those, the five systems shared with the utilization benchmark largely reverse order (Spearman -0.6 by the author's arithmetic); against a shared harness maintained by the developer of one of them, MemOS, the three shared systems keep most of their order, and harnesses run by third parties place the system at the bottom of the utilization ranking low on recall as well. The third claim is implicit and is the one the evidence most undermines: a recall score is not a stable property of a system. The same system's published LoCoMo score spans 10 to 21 points across harnesses, and two harnesses using the same answering model rank four shared systems with zero rank correlation. The paper separates four places where use can fail after access succeeds, consolidates the measured values in a table, and states a within-harness transfer test as an algorithm.

Keywords: long-term memory; conversational agents; memory utilization; LoCoMo; LongMemEval; benchmark validity; evaluation harness; retrieval-augmented generation

1. Introduction

A conversational assistant that remembers a user across weeks has to keep something from each session and bring it back when it matters. Early work on multi-session open-domain dialogue already found that models able to retrieve or summarise earlier sessions beat standard architectures in that setting (Xu et al., 2021). The designs that do this have multiplied. MemGPT pages old context out of the window and back in on demand (Packer et al., 2023). MemoryBank, recursive summarisation and the generative-agents architecture store, condense and reflect on past interactions (Zhong et al., 2023; Wang et al., 2023; Park et al., 2023). A second generation of systems extracts facts or graph triples from each turn and retrieves them later: Mem0, Zep, A-Mem, MemoryOS, MemOS, LightMem and EverMemOS among them (Chhikara et al., 2025; Rasmussen et al., 2025; Xu et al., 2025; Kang et al., 2025; Li et al., 2025a; Fang et al., 2025; Hu et al., 2026). Surveys of the area now count dozens of such systems and a matching number of benchmarks (Zhang et al., 2024; Hu et al., 2025b).

The benchmarks these systems are compared on are mostly question-answering suites built over long dialogue histories. LoCoMo supplies long synthetic conversations between two personas with questions about single facts, multi-hop links, timing and open-domain knowledge (Maharana et al., 2024). LongMemEval embeds 500 questions about user-assistant chats in histories of about 115,000 tokens and scores five abilities, including multi-session reasoning and knowledge updates (Wu et al., 2025). Later benchmarks add reflective as well as factual memory and observation as well as participation scenarios (Tan et al., 2025), project-oriented histories in which goals evolve (Bian et al., 2026), and conversations of up to ten million tokens with probes beyond simple recall (Tavakoli et al., 2025). Systems papers report their gains on these two benchmarks above all others, and the headline numbers are high. A raw-retrieval system reports 93.5 percent on LoCoMo (Derehag et al., 2026); another reports 91.7 percent in its developer's own run with OmniMemEval's answering model, set beside OmniMemEval's published numbers for 14 other systems (Wang, 2026).

In 2026 Qing, Shi and Vosoughi asked whether these numbers say anything about what deployed assistants are actually asked to do. Their UtilMem benchmark poses 1,717 open-ended requests across five domains, each requiring the system to combine evidence from

© 2026 Pranay Mahendrakar. Licensed under CC BY 4.0.

several sessions into a summary, analysis or plan, inside histories padded with distractor sessions built to share vocabulary with the evidence (Qing et al., 2026). Their headline is that "strong performance on conventional factual-memory benchmarks does not reliably translate into effective memory utilization", and that retrieval alone does not suffice because systems that recover the relevant evidence still fail to integrate it or are pulled off course by distractors. The most extreme case in their table is Mem0, which its developers report, with its graph variant, as the strongest memory system on LoCoMo short of reading the full context (Chhikara et al., 2025) and which retains 17.3 percent of the reference score on UtilMem, against 58.9 to 59.3 percent for plain retrieval over raw sessions.

The result is plausible, and the paper's diagnosis of what recall benchmarks leave out is careful. But the sentence that carries it, strong on recall benchmarks and weak on use, is three claims in one. It could mean that, for a given system, getting the evidence back does not guarantee using it well. It could mean that the ordering of systems on recall benchmarks is not the ordering on use, so that a practitioner choosing a memory system by its LoCoMo score chooses badly. Or, read most strongly, it could mean that recall benchmarks measure a capability distinct from the one that matters, so that progress on them is not progress on memory. The three have different evidence requirements. The first needs within-system measurements with retrieval controlled. The second needs both kinds of score for the same systems under the same harness. The third needs the second plus a showing that the recall score is itself a stable property of the system being ranked.

In this analysis the author reviews and synthesises published results to show that the first claim is well supported and long known, the second is supported mainly by comparison with developers' own reports and partly dissolves against shared-protocol and third-party measurements, and the third runs into a problem that comes before transfer: the recall scores themselves move by 10 to 21 points for the same system across evaluation harnesses, enough to reorder systems without any change in task. The paper argues four things.

First, the gap between access and use is old news inside the recall benchmarks themselves. LoCoMo reported in 2024 that session summaries retrieved with high recall did not improve answers, and that long-context models did no better than short-context ones at summarising events across sessions (Maharana et al., 2024). LongMemEval reported that 15 to 19 percent of all its questions were answered wrongly after correct retrieval, 40 to 50 percent of the errors (Wu et al., 2025). Personalization benchmarks show the same split between recalling a user fact and acting on it (Jiang et al., 2025; Zhao et al., 2025). What UtilMem adds is a task family where the post-retrieval share is larger, not a new kind of failure.

Second, the transfer failure in UtilMem's comparison is concentrated in one design choice and one source of scores. Fact extraction at write time, the design Mem0 shares with several systems, does poorly on UtilMem and also does poorly on recall benchmarks when third parties run them (Pakhomov et al., 2025; Pollertlam and Kornsuwannawit, 2026; Hu et al., 2025a). The reversal appears when UtilMem is set against vendor-reported recall scores (Figure 1b).

Third, the recall score is a harness property as much as a system property. Across five published harnesses, Mem0's LoCoMo score ranges from 57.7 to 77.7 and Zep's from 63.8 to 85.2 (Figure 1a). In each of four harnesses built by a system's own developers that the author located, that system ranks first on LoCoMo among the memory systems it is compared with. Transfer cannot be assessed against a quantity that moves this much with who runs it.

Fourth, between access and use there are at least four distinct places to fail: what the write step kept, what the read step assembled under interference, whether the reader can compose many pieces, and whether the system applies stored information without being asked to. UtilMem tests the first three. It does not test the fourth, because its requests name the topics to be combined, and the fourth is where personalization benchmarks find their largest gaps in short conversations, though in conversations of about 100,000 tokens a reminder no longer closes them for some models.

The argument is not that UtilMem is wrong or that recall benchmarks are adequate. A raw-retrieval pipeline reaches 93.5 percent on LoCoMo and 88.4 percent on LongMemEval-S (Derehag et al., 2026), which the author reads as close to the ceiling, and most of their questions need evidence from at most two sessions (Uddin et al., 2026). UtilMem's diagnosis of what they omit is right. The argument is about what a comparison between the two kinds of score licenses, and about the controls that comparison needs before it can say that recall rank fails to predict use rank.

Section 2 fixes terms. Section 3 goes through what the anchor measured and how. Sections 4 to 6 take the three claims in turn. Section 7 separates the four failure places. Section 8 examines what the utilization score measures. Section 9 consolidates the evidence in Table 1, and Section 10 states the within-harness test as Algorithm 1. Sections 11 to 14 give the strongest objections, the studies that would settle the open part, what is not known, and the limitations.

2. Scope and terms

This paper concerns external memory for conversational agents: systems that write records derived from past dialogue to a store outside the model's weights and read some of them back when answering later. Parametric memory, editing and unlearning are outside its scope, as are agent memories of tool trajectories and environment states.

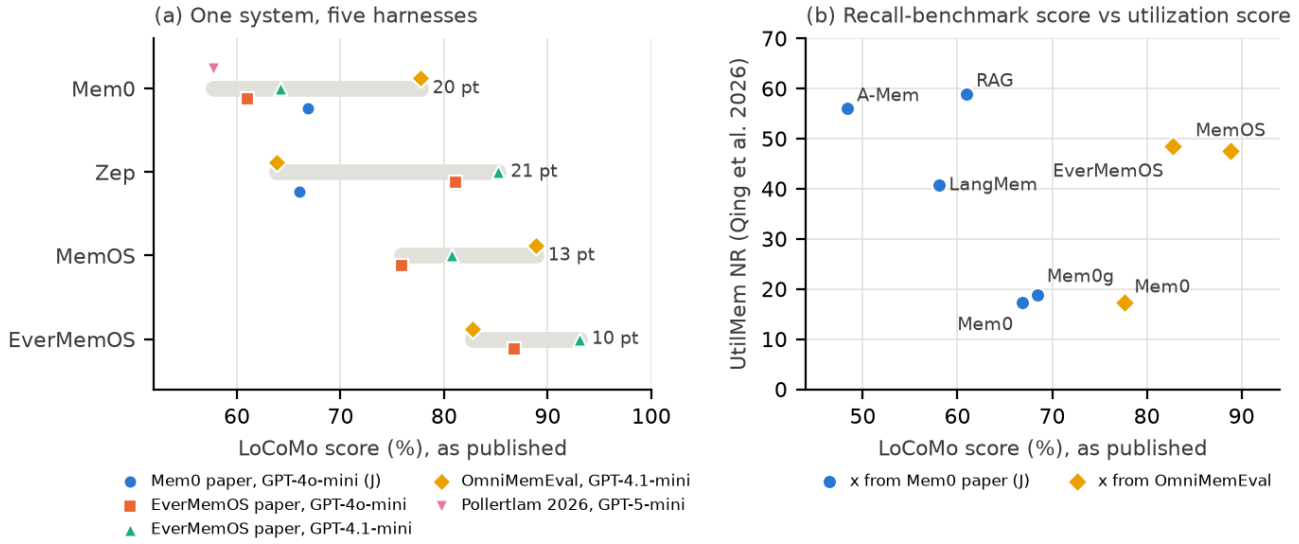


Figure 1. (a) LoCoMo scores published for the same four memory systems under five harnesses: the Mem0 paper's LLM-judge score with GPT-4o-mini (Chhikara et al., 2025, Table 2); the EverMemOS paper's accuracy with GPT-4o-mini and GPT-4.1-mini backbones (Hu et al., 2026, Table 1); OmniMemEval's re-evaluation with gpt-4.1-mini answering, as reported by Wang (2026, Table 1); and the Mem0 pipeline with a GPT-5-mini reader (Pollertlam and Kornsuwannawit, 2026, Table 3). Grey bars span each system's range; the label gives the range in points. (b) Each system's published LoCoMo score against its UtilMem normalised robustness, averaged over five domains (Qing et al., 2026, Table 2). Circles take LoCoMo from the Mem0 paper; diamonds from OmniMemEval. RAG on the y-axis is UtilMem's retrieval baseline with its default embedding; on the x-axis it is the Mem0 paper's best retrieval configuration. Metrics and graders differ across sources.

Four terms are used in fixed senses.

A recall score is a benchmark score on questions whose answers are short and are stated, or directly entailed, by a small number of turns in the history. LoCoMo's question-answering split and most of LongMemEval are recall benchmarks in this sense. The definition is about the task, not the benchmark's name: LoCoMo also contains an event-summarisation task that is not a recall task (Maharana et al., 2024), and the systems literature almost never reports it.

Access means that the evidence needed for a response is available to the answering model at answer time: it was kept by the write step and returned by the read step. Access can be measured as unit-level recall of gold evidence in what the reader receives, as UtilMem does for its retrieval baseline and LongMemEval does with Recall@k (Qing et al., 2026; Wu et al., 2025).

Use is the quality of the response conditional on access. A system that has access and responds well uses its memory; one that has access and responds badly does not. Use, so defined, is a property of the reader and of the form in which evidence reaches it, not only of the memory store.

A harness is everything between a memory system and its reported score: the ingestion procedure, chunking, retrieval budget, answering model, prompts, grader model and grading rubric, the subset of questions scored, and the version of the system. Two scores for the same system on the same benchmark under different harnesses are two measurements, not one.

With these terms, the three claims of Section 1 are: (C1) access is necessary and not sufficient for use; (C2) system rankings by recall score do not predict system rankings by use; (C3) recall scores do not measure the capability that use requires. C1 is a within-system claim. C2 and C3 are between-system claims, and both presuppose that a system has a recall score, in the singular.

3. What the anchor measured, table by table

UtilMem builds multi-session histories from five public sources: student conversations with an AI tutor, personal-finance questions, mental-health posts, fitness questions and S&P 500 annual filings (Qing et al., 2026). Sessions are rewritten into multi-turn dialogue by a pair of model agents. A question generator reads all evidence sessions of a bundle and writes ten candidate requests, constrained to use terms from the conversation so that retrieval-based systems can find the evidence. A counterfactual filter keeps a request only if two judge models agree that the answer degrades at each of two steps of random session removal. Distractor sessions are synthesised to share five to ten keywords with each evidence session while differing in domain and intent, and histories from the five domains are merged so that each instance carries its own strong distractors and the other domains' sessions as weak ones. The median number of evidence sessions per instance is 3 to 6 across domains, against 2 for LongMemEval-S, and the mean history length, about 120,000 tokens, is matched to LongMemEval-S.

Scoring is relative to a reference. For each instance the frozen answering model, Qwen3-235B, is given only the

evidence sessions and the request, and its answer is fixed as the reference with score 10. Each memory system ingests the full history, the same answering model is conditioned on what the system retrieves, and two judge models score the result from 1 to 10 against the reference after classifying it on five sub-dimensions. Normalised robustness (NR) rescales the mean score to 0 to 100; the degradation rate (DR) is the share of instances scored 6 or below.

Table 2 of the anchor reports the following averages over the five domains. Retrieval over raw sessions with three embedding models reaches 58.9 to 59.3 NR and 49.2 to 49.6 DR. A-Mem reaches 56.0 NR. MemOS, LangMem and EverMemOS fall between 40.7 and 48.4 NR. Mem0 reaches 17.3 NR with 98.3 percent of instances degraded, and its graph variant 18.9 NR. The authors read this as showing that keeping the original evidence matters more than building more compact or more structured representations.

Two further analyses address the access-use split directly. For the retrieval baseline, where retrieved units are observable, the mean score rises monotonically with the fraction of evidence sessions retrieved, in every domain. Among instances where at least 80 percent of evidence sessions were retrieved, mean scores remain at 6.4 (mental wellness) to 7.9 (finance) out of 10, and 53 percent of such instances in mental wellness and 29 percent in document analysis still score 6 or below. In a controlled study that holds recall at 1.0 and adds distractor sessions, NR falls with the number of distractors: at 40 strong distractors it is 47.4, 56.0 and 57.2 for three answering models of increasing size, and the first two or so distractors remove a quarter of the reference score. Weak distractors also hurt, leaving NR around 57 to 60 at 40 distractors.

Six features of the design matter for what follows.

First, the anchor reports no recall-benchmark scores of its own. The sentence that UtilMem performance does not track recall-benchmark performance is a comparison between the anchor's measurements and other papers' reported numbers, under other harnesses, answering models and graders. The anchor says as much in its limitations and conclusion, which scope the finding to "the systems and configurations tested".

Second, there is no full-context condition: no system reads the whole 120,000-token history. The distractor study is a partial substitute, since it shows the answering model degrading as irrelevant sessions are added to complete evidence, but it caps at 40 distractor sessions rather than the full haystack of roughly 60 to 70 sessions.

Third, the requests name what is to be combined. The published examples ask for "one combined recap" covering a listed set of topics, or a plan "using everything we discussed about" named items (Qing et al., 2026, Appendix D). This follows from the retrieval-compatible phrasing constraint, and it means the benchmark's "implicit retrieval" dimension is implicit in the evidence (preferences and patterns not stated as facts) but explicit in the trigger: the user has asked for the memory to be used.

Fourth, the reference is the same answering model's output from the oracle evidence, and the evidence-sensitivity filter used that model as its target. Requests were kept when this model's answer got worse as sessions were removed. The benchmark is therefore selected for instances where this generator benefits from complete evidence, and the score measures distance from this generator's own best answer. The authors note the selection effect explicitly.

Fifth, the zero-distractor anchor in the interference study is a judge's light paraphrase of the reference, not a fresh generation, because regenerating from identical input at temperature zero reproduces the reference. No condition reports the score of an independently generated answer from the same complete evidence, for instance with sessions reordered or at non-zero temperature. Without that condition, the part of the measured loss that reflects ordinary variation in how a long answer is organised, rather than lost evidence, is not identified.

Sixth, the validation is model-based. The two judges agree closely (Pearson 0.848, Spearman 0.843 over 1,717 retrieval-baseline outputs), but only 42.98 percent of their scores match exactly, and human inspection covered 50 development instances before the final generation run, at least 45 of which passed.

None of these features makes the anchor's measurements wrong. They set what the measurements can be compared with.

4. Claim one: access is necessary and not sufficient

C1 is the claim the anchor's data support most directly, and it is the claim with the longest prior record. This section collects that record, because its age matters for how the anchor's contribution is read.

4.1 The record inside the recall benchmarks

LoCoMo's authors evaluated retrieval over raw dialogue, over extracted observations and over session summaries. They found that retrieving the top five observations improved a model by about five points over raw logs, that the gain faltered as more observations were retrieved, and that session summaries did not help significantly despite high recall of the relevant session, which they attributed to information lost in summarising (Maharana et al., 2024). The same paper's event-summarisation task asked models to reconstruct the events in a speaker's life across sessions. The long-context model did worse than its short-context sibling, with lower precision by 3.0 points and lower recall by 8.7 points, and the authors concluded that long-context models "may not be proficient at utilizing their context appropriately". The benchmark that UtilMem describes as pointwise recall contained, at release, a task of the kind UtilMem argues is missing. It is the field's adoption

of LoCoMo that is pointwise: the Mem0 evaluation scores four question categories, excludes the adversarial category and does not report event summarisation (Chhikara et al., 2025).

LongMemEval decomposed memory into indexing, retrieval and reading and measured each (Wu et al., 2025). With the best memory design, correct retrieval followed by a wrong answer accounted for 15 to 19 percent of all instances and 40 to 50 percent of errors, more with weaker readers. Under oracle retrieval, where only evidence sessions are given, a poor reading strategy cost up to 10 points for GPT-4o, and extracting notes before answering, the Chain-of-Note procedure (Yu et al., 2023), combined with structured formatting of the retrieved items, was the best of the four reading designs tested. Long-context models answering from the full 115,000-token history fell 30 to 60 percent below their oracle-retrieval accuracy. Correct retrieval was necessary for a correct answer about 90 percent of the time. This is C1, stated with numbers, in 2024.

Operation-level diagnostics have since put shares on the layers. EvalMem runs separate examiners for whether a fact was stored, whether the native retriever returned usable evidence, and whether the answering model can answer from oracle evidence (Liu et al., 2026a). On LoCoMo, retrieval defects affected 22.1 percent of queries, encoding defects 7.7 percent and generation defects 6.5 percent; on LongMemEval-S the figures were 18.3, 8.5 and 5.4 percent. HaluMem scores extraction, updating and question answering separately and reports that all but one system extracted fewer than 60 percent of reference memory points, that update omission rates mostly exceeded 50 percent, and that question-answering accuracy stayed below 70 percent for every system, with errors propagating from the extraction stage (Chen et al., 2025).

4.2 The record in personalization benchmarks

Benchmarks built around applying what the user said, rather than restating it, show the same split more sharply. PersonaMem reports that frontier models score 60 to 70 percent on recalling user facts and tracking how preferences changed, but 30 to 50 percent on responses that incorporate the user's latest situation; models did better at describing how a preference had evolved than at acting on its current value (Jiang et al., 2025). CUPID gives models prior interaction sessions in which a user revealed context-dependent preferences through feedback, then a new request; no model exceeded 50 percent precision or 65 percent recall in inferring the relevant preference, and even with only the relevant sessions supplied, precision stayed under 70 percent (Kim et al., 2025). PRAGMA asks for personalized guidance grounded in longitudinal histories and reports that systems fail both to recover the right evidence and to use it once recovered (Yu et al., 2026). Memora finds memory agents strongest on remembering and weakest on reasoning, with language models competitive on recommendation, and documents frequent reuse of invalidated memories that ordinary accuracy does not penalise (Uddin et al., 2026). ImplexConv constructs cases where the relevant information is connected to the request only through semantically distant cues, so that similarity-based retrieval misses it (Li et al., 2025b).

4.3 The record in long-context evaluation

The general version of C1, that a model which can locate information in its context need not be able to use it, is established in long-context evaluation. Accuracy depends on where relevant information sits in a long input (Liu et al., 2024). Irrelevant sentences added to a math problem degrade accuracy substantially (Shi et al., 2023). RAG benchmarks separate noise robustness from information integration and find the latter hard (Chen et al., 2024). RULER found that models nearly perfect at the vanilla needle test dropped sharply on multi-needle, aggregation and tracing tasks, and only half of the models claiming 32,000-token contexts held up at that length (Hsieh et al., 2024). BABILong estimates that popular models effectively use 10 to 20 percent of their context for reasoning over scattered facts (Kuratov et al., 2024). NoCha's true-or-false claims about recent novels, the largest share of which (47.9 percent) require global reasoning over the book, left no open-weight model above chance despite strong synthetic scores (Karpinska et al., 2024). Michelangelo builds tasks that require discarding most of a context to reveal a latent structure (Vodrahalli et al., 2024), and LongBench v2 includes long-dialogue history understanding among its six categories, on a benchmark where the best model answering directly reached 50.1 percent overall (Bai et al., 2024). NoLiMa removes literal overlap between question and needle, and 11 of 13 models fell below half their short-context score at 32,000 tokens (Modarressi et al., 2025). Du et al. (2025) show performance falling by 13.9 to 85 percent with input length even when every relevant token is retrievable, even when irrelevant tokens are replaced by whitespace or masked. Joren et al. (2025) find that strong models answer well when the context is sufficient and fail to abstain when it is not; the author has argued elsewhere that sufficiency in this sense is a property that cannot be computed before the answer is known (Mahendrakar, 2026b). In retrieval-augmented generation more broadly, query-document relevance labels correlate only weakly with downstream task performance, and scoring each retrieved document by what it does to the generator's output correlates better (Salemi and Zamani, 2024).

Goldman et al. (2024) give this literature a vocabulary that fits the memory case. Long-context tasks differ along dispersion, how hard the needed information is to find, and scope, how much of it there is. Needle tests are low-scope and low-dispersion. Recall benchmarks for conversational memory are low-scope: in LoCoMo 94 percent of questions need evidence from at most two sessions, and in LongMemEval 85 percent (Uddin et al., 2026). UtilMem is designed to be higher-scope (medians of 3 to 6 sessions) and, through its distractors, higher-dispersion.

4.4 What the anchor adds to C1

Against this record, UtilMem's contribution to C1 is a measurement of size in a task family, not a discovery of kind. Its controlled interference study isolates something the earlier record shows only partially: with all evidence present, semantically similar distractors cost more than unrelated ones, and the cost is front-loaded in the first few distractors. Its high-recall analysis shows a residual gap, but with a caveat: at a recall of 0.8, up to a fifth of the evidence is missing and the retrieved set also contains distractors, so the gap mixes three causes the analysis does not separate.

One result in the record complicates the size of C1. On LoCoMo and LongMemEval-S, EvalMem attributes only 5 to 7 percent of queries to generation failure from oracle evidence (Liu et al., 2026a). On those benchmarks, access is most of the problem. The post-access share grows with scope: about a sixth of all LongMemEval questions under its authors' best design, 29 to 53 percent of UtilMem's high-recall instances in two domains, and half or more of PersonaMem's application questions, where the history is given to the model directly. C1 is true everywhere; its size is a property of the task. A recall benchmark is a task on which the post-access share happens to be small.

5. Claim two: does recall rank predict use rank?

C2 is the claim a practitioner cares about. It says that choosing a memory system by its LoCoMo or LongMemEval score will mislead someone who needs the system to synthesise. The anchor does not test C2 directly, since it reports no recall scores. It can be tested, imperfectly, by pairing the anchor's numbers with published recall scores for the same systems.

5.1 Against developers' own reports

The Mem0 paper reports an LLM-as-judge score on LoCoMo for its own two variants and for several baselines it ran itself, all with GPT-4o-mini answering (Chhikara et al., 2025). Five of these appear in UtilMem: Mem0, Mem0 with graph memory, retrieval-augmented generation over raw text, LangMem and A-Mem. On LoCoMo their order is Mem0-graph (68.44), Mem0 (66.88), best retrieval configuration (60.97), LangMem (58.10), A-Mem (48.38). On UtilMem it is retrieval (58.9 with the default embedding), A-Mem (56.0), LangMem (40.7), Mem0-graph (18.9), Mem0 (17.3). The Spearman correlation between the two orderings, computed by the author, is -0.6 (Figure 1b). The order is nearly reversed. This is the strongest published support for C2 that the author located, and it is what a reader of the anchor would naturally construct.

5.2 Against shared-protocol and third-party measurements

Shared-protocol harnesses give a different picture. Under OmniMemEval, a shared protocol that re-evaluated 14 memory systems with gpt-4.1-mini answering, as reported by Wang (2026), three of the UtilMem systems appear: MemOS (88.83 on LoCoMo), EverMemOS (82.75) and Mem0 (77.68). OmniMemEval is maintained by the company that develops MemOS (Wang, 2026; Li et al., 2025a), so it is a third-party harness for EverMemOS and Mem0 but not for MemOS, and MemOS's first place there is a developer-harness result. On UtilMem their order is EverMemOS (48.4), MemOS (47.5), Mem0 (17.3). The two leaders swap places by 0.9 NR, and Mem0 is last on both. The rank correlation is 0.5, computed by the author, on three points that cannot carry significance.

Measurements by groups outside every system they score place Mem0 low on recall tasks too. ConvoMem, a 75,336-question benchmark from authors outside the systems it compares, reports full-context reading at 70 to 82 percent even on its hardest multi-message cases, against 30 to 45 percent for Mem0, for histories under 150 conversations (Pakhomov et al., 2025). A cost-performance study running the Mem0 pipeline with a GPT-5-mini reader reports 57.68 percent on LoCoMo and 49.00 percent on LongMemEval, against 92.85 and 82.40 percent for the same reader with the full history (Pollertlam and Kornsuwannawit, 2026). In MemoryAgentBench, which runs all systems under one protocol with GPT-4o-mini, Mem0 averages 32.6 on accurate-retrieval tasks against 60.5 for BM25 retrieval over raw chunks (Hu et al., 2025a). PersonaMem reports retrieval over raw messages beating Mem0 across most question types (Jiang et al., 2025).

Read together, the shared-protocol and third-party record does not show a system that is strong at recall and weak at use. It shows a system that extracts facts at write time and is comparatively weak at both: about 11 points behind the leader, MemOS in its developer's harness, on LoCoMo under OmniMemEval, 33 to 35 points behind full context in the cost study, about 40 points behind retrieval on UtilMem. The reversal in Section 5.1 comes from setting a third-party utilization score against a developer-reported recall score.

5.3 The write policy as the interaction term

There is a coherent reason for a write-time extraction design to look better on recall than on use, and it does not depend on any harness. A fact extractor keeps the kind of content recall questions ask about: names, dates, stated preferences, events. It drops what synthesis needs: the reasoning in a conversation, the order in which things happened, the hedges and constraints a user attached, and anything the extractor did not recognise as a fact. LongMemEval measured this directly. Replacing sessions or rounds with extracted summaries or facts reduced accuracy because of information loss, with one exception: on multi-session reasoning questions, fact decomposition helped, plausibly because it normalised the format of the items to be aggregated (Wu et al., 2025). The cost study attributes Mem0's deficit to compression of roughly 101,600-token conversations into about 2,909 tokens of atomic facts per user (Pollertlam and Kornsuwannawit, 2026). UtilMem's own

reading is the same: fact extraction discards context that relational structure cannot restore (Qing et al., 2026).

The author's earlier analysis of write-time compression argued that every located query-agnostic compressor is tuned for, or scored against, questions drawn from the same generator it will be scored on, while a memory write must choose what to keep before the future question exists (Mahendrakar, 2026a). The present paper adds the converse consequence. A compressor tuned on recall-style questions will look good on recall benchmarks for exactly the reason it fails on synthesis: it keeps what those questions ask for. That is an interaction between write policy and task, and an interaction of this kind can reverse rankings between benchmarks without either benchmark being invalid. It predicts that the systems most penalised on UtilMem are those that compress hardest at write time, which is what the anchor's table shows, and it predicts that the same systems will lose less on recall benchmarks than on synthesis benchmarks, which is what the third-party record shows.

Two results warn against reading this as "compression hurts use". PRAGMA gives the answering model either the gold evidence sessions or concise summaries of the same evidence distilled from annotations. Under gpt-5-mini on its trajectory-alignment task, the summaries scored 86.00 on grounding against 17.00 for the raw sessions (Yu et al., 2026). A well-targeted compression helped use a great deal. MemSuit trains a summariser to produce memory entries that preserve what observed question-answer pairs needed, and reports gains over strong baselines across query types (Lei et al., 2026). PERMA reports that memory systems linking related interactions extract preferences more precisely than retrieval over raw dialogue (Liu et al., 2026b), and SeCom reports that segment-level memory units with compression-based denoising outperform turn-level, session-level and summary-based baselines on LoCoMo (Pan et al., 2025). The anchor's conclusion that keeping raw evidence matters more than compact representation is true of the compressors it tested, which compress without knowing the future request. It is not true of compression as such. What matters for use is whether the write step kept what the read step will need, and that is a property of the match between the write policy and the task distribution.

5.4 Where C2 stands

C2 holds, on present evidence, in a narrow form. Systems that compress aggressively at write time lose more on high-scope synthesis than on low-scope recall, so the gap between them and raw-retrieval systems widens, and a ranking that mixes compression-heavy and raw systems can reorder between the two kinds of benchmark. Whether rankings among systems with similar write policies also fail to transfer is not shown by any published comparison. The one shared-protocol pairing available, in a harness maintained by the developer of the system it places first, preserves the order of the bottom system and swaps the top two by less than one NR point.

6. Claim three: whose recall score?

C2 and C3 presuppose that a system has a recall score. The published record does not support that presupposition at the precision the comparisons need.

6.1 One system, several numbers

Figure 1a collects the LoCoMo scores published for four systems under five harnesses. Mem0 is reported at 66.88 by its developers with GPT-4o-mini (Chhikara et al., 2025), at 61.00 and 64.20 by the EverMemOS developers with GPT-4o-mini and GPT-4.1-mini (Hu et al., 2026), at 77.68 under OmniMemEval with gpt-4.1-mini (Wang, 2026), and at 57.68 in the cost study with a GPT-5-mini reader (Pollertlam and Kornsuwannawit, 2026): a 20-point range. Zep is reported at 65.99 by the Mem0 developers, at 81.06 and 85.22 by the EverMemOS developers, and at 63.83 under OmniMemEval: a 21-point range. MemOS ranges from 75.87 to 88.83 and EverMemOS from 82.75 to 93.05.

Some of the spread is model capability and is expected: a stronger answering model raises every system. But the spread does not order itself by answering model. The EverMemOS harness and OmniMemEval both answer with GPT-4.1-mini. Under the first, the four shared systems rank EverMemOS (93.05), Zep (85.22), MemOS (80.76), Mem0 (64.20). Under the second, they rank MemOS (88.83), EverMemOS (82.75), Mem0 (77.68), Zep (63.83). The Spearman correlation between the two orderings, computed by the author, is zero. Zep moves from second to last and Mem0 from last to third, with the answering model held fixed. Mem0 leads Zep in two of the four harness-and-model combinations that report both and trails it, by about 20 points, in the other two.

LongMemEval shows the same pattern where several measurements exist. Zep reports 63.8 percent with GPT-4o-mini and 71.2 percent with GPT-4o on LongMemEval-S (Rasmussen et al., 2025); OmniMemEval reports 79.8 percent (Wang, 2026); MemoryAgentBench reports 38.3 on its LongMemEval-derived subset (Hu et al., 2025a); and the EverMemOS paper, which takes its LongMemEval baselines from the official MemOS results, lists 63.80 (Hu et al., 2026). Mem0 is at 66.40 in that EverMemOS table, 56.0 under OmniMemEval, 49.00 in the cost study and 36.0 in MemoryAgentBench's subset, a spread of 30 points. Wang (2026) summarises the OmniMemEval exercise as showing that "several self-reported scores fell by 10-40 points" under one protocol.

6.2 What moves the number

The harness differences are documented, even where their effects are not separated. Graders differ: OmniMemEval grades with gpt-4o-mini and Mnemon with gpt-4.1-mini, which Wang (2026) reports changes scores by one to two points on LoCoMo and more on HaluMem. Question subsets differ: the Mem0 evaluation drops LoCoMo's adversarial category (Chhikara et al.,

2025). Integration differs: EverMemOS runs some baselines through their official APIs with the vendor's memory configuration and substitutes only the answering model (Hu et al., 2026), while UtilMem follows a shared ingestion pipeline drawn from another memory project (Qing et al., 2026; Fang et al., 2025). Retrieval budgets differ: on LoCoMo, the context sent to the answering model per question ranges from about 1,200 to 32,500 tokens across the OmniMemEval systems (Wang, 2026), and SmartSearch shows that the budget interacts with ranking so strongly that, without a reranker, only 22.5 percent of gold evidence survived truncation even though raw recall was 98.6 percent (Derehag et al., 2026).

There is also an observation the author cannot explain away and does not overinterpret. In each of the four harnesses built by a system's developers that the author located, the developers' system ranks first on LoCoMo among the memory systems compared: Mem0-graph in the Mem0 paper, EverMemOS in the EverMemOS paper, MemOS in OmniMemEval, which is maintained by the company that develops MemOS (Wang, 2026; Li et al., 2025a), and Mnemon in the Mnemon paper, which sets its own run beside OmniMemEval's published numbers. Four cases are too few to estimate an effect, and developers are expected to configure their own systems better than anyone else's. That is the point: a developer harness measures the system together with its best configuration, and a competitor configured by the same developer is measured together with a worse one. Neither is the system's recall score.

6.3 Consequence for C2 and C3

If a system's recall score varies by 10 to 21 points across harnesses and rankings among systems are uncorrelated between two harnesses using the same answering model, then any cross-paper claim that recall rank does or does not predict use rank is measuring harness disagreement as much as transfer. This applies to both directions. The reversal in Figure 1b is not evidence for C2 until the recall scores come from the harness that produced the use scores. The partial preservation under OmniMemEval is not evidence against C2 for the same reason.

C3, that recall benchmarks measure the wrong construct, cannot be settled by any of this. The construct-validity critique of benchmarks argues that a benchmark stands in for a capability only under an argued link between task and capability (Raji et al., 2021; Bowman and Dahl, 2021), and a review of 445 LLM benchmarks found that only 53.4 percent of the reviewed articles present evidence for that link (Bean et al., 2025). Recall benchmarks have an obvious link to one part of memory, access, and the record in Section 4 shows that access is necessary for use. What they lack is coverage of the post-access part. That is a scope limitation, not a measurement of the wrong thing, and it implies that recall scores should predict use where the post-access share is small and fail to where it is large. That prediction is testable, and Section 10 states the test.

7. Four places between access and use

C1 says use can fail after access succeeds. The record locates the failure in four places that a single "utilization" score combines. Separating them matters because they have different remedies and are covered by different benchmarks.

7.1 Write: what was kept

A write step that summarises, extracts or overwrites decides what the reader can ever see. Use fails here when the kept form lost what the task needs, even though a record about the right topic is retrieved. LoCoMo's summaries retrieved with high recall but no gain (Maharana et al., 2024), LongMemEval's fact and summary values (Wu et al., 2025), HaluMem's extraction recall below 60 percent (Chen et al., 2025) and UtilMem's compression-heavy systems (Qing et al., 2026) are failures here. Strictly, a write failure is an access failure at the level of content, even when it is an access success at the level of the record. Unit-level recall, which counts whether the right session or record was retrieved, cannot see it. This is one reason recall metrics and use metrics diverge for compression-heavy systems: the retriever finds the record, and the record no longer contains the detail.

7.2 Compile: what reached the reader, in what company

Between retrieval and reading, a system decides how many items to pass, in what order, at what budget, and mixed with what. Use fails here when the right evidence is retrieved but is truncated, buried, or accompanied by plausible distractors. SmartSearch's gap between 98.6 percent recall and 22.5 percent survival after truncation is a compile failure (Derehag et al., 2026). LoCoMo's observation that gains faltered as more observations were retrieved is one (Maharana et al., 2024), and LongMemEval's finding that the best retrieval budget depends on the reader's capability is another (Wu et al., 2025). UtilMem's interference study is the cleanest measurement of compile failure in conversational memory: evidence complete, distractors added, quality falling (Qing et al., 2026). The general RAG record says what kind of distractor matters. Passages the retriever ranks highly that lack the answer hurt, while in one study random passages raised accuracy by up to 35 percent (Cuconasu et al., 2024). UtilMem's weak noise was not neutral, but its strong noise did more damage, consistent with that record.

7.3 Compose: whether the reader can combine many pieces

With the right evidence present and nothing else, the reader still has to combine it. Use fails here when evidence is complete and clean and the answer is still poor. LongMemEval's oracle-retrieval condition measures compose failure directly, and Chain-of-Note recovers up to 10 points of it (Wu et al., 2025). PRAGMA's gap between oracle sessions and oracle summaries is compose failure that a better presentation removes (Yu

et al., 2026). The long-context record on aggregation, global reasoning and latent structure (Hsieh et al., 2024; Karpinska et al., 2024; Vodrahalli et al., 2024; Kuratov et al., 2024) and on length alone (Du et al., 2025) bears on this place. MemoryAgentBench's long-range-understanding tasks are compose-heavy, and there the ordering of agent types inverts: retrieval agents with GPT-4o-mini beat the same model reading long context on accurate retrieval (60.5 against 49.2 for BM25) and lose on novel summarisation (19.0 against 28.9) (Hu et al., 2025a). That is a within-harness reversal between a recall task and a composition task, and it is the best published evidence for C2 in its proper form. Its scope should be stated plainly: it is a reversal between agent types (retrieval versus full context), not among memory systems with similar designs.

UtilMem cannot isolate compose failure, because its reference is the answering model's own output from clean complete evidence. By construction, a pipeline that hands the reader exactly the evidence set, in the form used to make the reference, reproduces the reference, whatever the reference's quality. Compose quality is held at the generator's ceiling and subtracted out. That is a reasonable design for comparing memory systems with a frozen reader, but it means UtilMem's NR is a measure of write and compile failures relative to the reader's own composition, not of composition.

7.4 Trigger: whether stored information is applied without being asked

The fourth place is the one recall and utilization benchmarks share a blind spot about. A user who says early on that they are vegetarian, then later asks for dinner suggestions without mentioning it, needs the assistant to recognise that the stored fact applies. Use fails here when evidence is available and the reader could use it if prompted, but does not. PrefEval measured this directly. In conversations of about 10 turns and 3,000 tokens, preference-following accuracy on the restaurant topic without any prompt was 0.07 for both Claude 3.5 Sonnet and Gemini 1.5 Pro, and a single sentence reminding the model to consider earlier preferences raised Gemini to 0.91 and Claude to 0.45 (Zhao et al., 2025). At about 300 turns and 103,000 tokens, the setting closest to UtilMem's histories, the reminder no longer helped either model: Claude scored 0.02 with and without it, and Gemini fell from 0.09 to 0.05. The authors' own summary is that the reminder brings improvement, "however, accuracy still deteriorates with longer context lengths" (Zhao et al., 2025). In the short conversations the model had access and could compose; it did not trigger. In the long ones, prompting it to trigger was not enough. CUPID's finding that models fail to discern which prior context is relevant to a new request (Kim et al., 2025), PersonaMem's gap between recalling a preference change and acting on the latest preference (Jiang et al., 2025), and ImplexConv's semantically distant cues (Li et al., 2025b) are trigger failures. Multi-turn work outside memory finds a related unreliability: models commit early in a conversation to assumptions they do not revisit (Laban et al., 2025).

Recall benchmarks never test trigger, because the question asks for the stored fact. UtilMem does not test it either: its requests name the sessions' topics, and its retrieval-compatible phrasing constraint exists so that the query can find the evidence. The "implicit" in UtilMem's implicit-retrieval dimension refers to evidence that states a preference or pattern indirectly; the request to use memory is explicit. The place where personalization benchmarks record their largest gaps in short conversations is the place neither kind of conversational-memory benchmark examines.

Table 2 gives the author's reading of which places each benchmark exercises. A cell marked "yes" means the benchmark varies or stresses that place in some of its items; "partly" means it does so incidentally or in a minority of items; "no" means its design holds that place fixed or bypasses it.

Table 2. Which of the four failure places each benchmark exercises, by the author's reading of the cited designs.

Benchmark	Write	Compile	Compose	Trigger
LoCoMo QA (Maharana et al., 2024)	partly	partly	no	no
LongMemEval (Wu et al., 2025)	yes	yes	partly	no
UtilMem (Qing et al., 2026)	yes	yes	no	no
MemoryAgentBench (Hu et al., 2025a)	yes	yes	yes	no
PrefEval (Zhao et al., 2025)	no	partly	no	yes
PersonaMem (Jiang et al., 2025)	partly	partly	partly	yes
CUPID (Kim et al., 2025)	no	yes	partly	yes

UtilMem's compose cell reads "no" for the reason given in Section 7.3: its reference is the frozen reader's own composition.

8. What the utilization score measures

UtilMem's NR is a reference-matching score: two LLM judges grade a long answer against a reference produced by the same model from oracle evidence, after classifying deviations on five sub-dimensions. Three questions about it bear on the comparison with recall scores.

The first is noise floor. Long open-ended answers vary in organisation and emphasis even when produced from identical evidence. The judges are instructed to permit content supported by the evidence that the reference omits (Qing et al., 2026, Appendix E), but the score cap from any "clear" deviation on any sub-dimension means organisation differences can register as losses. The anchor's zero-distractor anchor is a paraphrase by the judge, so the score an independent regeneration from identical evidence would receive is not reported. Weak distractors, unrelated sessions from other domains, bring NR to about 57 to 60 at 40 distractors, within a few points of strong distractors at the same count for the two larger readers. Either a long list of unrelated sessions is

nearly as harmful as a list of related ones, or part of the measured loss at high distractor counts is a regeneration floor that any change of input produces. The data reported do not distinguish the two.

The second is validity against people. LLM judges agree with human preferences at roughly human-human rates on conversational benchmarks but show position, verbosity and self-preference effects (Zheng et al., 2023). HELMET validated its model-based long-output metric against human judgment before relying on it, reporting kappa of 0.91 for summary precision and 0.76 for recall (Yen et al., 2025). UtilMem reports agreement between two judges from the same model family and a development-time inspection of 50 instances. Agreement between two judges establishes reliability, not validity.

The third is what the score is relative to. Because the reference is the frozen reader's own answer from the evidence set, NR measures how much of that reader's best answer a memory pipeline preserves. It does not measure whether that answer is good, and it cannot reward a pipeline that gives the reader something better than the designated evidence, for example a well-targeted summary of the kind PRAGMA found more useful than raw sessions (Yu et al., 2026). A system whose write step produces PRAGMA-style distillations would be scored on how closely its reader's answer matches the reader's answer from raw sessions, and could be penalised for improving on it.

None of this makes NR a bad instrument for its stated purpose, comparing memory pipelines under a fixed reader. It does mean that NR and a recall benchmark's accuracy differ in more than task: one is accuracy against a human-written answer, the other is fidelity to a model's own answer, graded by two related models. A low rank correlation between them would be expected even for two tasks that tested the same capability.

9. Consolidated evidence

Each row of Table 1 is one published measurement on which the argument depends. The last column names the claim from Section 2 the row bears on. All values are as reported by the cited source except the three rank correlations, which are the author's arithmetic on the cited values and are labelled as such.

Four patterns stand out in Table 1. First, every row bearing on C1 points the same way, across benchmarks built by different groups over three years: access is necessary, and a substantial share of failure occurs after it. Second, among the rows bearing on C2 that hold the harness fixed, MemoryAgentBench shows a reversal between agent types, and ConvoMem and the cost study show a compression-heavy system weak on recall, which with UtilMem makes it weak on both recall and synthesis, not a system strong at one and weak at the other. Third, the only near-reversal among memory systems uses developer-reported recall scores. Fourth, the C3 rows show disagreement among recall harnesses as large as the disagreement between recall and use. HELMET's row is the general analogue: in long-context evaluation, synthetic retrieval tasks are imperfect predictors of downstream tasks across 35 models, while harder recall tasks with distracting needles predict better (Yen et al., 2025). That last finding carries over to conversational memory as a prediction: recall benchmarks with strong distractors should predict UtilMem-style use better than recall benchmarks without them.

10. A within-harness transfer test

Algorithm 1 states the procedure that would test C2 and the scope form of C3. The procedure follows from Sections 4 to 8: hold the harness fixed, run recall and use tasks over the same histories, measure each failure place separately, and estimate rank agreement with an interval.

Four points about the procedure.

First, steps 1 to 3 remove the problem of Section 6. A system's recall score is measured in the same harness that measures its use score, with the same reader and grader. The developer-reported score is still reported (step 11), but beside the harness score, so a reader can see how much of a system's reputation comes from its own harness.

Second, the oracle conditions of step 5 separate the four places of Section 7. O1, which gives the reader the system's own store filtered to the gold evidence, measures what the write step kept. O2, which gives the reader exactly the gold evidence, measures the compile and compose ceiling. O3, which repeats O2 with a different order or seed, gives the regeneration floor whose absence Section 8 noted; any loss smaller than the floor is noise. O4 rewrites the request so that it no longer names the topics to be combined, which turns a use task into a trigger task. EvalMem's three examiners are a version of O1 and O2 for recall tasks (Liu et al., 2026a), and PrefEval's reminder condition is the converse of O4 (Zhao et al., 2025).

Third, step 4 requires references checked by people. Model-generated references are cheaper, but the comparison in Section 8 shows that a reference produced by the reader from oracle evidence fixes compose quality at the reader's ceiling and can penalise pipelines that improve on raw evidence.

Fourth, steps 7 to 10 make C2 a measurable quantity with an interval, at a stated scope, and stratified by write policy. The interaction argument of Section 5.3 predicts a positive rho within strata and a lower or negative rho across strata when compression-heavy and raw systems are mixed. If rho is high within strata, recall benchmarks are adequate for choosing among similar designs and inadequate only for choosing between designs. If rho is low within strata, the strong form of C2 holds.

Applied to the published record, the procedure stops early. No study located here runs recall and use tasks over the same histories in the same harness for memory systems. MemoryAgentBench comes closest, running

Table 1. Published measurements bearing on whether recall scores predict use in conversational memory. NR is UtilMem's normalised robustness (0-100); LC is full-context reading.

Source	Setting	Condition	Comparison	Reported value	Bears on
Qing et al. 2026	UtilMem, 5 domains	raw-session RAG	Mem0 / Mem0 + Graph	58.9-59.3 vs 17.3 / 18.9 NR	C2
Qing et al. 2026	UtilMem, 5 domains	A-Mem; MemOS, LangMem, EverMemOS	raw-session RAG	56.0; 40.7-48.4 vs 58.9-59.3 NR	C2
Qing et al. 2026	UtilMem, RAG, recall ≥ 0.8	mean score / 10	oracle reference = 10	6.4-7.9; 29-53% scored ≤ 6 in two domains	C1
Qing et al. 2026	recall 1.0, 40 strong distractors	three readers	0 distractors	NR 47.4 / 56.0 / 57.2	C1 (compile)
Chhikara et al. 2025	LoCoMo, GPT-4o-mini, J	Mem0 + Graph; Mem0	RAG best; LangMem; A-Mem	68.44; 66.88 vs 60.97; 58.10; 48.38	C2, C3
Author, from rows 1-2, 5	5 shared systems	Mem0-paper LoCoMo	UtilMem NR	Spearman -0.6	C2
Wang 2026 (OmniMemEval)	LoCoMo, gpt-4.1-mini	MemOS; EverMemOS; Mem0	Zep	88.83; 82.75; 77.68 vs 63.83	C3
Author, from rows 1-2, 7	3 shared systems	OmniMemEval LoCoMo	UtilMem NR	Spearman 0.5	C2
Hu et al. 2026	LoCoMo, GPT-4.1-mini	EverMemOS; Zep; MemOS	Mem0	93.05; 85.22; 80.76 vs 64.20	C3
Author, from rows 7, 9	4 shared systems, same reader	EverMemOS harness	OmniMemEval	Spearman 0.0	C3
Pollertlam & K. 2026	LoCoMo / LongMemEval, GPT-5-mini	Mem0 pipeline	LC	57.68 / 49.00 vs 92.85 / 82.40	C2
Pakhomov et al. 2025	ConvoMem, < 150 conv.	Mem0	LC	30-45% vs 70-82%	C2
Hu et al. 2025a	MemoryAgentBench, GPT-4o-mini	BM25 RAG	LC	AR 60.5 vs 49.2; Summ. 19.0 vs 28.9	C2
Hu et al. 2025a	MemoryAgentBench	Mem0	BM25 RAG	AR 32.6 vs 60.5; Summ. 4.8 vs 19.0	C2
Wu et al. 2025	LongMemEval, best design	correct retrieval, wrong answer	all instances	15-19% (40-50% of errors)	C1
Wu et al. 2025	LongMemEval, oracle retrieval	weak reading strategy	Chain-of-Note + JSON	up to 10 pts lower	C1 (compose)
Maharana et al. 2024	LoCoMo, summary RAG	high session recall	raw logs	no significant gain	C1 (write)
Liu et al. 2026a	LoCoMo / LME-S defects	retrieval	encoding; generation	22.1 / 18.3 vs 7.7 / 8.5; 6.5 / 5.4%	C1
Derehag et al. 2026	LoCoMo, oracle	raw recall	gold kept after truncation	98.6% vs 22.5%	C1 (compile)
Jiang et al. 2025	PersonaMem, frontier LLMs	recall facts, track changes	apply latest state	60-70% vs 30-50%	C1 (trigger)
Zhao et al. 2025	PrefEval, 10 turns	zero-shot	one-line reminder	Gemini 0.07 vs 0.91; Claude 0.07 vs 0.45	C1 (trigger)
Zhao et al. 2025	PrefEval, 300 turns	zero-shot	one-line reminder	Gemini 0.09 vs 0.05; Claude 0.02 vs 0.02	C1 (trigger)
Kim et al. 2025	CUPID, 10 LLMs	preference inference	gold preference	precision < 50%, recall < 65%	C1 (trigger)
Yu et al. 2026	PRAGMA, gpt-5-mini	oracle summaries	oracle sessions	grounding 86.00 vs 17.00	write policy
Uddin et al. 2026	LoCoMo / LongMemEval	questions needing ≤ 2 sessions	all questions	94% / 85%	scope
Yen et al. 2025	HELMET, 35 models	synthetic tasks	downstream tasks	no average Spearman > 0.8	C3

accurate-retrieval and long-range-understanding tasks for many systems under one protocol (Hu et al., 2025a), but over different source texts for the two task families, without oracle conditions for memory systems, and without an interval on the cross-task correlation. UtilMem runs the use task and part of step 5 (O2, via its reference, and a version of the compile condition through its interference study), but not step 3. EvalMem runs the oracle decomposition for recall tasks only.

11. The strongest case against this reading

"The Mem0 numbers are from the vendor, but they are the numbers practitioners see. The anchor's comparison is the practically relevant one." This is right about practice and is the reason the comparison matters. But the practical lesson is then about vendor reporting, not about recall benchmarks. A practitioner misled by a developer's LoCoMo score would have been misled on LoCoMo itself: harnesses not run by Mem0's developers put it from 11 to 35 points below the leaders on recall, and the fully third-party cost study alone puts it 33 to 35 points below full context (Section 5.2). The

Algorithm 1. Within-harness test of recall-to-use transfer

```
Input: memory systems S, histories H, recall questions R(H),
       use requests U(H) with gold evidence E, readers M, grader G
1  fix harness: ingestion, budget b, reader m in M, grader G, prompts
2  for each s in S: ingest H once; store snapshot; log retrieved units
3  score recall: acc_R(s) on R(H) with reader m, grader G
4  score use: q_U(s) on U(H), judged against human-checked references
5  run oracle conditions on U(H) for each s:
     O1 store oracle   (reader sees s's store filtered to E)
     O2 compile oracle (reader sees exactly E, shuffled)
     O3 regen floor    (O2 repeated with new order or seed)
     O4 trigger probe  (request rewritten without naming topics)
6  attribute loss: write = O1 − O2; compile = q_U − O1;
     compose = human score of O2; floor = O2 − O3; trigger = q_U − O4
7  rho <- Spearman(acc_R, q_U) over S; bootstrap over H for interval
8  repeat 1–7 for >= 2 readers and >= 2 graders; report all rho
9  if interval of rho excludes 0 and is positive: recall predicts use
     at this scope; else if it includes 0: transfer not shown
10 stratify by write policy (raw, summary, fact extraction) and
     report rho within strata
11 report each system's developer-harness score beside acc_R(s)
```

remedy that follows is independent evaluation, which recall benchmarks support, not abandoning recall benchmarks.

"**Harness variance affects all benchmarks. UtilMem's scores would vary too, so the comparison is fair.**" UtilMem's scores would vary, and no second harness has measured them. That strengthens the point rather than answering it: a single-harness use score set against multi-harness recall scores compares one measurement with a spread. Section 6 shows that the spread is wide enough to reorder systems. Until UtilMem is run by a second group, its ordering carries the same caveat the recall orderings do.

"**Section 5.3's write-policy interaction concedes C2. If rankings reverse because of write policy, recall benchmarks do fail to predict use.**" They fail to predict use across write policies, at high scope. That is a real and useful limitation, and the paper states it. It is narrower than the anchor's headline, which a reader can take to mean that recall progress is not memory progress. Under the interaction reading, recall progress for raw-retrieval systems (better ranking, better budgets) should transfer to use, because those systems keep the evidence and the improvements act on compile and compose. SmartSearch's ranking gains on recall and the anchor's finding that raw retrieval leads on use are consistent with that (Derehag et al., 2026; Qing et al., 2026).

"**Trigger is a model property, not a memory property. A memory benchmark rightly asks for the memory explicitly.**" For evaluating memory stores in isolation, this is defensible, and it is UtilMem's choice. But the anchor motivates its benchmark by what deployed assistants do, and deployed assistants are not usually told which past conversations to consult. The evidence in Section 7.4 says that, in short preference-following conversations of about 10 turns and 3,000 tokens, failing to apply a stored preference unprompted was a far larger failure than failing to apply it when reminded; at about 103,000 tokens the reminder no longer helped Claude 3.5 Sonnet or Gemini 1.5 Pro (Zhao et al., 2025). A benchmark of use that names its evidence measures use conditional on trigger.

"**The rank correlations in this paper are on three to five points. They prove nothing.**" Agreed, and the paper uses them for that reason: to show that the published record cannot settle C2 in either direction. The correlation of -0.6 on five points and 0.5 on three are both compatible with zero. The correlation of zero between two harnesses with the same reader is compatible with modest positive agreement. Algorithm 1 asks for intervals because none of these numbers has one.

12. Studies that would settle the open part

The open part could be closed by five studies, none of which needs new data collection beyond what the public benchmarks and open-source memory systems already provide.

UtilMem with recall in the same harness. Run LoCoMo and LongMemEval-S through UtilMem's ingestion pipeline, reader and judges for the same systems, and report the rank correlation with a bootstrap interval over histories. This is steps 1 to 3 and 7 of Algorithm 1, and it is the single cheapest way to turn the anchor's cross-paper sentence into a measurement.

A regeneration floor for reference-matched scoring. Score the reader's answer from the oracle evidence with sessions shuffled, and at two non-zero temperatures, against the fixed reference. If the result is near the reference, the anchor's weak-noise curve measures harm from unrelated sessions. If it is well below, part of the curve measures regeneration variance.

A trigger split of a use benchmark. Rewrite a sample of UtilMem requests so they ask for the same output without naming the topics ("what should I tell my coach?" in place of "combine my notes on creatine, sleep and hydration"), and report the drop. PrefEval's results predict a large one in short histories. At about 103,000 tokens a reminder no longer helped the two models named in Section 7.4 (Zhao et al., 2025), so for histories of UtilMem's length PrefEval does not predict the size of

the drop. If the drop is small, trigger is not the bottleneck in synthesis tasks at that length; if it is large, use benchmarks that name their evidence overstate use.

Write-policy strata with targeted compression. Add to the UtilMem comparison a system whose write step is trained to preserve what future requests need, such as MemSuit (Lei et al., 2026), and one given PRAGMA-style distilled summaries. If these match or beat raw retrieval on use, the anchor's conclusion that raw evidence beats compact representation is about query-agnostic compressors only.

A second independent harness for recall. Re-run OmniMemEval's 14 systems under a protocol that is third-party for every system, with the grader varied and the answering model fixed, and report system-level rank agreement. If agreement is high, the variance in Figure 1a comes mostly from developer harnesses. If it is low, recall scores for memory systems are not yet stable enough to rank systems at all.

13. What is not known

It is not known whether, under one harness, the ranking of conversational memory systems on recall questions predicts their ranking on synthesis requests over the same histories. No study located here has measured both for the same systems.

It is not known how much of UtilMem's measured loss under distractors is regeneration variance rather than evidence loss. The anchor does not report the score of an independent regeneration from complete evidence.

It is not known how large the trigger gap is for synthesis tasks. Evidence on trigger comes from preference-following benchmarks with single preferences, not from multi-session synthesis.

It is not known how much of the 10 to 21 point spread in published LoCoMo scores comes from graders, question subsets, retrieval budgets, system versions or configuration expertise. No study has varied these one at a time for the same systems.

It is not known whether compression designed for future use, rather than for recall, closes the gap the anchor attributes to write-time compression. The two studies that point that way use annotated or observed queries (Yu et al., 2026; Lei et al., 2026), and neither was run on UtilMem.

14. Limitations

This analysis synthesises published experiments, so every design weakness in those experiments passes into it. The anchor was read as its first arXiv version, and a revision could supply the recall scores, the full-context condition or the regeneration control that Section 3 finds missing. Most cited memory-system and memory-benchmark papers are 2025 and 2026 preprints that have not been peer reviewed, several are by system developers evaluating their own systems, and the OmniMemEval numbers are taken from Wang (2026), who reports them from a results page maintained by the developer of one of the systems compared. The values in Figure 1 and Table 1 come from different metrics (LLM-judge J, judged accuracy, normalised robustness) and different graders, and rows should not be compared as if on one scale. The rank correlations are the author's arithmetic on three to five points each and carry no significance. The "harness" is treated as a single factor, though it has many components whose separate effects are unknown. The coverage judgements in Table 2 are the author's reading of benchmark designs, not properties their authors claim. Finally, the paper does not evaluate cost, latency or privacy, which are legitimate reasons to prefer compressed memory that do not depend on its use quality (Pollertlam and Kornsuwannawit, 2026; Pakhomov et al., 2025).

15. Conclusion

"Strong on factual recall, weak on utilization" is three claims. That access is necessary and not sufficient for use is true, measured, and older than the benchmark that restates it: LoCoMo, LongMemEval and the personalization benchmarks all show it, and its size grows with the scope of the task. That recall rank does not predict use rank is supported where write policies differ, because a compressor tuned to keep facts keeps what recall questions ask for and loses what synthesis needs, and it is otherwise untested; the clearest reversal in the published record sets developer-reported recall scores against a third-party use score. That recall benchmarks measure the wrong thing is not shown, because the recall scores themselves are not yet stable: the same system's published LoCoMo score moves by up to 21 points with the harness, and two harnesses with the same reader rank four systems with no rank agreement.

Between access and use sit four places to fail: write, compile, compose and trigger. Recall benchmarks see little of the last three. UtilMem sees write and compile clearly, holds compose at its reader's ceiling, and bypasses trigger by naming the evidence in its requests. The experiment that would decide how far recall predicts use is cheap: score both kinds of task over the same histories in one harness, with oracle conditions that locate each failure and a regeneration floor that bounds the noise. Algorithm 1 lists its steps in the order they should be run.

References

Bai, Y., Tu, S., Zhang, J., Peng, H., Wang, X., Lv, X., Cao, S., Xu, J., et al. (2024). LongBench v2: Towards Deeper Understanding and Reasoning on Realistic Long-context Multitasks. arXiv:2412.15204. doi:10.48550/arXiv.2412.15204.

Bean, A. M., Kearns, R. O., Romanou, A., Hafner, F. S., Mayne, H., Batzner, J., Foroutan, N., Schmitz, C., et al. (2025). Measuring what Matters: Construct Validity in Large Language Model Benchmarks. arXiv:2511.04703. doi:10.48550/arXiv.2511.04703.

Bian, H., Yao, Z., Hu, S., Xu, Z., Zhang, S., Guo, Y., Yang, Z., Han, X., Wang, H., & Chen, R. (2026). RealMem: Benchmarking LLMs in Real-World Memory-Driven Interaction. arXiv:2601.06966. doi:10.48550/arXiv.2601.06966.

Bowman, S. R., & Dahl, G. E. (2021). What Will it Take to Fix Benchmarking in Natural Language Understanding? arXiv:2104.02145. doi:10.48550/arXiv.2104.02145.

Chen, D., Niu, S., Li, K., Liu, P., Zheng, X., Tang, B., Li, X., Xiong, F., & Li, Z. (2025). HaluMem: Evaluating Hallucinations in Memory Systems of Agents. arXiv:2511.03506. doi:10.48550/arXiv.2511.03506.

Chen, J., Lin, H., Han, X., & Sun, L. (2024). Benchmarking Large Language Models in Retrieval-Augmented Generation. AAAI 2024. arXiv:2309.01431. doi:10.48550/arXiv.2309.01431.

Chhikara, P., Khant, D., Aryan, S., Singh, T., & Yadav, D. (2025). Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. arXiv:2504.19413. doi:10.48550/arXiv.2504.19413.

Cuconasu, F., Trappolini, G., Siciliano, F., Filice, S., Campagnano, C., Maarek, Y., Tonellotto, N., & Silvestri, F. (2024). The Power of Noise: Redefining Retrieval for RAG Systems. arXiv:2401.14887. doi:10.48550/arXiv.2401.14887.

Derehag, J., Calva, C., & Ghiurau, T. (2026). SmartSearch: How Ranking Beats Structure for Conversational Memory Retrieval. arXiv:2603.15599. doi:10.48550/arXiv.2603.15599.

Du, Y., Tian, M., Ronanki, S., Rongali, S., Bodapati, S., Galstyan, A., Wells, A., Schwartz, R., Huerta, E. A., & Peng, H. (2025). Context Length Alone Hurts LLM Performance Despite Perfect Retrieval. Findings of EMNLP 2025. arXiv:2510.05381. doi:10.48550/arXiv.2510.05381.

Fang, J., Deng, X., Xu, H., Jiang, Z., Tang, Y., Xu, Z., Deng, S., Yao, Y., et al. (2025). LightMem: Lightweight and Efficient Memory-Augmented Generation. ICLR 2026. arXiv:2510.18866. doi:10.48550/arXiv.2510.18866.

Goldman, O., Jacovi, A., Slobodkin, A., Maimon, A., Dagan, I., & Tsarfaty, R. (2024). Is It Really Long Context if All You Need Is Retrieval? Towards Genuinely Difficult Long Context NLP. EMNLP 2024. arXiv:2407.00402. doi:10.48550/arXiv.2407.00402.

Hsieh, C.-P., Sun, S., Kriman, S., Acharya, S., Rekesh, D., Jia, F., Zhang, Y., & Ginsburg, B. (2024). RULER: What's the Real Context Size of Your Long-Context Language Models? COLM 2024. arXiv:2404.06654. doi:10.48550/arXiv.2404.06654.

Hu, C., Gao, X., Zhou, Z., Xu, D., Bai, Y., Li, X., Zhang, H., Li, T., Zhang, C., Bing, L., & Deng, Y. (2026). EverMemOS: A Self-Organizing Memory Operating System for Structured Long-Horizon Reasoning. arXiv:2601.02163. doi:10.48550/arXiv.2601.02163.

Hu, Y., Wang, Y., & McAuley, J. (2025a). Evaluating Memory in LLM Agents via Incremental Multi-Turn Interactions. arXiv:2507.05257. doi:10.48550/arXiv.2507.05257.

Hu, Y., Liu, S., Yue, Y., Zhang, G., Liu, B., Zhu, F., Lin, J., Guo, H., et al. (2025b). Memory in the Age of AI Agents. arXiv:2512.13564. doi:10.48550/arXiv.2512.13564.

Jiang, B., Hao, Z., Cho, Y.-M., Li, B., Yuan, Y., Chen, S., Ungar, L., Taylor, C. J., & Roth, D. (2025). Know Me, Respond to Me: Benchmarking LLMs for Dynamic User Profiling and Personalized Responses at Scale. COLM 2025. arXiv:2504.14225. doi:10.48550/arXiv.2504.14225.

Joren, H., Zhang, J., Ferng, C.-S., Juan, D.-C., Taly, A., & Rashtchian, C. (2025). Sufficient Context: A New Lens on Retrieval Augmented Generation Systems. arXiv:2411.06037. doi:10.48550/arXiv.2411.06037.

Kang, J., Ji, M., Zhao, Z., & Bai, T. (2025). Memory OS of AI Agent. arXiv:2506.06326. doi:10.48550/arXiv.2506.06326.

Karpinska, M., Thai, K., Lo, K., Goyal, T., & Iyyer, M. (2024). One Thousand and One Pairs: A "novel" challenge for long-context language models. EMNLP 2024. arXiv:2406.16264. doi:10.48550/arXiv.2406.16264.

Kim, T. S., Lee, Y., Park, Y., Kim, J., Kim, Y.-H., & Kim, J. (2025). CUPID: Evaluating Personalized and Contextualized Alignment of LLMs from Interactions. COLM 2025. arXiv:2508.01674. doi:10.48550/arXiv.2508.01674.

Kuratov, Y., Bulatov, A., Anokhin, P., Rodkin, I., Sorokin, D., Sorokin, A., & Burtsev, M. (2024). BABILong: Testing the Limits of LLMs with Long Context Reasoning-in-a-Haystack. NeurIPS 2024 Datasets and Benchmarks Track. arXiv:2406.10149. doi:10.48550/arXiv.2406.10149.

Laban, P., Hayashi, H., Zhou, Y., & Neville, J. (2025). LLMs Get Lost In Multi-Turn Conversation. arXiv:2505.06120. doi:10.48550/arXiv.2505.06120.

Lei, Z., Shi, M., Fu, X., He, H., Wang, Q. R., & Li, J. (2026). Grounding Memory Summarization in Utility Intent. arXiv:2609.33417. doi:10.48550/arXiv.2609.33417.

Li, Z., Xi, C., Li, C., Chen, D., Chen, B., Song, S., Niu, S., Wang, H., et al. (2025a). MemOS: A Memory OS for AI System. arXiv:2507.03724. doi:10.48550/arXiv.2507.03724.

Li, X., Bantupalli, J., Dharmani, R., Zhang, Y., & Shang, J. (2025b). Toward Multi-Session Personalized Conversation: A Large-Scale Dataset and Hierarchical Tree Framework for Implicit Reasoning. arXiv:2503.07018. doi:10.48550/arXiv.2503.07018.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., & Liang, P. (2024). Lost in the Middle: How Language Models Use Long Contexts. Transactions of the Association for Computational Linguistics, 12. arXiv:2307.03172. doi:10.48550/arXiv.2307.03172.

Liu, Z., Zhong, J., Han, R., Wu, Z., Tang, S., He, C., Yang, Y., Qiu, Y., et al. (2026a). EvalMem: An Operation-Level Diagnostic Framework for Long-Term Memory Systems. arXiv:2609.22231. doi:10.48550/arXiv.2609.22231.

Liu, S., Zhu, J., Shu, L., Lin, J., Chen, Y., Zhang, H., Zhang, C., Xu, D., et al. (2026b). PERMA: Benchmarking Personalized Memory Agents via Event-Driven Preference and Realistic Task Environments. arXiv:2603.23231. doi:10.48550/arXiv.2603.23231.

Maharana, A., Lee, D.-H., Tulyakov, S., Bansal, M., Barbieri, F., & Fang, Y. (2024). Evaluating Very Long-Term Conversational Memory of LLM Agents. ACL 2024. arXiv:2402.17753. doi:10.48550/arXiv.2402.17753.

Mahendrakar, P. (2026a). Compressed Once, Read Many Times: Why Prompt-Compression Results Do Not Transfer to Agent Memory, What the Query-Agnostic Line Already Showed, and the Two Properties of a Memory Write No Protocol Yet Scores. Zenodo. doi:10.5281/zenodo.22969435.

Mahendrakar, P. (2026b). Sufficient for Whose Answer? Three Predicates Under One Word in Selective Retrieval-Augmented Generation, and Why the Only One Tied to Correctness Cannot Be Computed Where the Gate Runs. Zenodo. doi:10.5281/zenodo.22840565.

Modarressi, A., Deilamsalehy, H., Dernoncourt, F., Bui, T., Rossi, R. A., Yoon, S., & Schutze, H. (2025). NoLiMa: Long-Context Evaluation Beyond Literal Matching. ICML 2025. arXiv:2502.05167. doi:10.48550/arXiv.2502.05167.

Packer, C., Wooders, S., Lin, K., Fang, V., Patil, S. G., Stoica, I., & Gonzalez, J. E. (2023). MemGPT: Towards LLMs as Operating Systems. arXiv:2310.08560. doi:10.48550/arXiv.2310.08560.

Pakhomov, E., Nijkamp, E., & Xiong, C. (2025). Convomem Benchmark: Why Your First 150 Conversations Don't Need RAG. arXiv:2511.10523. doi:10.48550/arXiv.2511.10523.

Pan, Z., Wu, Q., Jiang, H., Luo, X., Cheng, H., Li, D., Yang, Y., Lin, C.-Y., et al. (2025). On Memory Construction and Retrieval for Personalized Conversational Agents. arXiv:2502.05589. doi:10.48550/arXiv.2502.05589.

Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative Agents: Interactive Simulacra of Human Behavior. arXiv:2304.03442. doi:10.48550/arXiv.2304.03442.

Pollertlam, N., & Kornsuwannawit, W. (2026). Beyond the Context Window: A Cost-Performance Analysis of Fact-Based Memory vs. Long-Context LLMs for Persistent Agents. arXiv:2603.04814. doi:10.48550/arXiv.2603.04814.

Qing, P., Shi, F., & Vosoughi, S. (2026). UTILMEM: Benchmarking Evidence Utilization in Long-Term Conversational Memory. arXiv:2608.30508. doi:10.48550/arXiv.2608.30508.

Raji, I. D., Bender, E. M., Paullada, A., Denton, E., & Hanna, A. (2021). AI and the Everything in the Whole Wide World Benchmark. arXiv:2111.15366. doi:10.48550/arXiv.2111.15366.

Rasmussen, P., Paliychuk, P., Beauvais, T., Ryan, J., & Chalef, D. (2025). Zep: A Temporal Knowledge Graph Architecture for Agent Memory. arXiv:2501.13956. doi:10.48550/arXiv.2501.13956.

Salemi, A., & Zamani, H. (2024). Evaluating Retrieval Quality in Retrieval-Augmented Generation. arXiv:2404.13781. doi:10.48550/arXiv.2404.13781.

Shi, F., Chen, X., Misra, K., Scales, N., Dohan, D., Chi, E., Scharli, N., & Zhou, D. (2023). Large Language Models Can Be Easily Distracted by Irrelevant Context. ICML 2023. arXiv:2302.00093. doi:10.48550/arXiv.2302.00093.

Tan, H., Zhang, Z., Ma, C., Chen, X., Dai, Q., & Dong, Z. (2025). MemBench: Towards More Comprehensive Evaluation on the Memory of LLM-based Agents. Findings of ACL 2025. arXiv:2506.21605. doi:10.48550/arXiv.2506.21605.

Tavakoli, M., Salemi, A., Ye, C., Abdalla, M., Zamani, H., & Mitchell, J. R. (2025). Beyond a Million Tokens: Benchmarking and Enhancing Long-Term Memory in LLMs. arXiv:2510.27246. doi:10.48550/arXiv.2510.27246.

Uddin, M. N., Shubham, K., Blanco, E., Baral, C., & Wang, G. (2026). From Recall to Forgetting: Benchmarking Long-Term Memory for Personalized Agents. arXiv:2604.20006. doi:10.48550/arXiv.2604.20006.

Vodrahalli, K., Ontanon, S., Tripuraneni, N., Xu, K., Jain, S., Shivanna, R., Hui, J., Dikkala, N., et al. (2024). Michelangelo: Long Context Evaluations Beyond Haystacks via Latent Structure Queries. arXiv:2409.12640. doi:10.48550/arXiv.2409.12640.

Wang, G. (2026). Mnemon: Raw Records, Fast Judgments, Slow Thoughts. arXiv:2609.36059. doi:10.48550/arXiv.2609.36059.

Wang, Q., Fu, Y., Cao, Y., Wang, S., Tian, Z., & Ding, L. (2023). Recursively Summarizing Enables Long-Term Dialogue Memory in Large Language Models. arXiv:2308.15022. doi:10.48550/arXiv.2308.15022.

Wu, D., Wang, H., Yu, W., Zhang, Y., Chang, K.-W., & Yu, D. (2025). LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. ICLR 2025. arXiv:2410.10813. doi:10.48550/arXiv.2410.10813.

Xu, J., Szlam, A., & Weston, J. (2021). Beyond Goldfish Memory: Long-Term Open-Domain Conversation. arXiv:2107.07567. doi:10.48550/arXiv.2107.07567.

Xu, W., Liang, Z., Mei, K., Gao, H., Tan, J., & Zhang, Y. (2025). A-MEM: Agentic Memory for LLM Agents. NeurIPS 2025. arXiv:2502.12110. doi:10.48550/arXiv.2502.12110.

Yen, H., Gao, T., Hou, M., Ding, K., Fleischer, D., Izsak, P., Wasserblat, M., & Chen, D. (2025). HELMET: How to Evaluate Long-Context Language Models Effectively and Thoroughly. ICLR 2025. arXiv:2410.02694. doi:10.48550/arXiv.2410.02694.

Yu, H., Koh, H., Kim, M., Jang, Y., & Jung, K. (2026). PRAGMA: Evaluating Personalized Guidance with Memory Alignment in Lifelong Conversations. arXiv:2609.09664. doi:10.48550/arXiv.2609.09664.

Yu, W., Zhang, H., Pan, X., Ma, K., Wang, H., & Yu, D. (2023). Chain-of-Note: Enhancing Robustness in Retrieval-Augmented Language Models. arXiv:2311.09210. doi:10.48550/arXiv.2311.09210.

Zhang, Z., Bo, X., Ma, C., Li, R., Chen, X., Dai, Q., Zhu, J., Dong, Z., et al. (2024). A Survey on the Memory Mechanism of Large Language Model based Agents. arXiv:2404.13501. doi:10.48550/arXiv.2404.13501.

Zhao, S., Hong, M., Liu, Y., Hazarika, D., & Lin, K. (2025). Do LLMs Recognize Your Preferences? Evaluating Personalized Preference Following in LLMs. ICLR 2025. arXiv:2502.09597. doi:10.48550/arXiv.2502.09597.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., et al. (2023). Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. arXiv:2306.05685. doi:10.48550/arXiv.2306.05685.

Zhong, W., Guo, L., Gao, Q., Ye, H., & Wang, Y. (2023). MemoryBank: Enhancing Large Language Models with Long-Term Memory. arXiv:2305.10250. doi:10.48550/arXiv.2305.10250.

© 2026 Pranay Mahendrakar. Licensed under CC BY 4.0.

DOI: 10.5281/zenodo.23899944