

Evaluating LoreWeaver’s Story Memory on LoCoMo

The LoreWeaver team

October 18, 2026 · loreweaverai.com/research/locomo-memory-benchmark

Abstract

LoreWeaver is an interactive fiction platform in which a language model tells long stories with a player. Long stories outgrow any model’s context window, so the platform keeps a long-term memory: after every four storyteller replies it writes short, self-contained memories from the new turns, and before each reply it ranks them against the moment by embedding similarity, text and keyword match, the characters involved, and importance. We evaluate this system on LoCoMo, a benchmark of ten long multi-session conversations with 1,540 scored questions. To make the result comparable with published numbers, we adopt the protocol of Mem0’s public benchmark repository without changes: its answer and judge prompts, GPT-5 as answerer and judge, and the 200 top-ranked memories per question.

LoreWeaver answers 1,387 of 1,540 questions correctly (90.1%, 95% CI 88.6 to 91.5). Mem0’s published per-question results for its managed platform score 91.6% under the same protocol, a paired difference that is not significant (McNemar $p = 0.069$). Those results merge a second pass over 156 questions; against the first pass they appear to replace, the two systems are level (90.1% and 89.9%). On a 598-question subset, LoreWeaver’s production configuration of 40 memories plus a running recap matches its 200-memory score. An audit of the 153 errors attributes about a third to questionable gold answers, a quarter each to answering mistakes and to details the memories never recorded, and an eighth to relative dates the memories left unresolved.

1 Introduction

A language model reads only what is in its prompt. In interactive fiction the prompt cannot hold the whole story for long: a campaign of a few hundred exchanges runs to tens of thousands of words, and models use material in the middle of a long input less reliably than material at either end [5]. A storyteller that is to remember a promise from the first session, or the name of a minor character met weeks earlier, needs a memory system that records what happened and brings back the right part for each reply.

Memory layers for assistants address the same problem. MemGPT pages information between a model’s context and external storage [6], Mem0 extracts and updates facts from conversations [2], and Zep keeps them in a temporal knowledge graph [7]. Stories add requirements of their own. Memories must follow the story’s calendar rather than the real one, keep a fictional cast’s names stable, and disappear when a player rewinds the scenes they came from.

LoreWeaver’s memory system was built for stories, and until now we had measured it only on them. Here we test it on LoCoMo [1], the benchmark most often reported for long-term conversational memory. LoCoMo’s conversations are between two personas rather than a player and a storyteller, but their shape is close to a long story: many sessions over several months, events mentioned once in passing, and questions that need single facts, facts combined across sessions, dates, and inferences from what was said. LongMemEval [4] tests assistants on longer chat histories; we leave it for future work.

LoCoMo scores are hard to compare across papers. Systems differ in the model that answers from memories, the model that judges the answers, the judge’s instructions, the number of memories retrieved per question, and which question categories count. The original Mem0 paper reported judge scores between 66% and 69% with GPT-4o-mini answering [2], while Mem0’s current repository reports 92.5% with GPT-5 and a more lenient judge [3]. These numbers do not measure the same thing. We therefore made no protocol choices of our own: we reproduced Mem0’s published protocol around LoreWeaver, changed only the memory system, and compared against Mem0’s own per-question results.

Our contributions are:

- A same-protocol evaluation of a production story-memory system on LoCoMo, with results by category and by conversation and paired tests against Mem0’s published run (Section 4).
- An analysis of Mem0’s published results file, which merges a second pass over 156 questions, and of what that means for comparisons (Section 4.4).
- A comparison at production settings, and an audit of every error with a hand-checked sample (Sections 4.5 and 5).

2 The memory system

LoreWeaver gives its storyteller a story in layers: the most recent turns word for word, memories, a running recap, lore cards written by the world’s author, and, in some worlds, game state such as quests and inventory. This report concerns memories; Section 4.5 measures the recap separately. The benchmark called the same code that serves players, and the only change was the number of memories recall returns.

2.1 Writing memories

After every fourth storyteller reply, a capture step reads the new stretch of the story, the player’s turns and the storyteller’s, and writes a handful of memories with a language model. Along with the new turns the writer receives the story’s four latest memories, so that time and place carry forward, and up to 40 names already used in the story, most frequent first, so that spellings stay consistent.

Each memory is one or two sentences that stand on their own: they name everyone involved and say who did or said what to whom. A memory also records its kind (fact, event, plan, relationship, revelation or state), an importance rating, the characters involved, the story time and place when the text states them, and the range of turns it was written from. The writer copies dates and times as the story gives them and resolves relative expressions against the story’s own clock. It is told never to use the real-world calendar, and to leave the time empty when the story gives none. Each memory is embedded from its text together with its characters, time and place, as a 1,536-dimensional vector. Table 1 shows a memory from the benchmark.

Table 1: A memory LoreWeaver wrote from LoCoMo conversation 1, shortened. Its story time stays relative because the conversation never states a date (Section 5).

Field	Value
Text	Melanie told Caroline about her family camping trip in the mountains the week before, describing it as an awesome time where they explored nature, roasted marshmallows around the campfire, and went on a hike with an amazing view from the top.
Kind	Event
Story time	The week before this conversation
Place	The mountains
Characters	Melanie, Caroline
Source	Storyteller replies 33 to 36

Memories are tied to the turns they came from. A memory is saved only if its source turns still exist unchanged when writing finishes, and rewinding or editing the story removes the memories written from the turns it removes. These rules matter in a product where players rewind often. They play no part in this benchmark.

2.2 Recalling memories

Before each reply, recall fills the memory part of the prompt in three steps. Memories the player pinned come first. Next come the three memories written just before the oldest turn still visible in the prompt, so the story keeps its thread between the turns the storyteller can read and the memories it cannot. The rest of the space goes to the highest-scoring remaining memories, scored against the moment with five signals (Table 2).

Table 2: Signals in LoreWeaver’s memory ranking. A memory’s score is the weighted sum.

Signal	Weight	What it measures
Embedding similarity	0.50	Cosine similarity between the query’s embedding and the memory’s
Text match	0.30	PostgreSQL’s cover-density rank of the memory text against any of the query’s terms, tripled and capped at 1
Keyword overlap	0.15	Share of the query’s keywords found among the memory’s keywords
Characters	0.12	1 when the memory involves a character named in the query or the recent story
Importance	0.06	The memory’s importance rating, scaled to between 0 and 1

The query is the player’s new turn, prefixed with the proper nouns of the recent story; when the player asks the storyteller to continue without writing, it is the recent story itself. Scoring is exhaustive within a story, so no approximate index is involved. In production the 40 highest-scoring memories are candidates, and the chosen memories reach the storyteller in story order with their time and place. In the benchmark nothing is pinned and no turns are visible, so recall reduces to the ranked step, with the question as the query.

2.3 The recap

Alongside memories, LoreWeaver keeps a running recap of each story: the events so far in order, with recent ones in more detail, and the current state of things, including the time and place, relationships,

and open plans and promises. Each capture cycle folds the new turns into it, and the storyteller reads it on every reply. Mem0’s protocol gives the answerer retrieved memories only, so the headline run leaves the recap out.

3 Experimental setup

3.1 Data

We use LoCoMo’s released file [1]: ten conversations of 19 to 32 sessions each, 272 sessions and 5,882 turns in all, with 1,986 questions. As in [3], we score categories 1 to 4 and leave out category 5, whose adversarial questions ask about things never said. That leaves 1,540 questions, which we name as Mem0’s harness does: 841 single-hop, 282 multi-hop, 321 temporal and 96 open-domain.

3.2 Building the memories

Each conversation became one LoreWeaver story, on an account on the Legendary plan, which keeps every memory a story writes. Speaker A took the player’s part and speaker B the storyteller’s; Mem0’s harness likewise gives speaker A the user’s role. Turns were formatted as Mem0’s harness formats them, as “Name: text” followed by the image’s caption and search query in brackets when a turn shares a photo, and each turn was dated with its session’s date. Capture then ran over each story through the production code path until it had read every turn, which took between three and seven minutes per conversation.

LoCoMo’s session dates are metadata and never appear in the turns. LoreWeaver’s memory writer reads only the turns, so it never saw them. Mem0’s harness passes each session’s date to Mem0 when it adds that session’s messages.

3.3 Answering and judging

For each question we called recall with the question as the player’s turn and speaker A as the player, and took the 200 top-ranked memories. As in [3], the memories were sorted by date and shown to the answerer as “(Weekday, Month DD, YYYY) memory”, the date being that of the session the memory’s last source turn belongs to. Where a memory had a story time or place, we appended it in parentheses.

The answer and judge prompts are Mem0’s, used verbatim, with the last session’s date as the reference date. GPT-5 answered and judged at default sampling settings with a 4,096-token output limit, and the answer is the text after “ANSWER:”. The judge returns JSON with a CORRECT or WRONG label [10]. It is lenient by design: an answer is correct if it contains at least one item of a list answer, paraphrases count, extra detail is never penalized, and dates within 14 days and durations within 50% match. For open-domain questions the gold answer is the text before the first semicolon, as in [3].

3.4 Retries

Mem0’s client retries a missing completion up to five times; ours retried up to three times within the run. Seven questions were still blank afterwards and were run once more with the same settings, and four received answers. The other three again produced no text within the 4,096-token limit, which GPT-5 shares between its reasoning and its answer, and were run with a 16,000-token limit. No question was re-run because its answer was wrong. If the three long-limit answers are counted as wrong, the score is 1,386 (90.0%). If all seven questions that first came back blank are, it is 1,383 (89.8%).

3.5 Comparison and statistics

Mem0 publishes per-question results for its managed platform in its repository [3]: a run dated April 6, 2026, with GPT-5 answering and judging through Azure and 200 memories per question. We paired its labels with ours by question ID. Confidence intervals come from 10,000 bootstrap resamples of the 1,540 questions [9], and differences are tested with exact two-sided McNemar tests on the paired labels [8]. Answering and judging cost \$44.03 at OpenRouter’s GPT-5 prices.

4 Results

4.1 Overall accuracy

Table 3: Accuracy on LoCoMo categories 1 to 4 (1,540 questions) under Mem0’s protocol.

Run	Correct	Accuracy	95% CI
LoreWeaver	1,387	90.1%	88.6 to 91.5
Mem0, published per-question file	1,410	91.6%	90.1 to 92.9
Mem0, repository README	1,425	92.5%	Not reported
Mem0, first pass (inferred, Section 4.4)	1,384	89.9%	Not available

LoreWeaver scores 1.5 points below Mem0’s published file (95% CI -3.1 to $+0.1$). The two systems agree on 1,393 questions: 1,325 that both answer correctly and 68 that neither does. Of the 147 on which they disagree, LoreWeaver alone is correct on 62 and Mem0 alone on 85 (McNemar $p = 0.069$). The README’s headline of 92.5% is higher than the file it accompanies, which scores 91.6% with one question marked as an error. We compare against the file because it allows paired tests.

4.2 By category

Table 4: Accuracy by question category (%). The first-pass column assumes Mem0’s second pass covered exactly the questions its first pass missed.

Category	Questions	LoreWeaver	Mem0, file	Mem0, first pass
Single-hop	841	90.2	92.3	90.2
Multi-hop	282	91.8	93.3	91.8
Temporal	321	92.5	92.8	91.9
Open-domain	96	75.0	76.0	74.0

Open-domain questions are the hardest for both systems. They ask for inferences from what was said, such as whether a speaker would be considered religious, and their gold answers are the least consistent in the dataset (Section 5). Temporal questions end up close to level even though LoreWeaver’s memories often kept relative dates, which suggests the answerer recovered most dates from the session date attached to each memory.

4.3 By conversation

Accuracy per conversation ranges from 83.1% to 96.3% for LoreWeaver and from 86.9% to 96.1% for Mem0 (Figure 1). LoreWeaver is ahead on four conversations and behind on six, and the spread between conversations is several times the gap between the systems.

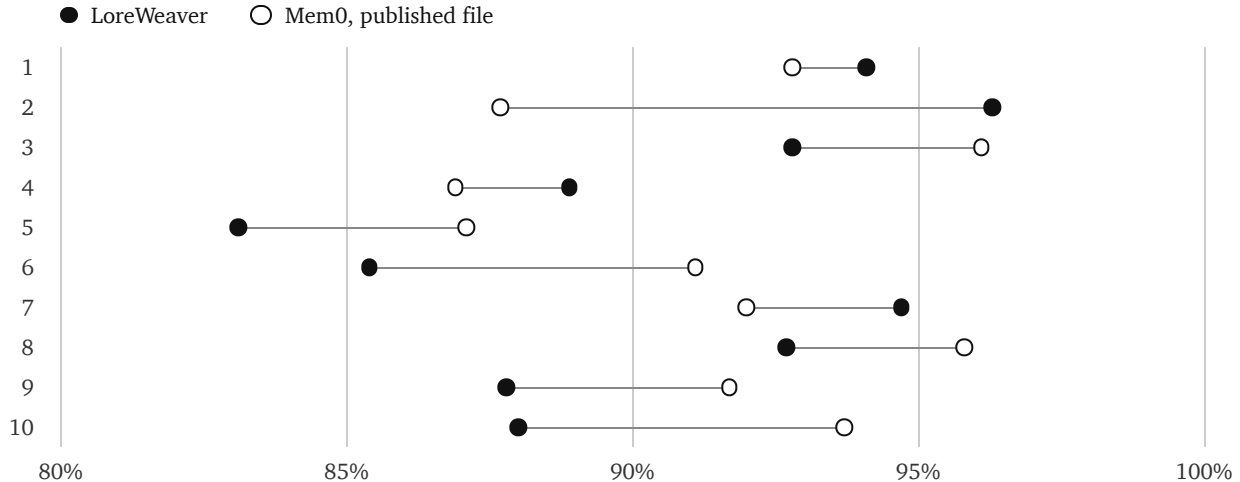


Figure 1: Accuracy per conversation, in file order, for LoreWeaver and for Mem0’s published file. Conversations have between 81 and 199 scored questions.

4.4 Mem0’s second pass

The metadata of Mem0’s file lists 156 question IDs under `merged_from_questions`. Every one of the 1,384 questions outside that list is labeled correct, and 26 of the 156 inside it are. The likely reading is that a second run over the questions the first run got wrong was merged into the file. On that reading Mem0’s first pass scored 1,384 of 1,540 (89.9%), and LoreWeaver’s single pass is level with it: LoreWeaver alone is correct on 79 questions and the first pass alone on 76 ($p = 0.87$). If the second run also covered questions the first got right, the first pass scored higher. The file does not say.

Re-running failures is a reasonable way to estimate how well a system can do when both the answerer and the judge sample. It does mean that the published file should be compared with a run treated the same way. We did not re-run LoreWeaver’s wrong answers.

4.5 Production settings

The protocol run gives the answerer 200 memories and no recap. In production LoreWeaver takes 40 ranked memories and adds the recap. Before switching to the protocol’s settings we had answered 598 questions with the production configuration, and Table 5 compares the runs on those questions. Of them, 574 come from the first four conversations, so they are not a random sample.

Table 5: Accuracy on the 598 questions answered in both LoreWeaver configurations.

Configuration	Memories	Recap	Accuracy
LoreWeaver, production	40	Included	92.0%
LoreWeaver, Mem0’s protocol	200	Left out	92.0%
Mem0, published file	200	None	91.0%

With a fifth of the memories and the recap, which averaged about 14,000 characters on these stories, LoreWeaver scores the same. For the product this is the more relevant result, because production recall has to choose 40 memories rather than return most of them (Section 6).

4.6 Retrieval coverage

LoCoMo marks the turns that support each answer. For 1,532 of the 1,540 questions it lists at least one we could locate, and for 99.8% of those at least one retrieved memory was written from a stretch of the story containing such a turn. The same holds for 151 of the 153 errors. The measure is coarse: it shows that recall reached the right part of the story, not that the needed fact was in the memory.

5 Error analysis

We assigned each of the 153 errors one main cause. A model, Claude Sonnet 5.5, read the question, the gold and generated answers, LoCoMo’s evidence turns with their dates, and every memory written from the stretch of the story containing them, and chose one of the five causes in Table 6. We then read the evidence for 30 of the 47 errors it attributed to the gold answer, and agreed with 27.

Table 6: Main cause of each of the 153 errors.

Cause	Errors	Share	Meaning
Questionable gold answer	47	31%	The evidence does not support the gold answer, the question names the wrong speaker, or the generated answer is defensible
Answering mistake	39	25%	The memories held what was needed, and the answer chose the wrong item, miscounted or reasoned wrongly
Detail not recorded	38	25%	No memory from the evidence turns contains the needed fact
Relative time	20	13%	The memory kept a relative time such as “last Sunday”, and the answer’s date was off
Partly recorded	9	6%	The memories held part of a multi-part answer

Questionable gold answers. Several questions attribute a statement to the wrong speaker. “What did Tim say about his injury on 16 November, 2023?” has the gold answer “The doctor said it’s not too serious”, but the evidence turn is John’s, and LoreWeaver answered that the injury was John’s. Five errors in conversation 5 alone are of this kind. Open-domain gold answers are often one guess among several plausible ones, such as “Hatha Yoga” for a style of yoga that builds core strength, or a particular national park the speakers never name. These problems affect every system on LoCoMo equally. We adjust no score for them, and they put the attainable score below 100%.

Details not recorded. LoreWeaver wrote 0.44 memories per turn, and the details it missed are mostly passing remarks: what a speaker said sunflowers represent, or a comparison of a business partnership to dancing together. Capture favors events, plans and facts about people over passing sentiment. That suits stories and costs points on questions about a single phrase. Of these 38 errors, 29 are single-hop questions.

Relative time. LoreWeaver’s memory writer is told never to use the real-world calendar, because in fiction the calendar belongs to the story. LoCoMo’s dates exist only as session metadata, so a remark like “last Sunday” stayed relative in the memory, and only 15% of memories carry a story time at all. The answerer sees each memory’s session date and has to work out the date itself. In 20 cases, 16 of them temporal questions, it got the date wrong. Giving the writer the session date when a story uses a real calendar would address most of these.

Answering mistakes. Most are counts, such as “How many times has Melanie gone to the beach in 2023?” answered 1 where the memories show 2, and choices between similar items, such as a castle shadow box given in place of a painting of a castle on a hill. With 200 memories in view, the answerer has many near-duplicates to choose between.

6 Discussion

What top-200 retrieval measures. Each LoreWeaver story held between 155 and 319 memories, 258 on average. Retrieving 200 per question therefore returned 76% of a story’s memories on average, and all of them for the two shortest conversations. At that depth the benchmark mostly measures what the memories contain and how well the answerer reads them, and ranking matters little. Production recall chooses 40 candidates, where ranking does matter, and the production configuration scored as well as the protocol run (Section 4.5). We consider that the more informative result for LoreWeaver, with the caveat that it rests on a smaller, non-random set of questions.

How much memories compress a story. The ten stories produced 2,584 memories averaging 30 words: about 102,000 tokens of memory for about 217,000 tokens of conversation. Half are facts and a quarter are events, and plans, relationships, revelations and states make up the rest. The writer is told to keep names, roles and exact wording because a storyteller needs them, so it compresses lightly. A memory layer that only has to answer questions could compress further.

Against Mem0. Under one protocol, with the same answerer and judge, a memory system built for interactive fiction scores within noise of a dedicated memory platform on conversational recall. We read this as evidence that both systems record and find much the same information. It does not rank them; that would take repeated runs of both systems and a cleaner dataset.

7 Limitations

- **One run.** GPT-5 samples at its default temperature, so answering and judging both vary between runs, and we did not measure by how much. Mem0’s file combines two runs.
- **A reported comparison.** We did not run Mem0. Its numbers come from its repository, and its platform may have changed since April 2026.
- **A lenient judge.** Scores under this protocol are not comparable with token F1 scores, or with stricter judges such as those of the original LoCoMo and Mem0 papers.
- **The dataset.** LoCoMo has ten conversations, and they are chat between two personas rather than fiction. Its gold answers contain errors (Section 5).
- **Dates.** Our memory writer did not receive session dates, which Mem0’s harness passes to Mem0. This costs LoreWeaver on temporal questions.
- **The production comparison** rests on 598 questions drawn mostly from four conversations.
- **The error audit** is a model’s judgment, checked by hand for a sample of one cause.

8 Conclusion

Run inside Mem0’s published LoCoMo protocol, LoreWeaver’s story memory answers 90.1% of 1,540 questions. That is 1.5 points below Mem0’s published platform results, a difference that is not signifi-

cant, and level with the first pass those results appear to have been built on. At production settings, with a fifth of the memories and a recap, it scores the same on the questions we ran both ways. The clearest room for improvement is in the memories themselves: recording passing details, and resolving relative times when a story uses a real calendar.

Appendix A. Settings

Setting	Value
Dataset	LoCoMo, locomo10.json, categories 1 to 4, 1,540 questions
Prompts	Mem0 memory-benchmarks, LoCoMo answer and judge prompts, verbatim
Answerer	GPT-5, default sampling, 4,096 output tokens (16,000 for 3 questions)
Judge	GPT-5, JSON output
Memories per question	200 top-ranked (197 on average)
Recap	Left out
Plan	Legendary, which keeps every memory
Capture	Every four storyteller replies, production code path
Search	Exhaustive within each story
Run	October 17 and 18, 2026

References

- [1] A. Maharana, D.-H. Lee, S. Tulyakov, M. Bansal, F. Barbieri, and Y. Fang. Evaluating Very Long-Term Conversational Memory of LLM Agents. ACL 2024. arXiv:2402.17753.
- [2] P. Chhikara, D. Khant, S. Aryan, T. Singh, and D. Yadav. Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. arXiv:2504.19413, 2025.
- [3] Mem0. memory-benchmarks. <https://github.com/mem0ai/memory-benchmarks>. Platform results dated April 6, 2026; accessed October 2026.
- [4] D. Wu, H. Wang, W. Yu, Y. Zhang, K.-W. Chang, and D. Yu. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. ICLR 2025. arXiv:2410.10813.
- [5] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang. Lost in the Middle: How Language Models Use Long Contexts. TACL, 2024. arXiv:2307.03172.
- [6] C. Packer, S. Wooders, K. Lin, V. Fang, S. G. Patil, I. Stoica, and J. E. Gonzalez. MemGPT: Towards LLMs as Operating Systems. arXiv:2310.08560, 2023.
- [7] P. Rasmussen, P. Paliychuk, T. Beauvais, J. Ryan, and D. Chalef. Zep: A Temporal Knowledge Graph Architecture for Agent Memory. arXiv:2501.13956, 2025.
- [8] Q. McNemar. Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages. Psychometrika 12(2):153–157, 1947.
- [9] B. Efron and R. J. Tibshirani. An Introduction to the Bootstrap. Chapman & Hall, 1993.
- [10] L. Zheng, W.-L. Chiang, Y. Sheng, et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. NeurIPS 2023 Datasets and Benchmarks. arXiv:2306.05685.