MemMooring: Keeping the History and Cutting the State in Conversational Memory

Anonymous authors
Paper under double-blind review

Abstract

A sentence about a life already carries the shape of the world it is about: the people and places, what changed, in what order, and what has not yet happened. Language is a structured abstraction of the world and of its changes, so a memory built from conversation can recover what was talked about rather than the words used. We present MemMooring, which rebuilds a conversation as a world representation that grows over time and is corrected in place across sessions rather than overwritten. Events retain world time, recording time and their source utterances; the states they change are kept as versions carrying intervals of validity and of recording. The evaluated stores are extracted with Ollama qwen3.6:latest (36.0B MoE, GGUF Q4_K_M, 262,144-token context); answering uses MiniMax-Text-01 at temperature 0, with DeepSeek-chat and GPT-4o as separate judges. The representation is complete enough to be read in two directions: a horizontal cut returns everything about one person at one moment – the facts then in force, the events under way or still consequential, and the plans already known whose time still lay ahead – and a longitudinal cut follows one matter from the decision that opened it through its recorded steps to where it stands. We evaluate the implemented storage, retrieval and answering paths on ten LoCoMo conversations. Under that protocol MemMooring reaches 86.6% and 83.1% against 85.5% and 82.2% for current Mem0. On the 181 questions that name a calendar date, adding a channel that orders events by the distance between their stored interval and the named day raises accuracy from 77.3% to 83.4% and from 75.7% to 81.8% at an unchanged limit of thirty records, and raises strict evidence coverage from 78.3% to 88.9%; on the 1356 questions that name no date the two retrievals return the same 81.4% coverage. The two cuts are shown as transcript-linked worked examples over one conversation; their set and process accuracy are not annotated and are not claimed.

Keywords. long-term memory, conversational agents, bitemporal data, event memory, episodic memory, provenance, state reconstruction, retrieval-augmented generation, LoCoMo, versioned stores

1 What a Memory Must Keep and What It Must Give Back

Conversation conveys information about entities, their states, and the events through which those states change over time. We investigate how conversational memory can reconstruct and maintain a structured representation of this evolving world from successive utterances. Such a representation should support both reconstructing a person's situation at a given time using only information available then, and tracing a continuing matter from its beginning through its development to its end or current state.

Existing conversational memories stop short of this on one side or the other. Snippet memories extract statements and retrieve them by similarity Chhikara (2025): the details survive, but what is returned is a set ranked for the question being asked now, carrying neither the span over which each statement held nor the order in which they were learned.

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

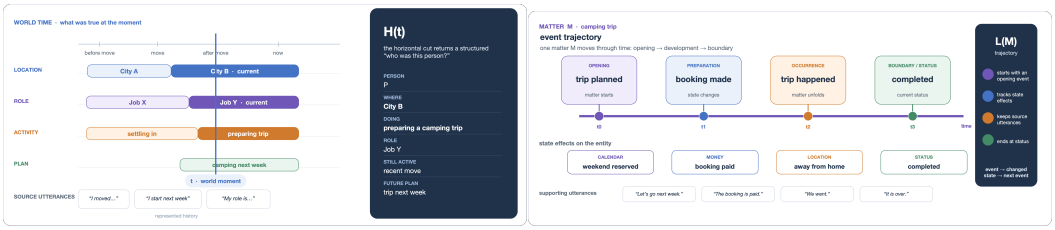

Figure 1: Two reads of one event–state store. The left panel is the horizontal world-time cut $H(t)$: it reconstructs the states, roles, activities and plans in force at a queried moment. The right panel is the longitudinal matter read $L(M)$: it follows one matter from opening through development to its current status, retaining the source utterances that license each step. Both reads are drawn from the same event–state store, and both are cut on the recording axis first: nothing learned after the moment of knowing may appear in either, which is what allows the upper panel to return a plan whose time still lies ahead. The panels are schematic; the measured worked examples are in Section 4.5.

Mutable-state memories keep a compact current state and discard much of the trace Badhe (2026): the prompt stays bounded, and a question about an earlier state depends on history the store no longer holds. Bitemporal graph and fact systems establish that valid time and transaction time together recover past belief Wang (2026); Zhang (2026), and we take that machinery as given rather than as a contribution. What remains open is building such a structure out of ordinary conversation, where dates are spoken as phrases and later sessions revise earlier ones, and then reading the one structure in both of the directions above.

We present MemMooring, which organizes conversation into entities, events, and states. Entities represent the people, places, and things discussed. Events record what happened, the entities involved, and the interval in which it occurred, while retaining the source utterances. Each state is attached to an entity, records an event's effect on that entity and the interval over which the effect holds, and links to the event that produced it. As subsequent sessions supplement or revise earlier accounts, the system preserves historical state versions. Separate recording intervals specify when each version was accepted by the system, supporting reconstruction of past situations using the information available at the time. Recall over this history is layered by a fact's grain: the state table is always present, events whose grain still covers the current moment remain shallow, and older events are reached through deep recall when triggered.

Both reads are drawn from the stored intervals when the question arrives rather than kept as snapshots, so either can be taken at a moment where nothing happened to be recorded, and the reconstruction of a past situation admits only records whose recording interval had already opened. We present each read as a worked example checked against the transcript rather than as an accuracy claim.

The two reads differ from retrieval in what selects the material. A retrieval is selected by the question being asked now, so what comes back is a sample of the record bearing on that question; a reconstruction is selected by the moment being asked about, so what comes back is what was there at that moment. Only the second reproduces the information a person actually had, which is what a claim about what they believed or why they acted has to rest on.

On the ten LoCoMo conversations Maharana (2024), over all 1540 questions that remain once the adversarial category is excluded, MemMooring answers 86.6% and 83.1% correctly under two judges, against 85.5% and 82.2% for the current Mem0 implementation used in this study under an identical answering model, answer prompt, judge prompt and date rule, and 12.5% and 15.8% when the same model answers with no memory at all. On the 181 questions that name a calendar date, adding the interval channel at the same thirty-record limit raises accuracy from 77.3% to 83.4% and from 75.7% to 81.8% under the two judges; on the 1356 questions that name no date the retrievals return the same 81.4% evidence coverage. Keeping history is affordable: an indexed as-of lookup over one million state versions takes 0.44 ms against 26.1 ms for a linear scan of the same records held as JSON. The two reads are shown as worked examples checked against the transcript; their set and process accuracy

are not annotated and are not claimed. Section 2 defines the representation, Section 3 its read operations, Section 4 reports the measurements and separates them from the protocol that remains, and Sections 5 and 6 give comparisons and limits.

2 Building the Memory

This section fixes how the memory is built and holds recall shut until Section 3. It gives what a write produces, the time a record carries and where that time comes from, what every derived fact points back at, the contract under which a language model is allowed to write, how a later write updates what is already there, and how records that belong to the same thing are brought together.

2.1 What a Write Leaves Behind

A representation that is corrected in place across many sessions has to be made of records a later session can add to without rewriting what is already there. That requirement fixes both what a record is and what a write is permitted to do to one. Five kinds of node carry the representation. An entity is an identity anchor – a person, a pet, a place, an object; its label is for humans, identity is the node, so two people called Jon are two nodes and a renamed entity is the same node. A kind classifies entities and forms a shallow hierarchy. An event is something that happened: a one-sentence description, a time interval, the original utterance, a pointer to the turns it rests on, and edges to its participants, to the things and places it involves, and to the container it hangs in. A state is one version of one fact about one entity: a slot, a value, the two intervals defined next, and edges to its subject, to an optional object, to the event that produced it, and to the version that closed it. A scope is a container – a conversation, a session, a day, or the grouping of smaller events inside an undertaking (Section 3.2).

Time is carried on two axes that must not be collapsed. An event carries time: when the thing happened, at the granularity the words justified. A state carries valid, the window over which the fact holds in the world, and recorded, the window over which this memory believed it – opened when the fact is written, closed when a later version supersedes it. An open right endpoint on either axis means 'still, as far as this memory knows'. Keeping the two apart is what makes a past moment answerable: a question about January is answered by filtering recorded at January, which removes not merely the records made later but the revisions later records imposed.

A conversational date is almost never spoken. What a speaker says is 'last week', 'the Friday before', 'next month', and a date is a computation from that phrase and the day the phrase was said. The store therefore keeps three things for an event timed this way: the phrase verbatim, the day it was uttered, and the interval computed from the two. The first two are evidence and are never rewritten; the third is derived, and can be recomputed at any time from the first two by a reader who disagrees with the computation.

2.2 The Write Contract and Its Evidence

Extraction is the weakest link in every memory built from text, so the contract fixes what a language model is allowed to produce. The evaluated stores use Ollama qwen3.6:latest (36.0B MoE, GGUF Q4_K_M, 262,144-token context) as the extraction model. The model reads a small window of the conversation together with what is already known, and emits events only: a sentence saying what happened, who took part, what it involved, the words that place it in time, and the lasting changes it caused as (subject, slot, value) triples, with an object where the slot is object-shaped. It does not create entities, does not decide which version to close, does not resolve a phrase into a date, and does not name containers. Each of those is a choice among candidates that already exist, or an arithmetic, and code does it. One departure from the contract holds for the stores evaluated here: they were built by a pipeline that still resolved time phrases inside the extraction call, and the separate deterministic resolver was written afterwards. The intervals in those stores are therefore the extractor's arithmetic, not the resolver's.

Every event carries a pointer to the conversation and message it came from, and optionally the quoted span. Every state points at the event that produced it. A derived fact is therefore always two hops from the words that licensed it, and a system that surfaces a fact can surface its origin at no extra cost.

2.3 Update and Reconciliation

A write preserves every existing version's value and its window of validity; only the recording window may be closed. Asserting a value on a timeline – keyed by subject, by slot, and where the slot is object-shaped by the object – closes the recording window of the version covering that moment, appends a copy of it bounded at that moment, links the two by a closing edge, and splits around any later version that already exists. Asserting the same value again is a no-op, so repetition in conversation does not multiply versions. Losing something is written as a version whose value is empty, which closes a possession without rewriting the history that contained it. Nothing in this procedure chooses between two accounts: where the key admits both, both are kept, and the reader is given each with the utterance behind it.

Recording also preserves the event links and subject–slot timelines present in the store. Where two accounts occupy the same key, both versions remain linked to their source utterances rather than one being discarded. The longitudinal example in Section 4 follows these recorded event-to-event links, so the trajectory remains tied to the evidence that produced each step.

3 Using the Memory

This section describes the two reads implemented in MemMooring and the retrieval path used in our evaluation. A horizontal cut returns the state of the world at a chosen moment; a longitudinal read follows one recorded event through the events and changes that developed from it. The answering path uses embedding retrieval, with an interval channel added for questions that name a calendar date. The two reads are shown against the transcript in Section 4, while the reported accuracy numbers come from the retrieval path.

3.1 Recall in Layers, and the Cut in Time

Nothing is deleted, so forgetting has to be expressed as distance. Each fact carries a grain – how long it keeps mattering – with three levels: an occasion that matters for a day, a stretch that matters for weeks, an undertaking that matters for a year. Recall then has three layers: the state table, which is always present; shallow recall, which contains the events whose grain still covers the present moment; and deep recall, which contains everything else and is reached only when something triggers it.

Time can be cut along either axis. A cut on the world-time axis selects the facts and events in force at a chosen moment in the world; a cut on the recording-time axis selects what the memory held at a chosen point in its own history. The two cuts can be combined when a question asks what was true at one moment according to what had been recorded by another. The cut returns four things. The states in force: for every slot of every entity in question, the version whose validity covers the selected world moment and whose record remains in the selected memory history. The events under way: those whose interval spans the selected world moment. The events still telling: those that ended earlier but whose grain has not run out. The events still ahead: those already present in the selected memory history whose time lies after the selected world moment – the plans. Nothing outside the selected axis position may enter the result, and the cut is drawn from the stored intervals when the question arrives rather than kept as a snapshot, so it can be taken at a moment where nothing happened to be recorded.

3.2 The Chain an Event Develops Along

The cut in time returns a person at a moment. The implemented longitudinal read returns the development of one recorded event: where it began, what happened along it, and where it ended or, if it has not ended, the latest state recorded for it. It follows event links already present in the store and returns the linked events, the states they changed, and the source utterances behind them. An event is the only fact this store holds; a fact about a person is what an event changed. The walk starts from a seed event, follows the recorded links to the decision or earlier event connected to it, then follows later recorded steps and the facts each step changed. It does not join events merely because they share a participant or a day. This is the traversal used in the worked example in Section 4.

As an operation, the longitudinal read takes an event and a moment of knowing, and returns the recorded trajectory visible by then: the opening event when one is linked, the subsequent steps, the states they changed, and the source words for each link. If a boundary event is recorded, the trajectory ends there; otherwise the last licensed step is returned as the current state. A state version may be handed in instead of an event, in which case the event that set it is taken as the seed. The walk follows recorded links only. Events that merely share a participant or a day are not added, and no missing link is inferred to widen the answer.

3.3 What a Question Is Handed

Questions are answered through a retrieval-augmented generation (RAG) path: retrieval selects records from the event–state store and supplies them as context to the answering model. For ordinary question answering, records are ranked by embedding similarity. For questions that name a calendar date, the embedding list is interleaved with an interval channel that orders events whose stored interval intersects the named date by their distance in days from it. Both paths are measured against the same evidence pointers (Section 4.2), so a question whose evidence is in the store but is not returned is a retrieval failure rather than a storage failure. The horizontal and longitudinal reads are presented separately as transcript-linked outputs; the reported answering scores measure the RAG path.

4 Evaluation

This section reports the completed measurements: ordinary LoCoMo question answering, a record-limited two-channel retrieval comparison with its no-date control on date-naming questions, evidence coverage, store composition and indexed lookup latency. It closes with two transcript-linked worked case studies: a horizontal cut and a longitudinal undertaking trace.

We separate what the store keeps, what retrieval hands over, and what the answering model does with it. The evaluated stores are extracted with Ollama qwen3.6:latest (36.0B MoE, GGUF Q4_K_M, 262,144-token context). The end-to-end comparison uses ten LoCoMo conversations and 1540 answerable questions, MiniMax-Text-01 at temperature zero with the answer prompt used in the Mem0 evaluation, and DeepSeek-chat and GPT-4o as separate judges. The comparator is the current Mem0 implementation used in this study, not an earlier Mem0 release; its configuration and the shared answering protocol are held fixed. The memory is the system-level difference between MemMooring and current Mem0. Within MemMooring, the retrieval ablation changes record selection and order at the same record limit. It measures the combined effect of the time and lexical channels added to embedding retrieval.

Temporal scoring uses a deterministic one-day tolerance applied after the judge: an answer the judge marked wrong is counted correct only when its date lies within one day of the gold date. The same rule is applied to every system and ablation arm. We do not use the Mem0 evaluation prompt's fourteen-day date tolerance or fifty-percent duration tolerance. Expanding our deterministic tolerance from one to seven days changes the observed system gap by at most 0.2 point, so the reported comparison is not produced by choosing a wide window.

We report the no-memory condition first as the lower bound: it shows how much of the benchmark can be answered from the question alone.

Table 1: End-to-end accuracy on the 1540 LoCoMo questions that remain once the adversarial category is excluded. All rows share the answering model, the answer prompt, the judge prompt and a deterministic one-day date tolerance applied after judging, and no answer is left unjudged. The last row adds the interval and content-word channels on the 181 questions whose text names a calendar date, and coincides with the row above it on every other question.

System	DeepSeek judge	GPT-4o judge
No memory (question only)	12.5	15.8
Mem0	85.5	82.2
MemMooring, embedding retrieval	85.7	82.3
MemMooring, all channels	86.6	83.1

Table 1 places the systems on ordinary LoCoMo question answering. MemMooring reaches 86.6% and 83.1% against 85.5% and 82.2% for Mem0 and 12.5% and 15.8% for the same answering model given no memory. The no-memory row fixes how much of the benchmark is answerable from the question alone. The difference from Mem0 is 1.1 and 0.9 points and is not significant under either judge (exact paired $p = 0.27$ and 0.43), and the more informative quantity is the disagreement behind the totals: the two systems differ on 234 questions under the first judge and 268 under the second, gaining and losing in almost equal numbers. Two independently designed memories reaching the same total while disagreeing on one question in six is not evidence that they hold the same information, and we therefore read this table as locating the system rather than as a result about structure. The result about structure is the ablation that follows, where the system is compared against itself.

Of the 1540 questions, 181 name a calendar date. This subset is defined rather than selected: a question enters it when a year, or a year and a month or day, can be parsed from the question itself, so the same rule applied by anyone yields the same 181 items, and nothing about our answers takes part in the decision. At a limit of thirty records we compare embedding similarity alone against the same ranking interleaved with a channel that orders the events whose stored interval intersects the named date by distance in days from that date. Where a question names only a year the channel yields nothing and the selection falls back to the embedding ranking, so the baseline is never handed fewer records than the ablated system. Accuracy rises from 77.3% to 83.4% under DeepSeek (+17/−6 question flips, exact paired $p = 0.035$) and from 75.7% to 81.8% under GPT-4o (+16/−5, $p = 0.027$). On the temporal category, over the same 37 questions, it rises from 56.8% to 75.7% under both judges. The judges agree on 178 of the 181 items, no judge call failed, and the one-day tolerance rescues no question in either row. A timestamped record admits no such ordering: a distance in days requires an interval to measure from.

Table 2: Retrieval comparison on the 181 questions whose text names a calendar date. The baseline selects exactly thirty records and no row selects more; renderer, answering model, answer prompt and judge prompt are held fixed, so the only difference between rows is which records are selected and in what order. Channels are interleaved one record at a time, interval channel first. Adding the content-word channel on top of the interval channel changes 9 and 8 questions in one direction against 6 and 7 in the other ($p = 0.61$ and 1.00), so the gain is attributable to the interval channel rather than to the number of channels. p is the exact paired binomial test against the embedding baseline, and the deterministic one-day date tolerance changes no row.

	DeepSeek judge			GPT-4o judge		
Retrieval	Acc.	Gained / lost	p	Acc.	Gained / lost	p
Embedding similarity only	77.3	–	–	75.7	–	–
+ interval channel	83.4	+17 / −6	0.035	81.8	+16 / −5	0.027
+ interval + content-word	85.1	+22 / −8	0.016	82.3	+21 / −9	0.043

The retrieval measurements say where the answer gain enters, and where it does not. On the questions that name a date, strict coverage of the dataset's own evidence pointers rises from 78.3% to 88.9% at the same limit of thirty records. On the 1356 questions that name no date and carry an evidence pointer, the two retrievals return 81.4% coverage each: the interval channel has nothing to order by there, and it changes nothing rather than trading accuracy for noise. That pair is the control for the ablation above – the channel improves selection exactly where a date is available to order against, and nowhere else. The remaining rows describe the store itself. Across the ten stores, 28.9% of validity intervals are closed by a later assertion, 42.2% of timelines contain more than one version, and the longest chain contains 103 versions, so the history the write contract preserves is not a corner case. On a synthetic store of one million versions, indexed as-of lookup takes 0.44 ms against 26.1 ms for a linear scan of the same records held as JSON; index construction and JSON loading take 47 s and 1.6 s, so the index repays its construction at roughly 1760 as-of queries. These are storage and retrieval results. They are not scores for the two cut operations, which are shown as worked examples below.

Table 3: Storage and retrieval layers, measured without the answering model. Coverage is strict – every evidence pointer the dataset gives for a question must be covered by the selected records – and is measured at a limit of thirty records for both retrievals. The store statistics are over the ten built stores; the latencies are on synthetic stores of one million state versions.

Layer	Quantity	Value
Retrieval	Coverage on the 181 date-naming questions, embedding only	78.3%
Retrieval	Coverage on the same questions, embedding + interval channel	88.9%
Retrieval	Coverage on the 1356 questions naming no date, embedding only	81.4%
Retrieval	Coverage on the same questions, embedding + interval channel	81.4%
Storage	Validity intervals closed by a later assertion	28.9%
Storage	Timelines holding more than one version	42.2%
Storage	Longest version chain	103
Storage	As-of lookup, one million versions	0.44 ms
Storage	Linear scan of the same records as JSON	26.1 ms
Storage	Index construction / JSON load	47 s / 1.6 s

4.1 The Benchmark

LoCoMo Maharana (2024) is a set of long multi-session conversations between two speakers, each session dated, with questions annotated for the turns that license them. We use the four answerable categories and treat the adversarial one separately: its questions carry no gold answer, the field that looks like one states the opposite of the correct behaviour, and what is scored there is refusal. Sessions are separated by days to weeks, so a judge accepting a date within a fortnight accepts most of the wrong dates in the store as well, and many turns carry an image whose caption some questions were written against. Corpus counts, duplicate handling and the input boundary are fixed in the artifact manifest.

LoCoMo's ordinary questions do not directly request a complete cross-section or a process trajectory. We therefore use its aggregate score to locate the systems, its evidence pointers to measure retrieval, and its date-naming subset for the reported two-channel retrieval comparison.

4.2 Evaluation Layers and Measured Results

Every question is scored at three layers, and the reported quantities are kept separate. A percentage belongs to the last hop that produced it.

A percentage belongs to the last hop that produced it, and numbers whose inputs or models differ never share a sentence and are never subtracted: a difference between them is not a loss attributable to a layer. Attribution is done by reading the three columns of one question together. Matching rules are fixed in the evaluation scripts; judge sensitivity and failed judge calls are stated with the ablation results below.

4.3 Ablations

The reported ablation compares embedding-only retrieval with the interval channel interleaved into it, and then with a weighted content-word channel added on top of both, all at the same limit of thirty records (Table 2). The two steps separate cleanly: the interval channel carries the gain, and the channel added above it does not move the outcome by a measurable amount.

4.4 Cost and Composition

Table 3 reports the measured store composition, version-chain statistics and indexed as-of latency, including the cost of index construction.

4.5 Worked Case Studies

The first worked example is a horizontal cut, taken from the store built for the second conversation. One speaker, Jon, is cut at 20 January 2023 as the memory stood at the end of that day. The cut returns thirty-three events then under way, one earlier event whose effect had not yet run out, thirteen facts then in force and none merely possible, and three events that had not yet happened: two records placing a demonstration of new dance moves on 27 January and one placing a performance of his own choreography at a nearby festival on 1 February. Those two dates were not spoken. The speaker said 'next Friday' and 'next month'; the phrases are stored verbatim with the day they were uttered, and the dates were computed from them in code. Both events lie after the moment of the cut on the world axis and before it on the recording axis, which is why they can be returned as things then expected rather than as things then done. A cut of the same speaker one day earlier returns none of these records, because all of them were learned during 20 January. This example is checked against the transcript; it is not an accuracy estimate.

The second worked example follows one undertaking of the same speaker longitudinally. Its opening decision is recorded on 20 January 2023 from turns D1:1 and D1:2 – he intends to start his own business – and twenty-two further records hang from it, three levels deep, through 13 June. The walk returns that decision, then the steps as they were told: on 1 February he is still looking for premises, on 4 February he is rehearsing for a show while searching, on 8 February he says he loves running the studio, on 11 May he is teaching there, and on 13 June he states what he wants the place to become. One record on 11 May supplies the cause of the January decision – losing his job gave him the push – so an explanation of the chain's root arrives five months after the root itself; the two axes keep that order without contradiction. Underneath the chain, the speaker's occupation slot holds thirty-seven versions, thirteen of them with a closed interval of validity, running from unemployed on 19 January to a temporary job taken in late July while investors are sought. The trajectory is not monotone and is not single-valued. The slot resolves into three lines, distinguished by which speaker's turns asserted them: on 1 July the line written from his own turns holds that he is opening a dance studio, while the line written from his interlocutor's turns holds that he has lost his job. Neither closed the other, and the store returns both with the utterance behind each. A representation that resolved the disagreement at write time would have to discard one of them.

5 Related Work

Memory systems differ in what a write produces and therefore in what a read can return. Production memory layers extract statements and retrieve them by similarity Chhikara (2025). Bitemporal engines and graph systems attach valid and transaction intervals, supersession and provenance, and can recover a past snapshot or belief state Wang (2026); Rasmussen et al. (2025); Zhang (2026); a typed algebra specifies the corresponding write-path guarantees Wang (2026). State-aware banks keep superseded records with explicit roles and successor links Shi (2026), while editable-core systems keep a compact state with less lineage Packer et al. (2023); LangMem (2025). MemMooring does not claim as-of querying as unique. Its distinction is the object reconstructed from conversation and exposed to the

reader: a horizontal set containing states, ongoing and consequential events, and already-known future plans, plus a longitudinal undertaking whose members record how one matter developed. The current experiments quantify the time-aware retrieval consequence; the completeness of those two structured reads remains to be measured.

6 Limitations

Slot assignment is the dominant failure. A value filed under the wrong slot does not merely add noise: because a new version closes the old one, it replaces a correct fact and becomes the current answer. Pairing states with their events (Section 3.3) exposes the error but does not prevent it. The stronger remedy is a closed vocabulary for status-like slots, enforced in code at write time, with values outside the vocabulary routed to an open slot; this is specified but not yet part of the measured system.

Building the undertaking layer is difficult. Asking a model to induce undertakings globally yields themes rather than happenings, and linking a new thing to an open undertaking requires candidate filtering in code. The current evaluation keeps this as a transcript-linked worked case study and does not claim an aggregate structure score.

The cut evaluation is narrower than the retrieval evaluation. Ordinary question answering and the date-channel ablation use all ten LoCoMo conversations, while the horizontal and longitudinal cuts are shown on one transcript-linked example. Their output shape is inspectable; their completeness and process accuracy are not yet quantified.

7 Conclusion

MemMooring treats conversation as evidence from which a timed world representation is rebuilt. Events retain when they happened, when they were recorded and the words that reported them; the states they change remain as versioned intervals. This representation defines two complementary reads: a horizontal cut that asks who someone was and what surrounded them at a moment, and a longitudinal cut that asks where one undertaking began, how it developed and whether it ended or remains in progress. The current evidence is narrower than the full claim. On ten LoCoMo conversations the stores hold the required version structure and indexed as-of lookup scales to one million versions; adding the interval channel at a fixed record limit raises accuracy on date-naming questions by 6.1 points under both judges and changes nothing where no date is named. Ordinary question answering is at parity with a production snippet memory. Quantifying the two cuts is the next step.

Use of Large Language Models (required by ICLR 2027; does not count toward the page limit)

In this work, generative AI tools assisted with implementation of the memory system and evaluation harness, data cleaning and replay, feedback on experimental protocols, qualitative failure analysis, interpretation of results, literature discovery, reference formatting, figure preparation, and drafting and editing portions of the manuscript. They were not used to generate synthetic datasets, formulate mathematical claims or proofs, or translate research materials. The authors supplied the central conceptual framework and hypotheses, checked cited sources and numerical results, reviewed all AI-assisted text and artifacts, and take responsibility for the final content.

Reproducibility Statement (does not count toward the page limit)

Section 4 and the artifact manifest state the public dataset, evaluated question counts, model names, retrieval budget, deterministic date rule, matching metrics, question-level flips and paired tests used in the reported experiments. The submission package includes the result records under artifacts/results/, the evaluation scripts under artifacts/scripts/,

and the run logs under artifacts/logs/. The exact extraction, answering and judge prompts are documented by configuration but are not reproduced in the manuscript.

LoCoMo is public. Replaying a conversation sets recording time from session dates rather than ingestion wall-clock time. We checked one conversation for agreement in node and edge counts and in event and state content after replay. The synthetic one-million-version latency experiment and the remaining cost measurements are reported in Section 4.4 and Table 3.

Ethics Statement (does not count toward the page limit)

The system stores a durable record of what a person has said about their life, including sensitive attributes, that is never rewritten or deleted, and makes every stored fact traceable to the utterance that produced it. This is the property that makes the memory auditable and also the property that makes it dangerous: a design in which nothing is deleted is in direct tension with a person's right to have information about them erased. Closing a belief window is not erasure, and we do not claim it is. A deployment would need a destructive delete path that removes an entity's events, the states derived from them and the evidence pointers into the transcript, and we regard designing that path – and stating what it costs the audit trail – as required work rather than future work. The evaluation in this paper uses public benchmark conversations with fictional participants.

Funding

This information is provided through the submission system and omitted from the anonymous manuscript.

Conflict of Interest

This information is provided through the submission system and omitted from the anonymous manuscript.

Data Availability

LoCoMo is publicly available. This submission package includes the experiment configuration, per-question evaluation records, evaluation scripts, run logs and generated paper sources under artifacts/. The implementation and exact service prompts are maintained with the accompanying project release.

References

Allan, J. et al. (1998). Topic detection and tracking pilot study: Final report. DARPA Broadcast News Transcription and Understanding Workshop.

Allen, J. F. (1983). Maintaining knowledge about temporal intervals. Communications of the ACM, 26(11), 832-843.

Ding, T., Nannapaneni, A., Liu, B., & Zhang, L. (2026). Always-On Agents: A survey of persistent memory, state, and governance in LLM agents. arXiv:2606.30306. https://arxiv.org/abs/2606.30306

Shen, Z., Cheng, S., Guo, Z., Wang, W., Wang, Y., & Huang, H. (2026). AnchorMem: Anchored facts with associative contexts for building memory in large language models. Findings of ACL 2026. arXiv:2604.17377. https://aclanthology.org/2026.findings-acl.1736/

Shi, Z., Tang, Y., & Tung, A. K. H. (2026). A-TMA: Decoupling state-aware memory failures in long-term agent memory. arXiv:2607.01935v2.

Badhe, S., Tiwari, P., & Chung, J. (2026). SKILL.state: Scalable long-horizon agent skills. arXiv:2608.26263.

Brown, R., & Kulik, J. (1977). Flashbulb memories. Cognition, 5(1), 73-99.

Bugert, M., Reimers, N., & Gurevych, I. (2021). Generalizing cross-document event coreference resolution across multiple corpora. Computational Linguistics, 47(3), 575-614.

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.

Razniewski, S., Arnaout, H., Ghosh, S., & Suchanek, F. (2024). Completeness, recall, and negation in open-world knowledge bases: A survey. ACM Computing Surveys. arXiv:2305.05403.

Datomic documentation. Transaction data reference: assertion and retraction of datoms. https://docs.datomic.com/transactions/transaction-data-reference.html

Chen, Y., Zheng, Y., Li, Y., Xiao, B., Xu, B., Qiao, S., Fang, J., Deng, X., Yao, Y., Wang, X., Zhang, L., Li, H., Chen, H., & Deng, S. (2026). EM^2Mem: Event-centric multimodal memory for large language models. arXiv:2609.00551.

Fountas, Z., Benfeghoul, M. A., Oomerjee, A., Christopoulou, F., Lampouras, G., Bou Ammar, H., & Wang, J. (2025). Human-inspired episodic memory for infinite context LLMs. ICLR 2025. arXiv:2407.09450.

Wang, L. (2026). Less context, more accuracy: A bi-temporal memory engine for LLM agents (Engram). arXiv:2606.09900. Code: github.com/ly-wang19/engram.

Zou, H., Sun, T., He, C., Tian, Y., Li, Z., Jin, L., Liu, N., Zhong, J., & Wei, K. (2026). ES-Mem: Event segmentation-based memory for long-term dialogue agents. arXiv:2601.07582.

Fowler, M. Event sourcing and retroactive event. https://martinfowler.com/eaaDev/EventSourcing.html

Wu, Z., Zhang, H., Lin, F., Xu, W., Xu, X., Chen, Y., Zou, H. P., Chen, S., Zhang, W., Liu, X., Yu, P. S., & Wang, H. (2026). GAM: Hierarchical Graph-based Agentic Memory for LLM Agents. In Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 34647-34664. Association for Computational Linguistics. https://doi.org/10.18653/v1/2026.acl-long.1600

Humaidan, D., Otte, S., & Butz, M. V. (2020). Fostering event compression using gated surprise. arXiv:2005.05704.

Gorlo, N., Wise, D. K., Speranzon, A., & Carlone, L. (2026). Worth Remembering: Surprise-Gated Robot Episodic Memory. arXiv:2606.03787.

Yue, J., Hu, C., Sheng, J., Zhou, Z., Zhang, W., Liu, T., Guo, L., & Deng, Y. (2026). HyperMem: Hypergraph memory for long-term conversations. arXiv:2604.08256.

Xu, R., Yang, T., & Huang, W.-C. (2026). HyperSkill: Self-evolving LLM agents via hypergraph-structured skill memory. arXiv:2608.16114.

Kurby, C. A., & Zacks, J. M. (2008). Segmentation in the perception and memory of events. Trends in Cognitive Sciences, 12(2), 72-79.

LangMem (2025). LangChain. https://langchain-ai.github.io/langmem – memory items with in-place patching and no version history.

Li, D., Liu, Z., Wang, J., Huang, J., Li, F., Jia, B., Hu, B., & Zhang, M. (2026). LycheeMemory V2: Efficient long-term memory for LLM agents via semantic segment-level consolidation. arXiv:2608.12990.

Maharana, A., Lee, D.-H., Tulyakov, S., Bansal, M., Barbieri, F., & Fang, Y. (2024). Evaluating very long-term conversational memory of LLM agents (LoCoMo). ACL 2024.

McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex. Psychological Review, 102(3), 419-457.

Chen, H. et al. (2026). MemForest: An efficient agent memory system with hierarchical temporal indexing. arXiv:2605.23986 (v3, 6 Sep 2026). [verified: Appendix B.2 fixed-retrieval dual-time control; current-state / historical-state / transition framing]

Wang, J., Sun, T., Zhu, J., Zhang, C., Li, S., Liu, X., Wen, Z., Tu, X., Ren, J., Zhang, X., Yuan, Z., & Zhang, L. (2026). MemForest: Efficient Agent Memory Management via EventTree Partitioning and Progressive Merging. arXiv:2609.08273. Distinct work sharing the name MemForest; cite by arXiv identifier.

memnode. Persistent memory for AI agents with typed provenance and correction lineage. https://memnode.dev/ [product, not a paper]

Memobase. LoCoMo benchmark judge prompt (docs/experiments/locomo-benchmark/metrics/llm_judge.py). https://github.com/memodb-io/memobase

Ma, E., Zhou, Y., Huang, W.-C., Yang, J., Ma, H., Wang, Z., Li, C., Miao, C., Yu, P. S., & Wang, Z. (2026). MEMPROBE: Probing Long-Term Agent Memory via Hidden User-State Recovery. arXiv:2606.24595. https://arxiv.org/abs/2606.24595

Massari, A., Mariani, F., Heibi, I., Peroni, S., & Shotton, D. (2024). OpenCitations Meta. Quantitative Science Studies, 5(1), 50-75. https://doi.org/10.1162/qss_a_00292

W3C. Time ontology in OWL (OWL-Time). W3C Recommendation. https://www.w3.org/TR/owl-time/ – sections on temporal vagueness and on validity state both as explicitly out of scope.

Packer, C. et al. (2023). MemGPT: Towards LLMs as operating systems. arXiv:2310.08560.

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. UIST 2023. arXiv:2304.03442.

Amouyal, S. J. et al. (2023). QAMPARI: A benchmark for open-domain questions with many answers. GEM@ACL 2023. arXiv:2205.12665.

Zhou, Y., Huang, H., Liu, Y., Dai, R., Wang, X., Zhang, X., Shi, S., & Deng, Y. (2026). Do retrieval augmented language models know when they don't know? AAAI 2026. arXiv:2509.01476. https://ojs.aaai.org/index.php/AAAI/article/download/40822/44783

Rasmussen, P. et al. (2025). Zep: A temporal knowledge graph architecture for agent memory. arXiv:2501.13956.

Rogers, T. B., Kuiper, N. A., & Kirker, W. S. (1977). Self-reference and the encoding of personal information. Journal of Personality and Social Psychology, 35(9), 677-688.

Zhong, V. et al. (2022). RoMQA: A benchmark for robust, multi-evidence, multi-answer question answering. arXiv:2210.14353.

Wu, Y., Li, J., Liang, X., Chen, Y., Liang, Y., Mo, L., & Li, G. (2026). SAG: SQL-Retrieval Augmented Generation with Query-Time Dynamic Hyperedges. arXiv:2606.15971.

Lu, Z., Li, D., Shi, Y., Wang, B., Wang, L., & Hu, B. (2026). Structured episodic event memory. arXiv:2601.06411.

Fan, X., Liu, M., Yang, R., Ouyang, S., & Han, J. (2026). Can agent memory systems track evolving state? (StateMemBench / StateMem). arXiv:2608.19652.

Piryani, B., Abdallah, A., Mozafari, J., Anand, A., & Jatowt, A. (2026). It's high time: A survey of temporal question answering. ACL 2026. arXiv:2505.20243.

Zhang, X. (2026). TGMS: An agent-native bi-temporal graph management system. arXiv:2607.10265.

Sun, Q., Li, S., Huynh, D., Reynolds, M., & Liu, W. (2025). TimelineKGQA: A comprehensive question-answer pair generator for temporal knowledge graphs. In WWW '25: Companion Proceedings of the ACM on Web Conference 2025 (pp. 797-800). ACM. https://doi.org/10.1145/3701716.3715308.

Wang, Z. (2026). TOKI: A Bitemporal Operator Algebra for Contradiction Resolution in LLM-Agent Persistent Memory. arXiv:2606.06240.

Shu, Y., Liu, P., Zhang, T., Gao, R., Ma, J., & Sun, C. (2026). TraceMem: Weaving narrative memory schemata from user conversational traces. arXiv:2602.09712.

Tulving, E. (1972). Episodic and semantic memory. In E. Tulving & W. Donaldson (Eds.), Organization of Memory. Academic Press.

Tulving, E., & Thomson, D. M. (1973). Encoding specificity and retrieval processes in episodic memory. Psychological Review, 80(5), 352-373.

W3C (2013). PROV-O: The PROV ontology. W3C Recommendation, 30 April 2013. https://www.w3.org/TR/prov-o/

Wu, D. et al. (2025). LongMemEval: Benchmarking chat assistants on long-term interactive memory. ICLR 2025.

XTDB documentation. Time in XTDB: valid time and system time (SQL:2011 bitemporality). https://docs.xtdb.com/about/time-in-xtdb.html

Zwaan, R. A., Langston, M. C., & Graesser, A. C. (1995). The construction of situation models in narrative comprehension: An event-indexing model. Psychological Science, 6(5), 292-297.