ATTESTMEM: TURNING LONG CONVERSATIONS INTO TRACEABLE, REVISABLE MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon language agents require persistent memory. Existing systems reduce dialogue histories to collections of independently embedded chunks: outdated or contradictory information can coexist at retrieval time, and semantic similarity alone decides which memories enter the answering context. We introduce AttestMem, a training-free memory framework that represents conversations as a revisable collection of attested statements and compiles a bounded answering context from recalled candidates. Each statement carries its extraction provenance, utterance time, validity interval, and ingestion order; when deterministic source matching succeeds, the row additionally carries turn-level references and its source-match grade, and ambiguous or unsupported cases are explicitly marked rather than guessed. An append-only event stream incorporates updates so that they supersede rather than overwrite earlier values, while rejected writes and historical versions remain queryable. At inference time, candidates are nominated through complementary routes and compiled under a character budget that prioritizes assertions and recovered source quotations while trimming transcript text. With GPT-4o-mini for memory construction and answering, AttestMem reaches 89.03% accuracy on LoCoMo and 86.00% on LongMemEval-S; with GPT-4.1-mini, 93.44% and 89.10%, using 3.6–3.9k prompt tokens per question. These results match or exceed the strongest reported baseline in each backbone group on both benchmarks. Paired ablations show the largest accuracy decrease when the ledger is removed, and removing the context limit lowers LoCoMo accuracy by 14.25 points while tripling the prompt.

1 INTRODUCTION

A conversational agent is judged, over months of use, by what it can be relied on to remember. It must hold a stable model of a user, answer about facts that have since changed, and reason over evidence scattered across dozens of sessions. A longer context window alone does not ensure reliable memory use: adding input is not the same as using the information already available (Liu et al., 2023). Agents therefore keep memory outside the context and assemble a compact context per question (Wu et al., 2024; Maharana et al., 2024). Memory is consequently a *process* rather than a *store*, and the process has two halves: dialogue is turned into something persistent, and part of that persistence is selected, per question, into a bounded window.

Memory construction and retrieval have both improved, but the step that assembles the context has not. The write half is richly instrumented: a stored memory commonly carries the turns that support it, the moment it was uttered, the interval over which it holds, and the statement that replaced it (Rasmussen et al., 2025; Xu et al., 2025; Chhikara et al., 2025; Yao et al., 2026; Li et al., 2026a; Dai et al., 2026; Chen et al., 2026; Yue et al., 2026; Tang et al., 2026; Hu et al., 2026). The read half has improved in how candidates are ordered and how far a search may wander (Wang et al., 2025a;b; Jiang et al., 2026; Kang et al., 2026). What has not changed is the last step: however a candidate was produced and however well it is annotated, the set handed to the model is whatever scores highest until the space runs out. We call this the **similarity-sufficiency assumption**, that relevance to the question is the whole answer to whether a memory may enter the context. It concerns the decision rather than the annotation, which is why it survives every field the write half adds.

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.

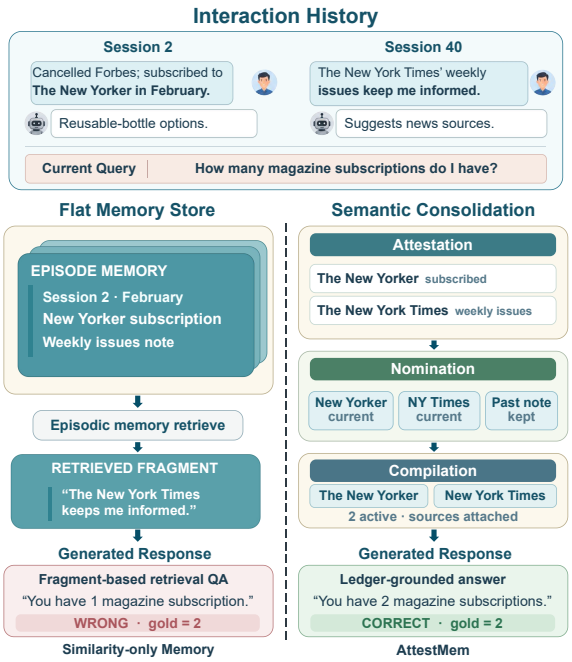


Figure 1: Similarity-only retrieval versus the attested ledger on one LongMemEval-S case.

Figure 1 shows the failure on one case. Coverage does not explain these failures: in our own runs the questions the system fails on every sampling draw are no less covered than the ones it answers, the answer turn already being cited by a rendered row in 98.1% of the former against 97.9% of the latter. The evidence is in the window and the answer is still wrong. What is missing is a decision about which of the assembled memories should enter the answering context, and a way to verify it. We build that decision into the step that fills the window, over a single object, an **attested-statement ledger**, and distinguish it from the three nearest lines of work: systems that revoke stale memories in the store (Zhou et al., 2026; Cui et al., 2026), systems that validate memory-management transitions with verifiers (Sun et al., 2026; Yang et al., 2026), and systems that treat the context as a budget to be restored (Sheverev et al., 2026; Kang et al., 2026). Each decides on content before retrieval, when memory is written, or after ranking; in each, the step that fills the window is still a ranking step. **Attestation** turns dialogue into statements tied to the turns that support them, each graded by deterministic source matching rather than by asking a model to cite, and appends revisions so that a later statement supersedes an earlier one. **Nomination** retrieves candidates through several routes and deletes none. **Compilation** then fills a character budget that prices classes of text rather than ranking them uniformly, while a separate budget keeps the verbatim source of the assertions that survive, so what was paraphrased into the context can still be checked. Two invariants hold the ledger together: no statement enters it without a provenance record, and no read-time judgement deletes one.

On LoCoMo and LongMemEval-S, AttestMem reaches 89.03% and 86.00% with GPT-4o-mini, and 93.44% and 89.10% with GPT-4.1-mini, using each backbone for both construction and answering. Its largest module loss comes from removing the ledger; on LoCoMo, removing the context limit costs even more accuracy while tripling the prompt. Our contribution is the complete path from source-linked statements to a bounded answering context, together with paired ablations that separate the value of the ledger, candidate nomination, and the final cut — and a measurement the field list does not suggest: the ledger's own fields make poor admission rules, and letting them decide costs 7.47 points while rendering more.

2 Related Work

Memory construction and retrieval. Conversational memory has progressed from retrievable records (Zhong et al., 2024; Packer et al., 2023) to structured designs that impose entities, relations and hierarchies — recursive trees, community summaries, temporal knowledge graphs and hyperedges over co-occurring episodes (Sarthi et al., 2024; Edge et al., 2024; Jiménez Gutiérrez et al., 2024; Rasmussen et al., 2025; Xu et al., 2025; Yue et al., 2026; Tang et al., 2026; Hu et al., 2026; Chen et al., 2026) — and consolidation procedures (Dai et al., 2026; Li et al., 2026a; Jiang et al., 2026). Hybrid scoring, multi-route search, and decomposition improve which records are found (Tang et al., 2026; Wang et al., 2025a;b). Yet audits find failures even when the supporting evidence is retrieved (Ma et al., 2026; Huang et al., 2026), and raw-log retrieval can rival structured memory (Li et al., 2026b). These results motivate using stored structure in the final context decision, rather than treating its existence as the contribution.

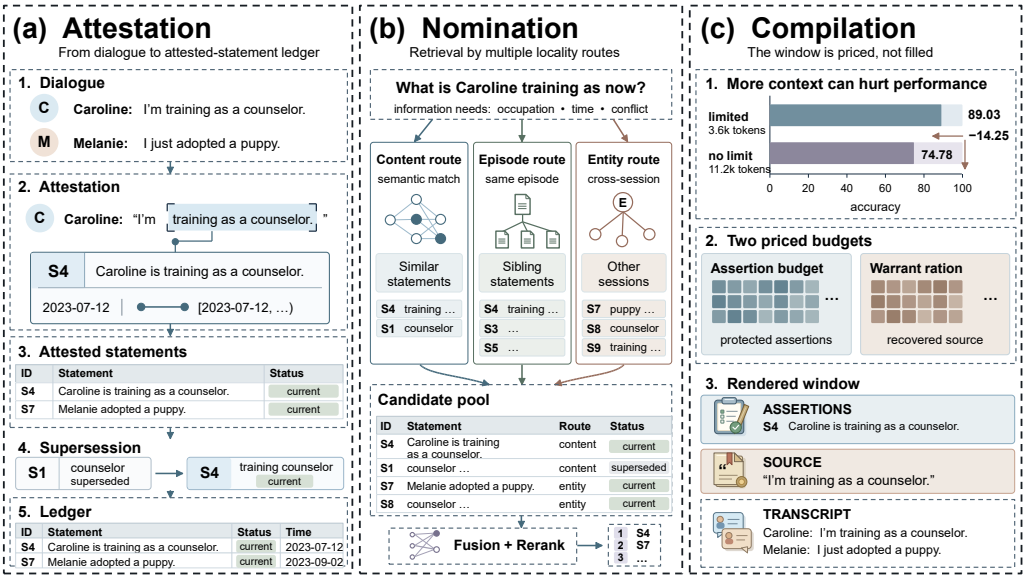


Figure 2: The three operations. Attestation writes source-linked statements with append-only revisions; Nomination proposes candidates without deleting history; Compilation spends a bounded window across assertions, verbatim source, and transcript.

Where the decision is made. The closest systems use temporal state to revoke or supersede stored memories (Zhou et al., 2026; Cui et al., 2026), select applicable historical states through temporal profiles and query-time temporal constraints (Yao et al., 2026), or verify memory-consolidation transitions with verifiers (Sun et al., 2026; Yang et al., 2026). Budget-aware systems compress context or select operators under a limit (Sheverev et al., 2026; Kang et al., 2026). AttestMem instead preserves historical candidates and decides how much assertion, source, and transcript text to render when filling the answering window.

3 Problem Formulation and Framework Overview

Setting. A conversation arrives as a dated stream of turns. We write $\mathcal{D} = \{u_1, \ldots, u_T\}$ for the turns in utterance order, each u_i carrying a speaker from {user, assistant}, a text span, and the date on which it was said. A question q arrives with a query date t_q and asks about some state of the user. The system must produce an answer a from a context $C(q)$ that it assembles from its own memory, under a nominal capacity that bounds the transcript portion of $C(q)$. Nothing is trained: one model both constructs the memory and answers from it, so the only thing that can carry information from $\mathcal{D}$ to a is what the construction step chose to write down and what the assembly step chose to render.

The object. Memory is an *attested-statement ledger*: an append-only collection of *statements*, defined in §4.1. The ledger is neither a cache of the conversation nor a summary of it: statements are the unit that can be judged, quoted and revised, and every later decision is expressed over them. Here *statement* names the persistent ledger object, *assertion* its textual proposition x, *row* its rendered or stored representation, and *warrant* the verbatim source recovered for it. Two invariants define admissible ledgers, both properties of the artefact rather than of a run: every statement carries an extraction provenance record, $\mathrm{prov}(s) \neq \emptyset$ (I1) — the extraction pointer, distinct from the turn-level source references of §4.1 — and no read-time decision removes one, $L_{\text{after read}} = L_{\text{before read}}$ (I2). A system satisfying I1 can always be audited; I2 keeps every historical record retrievable, so a read may decline to *render* a row but cannot remove it, although a correct answer still depends on extraction, retrieval, and reasoning on top of what is kept.

The three moves. Attestation builds the ledger from $\mathcal{D}$; Nomination proposes a candidate set $C_0(q)$ without filtering by ledger state; Compilation renders $C(q)$ under a character budget (Figure 2). This order separates failure to record a claim, failure to find it, and failure to make it usable in the final context. Section 5.1 defines the evaluation and paired ablations for these operations.

4 METHOD

The three operations answer the three questions the introduction raised: is the claim checkable (§4.1), is the right claim reachable (§4.2), and does context construction use the window's budget effectively (§4.3).

4.1 ATTESTATION: DIALOGUE INTO CLAIMS

Segmentation and extraction. The conversation is divided into processing units, each a dated span of turns, recursively split when oversized. Two extraction passes yield standalone assertions: the first preserves specific details, resolves relative event time while retaining the speaker's phrase, and writes a sample question as an index key; the second looks for missed numbers, dates, places, and negations, whose omission is invisible in a summary and fatal in an answer. That question is what lets a statement be retrieved by the shape of the question it answers rather than by lexical overlap with its own wording. The span and its turns remain attached to every assertion. The full instruction is in Appendix A. Similar second-pass extraction appears in prior memory systems (Zhong et al., 2024; Xu et al., 2025; Yue et al., 2026); here it also supplies material for source checking at read time.

The statement and its grade. A statement is a tuple $s = (x, P, \pi, t_u, \tau, \chi, \sigma)$: the assertion x, the turns P supporting it, its *attestation grade* π, utterance time t_u, validity interval $\tau = [t_0, t_1)$, revision links χ, and current status σ. The append-only stream supplies ingestion order. Temporal and atomic-fact stores have related fields (Yao et al., 2026; Rasmussen et al., 2025; Luo et al., 2026); our grade is computed by deterministically matching x to source turns rather than accepting a generated citation (Chhikara et al., 2025). A unique literal match is `exact`; multiple matches retain all candidate turns as `ambiguous`; a sufficiently distinct token-overlap match is `fuzzy`; and a weak match is `unmapped`, with an empty reference. A row without a secure attribution remains visible but cannot masquerade as source evidence. The grade measures source-match status; it does not certify semantic entailment or factual correctness. On the recorded LongMemEval-S build — 1.09M statement rows across the 500 per-question stores — 35.7% of rows grade `exact`, 47.8% `fuzzy`, 14.8% `ambiguous` and 1.7% `unmapped`; every row carries an extraction pointer and 98.3% a turn-level reference (Appendix B, Figure 6). Merge and supersession preserve the union of turn references.

Settlement. A new statement s is compared with its nearest predecessor, the maximum-cosine existing row $p(s) = \arg\max_{s' \in L_{<t}} \cos(e(s), e(s'))$. With $\text{sim} = \cos(e(s), e(p(s)))$: $\text{sim} \geq 0.92$ with identical normalised text confirms the statement; $0.80 \leq \text{sim} < 0.92$ goes to an LLM classifier that returns one operation $o = g(s, p(s)) \in \{\text{SUPERSEDE}, \text{UPDATE}, \text{MERGE}, \text{CONTRADICTION}, \text{DISTINCT}\}$ (if the classifier fails, MERGE at $\text{sim} \geq 0.88$, otherwise ADD); below 0.80 the statement is added as a new row. Every accepted operation appends an event, $L_{t+1} = L_t \cup \{e(o, s, p(s))\}$ — the old state is never mutated or deleted. Operations are validated before application, rejecting missing predecessors, empty statements, and invalid dates; refusals are recorded. SUPERSEDE and CONTRADICTION close the older interval without erasing it, and both rows remain queryable — preserving past-query access and making every revision auditable (Cui et al., 2026).

What the ledger is worth. Removing the entire ledger costs 7.00 points on LongMemEval-S and 6.56 on LoCoMo. Dropping only its status annotations costs 2.13, adding the dates to the drop 3.40; the larger decrease from removing the whole ledger indicates that the annotations alone do not carry its benefit.

4.2 NOMINATION: PROPOSE, NEVER DELETE

The ledger makes a claim inspectable; it does not make the right claim *present*. A multi-hop question is decomposed into information needs, each used as a retrieval key. Dense and sparse routes cover semantic and lexical overlap. An episode route pools the gathered facts onto their episodes and adds other statements from the same episodes as the most-represented ones; among retrieval misses, three quarters have the answer row in an episode already reached by another row, so what is missing is a

sibling of something already found rather than a distant semantic match. An entity route adds facts linked to named entities, including cross-session candidates missed by similarity. Entity edges are used as an index, not rendered as evidence. That distinction is measured, not stylistic: 116 of 121 entities in one conversation have degree one, and two person-hubs carry 62% of its facts, so the graph has no entity-to-entity structure for a second hop to walk — which is why the entity route stops at the linked facts. Each route is a locality assumption about where a row sits when the content route misses it — in the span of something already gathered, or behind an entity the question names — and they fail in complementary ways, which is why they are kept separately rather than fused into one score. The three routes are unioned into the candidate set $C_0(q) = R_{\text{content}}(q) \cup R_{\text{episode}}(\cdot) \cup R_{\text{entity}}(q)$, formed independently of ledger status ($s \in C_0(q) \not\Rightarrow \sigma(s) = \text{current}$), fused by reciprocal-rank fusion ($k = 60$) over the dense, sparse and entity channels, and reranked by a cross-encoder (Qwen3-Reranker-8B) before compilation; collapsing them lowers LongMemEval-S accuracy from 86.00 to 80.60 and LoCoMo from 89.03 to 85.63.

The last of the three is the one a multi-hop question needs, because its two hops usually lie in different sessions and share only an entity; the store's entity graph is the only index that follows an entity by name across that boundary. Nomination does not filter by ledger status. A past-tense question may need a superseded row: excluding non-current rows at retrieval costs 14.3 points on LoCoMo multi-hop questions. The candidate set therefore preserves historical versions; compilation decides how much of each candidate to show (Yao et al., 2026; Zhou et al., 2026).

4.3 Compilation: the window is priced, not filled

Allocating the budget across text classes. The rendered context is not a ranked list but a string whose parts have different standing. **Assertion** rows are the dated, graded statements the ledger produces, and are what a question is answered from. **Source** lines are the verbatim text recovered on their behalf, and are what makes an assertion checkable. **Transcript** is the raw session dump situating both. Assertions and source are protected; the transcript is the region the cut may take, so the nominal budget binds only the transcript and the rendered string can exceed it when protected content is long (Appendix C). Formally, with $A(q)$, $W(q)$ the protected assertion and warrant blocks and B the nominal character budget (8,000 by default), $C(q) = A(q) \oplus W(q) \oplus T_B(q)$ with $|T_B(q)|_{\text{char}} \leq B$, where each transcript line ℓ scores $s(\ell, q) = |\text{terms}(q) \cap \ell| + \lambda_v \mathbf{1}[\ell \text{ carries a value}] + \lambda_h \mathbf{1}[\ell \text{ heads a session}]$ with $\lambda_v = 0.75$, $\lambda_h = 2.0$ (the released defaults), and $T_B(q)$ fills greedily in descending score. Two rules run before the cut and are unconditional: identical row bodies are folded with the current version kept, and a row whose body already appears in the surviving transcript is dropped last, so nothing can justify its own removal. No line is ever reordered; ordering was settled by the previous move.

The limit is not a compromise. The natural reading of a capacity is that it buys tokens at the price of accuracy; the measured direction is the opposite (Section 5.5): removing the limit costs 14.25 points on LoCoMo while the prompt more than triples, and both errors exceed any single module removal. A statement whose value has changed can be answered only if the window contains the row that says so and excludes enough of the surrounding material for it to be found at all.

Warrant: evidence as a separate budget. An assertion that survives the cut is still a paraphrase. A second, independent budget recovers verbatim turns for the surviving assertions (the stored fields are in Appendix B); each selected turn is accompanied by one adjacent turn to preserve context. Keeping this budget separate prevents quotations from displacing assertions. Removing it costs 3.38 points on LoCoMo and 3.00 on LongMemEval-S. We do not gate source recovery by attestation grade: weakly matched rows can be precisely the ones whose source needs checking.

5 Experiments

5.1 Setup

We evaluate LongMemEval-S (Wu et al., 2024) on 500 questions and LoCoMo (Maharana et al., 2024) on 1,540 non-adversarial questions — the 446 adversarial items are excluded for denominator comparability with the published rows we quote; Appendix F scores these 446 items for refusal

Method	Avg. Tokens	Single Hop	Multi Hop	Temporal	Open Domain	Overall
GPT-4o-mini backbone						
Full Context	31.5k	90.01	69.86	52.96	55.21	76.43
Naive RAG	6.2k	57.55	54.61	24.30	51.04	49.68
MemoryOS	5.2k	62.43	56.50	37.18	40.28	54.70
Mem0	1.0k	66.71	58.16	55.45	40.62	61.00
A-Mem	3.0k	66.83	52.13	61.99	30.21	60.84
RecMem	2.7k	79.79	72.70	57.32	58.33	72.47
EverMemOS	2.5k	<u>91.08</u>	**86.17**	<u>81.93</u>	<u>66.67</u>	<u>86.76</u>
AttestMem (ours)	3.6k	**93.46**	<u>85.46</u>	**86.29**	**67.71**	**89.03**
GPT-4.1-mini backbone						
Full Context	31.5k	90.84	82.62	79.13	57.29	84.18
Naive RAG	6.3k	61.24	58.87	33.96	50.00	54.42
MemoryOS	5.5k	67.30	59.34	42.26	59.03	60.11
Mem0	1.0k	68.97	61.70	58.26	50.00	64.20
A-Mem	5.6k	73.25	59.93	72.90	42.71	68.83
RecMem	2.8k	86.92	78.37	75.39	57.29	81.10
EverMemOS	2.3k	**96.67**	**91.84**	<u>89.72</u>	<u>76.04</u>	<u>93.05</u>
AttestMem (ours)	3.9k	<u>96.08</u>	<u>91.13</u>	**90.34**	**87.50**	**93.44**

Table 1: Main results on LoCoMo (1,540 questions; accuracy in percent, mean over three answer draws, majority vote over three `gpt-4o-mini` judgments per draw). Baseline rows are published results: for systems both source papers report we quote EverMemOS, otherwise RecMem (§5.1, Appendix E). Bold marks the best per column and underline the second best.

under REMem's protocol (Shu et al., 2026), where AttestMem attains the comparison's best refusal F1. Experiments are training-free. Each main-table configuration uses the indicated backbone for both memory construction and answering.

Judging and aggregation. The evaluation protocol is EverMemOS's official one, applied unchanged: the released evaluation code (Hu et al., 2026) judges each fixed answer three times with `gpt-4o-mini` at temperature zero, and two `CORRECT` verdicts suffice. Each configuration's accuracy is the mean of the per-draw majority-vote accuracy over three independent answer generations; category accuracy applies the same rule within each category, and overall accuracy weights questions equally. Appendix D specifies the dataset-specific grading prompts.

Published baselines. Baseline rows are published results, not our reruns. Where both source papers report a system, we quote EverMemOS (Hu et al., 2026), whose evaluation code is the protocol we adopt; rows it does not report come from RecMem (Dai et al., 2026), the only other source, and retain RecMem's published aggregation over three runs. Appendix E reproduces both sources' complete tables and records their protocol conventions; comparisons against the RecMem-only rows should be read with that difference in mind.

Each ablation changes one component against a same-batch control on identical stores, questions, and judge calls, under the same three-draw aggregation. Appendix D records paired tests and answer re-rolls.

5.2 Main results

AttestMem matches or outperforms the strongest published memory system under both backbones (Table 1). On LoCoMo, AttestMem reaches 89.03 with `gpt-4o-mini`, 2.27 points above the published EverMemOS result, and 93.44 with `gpt-4.1-mini`, on par with its 93.05. The clearest category gain with the latter backbone is open-domain accuracy: 87.50 versus 76.04.

On LongMemEval-S (Table 2), AttestMem reaches 86.00 with `gpt-4o-mini`, 16.80 points above the published RecMem result. With `gpt-4.1-mini`, it reaches 89.10, 6.10 points above EverMemOS; temporal reasoning rises from 77.44 to 89.85. The gains over Full Context and Naive RAG are 40.40 and 26.60 points on LongMemEval-S, and 12.60 and 39.35 on LoCoMo. The largest category gains over Full Context are temporal on LoCoMo (86.29 against 52.96) and single-session preference on LongMemEval-S (96.67 against 53.33).

Method	Token	SS-User	SS-Asst	SS-Pref	Multi-S	Know.Upd	Temp.Reas	Overall
GPT-4o-mini backbone								
Full Context	112.3k	44.29	80.36	53.33	35.34	55.13	34.59	45.60
Naive RAG	11.9k	85.71	83.93	46.67	52.63	65.38	41.35	59.40
MemoryOS	8.9k	**97.14**	89.29	70.00	58.65	73.08	48.87	67.80
Mem0	1.9k	95.71	55.36	70.00	57.89	69.23	54.14	64.00
A-Mem	13.1k	92.54	**98.21**	36.67	53.38	70.51	44.36	63.80
RecMem	5.6k	91.43	85.71	46.67	56.39	80.77	61.65	69.20
AttestMem (ours)	3.6k	96.43	95.54	**96.67**	**75.56**	**85.26**	**85.34**	**86.00**
GPT-4.1-mini backbone								
Full Context	112.3k	91.43	**100.00**	56.67	53.38	75.64	48.12	66.20
Naive RAG	11.9k	85.71	82.36	83.33	63.91	75.64	45.86	67.00
MemoryOS	9.2k	94.29	89.29	**100.00**	67.67	80.77	54.89	74.40
Mem0	1.1k	82.86	26.78	90.00	63.15	66.67	72.18	66.40
A-Mem	15.5k	95.71	**100.00**	63.33	61.65	82.05	52.63	71.60
RecMem	5.9k	95.71	96.43	86.67	65.41	75.64	68.42	76.80
EverMemOS	2.8k	**97.14**	85.71	93.33	73.68	89.74	77.44	83.00
AttestMem (ours)	3.6k	94.29	98.21	80.00	**80.45**	**94.23**	**89.85**	**89.10**

Table 2: LongMemEval-S results (500 questions; accuracy in percent, same protocol as Table 1). Baseline rows are published results (Section 5.1). Bold marks the best per column and underline the second best.

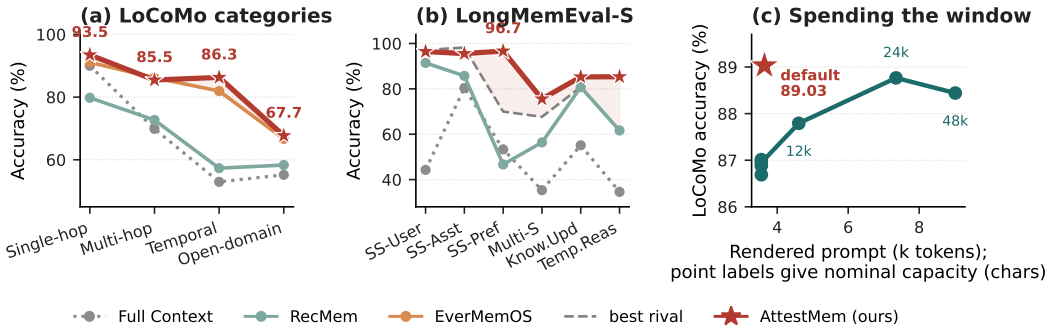


Figure 3: Per-category profiles on both benchmarks against the strongest published systems, and the LoCoMo capacity sweep (GPT-4o-mini backbone). The shaded band is the margin over the best rival per category; the star marks the default setting.

Figure 3 reads the two tables as per-category profiles: the margin over the best rival opens on the categories whose answer depends on a statement's state, and panel (c) previews the capacity sweep that Section 5.5 analyzes.

5.3 Ablation

Removing one operation at a time locates the value (Figure 4), with majority-vote judging at the full denominator. Table 3 lists the prompt cost by configuration; the complete grids are in Appendix G. Each configuration is paired against a same-batch control on identical stores, questions, judge and draw count. Full uses the GPT-4o-mini configuration reported in the main tables; the ablation tables retain their same-batch controls, distinct from the main-table runs.

Removing the ledger costs 7.00 points on LongMemEval-S and 6.56 on LoCoMo, the largest module loss on both. Removing Nomination costs 5.40 and 3.40 points, respectively; removing Warrant costs 3.00 and 3.38. Removing only status tags, while keeping all rows, costs 2.13 points on LongMemEval-S. This separation locates most of the ledger's value in constructing source-linked claims, with additional value in their annotations. The readings differ in kind: the ledger's loss concentrates on knowledge update and temporal reasoning, the questions whose answer depends on a statement's state rather than on a similar sentence; the routes and the ration earn smaller but real shares, because the paraphrase carries most of the answer and the verbatim source makes it check-

	LongMemEval (500)		LoCoMo (1,540)	
Configuration	Acc. (%)	ptok	Acc. (%)	ptok
Full (default limit)	**86.00**	3,550	**89.03**	3,636
w/o Nomination (routes collapsed)	80.60	3,477	85.63	3,633
w/o Attestation (no ledger)	79.00	3,259	82.47	3,511
w/o context limit	82.07	11,144	74.78	11,248
One session	81.73	6,401	73.07	2,695
capacity 4,000 chars	83.60	3,407	85.17	3,542
capacity 2,000 chars	83.60	3,409	85.43	3,541

Table 3: Prompt tokens (ptok) and majority-vote accuracy for every configuration, at the full denominator.

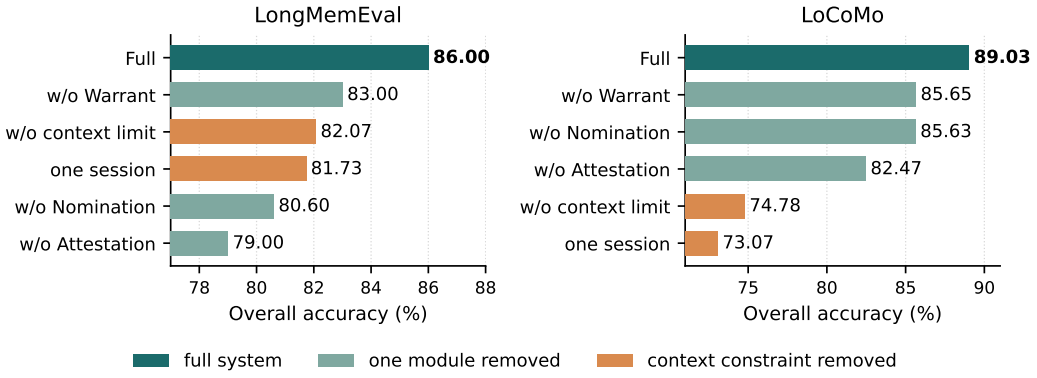


Figure 4: Overall accuracy under one-module removals and two context-limit changes. The full system is highlighted; context conditions are separated from module ablations.

able; and the context conditions move the other way, since admitting more text lowers LoCoMo accuracy — a stale window is worse than a short one (Table 3).

5.4 Case studies

Each operation fails in a characteristic way when it is removed (Figure 5). Attestation keeps the date of a ukulele lesson distinct from a later guitar repair; Nomination surfaces the most recent family trip when similarity prefers an older one; and Compilation retains all three charity amounts in a bounded context. For each item, the full system answers correctly in all three recorded draws while the paired ablation fails in all three; Section 5.3 quantifies these losses, and Appendix H traces the first item end to end.

5.5 Analysis

The context limit fails in both directions. Table 3 tests the capacity around its default; Figure 3(c) plots the full sweep. Removing the limit lowers LoCoMo from 89.03 to 74.78 as the prompt grows from 3.6k to 11.2k tokens. Keeping only one session lowers it to 73.07. Both errors are larger than removing any module, and they are not diffuse: temporal questions fall from 86.29 to 66.46 and 67.39, single-hop questions least, from 93.46 to 82.56 and 80.54. The context must contain the changed value without burying it in surrounding material. This is why we describe compilation as a contribution rather than a cost control: the priced cut improves accuracy while spending fewer tokens, and a system that renders everything it retrieves loses information to its own noise.

Two readings bound that claim. At 2,000 characters, the protected assertions and source already fill much of the window: the prompt shrinks only 4% from the 8,000-character setting while LongMemEval-S accuracy falls 2.40 points. Class protection adds 3.87 points on LongMemEval-S and 2.77 on LoCoMo. The lever is deliberately insensitive in this region: the protected classes fill the window first, so loosening the budget changes the prompt before it changes what the model can read. What matters is whether the context is curated, not how tightly.

Case 1: Event-Time Attestation

[Feb 1] **User:** "Started **ukulele lessons** today."
[Feb 25] **User:** "Taylor GS Mini to the **guitar tech** today."
LEDGER *item 4dfccbf7*
f1650 ukulele lessons start · Feb 1 · fuzzy · current
f1651 GS Mini → tech plan · Feb 1 · fuzzy · superseded
f1743 GS Mini → tech · Feb 25 · fuzzy · current
Query [Apr 1] **How many days from lessons to servicing?**
AttestMem 24 days: lessons Feb 1; servicing Feb 25.
w/o Attestation 0 days; both events dated Feb 25.

Case 2: Cross-Session Nomination

[May 27] **User:** "… 10 days **island-hopping in Hawaii**."
[May 28] **User:** "… went to **Paris** as a family last month."
LEDGER *item 9ea5eabc (dates = utterance)*
f324 family Hawaii trip, no date · May 26 · fuzzy · current
f1092 Hawaii trip, 10 days · May 27 · fuzzy · current
f1901 Paris family trip, April · May 28 · fuzzy · current
Query [Jun 8] **Which was my most recent family trip?**
AttestMem Paris — the April family trip.
w/o Nomination Hawaii — an older trip.

Case 3: Compilation Keeps Three Events in View *LongMemEval-S item 129d1232*

[May 1, 10:42] **User:** "charity walk … raised **$250**."
[May 1, 21:16] **User:** "charity yoga … raised **$600**."
[May 1, 22:24] **User:** "Bike-a-Thon … raised **$5,000**."
Query [May 1] **How much did I raise through all charity events?**
AttestMem (7.5k) **$5,850**: $5,000+$600+$250.

LEDGER
f850 charity walk --- $250 · May 1 · fuzzy · current
f1670 charity yoga --- $600 · May 1 · exact · current
f1969 Bike-a-Thon --- $5,000 · May 1 · fuzzy · current
Gold: $5,850
w/o limit (37.8k) **$5,600**; omits $250 walk.

Figure 5: Three paired LongMemEval-S cases: dialogue excerpted, answers condensed; each panel shows a representative draw, and the correct/wrong split holds in all three.

Where the ledger helps. Its removal hurts knowledge-update and temporal questions most, where the answer depends on a statement's changing state. A single-session preference is the other side: usually stated plainly, it rewards restating least, and AttestMem still reaches 96.67 on that category with GPT-4o-mini.

5.6 Further analysis: decidability is not entitlement

We also tested whether ledger fields should directly decide admission: retrieval proposes, and a rule using temporal applicability, revision currency, source support, and incremental information — each evaluated from explicit statement and admission state rather than from similarity — decides. It reaches 78.53 on LongMemEval-S against 86.00 for the priced cut at the same capacity, and renders more rows per item (67.6 versus 64.4), including material already partly represented: the predicates admit what is eligible, and the utility then favours coverage of needs the window already carries, the opposite of the priced cut's preference for material it does not yet hold. Per question type the loss concentrates on temporal reasoning and knowledge update. Computable state does not by itself make a good admission rule; Appendix G details the row composition the rejected rule renders.

6 Conclusion and limitations

AttestMem connects source-linked, revisable statements to a bounded answering context. Across LoCoMo and LongMemEval-S, its ablations assign the largest module loss to the ledger, while LoCoMo's context-limit ablations show that admitting more text can lower accuracy. The design preserves inspectable claims and spends context on what the question needs, without treating a computable field as an automatic veto.

Limitations. Ablations resolve single modules on both benchmarks against same-batch controls on the shipped store build, at the GPT-4o-mini backbone, and do not resample store construction; repeated judge calls assess a fixed answer and do not remove variation in answer generation. Stored provenance is complete, but verbatim warranting at answer time is bounded by source retrieval: only about a quarter of statement rows can be matched back to episode text, so the grade records how an assertion was matched to source, not a promise that the source is recoverable verbatim inside the warrant budget.

AI USE STATEMENT

AI coding assistants were used for language polishing and for assistance with analysis and figure code; the scope matches the declaration on the submission form. The experiments were configured, run and recorded by the authors; every quantity reported here traces to a logged run configuration, an evaluation manifest or an analysis artifact, and the case items in Figure 5 carry their item identifiers, with the gold answer shown for the third case. The authors take full responsibility for the final text, the experimental design and all conclusions.

ETHICS STATEMENT

This work studies measurement methodology for long-horizon conversational memory. It uses two public benchmarks (LoCoMo and LongMemEval-S) whose dialogues are published research data; it involves no new human-subject collection, releases no new dataset, and introduces no technology with direct harmful application. Long-conversation memory is nonetheless personal data: an append-only ledger that retains history and rejected writes serves auditability here but would have to be reconciled with deletion and consent requirements outside this evaluation setting. The primary ethical consideration is research integrity: all ablations are paired against same-batch controls on identical stores, the judge calls are recorded with their prompts and temperatures, identified quantities are distinguished from extrapolated ones throughout, and a negative result (the admission-rule evaluation in §5.6) is reported as rejected rather than retuned.

REPRODUCIBILITY STATEMENT

Section 4.3 and Appendices A–B give the complete extraction instructions, the stored-record schema and the settlement semantics; §5.1 and Appendix D pin the judging protocol to the EverMemOS release and specify the dataset-specific grading prompts; Appendices E and G give the complete published comparisons and ablation grids. Every reported number traces to a recorded run configuration, an evaluation manifest or an analysis artifact, and the case items carry their item identifiers, with the gold answer shown for the third case (Figure 5). The submission is accompanied by a package that releases the complete pipeline — store construction, answering, judging, and the aggregation scripts that produce every table from the evaluation outputs — together with the case items, available at `https://anonymous.4open.science/r/attestmem-anon-22D2`; the recorded evaluation rollouts will be released upon acceptance.

REFERENCES

Han Chen, Zining Zhang, Wenqi Pei, et al. MemForest: An efficient agent memory system with hierarchical temporal indexing, 2026. URL https://arxiv.org/abs/2605.23986.

Prateek Chhikara, Dev Khant, Saket Aryan, et al. Mem0: Building production-ready AI agents with scalable long-term memory, 2025. URL https://arxiv.org/abs/2504.19413.

Hanshuai Cui, Zhiqing Tang, Zhi Yao, et al. MemTxn: A transaction boundary for source-supported updates and complete-state recovery in agent memory, 2026. URL https://arxiv.org/abs/2607.27834.

Zijie Dai, Shiyuan Deng, Sheng Guan, Yizhou Tian, Xin Yao, Xiao Yan, and James Cheng. RecMem: Recurrence-based memory consolidation for efficient and effective long-running LLM agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 32353–32376, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1619. URL https://aclanthology.org/2026.findings-acl.1619/.

Darren Edge, Ha Trinh, Newman Cheng, et al. From local to global: A graph RAG approach to query-focused summarization, 2024. URL https://arxiv.org/abs/2404.16130.

Chuanrui Hu, Xingze Gao, Zuyi Zhou, Dannong Xu, Yi Bai, Xintong Li, Hui Zhang, Tong Li, Chong Zhang, Lidong Bing, and Yafeng Deng. EverMemOS: A self-organizing memory operating system for structured long-horizon reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 45836–45853, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.2125. URL https://aclanthology.org/2026.acl-long.2125/.

Wei-Chieh Huang, Weizhi Zhang, Yuchen Wu, et al. Harness the memory: A holistic evaluation of memory substrates in memory agents, 2026. URL https://arxiv.org/abs/2608.15008.

Eric Hanchen Jiang, Zhi Zhang, Yuchen Wu, et al. Memory as a controlled process: Learned adaptive memory management for LLM agents, 2026. URL https://arxiv.org/abs/2607.13591.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models, 2024. URL https://arxiv.org/abs/2405.14831.

Qingcan Kang, Mingyang Liu, Shixiong Kai, et al. Retain or consolidate? budget-dependent operator selection for language agent memory, 2026. URL https://arxiv.org/abs/2607.17545.

Kai Li, Xuanqing Yu, Ziyi Ni, Yi Zeng, Yao Xu, Zheqing Zhang, Xin Li, Jitao Sang, Xiaogang Duan, Xuelei Wang, Chengbao Liu, and Jie Tan. TiMem: Temporal-hierarchical memory consolidation for long-horizon conversational agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 21700–21720, San Diego, California, United States, 2026a. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1091. URL https://aclanthology.org/2026.findings-acl.1091/.

Ruizhe Li, Licheng Zhang, Benfeng Xu, et al. When your agent opens the chat app: Agent-controlled search over raw chat logs rivals structured memory, 2026b. URL https://arxiv.org/abs/2608.12888.

Nelson F. Liu, Kevin Lin, John Hewitt, et al. Lost in the middle: How language models use long contexts, 2023. URL https://arxiv.org/abs/2307.03172.

YuFei Luo, Xiucheng Xu, and Zhen Yang. MemSIF: From structured interactions to dual-track fact memory for LLM agents, 2026. URL https://arxiv.org/abs/2608.01742.

Enze Ma, Yufan Zhou, Wei-Chieh Huang, et al. MemAudit: Auditing long-term agent memory via hidden user-state recovery, 2026. URL https://arxiv.org/abs/2606.24595.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, et al. Evaluating very long-term conversational memory of LLM agents, 2024. URL https://arxiv.org/abs/2402.17753.

Charles Packer, Sarah Wooders, Kevin Lin, et al. MemGPT: Towards LLMs as operating systems, 2023. URL https://arxiv.org/abs/2310.08560.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, et al. Zep: A temporal knowledge graph architecture for agent memory, 2025. URL https://arxiv.org/abs/2501.13956.

Parth Sarthi, Salman Abdullah, Aditi Tuli, et al. RAPTOR: Recursive abstractive processing for tree-organized retrieval, 2024. URL https://arxiv.org/abs/2401.18059.

Maksim Sheverev, David Finkelstein, and Sergey Nikolenko. Beyond memory leaderboards: Evaluating scientific memory as budgeted context restoration, 2026. URL https://arxiv.org/abs/2607.16848.

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

Xiaolong Sun, Qichao Wang, Hangyu Li, et al. Verifiable memory: Learning unified memory management with local and global verifiers for large language model agents, 2026. URL https://arxiv.org/abs/2608.03137.

Zihao Tang, Xin Yu, Ziyu Xiao, Zengxuan Wen, Zelin Li, Jiaxi Zhou, Hualei Wang, Haohua Wang, Haizhen Huang, Weiwei Deng, Feng Sun, and Qi Zhang. Mnemis: Dual-route retrieval on hierarchical graphs for long-term LLM memory. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23914–23928, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1096. URL https://aclanthology.org/2026.acl-long.1096/.

Liang Wang, Haonan Chen, Nan Yang, et al. Chain-of-retrieval augmented generation, 2025a. URL https://arxiv.org/abs/2501.14342.

Xiaoqiang Wang, Suyuchen Wang, Yun Zhu, and Bang Liu. R^{3}Mem: Bridging memory retention and retrieval via reversible compression. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 4541–4557, Vienna, Austria, 2025b. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.235. URL https://aclanthology.org/2025.findings-acl.235/.

Di Wu, Hongwei Wang, Wenhao Yu, et al. LongMemEval: Benchmarking chat assistants on long-term interactive memory, 2024. URL https://arxiv.org/abs/2410.10813.

Wujiang Xu, Zujie Liang, Kai Mei, et al. A-MEM: Agentic memory for LLM agents, 2025. URL https://arxiv.org/abs/2502.12110.

Tianyu Yang, Sudipta Paul, Vijay Srinivasan, et al. TRUSTMEM: Learning trustworthy memory consolidation for LLM agents with long-term memory, 2026. URL https://arxiv.org/abs/2606.25161.

Yanyu Yao, Shangze Li, Zhi Zheng, et al. AtomMem: Building simple and effective memory system for LLM agents via atomic facts, 2026. URL https://arxiv.org/abs/2606.19847.

Juwei Yue, Chuanrui Hu, Jiawei Sheng, Zuyi Zhou, Wenyuan Zhang, Tingwen Liu, Li Guo, and Yafeng Deng. HyperMem: Hypergraph memory for long-term conversations. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 35237–35254, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1627. URL https://aclanthology.org/2026.acl-long.1627/.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19724–19731, 2024. doi: 10.1609/aaai.v38i17.29946. URL https://ojs.aaai.org/index.php/AAAI/article/view/29946.

Yan Zhou, Yue Ouyang, Kaiyang Zheng, et al. TEPA: Revoking stale memories for conflict-robust language agents, 2026. URL https://arxiv.org/abs/2608.07429.

A ATTESTATION: THE FULL INSTRUCTION SET

The extraction instruction is a frozen string; the appositive clauses below are the ones that carry mechanism, not decoration. The model is told to process the messages strictly one at a time and in order, and to decide for each message whether it contains factual information, extracting it as standalone sentence(s) and never batch-skipping. It must not skip anything for looking minor, because the questions this memory later answers concern past events, current states, future plans, intentions, thoughts, opinions, attitudes, wants, hopes and preferences alike. It must preserve all specific details — full names with context, complete locations, specific event names, item details, numbers and quantities, organisation names — because a paraphrase that drops a proper noun is precisely what a later reader cannot verify. It must distinguish mention time from event time: the date on each line is when something was *said*, and every event fact must additionally end with an event tag giving the actual event time resolved against that line's date, with the speaker's own relative phrase retained

alongside the resolved value. Ongoing or timeless facts carry no event tag. Output is one fact per line, unnumbered, not JSON.

Two further instructions are less obvious. After writing the facts the model re-reads the fragment and lists any information containing numbers, dates, places, negations or comparisons that it missed, extracting those too, or emits a coverage-OK marker. And after each fact it appends one tab-separated sample question that the fact alone would answer, phrased the way a person *asking* would phrase it rather than the way the fact states it, kept under twelve words, with no tab if the fact is too vague to support a question. That question is what lets a statement be retrieved by the shape of the question it answers rather than by lexical overlap with its own wording; the extraction unit is tuned so that this is cheap enough to do for every statement.

The pass-1 instruction, verbatim. Salient events and states, one per line, each with an event tag and an index question:

```
You are a Personal Information Extractor.
Extract **all possible facts or information** about the speakers from this
conversation fragment. Messages are chronological, one per line:
[date] index.Speaker: message

Rules:
0. Process messages strictly one by one, in order. For each message decide
   whether it contains factual information; if yes, extract it as standalone
   sentence(s). Never batch-skip.
1. Do NOT skip anything because it looks minor or trivial. Extract ALL
   meaningful information: past events, current states, future plans,
   intentions, thoughts, opinions, attitudes, wants, hopes, preferences.
2. PRESERVE ALL SPECIFIC DETAILS: full names with context ("The Name of the
   Wind" by Patrick Rothfuss, not "a book"), complete locations (Galway,
   Ireland), specific event names (benefit basketball game), item details
   (vintage camera), numbers and quantities (4 years ago, last week),
   company/organization names.
3. Light contextual completion so each fact is a clear standalone statement.
4. TIME RULE: distinguish mention time (the [date] on each line = when it was
   SAID) from event time (when it happened/will happen). Every event fact must
   end with an event tag "(event: <DD Month YYYY>)" | the actual event time
   resolved against the line's [date], keeping the relative phrase too.
   Ongoing/timeless facts: no event tag.
5. Output: one fact per line, no numbering, no JSON.
6. COVERAGE AUDIT: after writing the facts, re-read the fragment and list any
   information containing numbers, dates, places, negations (didn't/cancelled/
   gave up) or comparisons that you missed | extract those too. If nothing is
   missing output COVERAGE_OK on its own line and continue with the facts.
7. QUERY PATTERN: after each fact line, append ONE tab-separated sample
   question that this fact alone would answer, phrased the way a person
   ASKING would phrase it (not the way the fact states it). Format per line:
   <fact sentence>\t<sample question>
   The question must be answerable from THAT fact alone. Keep it under 12
   words. No tab if the fact is too vague for a question.

Conversation fragment (all said around {date}):
{episode}
```

The pass-2 instruction, verbatim. The second pass re-reads the fragment for the five kinds of content pass 1 tends to drop:

```
You are a gap-finding editor. Below are a conversation
fragment and the facts already extracted from it. Find facts the list MISSED,
but ONLY these five kinds:
(a) numbers/amounts/durations  (b) dates/time spans
```

```
(c) negations and cancellations (didn't / cancelled / gave up / refused)
(d) comparisons and superlatives (more / best / first / last)
(e) ITEMS IN FLIGHT | anything the user still has to pick up, collect, return,
    drop off, exchange, repair, collect from a person, lend back, borrow, or is
    waiting to receive. Write the ITEM and, when stated, the shop/person and the
    date, e.g. "The user still needs to return the boots to Zara (bought 2023-02-05)"
    or "The user lent her green sweater to her sister".
    This kind exists because counting questions ask how many things are in flight
    and the answer is lost when only "I own X" statements are extracted: measured,
    22.4% of the gold items of counting questions were absent from the store.
Output in the exact same one-fact-per-line format as the list (with event tags
where applicable). If nothing is missing output NONE.

[Already extracted]:
{pass1_facts}

[Conversation fragment]:
{episode}
```

The pass-3 instruction, verbatim. The detail sweep runs per unit against the facts already extracted:

```
You are a detail-completion editor. Below are a conversation
fragment and facts already extracted from it. List every CONCRETE detail the
list MISSED | one fact per line, each a complete standalone statement:
- physical items/objects (what exactly was bought, made, given, found)
- named works, places, brands, people, pets
- likes/dislikes/preferences (who prefers what)
- plans, goals, wishes, future intentions
- quantities attached to items (three paintings, two dogs)
Do NOT repeat facts already in the list. Do NOT invent details. Use the
one-fact-per-line format of the list (with event tags where applicable).
If nothing is missing output NONE.

[Already extracted]:
{pass1_facts}

[Conversation fragment]:
{episode}
```

The cross-cell omission audit, verbatim. When a segment was extracted in several cells, one further call asks for what only the whole segment supports:

```
You are auditing a memory extraction for OMISSIONS.

The segment below was extracted in {n} consecutive pieces (cells). Each cell was
read on its own, so facts that need the WHOLE segment were likely missed.

SEGMENT (date-tagged turns, in order):
{segment}

FACTS ALREADY EXTRACTED from this segment's cells:
{facts}

Task: list ONLY the facts that a cell-by-cell reading could NOT have produced:
- statements that refer back across cells ("the same one as before", "that
  place", "it also...") and need the earlier cell to be understood;
- temporal relations between turns in different cells (order, "the day after",
  "since then", "three weeks later");
```

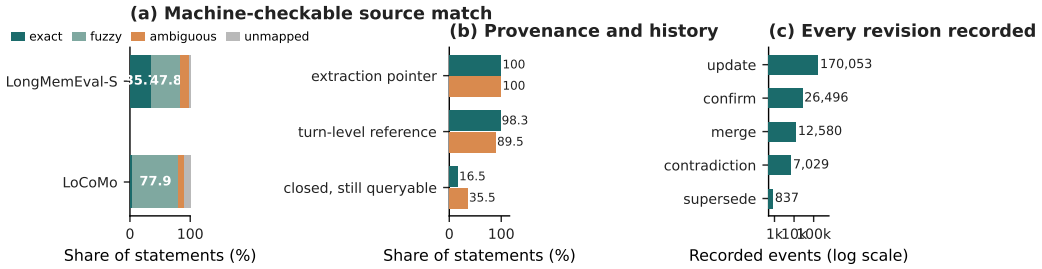


Figure 6: The recorded artefact behind the main tables (1,098,282 statements). (a) Source-match grades per benchmark; (b) provenance and history coverage (teal: LongMemEval-S; orange: LoCoMo); (c) the recorded revision stream, none erased.

```
- a value that is revised later in the segment, where the revision only makes
  sense with the earlier statement in view;
- a summary-level statement the segment supports but no single cell states.

Rules:
- Do NOT repeat any fact already listed above, and do not rephrase them.
- One fact per line, a complete standalone sentence with its date if the turn
  has one.
- If nothing is missing, output exactly: NONE
Output ONLY the lines.
```

B THE STORED RECORD

A row carries, in order of when it is written: the statement text; the resolved event time and the raw utterance date; the episode identifier and the session-chunk provenance it was extracted from; the validity interval; the status; the identifiers of its predecessor and successor; the attestation grade and the turn identifiers that grade was derived from; the resolved keywords used for indexing; the record of any evidence merged into the row; the recorded reason for a deprecation; and usage counters. Consolidation adds rather than replaces: when a row is superseded, its interval is closed and its successor recorded, but the row, its references and its grade remain queryable. The ledger is a separate append-only event stream in which every accepted operation stores both statements, the slot and the two values it concerns, the reason, the source identifiers and a timestamp; operations the validator refused are kept in a parallel rejected list rather than discarded. On the recorded artefact this gives every row a provenance pointer, with the grade resolved at turn level where the match allowed it and at chunk level otherwise, and a minority of rows closed and still queryable.

The recorded artefact. The main-table builds leave a measurable artefact: 500 per-question stores for LongMemEval-S and 10 per-conversation stores for LoCoMo, 1,098,282 statement rows in total. Every row carries an extraction pointer, 98.3% carry turn-level references, and the grade mix is `exact` 35.5%, `fuzzy` 48.0%, `ambiguous` 14.7%, `unmapped` 1.8% — the unmapped share matches the rows without turn references. The mix is benchmark-dependent: LoCoMo's shorter conversations grade 77.9% `fuzzy` and 3.2% `exact`, against 47.8% and 35.7% on LongMemEval-S. The event stream records 216,998 non-add operations — 170,053 updates, 26,496 confirmations, 12,580 merges, 7,029 contradictions, 837 supersessions and 3 backdates — and 16.7% of rows are closed and remain queryable. Figure 6 visualises the aggregate.

Recorded settlement events, one per operation. The reasons and slots below are verbatim from the conv-26 and 4dfccbf7 ledgers (the last from conv-42): CONFIRM f139→f139, reason "re-statement merged into existing fact" (slot *conference*, 2023-07-12); UPDATE f1537→f1650, reason "The new statement updates the old by specifying the date the user started ukulele lessons" (slot *lessons*, 2023-02-01); MERGE f116→f140, reason "Both statements express Caroline's desire to help others using her story" (slot *caroline*, 2023-07-03); CONTRADICTION f562→f577, reason "The old statement is about empowerment through art, while the new one focuses on the importance of art during transition" (2023-09-13); SUPERSEDE (conv-42) with reason "The new statement

provides a more complete version of Nate's." DISTINCT and BACKDATE events are recorded but vanishingly rare (3 backdates in 216,998 non-add events).

Per-conversation statistics. Table 4 gives the ten LoCoMo store builds. Non-current rows keep their grade, references and closed interval.

Conversation	Statement rows	Non-current	Ledger events
conv-26	725	265	775
conv-30	576	251	596
conv-41	1,022	373	1,094
conv-42	993	354	1,054
conv-43	1,101	334	1,147
conv-44	1,158	489	1,236
conv-47	1,069	298	1,140
conv-48	999	337	1,021
conv-49	824	292	881
conv-50	1,033	384	1,106
Total	9,500	3,377	10,049

Table 4: The ten LoCoMo store builds behind the main tables. Non-current rows are superseded or otherwise closed and remain queryable.

C BOUNDED READINGS OF THE CONTEXT LIMIT

The capacity ladder. Table 5 gives the readings on both sides of the default capacity.

Nominal capacity	**LongMemEval Acc.**	**ptok**	**LoCoMo Acc.**
2,000 chars	83.60	3,409	85.43
4,000 chars	83.60	3,407	85.17
8,000 chars (default)	**86.00**	3,550	**89.03**
no limit	82.07	11,144	74.78
One session (minimum context the pipeline can produce)			
one session	81.73	6,401	73.07

Table 5: The capacity ladder: majority-vote judge accuracy and rendered prompt tokens (ptok) on both sides of the default 8,000-character capacity.

Lowering the nominal capacity from the default value to 4,000 and then 2,000 characters moves the rendered string by about four percent and reduces LongMemEval-S accuracy by 2.40 points. The reason is that assertions and recovered source are protected: the cut stops at what is left of the transcript, so beyond a certain point the target is simply not binding. The default value is the first point above that floor, which is why we do not present it as a tuned optimum and why the main text makes no claim about the optimal capacity. The full seven-value sweep on LoCoMo against rendered-prompt usage is plotted in Figure 3(c).

Class protection at a fixed capacity. With the capacity held at its default value and only the class rules removed — every rendered line competing on one relevance score, the twin-collapse and entity-block rules disabled with them — LongMemEval-S falls from 86.00 to **82.13** while the prompt gets *shorter*, 3,550 to 3,310 tokens. On LoCoMo the same substitution moves 89.03 to 86.26. The difference in effect size reflects the benchmarks rather than the mechanism: the LongMemEval-S transcripts are roughly thirty times larger than the LoCoMo ones, so protected content fills the window there and does not here.

D PROTOCOL NOTES

One conversation, one arm. With the capacity limit removed, the prompts for one LoCoMo conversation grow to about fourteen thousand characters. Two of its items then exceed the answering

model's generation ceiling even at a raised limit, and re-running the conversation through the reranking service triggered sustained rate limiting that made the pass disproportionately slow. Those items were therefore answered with the fused candidate order and the reranker disabled, applied identically to the affected arms and to their control, so the comparison remains like-for-like; the alternative would have been to drop the conversation and report the arms on a smaller denominator.

Judge protocol. We use the GPT-4o-mini configuration of the official EverMemOS evaluation code, pinned to commit `462ebf9fd59b`: `benchmarks/config_hybrid_judge4o.toml` sets three judge calls and temperature zero, and `benchmarks/run.py` implements majority voting on a fixed answer. For verdicts $j_{ir} \in \{0, 1\}$ on question i, the reported statistic is

$$c_i = \mathbf{1}\left[\sum_{r=1}^{3} j_{ir} \geq 2\right], \qquad \text{Accuracy} = \frac{100}{N}\sum_{i=1}^{N} c_i.$$

Here $N = 1{,}540$ for LoCoMo and $N = 500$ for LongMemEval-S. This is a question-weighted accuracy, not a macro-average over categories or a mean of the three judge labels. Independent answer runs, if reported, are aggregated separately after each answer's judge vote.

Grading prompts. The benchmark adapters `benchmarks/adapters/locomo.py` and `longmemeval.py` define the grading rules. LoCoMo uses the source's generous semantic-matching prompt, including equivalent date expressions. LongMemEval-S uses its category-aware hybrid prompt: unanswerable questions accept an appropriate abstention; temporal duration answers allow the source's one-unit tolerance; knowledge-update answers may include an older value if the updated value is present; preference answers are judged by correct use of the user's preference rather than exhaustive rubric coverage. The source code allows five judge attempts per call. LoCoMo keeps exhausted failures as wrong; the LongMemEval-S adapter excludes unreadable judge rows. A full-denominator result therefore requires all LongMemEval-S verdicts to be resolved, with any unresolved count reported explicitly.

Published-result provenance. All AttestMem scores use the judge protocol above. The three-judge averaging protocol described in the EverMemOS paper and RecMem's average over three runs remain properties of their cited results. Baseline values are reproduced directly from those publications with their source identified in the table captions.

Paired controls. Every ablation is paired against a same-batch control on identical stores, question set, judge and draw count. Per-item flips are tested with an exact two-sided McNemar test. A flip that occurs on a byte-identical prompt is a re-roll rather than a mechanism effect, and such flips are counted and reported separately; no cell in this paper credits a flip of that kind.

The LoCoMo grading prompt, verbatim. The judge template is:

```
Your task is to label an answer to a question as 'CORRECT' or 'WRONG'. You will be gi
    (1) a question (posed by one user to another user),
    (2) a 'gold' (ground truth) answer,
    (3) a generated answer
which you will score as CORRECT/WRONG.

The point of the question is to ask about something one user should know about the ot
The gold answer will usually be a concise and short answer that includes the referenc
Question: Do you remember what I got the last time I went to Hawaii?
Gold answer: A shell necklace
The generated answer might be much longer, but you should be generous with your gradi

For time related questions, the gold answer will be a specific date, month, year, etc

Now it's time for the real question:
Question: {question}
Gold answer: {gold}
```

```
Generated answer: {answer}

First, provide a short (one sentence) explanation of your reasoning, then finish with
Do NOT include both CORRECT and WRONG in your response, or it will break the evaluati

Just return the label CORRECT or WRONG in a json format with the key as "label".
```

LongMemEval-S judging runs through the official adapter's category-aware hybrid prompt, whose per-category rules are summarised above; the released `judge_lme.py` delegates to that adapter so that parsing and retry behaviour match.

E THE COMPLETE PUBLISHED COMPARISON

Backbone	Method	Tokens	SH	MH	T	OD	Overall
4o-mini	Full Context	31.5k	90.01	69.86	52.96	55.21	76.43
4o-mini	Naive RAG	6.2k	57.55	54.61	24.30	51.04	49.68
4o-mini	MemoryOS	4.9k	74.55	58.87	44.24	46.88	63.64
4o-mini	Mem0	2.0k	65.52	52.84	52.02	37.50	58.64
4o-mini	A-Mem	3.0k	66.83	52.13	61.99	30.21	60.84
4o-mini	RecMem	2.7k	79.79	72.70	57.32	58.33	72.47
4o-mini	MemU	4.0k	72.77	62.41	33.96	46.88	61.15
4o-mini	MemOS	2.5k	81.45	69.15	72.27	60.42	75.87
4o-mini	Zep	1.4k	88.11	71.99	74.45	66.67	81.06
4o-mini	EverMemOS	2.5k	91.08	86.17	81.93	66.67	86.76
4.1-mini	MemoryOS	5.5k	67.30	59.34	42.26	59.03	60.11
4.1-mini	Mem0	1.0k	68.97	61.70	58.26	50.00	64.20
4.1-mini	MemU	4.0k	74.91	72.34	43.61	54.17	66.67
4.1-mini	MemOS	2.5k	85.37	79.43	75.08	64.58	80.76
4.1-mini	Zep	1.4k	90.84	81.91	77.26	75.00	85.22
4.1-mini	EverMemOS	2.3k	96.67	91.84	89.72	76.04	93.05

Table 6: Published LoCoMo results, grouped by backbone: the first six `gpt-4o-mini` rows from RecMem's Table 1, the remaining rows from EverMemOS's Table 1.

In the main tables, a baseline row quotes EverMemOS's figure wherever EverMemOS's tables report that system under that backbone — its evaluation code is the protocol we adopt unchanged — and quotes RecMem's tables otherwise, the only other source for Full Context, Naive RAG, A-Mem, RecMem itself, and the whole GPT-4o-mini block of Table 2. This appendix reproduces each source's complete table: in the LoCoMo comparison, the first six `gpt-4o-mini` rows (Full Context through RecMem) come from RecMem's Table 1 (Dai et al., 2026); all remaining rows come from EverMemOS's Table 1 (Hu et al., 2026). In the LongMemEval-S comparison, the `gpt-4o-mini` block is RecMem's Table 2 and the `gpt-4.1-mini` block is EverMemOS's Table 2. Values retain the source papers' scoring and token definitions; RecMem's scores are averages over three runs, and EverMemOS's LongMemEval-S baseline rows are themselves sourced from the official MemOS results. Differing values for one method name are different published measurements under different protocols, not repeated runs of one configuration.

Backbone	Method	Tok.	SSU	SSA	SSP	MS	KU	TR	Overall
4o-mini	Full Context	112.3k	44.29	80.36	53.33	35.34	55.13	34.59	45.60
4o-mini	Naive RAG	11.9k	85.71	83.93	46.67	52.63	65.38	41.35	59.40
4o-mini	A-Mem	13.1k	92.54	98.21	36.67	53.38	70.51	44.36	63.80
4o-mini	Mem0	1.9k	95.71	55.36	70.00	57.89	69.23	54.14	64.00
4o-mini	MemoryOS	8.9k	97.14	89.29	70.00	58.65	73.08	48.87	67.80
4o-mini	RecMem	5.6k	91.43	85.71	46.67	56.39	80.77	61.65	69.20
4.1-mini	MemU	0.5k	67.14	19.64	76.67	42.10	41.02	17.29	38.40
4.1-mini	Zep	1.6k	92.90	75.00	53.30	47.40	74.40	54.10	63.80
4.1-mini	Mem0	1.1k	82.86	26.78	90.00	63.15	66.67	72.18	66.40
4.1-mini	MemOS	1.4k	95.71	67.86	96.67	70.67	74.26	77.44	77.80
4.1-mini	EverMemOS	2.8k	97.14	85.71	93.33	73.68	89.74	77.44	83.00

Table 7: Published LongMemEval-S results, grouped by backbone: the `gpt-4o-mini` block from RecMem's Table 2, the `gpt-4.1-mini` block from EverMemOS's Table 2.

F REFUSAL ON THE ADVERSARIAL ITEMS

Method	Refusals	Precision	Recall	F1
Mem0	90	40.0	8.1	13.5
Graphiti	954	38.9	83.6	53.1
REMem	344	73.3	56.8	64.0
AttestMem (ours)	338	**83.1**	**63.0**	**71.7**

Table 8: Refusal on the 446 adversarial LoCoMo items under the protocol of REMem (Shu et al., 2026): every system answers all 1,986 questions under an explicit instruction to output "no information available" when the evidence lacks the answer, and an output containing the instructed phrase counts as a refusal. Precision is the share of refusals falling on the 446 truly unanswerable items; recall is the share of those 446 the system refuses. Mem0, Graphiti, and REMem rows are quoted from REMem's Table 6; the AttestMem row is our run with the same answering model, system prompt, and refusal rule over our frozen evidence.

The main tables exclude LoCoMo's 446 adversarial items for denominator comparability with the published rows; Table 8 evaluates exactly these items for refusal. The protocol is REMem's (Shu et al., 2026): every system answers all 1,986 LoCoMo questions under an explicit instruction to output "no information available" when the evidence does not contain the answer, and an output containing the instructed phrase counts as a refusal (REMem's scoring rule, applied verbatim). Precision is the share of refusals that fall on the 446 truly unanswerable items; recall is the share of those 446 that the system refuses.

Our row re-answers all 1,986 questions with the same answering model as REMem's comparison (`gpt-4.1-mini-2025-04-14`), REMem's system prompt, answer parsing, and refusal rule, over AttestMem's frozen evidence — the retrieval recorded by the first draw of the main configuration — in a single draw, matching REMem's single-run convention. AttestMem refuses 338 times, 281 of them on the adversarial items; the remaining 57 refusals are 3.7% of the 1,540 answerable items. It attains the comparison's highest refusal precision and F1.

G COMPLETE ABLATION TABLES

Table 9 gives the LongMemEval-S arms in full.

Configuration	What it changes	Accuracy	ptok
Full (default)	control	**86.00**	3,550
w/o Attestation	no ledger, raw transcript only	79.00	3,259
w/o Nomination	retrieval routes collapsed	80.60	3,477
w/o context limit	class-budget cap raised to 10M	82.07	11,144
one session	single session of context	81.73	6,401
w/o Warrant	no verbatim source recovered	83.00	3,489
w/o layered entitlement	one flat relevance score	82.13	3,310
w/o priced budget	capacity present, no class pricing	82.53	11,395
w/o status tags	status field dropped, rows kept	83.87	3,506
w/o statement dates and tags	dates and status dropped, rows kept	82.60	3,509
capacity 4,000 chars	nominal capacity lowered	83.60	3,407
capacity 2,000 chars	nominal capacity lowered further	83.60	3,409
L1 + L3 only	warrant layer dropped entirely	82.73	9,948

Table 9: LongMemEval ablations, full denominator, majority vote over three `gpt-4o-mini` judgments; Full is the control.

Table 10 gives the LoCoMo arms.

Configuration	What it changes	SH	MH	T	OD	Overall
Full (default)	control	93.46	85.46	86.29	67.71	**89.03**
w/o Attestation	no ledger, raw transcript only	88.94	68.91	82.24	66.32	82.47
w/o Nomination	retrieval routes collapsed	90.73	78.25	83.90	68.40	85.63
w/o context limit	class-budget cap raised to 10M	82.56	68.56	66.46	52.78	74.78
one session	single session of context	80.54	65.84	67.39	47.92	73.07
w/o Warrant	no verbatim source recovered	89.81	80.38	84.84	67.36	85.65
w/o status tags	status field dropped, rows kept	90.92	79.43	85.25	71.53	86.43
w/o statement dates, tags	dates and status dropped, rows kept	90.01	80.02	82.76	69.10	85.37
w/o layered entitlement	one flat relevance score	90.57	80.14	86.09	67.01	86.26
w/o priced budget	capacity present, no class pricing	92.23	83.22	85.46	68.06	87.66
L1 + L3 only	warrant layer dropped entirely	89.73	80.50	84.53	68.06	85.61

Table 10: LoCoMo ablations, full denominator (1,540 questions), same protocol; Full is the control.

The ordering that fails, in detail. Promoting the ledger's state into admission predicates and selecting greedily under the capacity renders *more* rows per item than the full system (67.6 against 64.4) with a slightly higher share of non-current rows, so its loss is not a matter of deleting history.

Recorded items. Verbatim from the run. Category-5 items the system refuses: "What did Caroline realize after her charity race?", answered "no information available" (representative of the 281 refusals on truly unanswerable items). Category-5 items it fails to refuse: "What type of individuals does the adoption agency Melanie is considering support?", answered "The adoption agency Melanie is considering supports LGBTQ+ individuals." — the retrieved rows describe the *friend's* agency, and the answering model attributes them to Melanie's. Over-refusals on answerable items: "When did Caroline and Melanie go to a pride fesetival together?"' (single-hop; the compiled evidence did not contain the visit) and "Would Melanie go on another roadtrip soon?"' (temporal). An answerable item answered correctly: "When did Caroline go to the LGBTQ support group?"' → "Caroline went to the LGBTQ support group on 7 May 2023."'.

Per-category refusals. Table 11 splits the 338 refusals by category. Over-refusal concentrates on temporal questions (14.6%), where date arithmetic against a closed validity interval can look unsupported; the single-hop and open-domain rates stay below 4%.

H AN END-TO-END RECORDED TRACE

This appendix walks one recorded run end to end: LongMemEval-S item `4dfccbf7`, the item behind the first panel of Figure 5, first of its three draws, answered by the full GPT-4o-mini configuration. Everything quoted below is verbatim from the stored artefacts: the built store, the answer record with its retrieval and compilation traces, and the judge verdicts.

Category	Questions	Refusals	Share
1 multi-hop	282	11	3.9%
2 single-hop	321	9	2.8%
3 temporal	96	14	14.6%
4 open-domain	841	23	2.7%
5 adversarial	446	281	63.0%

Table 11: Refusals by LoCoMo category under the REMem protocol (Table 8).

Dialogue to statements. In the source dialogue, turn 35 (1 February 2023) contains "I've just started taking ukulele lessons with my friend Rachel today (01 February 2023) and it's been really fun so far", alongside a plan to take the guitar to the tech by month's end. The extractor writes, among others, the statements f1651 (turn 35#t8, `fuzzy`): "The user plans to take their Taylor GS Mini to Joe at Guitar Central by the end of this month (February 2023)"; and f1650 (turn 35#t4, `fuzzy`): "The user just started taking ukulele lessons today (2023-02-01)". In turn 38 (25 February 2023) the plan becomes a decision — "I decided to take my Taylor GS Mini to the guitar tech for servicing today (25 February 2023)" — from which the extractor writes f1743 (turn 38#t8, `fuzzy`): "The user decided to take their Taylor GS Mini to the guitar tech for servicing today (25 February 2023)".

Settlement. The ledger records two UPDATE events. On 2023-02-01, f1537 (the Rachel sentence) is updated to f1650 with the recorded reason "The new statement updates the old by specifying the date the user started ukulele lessons" (slot *lessons*). On 2023-02-25, f1651 is updated to f1743 with the reason "The new statement updates the old by indicating that the user decided to take their Taylor GS Mini to the guitar tech for servicing today" (slot *february*; recorded old value "2023", new value "25"). f1651's validity interval closes at 2023-02-26; the row and its grade remain queryable, and nothing is deleted.

Nomination. At answer time (1 April 2023) the question is decomposed into the verbatim needs "Date when ukulele lessons started", "Date when acoustic guitar was taken to guitar tech for servicing", and "Calculation of the number of days between the two dates". Retrieval returns 108 candidate rows: the decision rows f1738 and f1743 rank first and second, the lesson row f1650 ranks third, and the superseded plan f1651 is retrieved fifth — nomination proposes without filtering by ledger state.

Compilation. The compiler renders 7,540 characters (the "(7.5k)" of Figure 5). Its first lines are:

```
- [2023-02-25] I decided to take my Taylor GS Mini to the guitar tech for servicing today (25
February 2023).
- [2023-02-25] The user decided to take their Taylor GS Mini to the guitar tech for servicing today
(25 February 2023).
- [2023-02-01] The user just started taking ukulele lessons today (2023-02-01).
- [2023-02-01] [superseded until 2023-02-02] I've just started taking ukulele lessons with my friend
Rachel today (01 February 2023) and it's been really fun so far.
- [2023-02-01] [superseded until 2023-02-26] The user plans to take their Taylor GS Mini to Joe at
Guitar Central by the end of this month (February 2023).
```

Both target dates are present, the superseded rows are rendered with their closed intervals rather than deleted, and they sit inside the protected classes the transcript cut cannot touch.

Answer and judging. The model answers: "24 days had passed since the user started taking ukulele lessons on February 1, 2023, when they decided to take their Taylor GS Mini to the guitar tech for servicing on February 25, 2023." All three judge calls return CORRECT, and all three draws give 24 days. The paired w/o-attestation draw, on the same question and the same store, answers "0 days had passed since the user started taking ukulele lessons on February 25, 2023, when they decided to take their Taylor GS Mini to the guitar tech for servicing on the same day" and is judged WRONG in all three draws: with attestation removed, both events are dated to the decision turn.

Trace 2: cross-session nomination (9ea5eabc). The question "Where did I go on my most recent family trip?" (asked 8 June 2023) is answered from the same 87-row candidate set and the same

7,999-character compiled context under the full system and under w/o Nomination; the difference is which rows the routes surface. The full system's context carries the April Paris row f1901 (turn 45, `fuzzy`) beside the two Hawaii rows, and answers: "The user went to Paris with family in April 2023 on their most recent family trip." The ablation answers: "The user went on their most recent family trip to Hawaii, where they had a great time snorkeling together." and is judged WRONG in all three draws.

Trace 3: compilation under capacity (129d1232). The question "How much money did I raise in total through all the charity events?" decomposes into the verbatim needs "Total amount of money raised", "List of charity events participated in", and "Amount raised in each charity event" — identically under both conditions, and retrieval returns the same 108 candidate rows. Compilation is where the conditions part: the full system renders 7,503 characters carrying all three charity statements, while w/o context limit renders 37,817 characters. The full system answers: "The user raised a total of $5,850 throughallthecharityeventstheyparticipatedin$:5,000 from the Bike-a-Thon for Cancer Research, $600 fromthecharityyogaeventforalocalanimalshelter, and$250 from the charity walk." The ablation answers: "You raised a total of $5,600 throughallthecharityeventsyouparticipatedin, whichincludes$5,000 from the Bike-a-Thon for Cancer Research and $600 fromthecharityyogaeventforalocalanimalshelter'' --- the$250 walk is dropped from an overfull window.