What Remains After Retrieval Saturates: Measuring the Delivery Gap in AI Memory Systems with an Open Evaluation Protocol for Tablet 1 and Scroll 1

Kim Sunwoo, Wontopos Contact: sunwoo.ceo@wontopos.com · wontopos.com/model/tablet-1 · wontopos.com/model/scroll-1

July 7, 2026 · Preprint

Abstract

Benchmark scores for AI memory systems are hard to compare. Systems differ in reader model, grading method, prompts, and reporting practice, so even numbers on the same benchmark measure different things. We evaluate two commercial memory models under a single fixed, fully published protocol on the full LongMemEval-S set (500 questions), five independent runs each, with every run published: **Tablet 1**, which delivers a minimal single-pass context, and **Scroll 1**, which expands the same retrieval hits with their session neighbors for a fuller delivery. Neither has an LLM in the retrieval path. Tablet 1 scores a mean of **85.2%** (σ=1.1; reader: Claude Opus 4.8); Scroll 1 scores **90.7%** (σ=0.5; reader: GPT-5.5), with no best-of selection of any kind and no self-grading; answers are graded by an independent model, GPT-4o. The central finding is that the difference between the two scores does not come from retrieval. The engine's session-level retrieval recall is saturated at **99.6%** regardless of top-k configuration, and both models find the same evidence. What separates them is how much of the found memory actually reaches the model, which we call the **delivery gap**. Tablet 1, which hands the reader about 1,200 tokens per query, leaves a gap of 14.4 points below the retrieval ceiling; Scroll 1, at about 3,700 tokens, leaves 8.9 points. The two configurations also differ in reader model; a development lineage that enabled the delivery components one at a time is consistent with most of the gain coming from delivery mechanics, but the lineage is single-run, so the definitive decomposition is left to a single-reader rerun (§ 6, § 7). The results suggest that the bottleneck in long-term memory has moved from finding to delivering. We publish the full protocol, the verbatim reader and judge prompts, and the raw per-run, per-category scores; every subsequent model will be measured against these two baselines under the same protocol.

---

1. Introduction

Long-term memory for conversational AI splits into two stages: **finding** the right piece in an accumulated record (retrieval), and **delivering** the found piece so the model can actually use it. Competition and reporting in this field have concentrated on the first stage, and leaderboard numbers are commonly read as proxies for retrieval quality.

This paper reports two things. First, an **evaluation protocol** that fixes the reader, judge, prompts, and number of repetitions, and publishes every run. As § 2 shows, the same memory system can swing by more than ten points depending on the reader model, and retrieval-based scores are cited alongside end-to-end QA scores as if they were the same metric. Our first claim is that comparison without a fixed protocol is meaningless.

Second, the **delivery gap** measured under that protocol. Our engine's retrieval recall is 99.6%, effectively at the ceiling. Yet end-to-end answer accuracy, given that same evidence, sits at 85.2% (Tablet 1) and 90.7% (Scroll 1). The memory is right there, and the model still does not always use it. Most of this gap cannot be closed by improving retrieval (§ 6.1 counts the deliberate residual retrieval failures separately); it is a question of what is handed over, how much, and in what shape: delivery. Because Tablet 1 and Scroll 1 share identical retrieval and differ in delivery design, they form a comparatively controlled setting in which this gap can be observed.

We make no leaderboard claims. As § 2 documents, published numbers for other systems come from incomparable setups.

The contributions of this paper are:

1. **An open evaluation protocol**: reader, judge, prompts, and repetition count fixed, with verbatim prompts and raw scores fully published, re-runnable by third parties at the black-box boundary (§ 4, § 8). There is no self-grading (the judge is an independent model, distinct from the reader), and retrieval metrics are reported explicitly separated from end-to-end metrics.
2. **Two baselines with every run published**: Tablet 1 (85.2% $\pm$ 1.1), single-pass delivery, and Scroll 1 (90.7% $\pm$ 0.5), session-expansion delivery. Neither has an LLM in the retrieval path, and all five runs and all per-category raw scores are published (§ 5, Appendix C).
3. **A quantification of the delivery gap**: with retrieval recall saturated at 99.6%, we measure the distance to end-to-end accuracy (14.4 / 8.9 points), showing that the bottleneck has moved from finding to delivering (§ 6).

2. Related Work and the Observed State of Reporting

Lineages of long-term memory systems. Approaches to LLM long-term memory broadly divide into retrieval-augmented generation (RAG) [13] applied to conversation records, self-paging context management (MemGPT/Letta [10]), temporal knowledge graphs (Zep/Graphiti [2]), and extraction-and-consolidation pipelines (Mem0 [12]). Our system belongs to the first lineage, but uses semantic retrieval only, with no lexical matching (§ 3).

Benchmarks. Standard evaluations for long-term conversational memory include LoCoMo [11] and **LongMemEval** [1]. LongMemEval tests six kinds of long-term conversational memory (single-session user facts, single-session assistant facts, knowledge updates, preference inference, temporal reasoning, multi-session synthesis); the S configuration attaches an average of ~140,000 tokens of conversation history to each question. A successor version (LongMemEval-V2 [16]) was released in 2026; this measurement uses the original benchmark's S configuration for comparability with existing reports.

Lexical components persist. Zep/Graphiti [2] explicitly lists Okapi BM25 full-text search as one of its three retrieval functions, a hybrid design. Lexical matching is sensitive to per-language tokenization and orthography, so uniformity across languages is not structurally guaranteed. Our system uses no lexical matching at all (§ 3), though our own multilingual performance is likewise unmeasured (§ 7).

Published numbers are not comparable. The same system's public score moves substantially with the reader model: Mastra's Observational Memory reports 84.2% with GPT-4o and 94.9% with GPT-5-mini [3], and gaps of tens of points between self-reports and third-party comparisons are observed for other systems [6]. Some comparisons have been criticized for placing retrieval-based scores and end-to-end QA scores in the same table [4]. The Memoria evaluation [5] fixed retrieval and swapped readers, demonstrating the need for reader separation, the same design principle as our protocol.

Reporting practice as of July 2026. At the time of writing, self-reported LongMemEval-family numbers reach the 92–96% range: Mem0 at 94.4% (~7,000 tokens per query, self-reported May 2026) [6], OMEGA at 95.4% [7], ByteRover at 92.8% [8], and MemPalace at 96.6% [9]. These figures share three properties. First, **all are self-reported**, with no common reader or judge configuration. Second, some setups use **the same model to generate answers and to grade them** (e.g., GPT-4.1 as both generation and grading LLM) [6]; that is self-grading. Third, metric mixing has been raised as a concern: MemPalace's 96.6% has been questioned as possibly an R@5 **retrieval** score [4][9], and retrieval scores cannot sit in the same table as end-to-end QA accuracy. As § 6 shows, our own engine's retrieval recall is also 99.6%, but we report it explicitly separated from

the end-to-end scores (85.2/90.7%). Even the relative ordering of the same pair of systems has been observed to flip depending on who reports [6]. On the surface, this paper's numbers are lower than those figures. But we claim no ranking; the point of this section is that those numbers and ours are not comparable, because the protocols differ, and that this is precisely why protocols, not headline numbers, must be published.

Known limits of LLM grading. LLM-as-judge exhibits self-enhancement, verbosity, and position biases [14], self-preference bias [15], and roughly ± 1-point judge disagreement; the recommendation not to use the same model for answer generation and grading is well established [14][15]. We separate the judge from the reader, fix the grading rules to the official LongMemEval per-category rules, and minimize degrees of freedom at temperature 0, returning only yes/no (§ 4).

One conclusion follows from this landscape: **publish the protocol, not just the number.** Every measurement below follows that principle.

3. System Overview

The WOS Memory Engine. Both models share one retrieval core: semantic embeddings with neural reranking, no BM25 or keyword matching, and no LLM in the retrieval path. The same records therefore always return the same memories for the same query (deterministic retrieval). Implementation details (embedding and reranker vendors and configuration, weights, thresholds) are proprietary and undisclosed. This paper describes what each stage does, not how; evaluation is performed entirely at the black-box API boundary, so re-verification requires no internal information (§ 8).

Tablet 1 is the first publicly available WOS model. It retrieves in a single pass, returning a bounded context with a median of about **1,200 tokens** (observed max 1,700, n=500) per query, with median engine recall latency of **320 ms** (170–580 ms). There is no LLM anywhere in the retrieval path.

Scroll 1 is the next model after Tablet. Its retrieval judgment is identical to Tablet's; **its delivery differs**: retrieved hits are expanded with their session neighbors and overlap-merged into coherent passages (session chunk-expansion delivery). The delivered context per query is about **3,700 tokens** (as measured on this benchmark run), roughly $3\times$ Tablet's, with engine recall latency of about **260 ms** (p50, API-measured, idle single request). **In this measurement, Scroll 1's retrieval path likewise contains no LLM.** The measured configuration is the pure engine baseline with agentic re-search disabled; the LLM query assistance offered by the commercial Scroll tier is an optional layer on top of this engine and was not part of this measurement. Scroll 1 is thus not a different search but a fuller delivery of the same search, and both models retain deterministic retrieval. This design is what enables the controlled comparison in § 6.

Every subsequent model will be measured against Tablet 1 under this same protocol.

4. Evaluation Protocol

Benchmark. The full LongMemEval-S set, 500 questions. Each question is isolated to its own conversation history (about 53 sessions, ~140,000 tokens on average).

Retrieval. Every question is retrieved with the WOS Memory Engine only. Retrieval is the sole component under test.

Reader. A reader model, instructed to answer from the retrieved memories only, was held constant across all five runs of each model. The reader for the Tablet 1 measurement was **Claude Opus 4.8** (via the Claude API); the reader for the Scroll 1 measurement was **GPT-5.5** (reasoning effort: high). The reader is thus **constant within each model but differs between the two models.** The verbatim reader prompts are in Appendix A; these also differ between models (Scroll 1's requires explicit enumeration and timeline structures before answering). Both are published

verbatim. The implications of the differing reader and prompt for cross-model interpretation are addressed honestly in § 6 and § 7.

Judge. Each answer was graded by **GPT-4o** (via OpenRouter) at temperature 0, returning only yes/no under the official LongMemEval per-category rules (Appendix B). **There is no self-grading**: the judge is a different model from the reader, is not a Wontopos system, and plays no part in generating answers. This distinguishes the protocol from 2026 reporting setups that reuse one model for both generation and grading (§ 2).

Repetition and reporting. The full set was run five independent times and we report the mean. Every run is published; there is no best-of selection and no cherry-picking. Because the engine's retrieval is deterministic, run-to-run variation comes from reader nondeterminism.

4.1 Position on Prompt Fit

The reader prompt's instructions (duration computation, updated-value handling, preference tailoring, enumeration) correspond to LongMemEval's category structure. We do not hide this; we publish the prompts verbatim and take the following position: these instructions do not leak answers to specific items, but are general guidance for the task of answering from dated memories. Nevertheless, the prompt's contribution to the score cannot be separated without new measurement, so an ablation against a generic, branch-free prompt is stated explicitly as future measurement (§ 7).

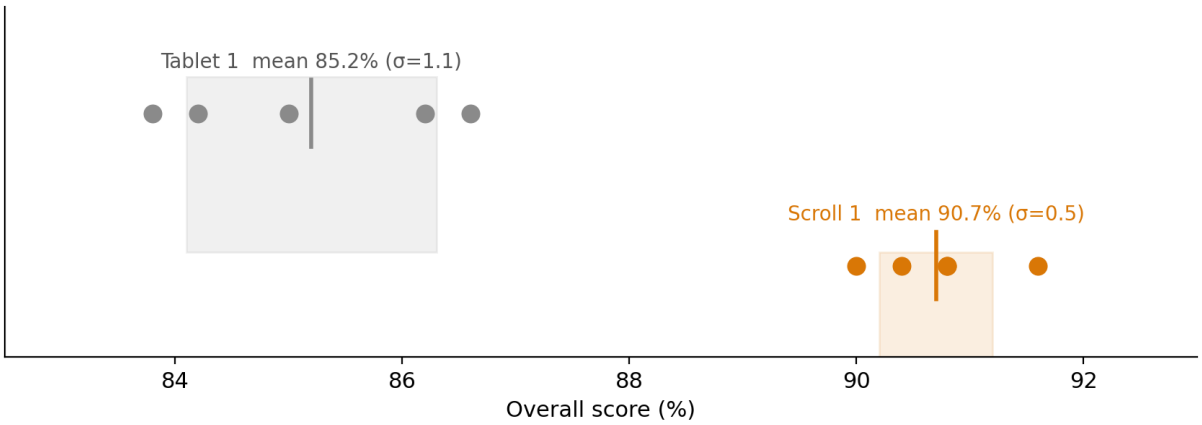
5. Results

5.1 Overall Scores

Model	Five runs	Mean	σ (population)	s (sample)	95% CI (t)	Bootstrap 95% CI	CV
Tablet 1	86.2, 84.2, 85.0, 83.8, 86.6	**85.2%**	1.09	1.22	[83.6, 86.7]	[84.2, 86.1]	1.43%
Scroll 1	90.4, 90.0, 91.6, 90.8, 90.8	**90.7%**	0.53	0.59	[90.0, 91.5]	[90.3, 91.2]	0.65%

The difference between the means is +5.6 points on raw values (90.72 − 85.16) or +5.5 on rounded figures (90.7 − 85.2); the body text uses the rounded 5.5 hereafter. A Welch t-test gives $t = 9.17$, $df \approx 5.8$, $p < 0.001$, and the confidence intervals do not overlap. All five Scroll 1 runs landed at or above 90.0 (Figure 2). Note that this test takes runs as the sampling unit and therefore captures uncertainty from reader nondeterminism only, not benchmark question-sampling variance; since both systems answered the identical 500 questions, question-level paired tests (e.g., McNemar) are the stronger analysis and will be reported when per-question correctness data is released. Because $n = 5$ is a small sample and the bootstrap underestimates the tails, we adopt the more conservative t interval as the default.

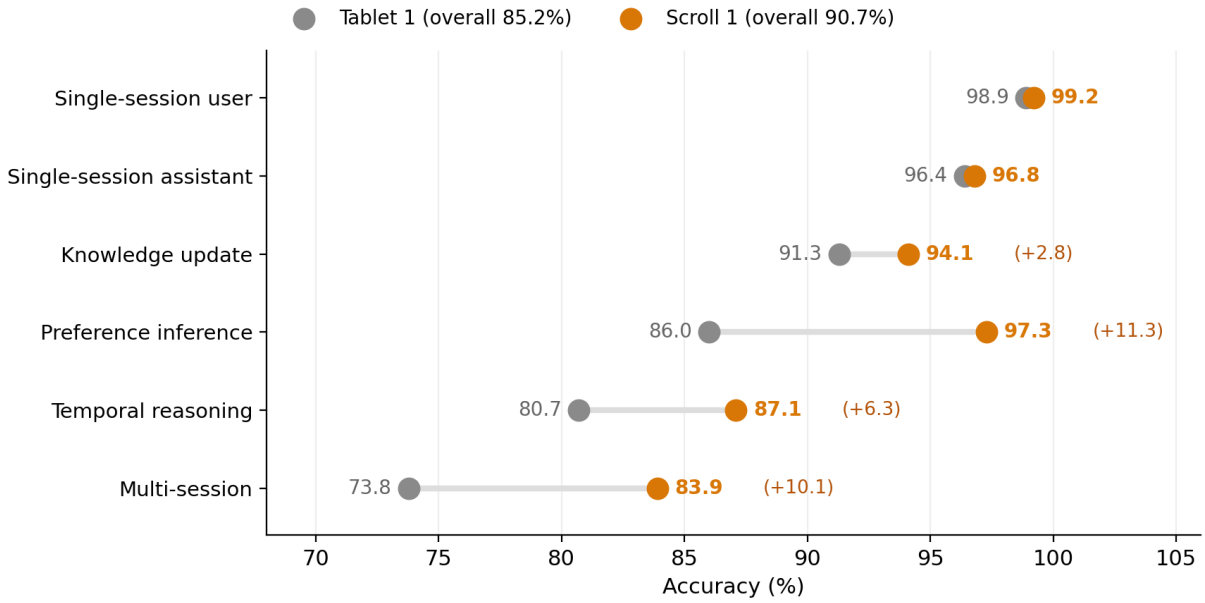
Figure 2. All five runs, both models (no best-of selection)



5.2 Per-Category Comparison (five-run means)

Category	Tablet 1	Scroll 1	Δ
Single-session user	98.9	99.2	+0.3
Single-session assistant	96.4	96.8	+0.4
Knowledge update	91.3	94.1	+2.8
Preference inference	86.0	97.3	+11.3
Temporal reasoning	80.7	87.1	+6.3
Multi-session	73.8	83.9	+10.1
Overall	**85.2**	**90.7**	**+5.5**

Figure 1. Per-category accuracy, five-run means (LongMemEval-S)



(Figure 1.)

The largest gains are in preference inference (+11.3), multi-session (+10.1), and temporal reasoning (+6.3): precisely the categories where **one scattered missing piece changes the answer**. On single-session recall, already near the ceiling, the two models stay within half a point of each

other: a fuller delivery has nothing left to add, consistent with the interpretation in § 6. The complete raw per-run, per-category scores are in Appendix C.

On variance, Tablet 1's largest instability was preference inference (s = 4.35, range 80.0–90.0); under Scroll 1 the same category tightens to s = 2.80 while the mean rises 11.3 points, consistent with reader nondeterminism being amplified when evidence pieces are missing.

5.3 Resource Consumption

	Tablet 1	Scroll 1
Delivered tokens per query	median ~1,200 (observed max 1,700)	~3,700 (measured on this run)
Engine recall latency (p50, idle single request)	~320 ms (170–580)	~260 ms
LLM in retrieval path	none	none (re-search disabled)

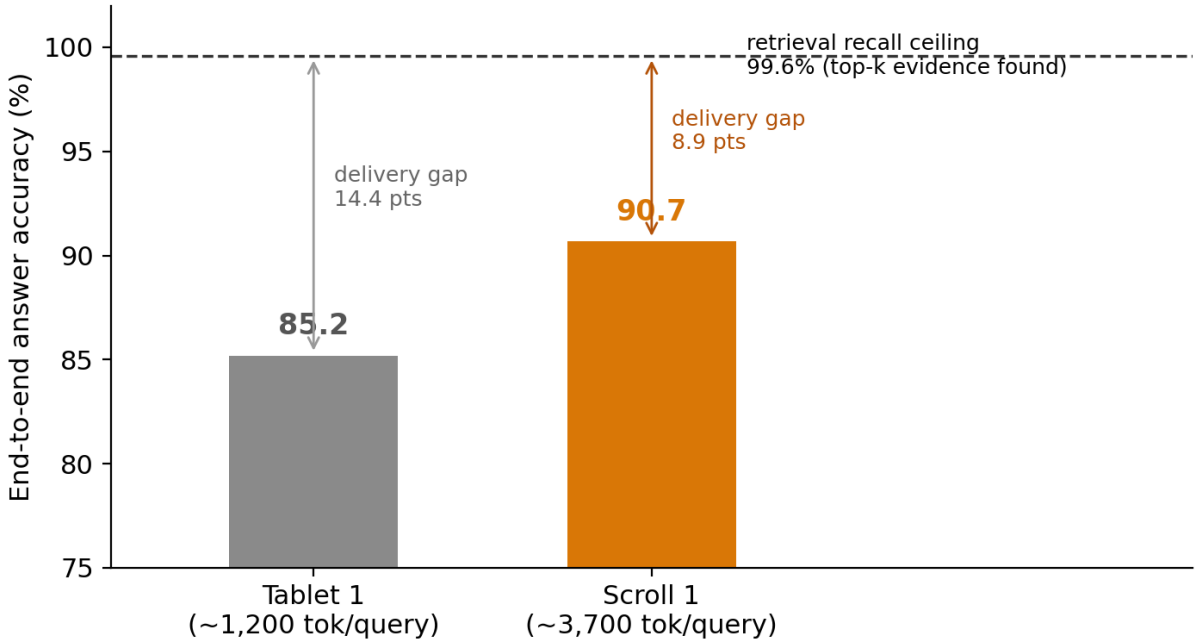
Latency is the p50 of idle single requests. Delivered-token size and latency are observables that anyone can re-measure at the API boundary; engine internals remain undisclosed (§ 3). Since each question carries an average history of ~140,000 tokens, Tablet 1 hands the model less than 1% of the accumulated record; Scroll 1, about 3%. As stored memory grows, the model-side input stays bounded.

6. The Delivery Gap: What Remains After Retrieval Saturates

Retrieval is saturated. Session-level retrieval recall measured on the engine, defined as follows, is **99.6–100%**: each question is isolated to its own conversation history (~53 sessions on average), and we score whether the session containing the gold evidence surfaces in the top results. This holds regardless of top-k configuration (5–20). The gold evidence is surfaced essentially every time. And this recall applies **equally** to Tablet 1 and Scroll 1, because their retrieval judgment is identical. One precision: this is a session-level metric while end-to-end accuracy is question-level, so the denominators differ; the "delivery gap" below should be read as an **approximate distance to a retrieval-conditional ceiling**, not an exact difference. In particular, for multi-session questions that require several evidence sessions, session-level recall is an optimistic ceiling; measuring the question-level "all required evidence delivered" rate is left as follow-up work alongside the per-question retrieval dumps (§ 8).

Yet answer accuracy does not reach that ceiling. Given the same evidence, Tablet 1 answers 85.2% correctly and Scroll 1 90.7% (Figure 3).

Figure 3. Retrieval is saturated; the remaining loss is delivery



The distance to 99.6% (14.4 points for Tablet 1, 8.9 for Scroll 1) is the **delivery gap**: the share of questions where the memory was found but not turned into a correct answer.

About the 5.5 points between the two models we must speak precisely. Retrieval being identical, it is certain that this difference **does not come from retrieval.** But the two configurations differ not only in delivery volume but also in reader prompt (intermediate-structure demands) and in the **reader model itself** (Opus 4.8 → GPT-5.5), so the five-run data alone cannot fully decompose the 5.5 points across these three factors.

A partial decomposition is, however, available. Scroll 1's development lineage recorded single-run scores as each delivery component was enabled in turn (all with the GPT-5.5 reader, same engine):

Configuration (cumulative)	Score
bare (no expansion in delivery)	84.4
+ hit-chunk preservation	88.0
+ structured-reasoning prompt	88.6
+ cross-passage dedup	89.0
+ session-context expansion (final config)	90.4

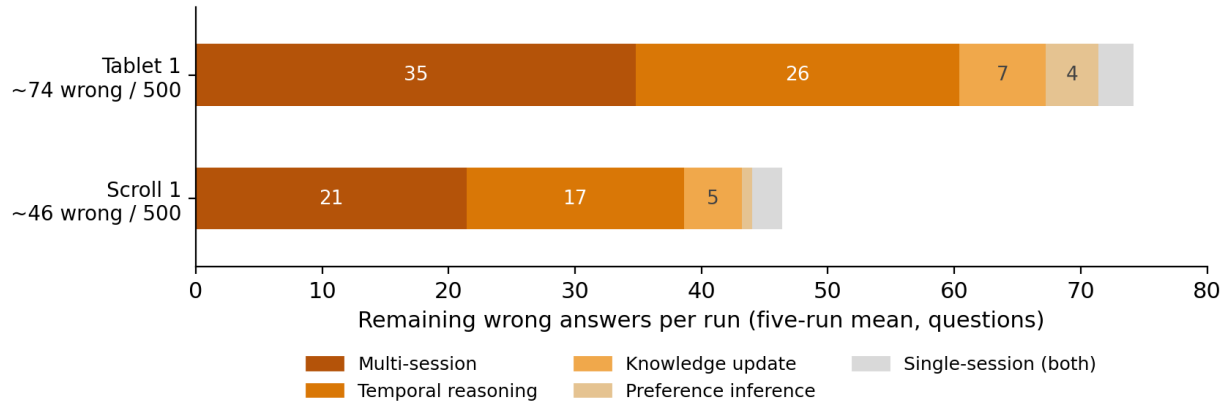
In this lineage, **delivery mechanics account for +5.4 points in total** (the 6.0-point lineage rise minus the structured prompt's +0.6 (88.0→88.6): 3.6+0.4+1.4). It also permits an indirect estimate of the reader effect: with bare delivery, GPT-5.5 scored 84.4, close to Tablet 1's 85.2 with Opus 4.8. This is consistent with **the reader swap alone having a small effect, with most of the gain coming from delivery mechanics.** But the lineage figures are all single runs, within reader nondeterminism (about ±1 point, § 5.1), and the bare configuration differs from Tablet 1 in both delivery and prompt simultaneously (one equation, three unknowns), so this is corroborating evidence rather than proof; the definitive decomposition belongs to the ablation in § 7. The gains concentrating in multi-session, preference inference, and temporal reasoning (§ 5.2) point the same way: these are the categories where a single missing piece quietly bends the answer, and fuller delivery is what brings that missing piece home.

6.1 Error Analysis: Why 90.7% When the Memory Is Right There at 99.6%

We dissect the delivery gap at the level of **questions**, not points. Multiplying the five-run mean error rate by each category's question count in LongMemEval-S (single-session user 70, single-session assistant 56, preference inference 30, knowledge update 78, temporal reasoning 133, multi-session 133; total 500; every per-run score in Appendix C is consistent with an integer correct-count over these counts [1]) gives the **expected number of wrong answers per run**. No single run realizes these fractional values. By construction the decomposition is consistent with the overall scores (Tablet 1: 74.2 wrong → 85.1%; Scroll 1: 46.4 wrong → 90.7%; table values rounded).

Category (questions)	Tablet 1 wrong	Scroll 1 wrong	Recovered by Scroll
Multi-session (133)	34.8	21.4	13.4
Temporal reasoning (133)	25.6	17.2	8.4
Knowledge update (78)	6.8	4.6	2.2
Preference inference (30)	4.2	0.8	3.4
Single-session (126)	2.8	2.4	0.4
Total (500)	**74.2**	**46.4**	**27.8**

Figure 4. Where the delivery gap lives: composition of remaining errors



Two things are immediately visible. First, **about 81–83% of the remaining errors live in temporal reasoning and multi-session.** Single-session recall is effectively solved (2–3 wrong out of 126). Second, the ~28 questions recovered by Scroll's expanded delivery concentrate in preference inference (3.4 of 30 recovered, 81% of its errors erased) and multi-session (13.4 recovered): precisely the categories where evidence is scattered and one missing piece changes the answer.

We now examine why the remaining errors occur, by failure type. Channels (i)–(iii) are loss paths within Scroll 1's remaining 46.4 questions; channel (iv) is the path **already closed** between the two models. The type inspection is based on a manual review of a sample of temporal-reasoning structural failures, not an exhaustive classification; knowledge update (4.6), single-session (2.4), and the remainder of temporal reasoning stay as an unattributed residual.

(i) Grading loss: questions the model got right but the score did not. Inspecting the structural failures in temporal reasoning, about three are judge artifacts: answers essentially equivalent to the ground truth that the judge failed to recognize as equivalent. This is a resolution limit of the measuring instrument, not a system failure, and rather than adjusting scores upward, we keep the grading rules fixed and absorb the loss as-is. To be clear about scope: this inspection audited only answers marked wrong (a one-directional audit); the rate of errors in the other direction (answers leniently marked correct) is unmeasured, and a two-directional judge–human agreement measurement on a sample is follow-up work.

(ii) Design cost: questions deliberately left unsolved on principle. About four temporal failures are of a type reachable only by anchoring retrieval on date strings. The engine keeps to pure semantic retrieval with no lexical or date matching (the basis of its language-equality design goal, § 3; unmeasured, § 7), so these questions remain the price of the principle. Precisely put, these are **deliberate residual retrieval failures**, an example showing that the 99.6% session-level recall of § 6 does not mean every piece of question-level required evidence is captured. Treating time as a numeric dimension rather than as vocabulary is the candidate solution that would not break the principle; it is left to future models.

(iii) Integration failure: questions where every piece arrives and the model still cannot put them together. The largest remaining block, multi-session (21.4 wrong), fails centrally on **counting and aggregation**: enumerating, deduplicating, and summing instances scattered across sessions. Even with the evidence in context, missing a single instance breaks the whole total. That language models under-use information positioned mid-context is a documented phenomenon [17], and all-or-nothing grading (no partial credit, Appendix B) amplifies the fragility. Notably, the structured-reasoning prompt, which forces enumerate → merge → sum, contributed only +0.6 points in the lineage (§ 6; a single-run figure within reader nondeterminism, but directionally suggestive). The failure therefore appears to be not one of output formatting but a **limit of aggregation itself over long contexts**, the hard core of the delivery gap.

(iv) Delivery truncation: the path Scroll 1 has already closed. The ~28-question difference between the models is the measured size of this channel. Retrieval had found the evidence, but single-pass delivery left neighboring pieces behind; when session expansion brought them along, the answers recovered. Preference inference is the emblematic case: a preference is laid down across multiple utterances, not one sentence, and expanded delivery alone cut its errors from 4.2 to 0.8.

In sum, Scroll 1's remaining 46.4 questions (about 44.5 measured against the 99.6% session-recall ceiling) are not one failure. Channels (i)+(ii), measurement and design costs, account for roughly seven questions verified on the temporal side alone, meaning that reading the whole residual as "the model failing to use evidence" overstates model failure by about three artifact questions; channel (iii), integration, is the largest block, centered on multi-session aggregation; the rest stays as an unattributed residual; and channel (iv) is the share Scroll 1 has already paid to recover. The decomposition points clearly at the next locus of work: not retrieval, not delivery volume, but **aggregation over delivered evidence**, and grading methods able to measure it with partial credit.

And after Scroll 1, 8.9 points still remain. This residue, evidence surfaced 99.6% of the time that the model still fails to use, is the current frontier that no retrieval improvement can close. We take it to be the next locus of work for long-term memory systems: the bottleneck is no longer finding, but getting the found memory fully used.

The cost, stated honestly. Fuller delivery is not free. Scroll 1 hands the reader roughly 3× the tokens per call, so reading costs rise accordingly (the optional LLM query assistance offered by the commercial Scroll tier was not part of this measurement, § 3). The trade-off is deliberate: for questions where one missing piece flips the answer, completeness matters more than cheapness. Conversely, for tasks already saturated, like single-session recall, Tablet 1 suffices and the extra cost is waste. That is why both models exist.

7. Limitations

This evaluation has clear limitations, which we state with the same weight as the results.

Single benchmark, and design selection on that benchmark. These are results on LongMemEval-S alone, and we make no claim of generalization to tasks with different distributions. More importantly: Scroll 1's delivery configuration and reader prompt were **selected while watching scores on this same benchmark** (the lineage in § 6 is the record of that process). There is no

run-level best-of, but design-level fitting exists, and we cannot rule out that it inflated the 90.7%. Held-out confirmation (e.g., LongMemEval-V2 [16]) is needed.

Different readers across models. The reader is fixed within each model's five runs but differs between models: Claude Opus 4.8 for Tablet 1, GPT-5.5 for Scroll 1. The development lineage in § 6 suggests the reader swap alone has a small effect (bare + GPT-5.5 = 84.4 ≈ Tablet 1's 85.2), but those figures are single runs and the bare delivery is not exactly Tablet 1's. A definitive decomposition of the between-model Δ requires (a) re-running both models five times under a single reader, and (b) re-running under a generic, branch-free prompt. Each model's delivery gap against its own ceiling (14.4 / 8.9 points) and the retrieval saturation figure (99.6%) stand independently of this confound. The small-sample limitation of n = 5 also remains.

Multilingual performance unmeasured. The engine design is language-neutral (no lexical matching), but this measurement used LongMemEval's English items only. "The same in every language" is a design claim, not a measured one.

Self-evaluation. The developer measured its own system (§ 9). Full publication of protocol, prompts, and raw scores does not remove this limitation; it only makes third-party re-verification possible.

No competitor runs. We did not run other systems under this protocol and therefore make no superiority claims. The figures in § 2 are each source's self-reported numbers, cited as-is.

8. Reproducibility

We state the scope of reproducibility precisely. **Published:** the benchmark (a public dataset), the full protocol, the verbatim reader and judge prompts (Appendices A–B), the raw per-run per-category scores (Appendix C), and the benchmark report pages (wontopos.com/model/tablet-1, wontopos.com/model/scroll-1). **Not published:** engine internals (embedding and reranking configuration). A third party can therefore re-run the identical protocol at the black-box boundary with a WOS API key and their own LLM keys and verify these results, but cannot re-implement the engine internals. Scroll 1 is invoked with the `X-WOS-Model: scroll-1` header. LLM keys are passed per request and never stored; memories are isolated per account and user. Per-question retrieval dumps are available on request. The published version will additionally specify model snapshot identifiers and run dates for reader and judge, reader sampling settings, and the ingestion procedure for conversation histories (session boundaries, timestamp handling). Contamination (both readers plausibly having seen LongMemEval in pretraining) cannot be ruled out, but applies equally to every system compared on this benchmark.

9. Conflict of Interest

The author is the founder and 100% equity holder of Wontopos, and the systems under evaluation, Tablet 1 and Scroll 1, are Wontopos's commercial products. Publication of this paper is tied to product launch. To offset this conflict we publish every run score, every prompt, and the full protocol, report no best-of numbers, and keep a third-party re-verification path open (§ 8).

10. Conclusion

When the protocol is fixed and every run is published, memory-system numbers finally become comparable. Under those conditions we measured two things: an honest single-pass, no-LLM retrieval baseline at 85.2% (Tablet 1), and the same retrieval with fuller delivery at 90.7% (Scroll 1). With session-level retrieval recall saturated at 99.6%, the difference between the two scores did not come from retrieval (the lineage suggests most of it came from delivery), and roughly nine points of gap remain even behind 90.7%. The problem of finding memories is nearly solved. The

problem of getting found memories fully used is just beginning, and every subsequent model will be measured against these two baselines with the same ruler.

References

[1] Wu, D. et al. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. arXiv:2410.10813. [2] Rasmussen, P. et al. Zep: A Temporal Knowledge Graph Architecture for Agent Memory. arXiv:2501.13956. (Explicitly lists BM25 full-text search as a retrieval function.) [3] Mastra Research. Observational Memory: 95% on LongMemEval. mastra.ai/research/observational-memory. (84.2% with GPT-4o vs 94.9% with GPT-5-mini.) [4] MemPalace benchmark methodology issue #29. github.com/MemPalace/mempalace/issues/29. (Criticism of mixing retrieval scores with end-to-end QA scores.) [5] MatrixOrigin. Benchmarking Memoria on LongMemEval. (Reader-separation experiment.) [6] Mem0. State of AI Agent Memory 2026. mem0.ai/blog/state-of-ai-agent-memory-2026. (94.4%, ~7,000 tokens/query, self-reported.) [7] OMEGA. LongMemEval Benchmark Leaderboard. omegamax.co/benchmarks. (95.4%, self-reported.) [8] ByteRover. Benchmark AI Agent Memory in Real Production. byterover.dev. (92.8%, self-reported.) [9] MemPalace. Benchmark Results: 96.6% LongMemEval. mempalace.tech/benchmarks. (96.6% reported as an R@5 retrieval score; for methodological criticism see [4] and arXiv:2604.21284.) [10] Packer, C. et al. MemGPT: Towards LLMs as Operating Systems. arXiv:2310.08560. [11] Maharana, A. et al. Evaluating Very Long-Term Conversational Memory of LLM Agents (LoCoMo). arXiv:2402.17753. [12] Chhikara, P. et al. Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. arXiv:2504.19413. [13] Lewis, P. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv:2005.11401. [14] Zheng, L. et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. NeurIPS 2023. arXiv:2306.05685. [15] Wataoka, K. et al. Self-Preference Bias in LLM-as-a-Judge. arXiv:2410.21819. [16] LongMemEval-V2: Evaluating Long-Term Agent Memory Toward Experienced Colleagues. arXiv:2605.12493. [17] Liu, N. F. et al. Lost in the Middle: How Language Models Use Long Contexts. TACL 2024. arXiv:2307.03172.

Appendix A. Verbatim Reader Prompts

A.1 Tablet 1 reader prompt (used to generate every answer)

```
Answer the question using ONLY the retrieved memories below (each is
prefixed with its [date]). This question is being asked on: {qdate}.
Apply whichever of these fits the question:
- For any 'how long ago' / 'how many days/weeks/months since' question,
  compute the duration relative to the asking date above (not any other
  today), using the memory dates.
- If the memories give CONFLICTING values for the same fact (different
  values as of different dates), mention BOTH and note which is more recent.
- If the question asks for ADVICE or a RECOMMENDATION, first identify this
  user's relevant preferences, interests, and past choices from the
  memories, then tailor your answer to them (not generic advice).
- Otherwise, answer the factual question concisely and directly.
If the answer is not in the memories, say you don't know. Answer in the
SAME LANGUAGE as the question.

Memories:
{mems}
```

```
Question: {q}

Answer:
```

A.2 Scroll 1 reader prompt (used to generate every answer)

```
Answer the question using ONLY the retrieved memories below (each
prefixed [date] (rel=relevance)). This question is asked on: {qdate}.
Apply whichever fits:
- 'how long ago' / 'how many days/weeks/months since': compute the
  duration relative to the asking date using memory dates.
- If memories give CONFLICTING values for the same fact as of different
  dates, give the MOST RECENT value (mention the prior only if asked).
- If the question asks for ADVICE/RECOMMENDATION, first identify this
  user's relevant preferences from the memories, then tailor to them.
- For COUNT/TOTAL/LIST questions: ENUMERATE every candidate item in the
  memories (including ones mentioned only once or in passing), MERGE
  duplicates (same item on different dates = one), then count/sum the
  distinct results and state the final number.
- BEFORE the final answer, write an explicit intermediate structure from
  the memories: for COUNT/TOTAL/LIST, list every candidate item (incl.
  ones mentioned once), merge duplicates (same item on different dates =
  one), then count/sum the distinct items; for TEMPORAL /
  values-changing-over-time, build a (value, date) timeline. Show the
  structure, THEN give the final answer.
If the answer is not in the memories, say you don't know. Answer in the
SAME LANGUAGE as the question.

Memories:
{mems}

Question: {q}

Answer:
```

{qdate} = the date the question is asked; {mems} = retrieved memories, each prefixed with its date; {q} = the question.

Appendix B. Judge Prompt and Grading Rules (temperature 0)

```
I will give you a question, the correct answer, and a model's response.
{RULE} Respond with ONLY 'yes' or 'no'.

Question: {q}
Correct answer: {gt}
Model response: {ans}

Is the model response correct?
```

{RULE} is the official LongMemEval per-category rule. **Temporal reasoning:** correct if the answer or an equivalent is present; off-by-one errors in days/weeks/months are not penalized. **Knowledge update:** correct if the updated answer is present (mentioning outdated information is fine). **Single-session preference:** need not cover every rubric point; counts if the user's personal information or preference is recalled and used correctly. **All other categories:** correct if the answer,

or an equivalent including all intermediate steps, is present; a subset of the required information is marked wrong.

Appendix C. Raw Per-Run, Per-Category Scores

C.1 Tablet 1 (five runs)

Category	Run 1	Run 2	Run 3	Run 4	Run 5	Mean
Single-session user	98.6	100.0	98.6	98.6	98.6	98.9
Single-session assistant	94.6	96.4	96.4	96.4	98.2	96.4
Knowledge update	92.3	89.7	93.6	88.5	92.3	91.3
Preference inference	90.0	86.7	83.3	80.0	90.0	86.0
Temporal reasoning	82.0	78.9	81.2	78.9	82.7	80.7
Multi-session	75.9	72.2	72.2	73.7	75.2	73.8
Overall	86.2	84.2	85.0	83.8	86.6	**85.2**

C.2 Scroll 1 (five runs)

Category	Run 1	Run 2	Run 3	Run 4	Run 5	Mean
Single-session user	98.6	98.6	98.6	100.0	100.0	99.2
Single-session assistant	98.2	96.4	96.4	98.2	94.6	96.8
Knowledge update	92.3	93.6	94.9	92.3	97.4	94.1
Preference inference	96.7	100.0	96.7	100.0	93.3	97.3
Temporal reasoning	85.7	84.2	90.2	86.5	88.7	87.1
Multi-session	85.0	84.2	84.2	84.2	82.0	83.9
Overall	90.4	90.0	91.6	90.8	90.8	**90.7**