Crossover Report:
Tablet 1 & Scroll 1

July 15, 2026

Contents

When to Deliver More: A Memory-Size Crossover Between Tablet 1 and Scroll 1 on LoCoMo and LongMemEval-S

Kim Sunwoo, Wontopos

Contact: sunwoo.ceo@wontopos.com • wontopos.com/research • Technical report, July 15, 2026

Abstract

Persistent-memory products face a tiering choice: hand the reader a minimal delivery, or a fuller one that preserves scattered evidence at a higher price. We measured that choice at two memory scales under one shared reader (GPT-5.6 Sol) and one independent judge (GPT-5.6 Luna), and combined the result with an earlier full-benchmark study. The finding is a **crossover**. On **LoCoMo** (about 20K tokens of history per question, 1,986 questions), **Tablet 1** and **Scroll 1** answer within 0.15 points of each other (76.5% vs. 76.7%), yet **Scroll 1** costs 2.2× more per query: its fuller delivery buys no accuracy at this scale, so **Tablet 1** is the efficient tier. On **LongMemEval-S** at about 140K tokens (500 questions, five runs), **Scroll 1** leads by +5.5 points (90.7% vs. 85.2%), concentrated in preference, multi-session, and temporal questions—the categories where one scattered piece decides the answer. We are explicit about what is and is not established: the small-memory result is a well-powered, same-reader measurement, while the large-memory advantage comes from a study whose two tiers used different readers (Claude Opus 4.8 vs. GPT-5.5), so it is directionally supported but reader-confounded. At large records, memory also decisively beats reading the full history: at ∼102K tokens both tiers answer 95.8% correctly against full context's 79.2%, at 17–37× lower cost. We publish every per-condition score, the cost ledger, and the prompts. The practical rule is not "Scroll is better" but "deliver more only when the record is large enough that fuller delivery buys accuracy." **This is a preliminary report with many limitations.** The new measurements are single-run, the only well-powered same-reader comparison (LoCoMo) is a statistical tie, and the large-memory advantage comes from a reader-confounded study. We present the crossover as an early observation that requires further verification—a same-reader rerun at 140K, a controlled length sweep, and multi-run repetition (§7)—not as a settled result.

1 Introduction

A long-term memory product does two things for a reader model: it *finds* the relevant pieces of an accumulated record, and it *delivers* them into the prompt. Two Wontopos tiers share one retrieval core and differ only in delivery. **Tablet 1** performs a single-pass delivery with a small bounded context. **Scroll 1** expands the same retrieval hits with their session neighbors, delivering roughly 1.5× as many tokens. A prior report showed that on long records **Scroll 1** answers more questions correctly; the natural next question is whether that is always true, and whether the extra delivery is worth its extra cost.

This report answers both with a single word: **it depends on how much memory the question carries**. We measured **Tablet 1** and **Scroll 1** at two memory scales and found a crossover. When the per-question record is small, the two tiers are statistically tied on accuracy and **Tablet 1** is cheaper, so **Tablet 1** wins. When the record is large, **Scroll 1**'s fuller delivery recovers answers that **Tablet 1** misses, so **Scroll 1** wins. Between those two regimes the right product choice flips.

We hold ourselves to the standard of the earlier Wontopos reports: publish the protocol and every run, separate retrieval-based numbers from end-to-end accuracy, never grade a model with itself,

and state confounds with the same weight as results. In that spirit we flag immediately the main limitation of the crossover: the well-powered, single-reader comparison we ran (LoCoMo, small memory) shows a *tie*, and the large-memory result where **Scroll 1** wins is taken from a study whose two tiers used different reader models. The direction is consistent across the category structure and an earlier delivery ablation, but a same-reader rerun at 140K and a controlled length sweep are the experiments that would turn this two-point picture into a proven curve (§7). The contributions are:

1. **A measured memory-size crossover** between two delivery tiers: at ∼20K tokens **Tablet 1** and **Scroll 1** tie on accuracy while **Tablet 1** costs 2.2× less; at ∼140K **Scroll 1** leads by +5.5 points (§5.1–§5.4).
2. **A lifecycle cost ledger** under one reader at ∼102K tokens, with retrieval metered and reader inference priced, showing both tiers 17–37× cheaper than reading the full record (§5.2).
3. **An honest decomposition of what is proven**: a well-powered small-memory tie, a reader-confounded large-memory win, and the two experiments that would close the gap (§7).

2 Related Work and Reporting Context

Approaches to long-term memory for conversational AI include retrieval-augmented generation over chat records [8], self-paging context managers (MemGPT/Letta [7]), and temporal knowledge graphs. Both tiers here belong to the retrieval lineage but use semantic retrieval only, with no lexical matching. Standard evaluations include **LoCoMo** [2], a set of long synthetic conversations whose per-question history is on the order of tens of thousands of tokens, and **LongMemEval** [1], whose S configuration attaches roughly 100–140K tokens of history per question across single-session facts, knowledge updates, preference inference, temporal reasoning, and multi-session synthesis. The two benchmarks differ in size *and* in question mix, which is why we treat them as two operating points rather than two samples of one distribution.

Reporting practice in this field mixes metrics: retrieval recall and end-to-end QA accuracy are sometimes cited in the same table, and some setups use one model both to answer and to grade [4]. We keep the reader and judge as separate models, publish every run, and report no best-of numbers. The point of long-context work such as "Lost in the Middle" [3] is directly relevant: a model with access to a long sequence does not use its contents uniformly, which is a plausible mechanism for the full-context degradation we observe at 102K (§5.2), though we do not measure evidence position directly.

3 Systems and Conditions

Both tiers are evaluated at the public API boundary; embedding, reranking, storage, and dimensionality choices are proprietary and undisclosed, so re-verification uses no internal information. **Tablet 1** returns a single-pass delivery with a median near 1,600–1,730 delivered tokens on these runs. **Scroll 1** expands retrieved hits with session neighbors and overlap-merges adjacent passages, delivering about 1.5× as many tokens. Their retrieval judgment is identical; **Scroll 1** is a fuller delivery of the same search, not a different search. The measured **Scroll 1** configuration uses the commercial route with agentic re-search disabled; the optional LLM query-understanding layer offered by the commercial tier was not part of this measurement.

We measured six conditions. Two are the products (`tablet_1`, `scroll_1`). Four are controls:

`full_nocache` delivers the entire record with a cache-busting prefix; `oracle` delivers the gold evidence sessions and controls for evidence selection; `sliding_window` delivers the most recent ∼8K tokens and tests recency without retrieval; `random_context` delivers non-evidence text at a comparable small budget and tests whether a small context is useful by itself.

4 Evaluation Protocol

Benchmarks. We use three runs. (i) **LoCoMo full**: 10 conversations, 1,986 questions, about 20K delivered full-context tokens per question. (ii) **LongMemEval-S stratified pilot**: 24 questions balanced across six categories, about 102K tokens per question. (iii) **LongMemEval-S single-category slice**: 20 single-session-user questions, about 102K tokens, used as an auxiliary check. For the large-memory quality frontier we additionally cite the earlier Wontopos report [9]: the full 500-question LongMemEval-S set, five runs, about 140K tokens per question.

Reader and judge. In all three runs measured here, every condition uses the same reader, **GPT-5.6 Sol** (reasoning effort high), with one generic prompt and a fixed output cap, so the reader is constant across conditions. Each answer is graded by **GPT-5.6 Luna** at temperature 0 under the official per-category rules, returning only yes/no. The judge is a different model from the reader; there is no self-grading. One dependence remains: reader and judge are from the same provider family, treated as a limitation (§7). Crucially, the earlier 140K benchmark [9] used *different* readers for the two tiers (Claude Opus 4.8 for **Tablet 1**, GPT-5.5 for **Scroll 1**); we carry that confound explicitly wherever we use its numbers.

Cost ledger. Every retrieval, reader, judge, and ingest call is recorded. WOS ingestion and retrieval are metered at real billing; reader and judge tokens are priced by the official July 2026 API table (projection, not invoice). Cost per query is reader inference plus metered retrieval; the judge and the one-time ingest charge are reported separately, not folded into cost per query. This matches the convention of the lifecycle study and reproduces its figures exactly.

5 Results

5.1 Small memory: LoCoMo

LoCoMo serves here as the small end of the memory-size axis: each question carries about 20K tokens of history, roughly a fifth of the LongMemEval-S record. At this scale the two delivery tiers are statistically indistinguishable on accuracy while **Tablet 1** costs far less (Table 1). Under a single shared reader over 1,986 questions, **Scroll 1** delivers 45% more context and costs 2.19× more per query, yet the two tiers finish within 0.15 points of each other—three questions out of 1,986—and stay within 1.6 points in every LoCoMo category. The 0.15-point difference has a 95% confidence interval of about ±2.6 points (normal approximation on the difference of proportions), comfortably spanning zero, so the two tiers are statistically indistinguishable at this scale; a paired question-level test would be tighter but does not change the conclusion. The fuller delivery has essentially nothing to add at this record length: retrieval already surfaces what the reader needs, so the extra session neighbors are redundant weight. Full per-condition results, including the non-memory controls, are in Appendix A; the raw per-question verdicts for the LME-S pilot are in Appendix D.

Table 1: LoCoMo at ∼20K tokens/question (1,986 questions, one shared reader). The two delivery tiers tie on accuracy; Tablet 1 is 2.2× cheaper. Cost per query is reader + metered retrieval.

Tier	Accuracy	Cost/query	Delivered tok	vs. Tablet cost
Scroll 1	**76.7%**	$0.0322	2,514	2.19×
Tablet 1	**76.5%**	$0.0147	1,730	1.00×

5.2 Large memory: LongMemEval-S

At about 102K tokens per question, both memory tiers answer 23 of 24 pilot questions correctly (95.8%), while full context drops to 79.2% and costs 17–37× more (Table 2). The 24-question tie is a power limitation, not evidence of equivalence: the two tiers produced *identical* correctness vectors on the pilot (both missed the same single preference question, so there are zero discordant pairs and no paired test can separate them), and with four questions per category the pilot cannot detect the category-level gains reported at full scale. Its value is the matched cost ledger, not a quality ranking. The well-powered quality comparison at large memory is the earlier 500-question benchmark [9], where **Scroll 1** scores 90.7% and **Tablet 1** 85.2%, a +5.5 point difference whose category breakdown is preference +11.3, multi-session +10.1, temporal +6.3, knowledge-update +2.8, and the two single-session categories within half a point. The gains land exactly where evidence is scattered across sessions and one missing piece bends the answer.

Table 2: LongMemEval-S stratified pilot (24 questions, one reader). The 500-question, five-run figures [9] used different readers per tier and are shown for the quality frontier only.

Condition	Accuracy	Cost/query	$/correct	Delivered tok	vs. Tablet cost
Full context	79.2%	$0.5185	$0.6550	103,448	36.7×
Oracle	91.7%	$0.0297	$0.0323	5,637	—
Scroll 1	**95.8%**	$0.0305	$0.0319	2,380	2.16×
Tablet 1	**95.8%**	$0.0141	$0.0147	1,625	1.00×
Sliding 8K	25.0%	$0.0344	$0.1377	6,638	—
Random	8.3%	$0.0116	$0.1394	2,097	—
500-question benchmark [9], different readers per tier:	Tablet 1 = 85.2%,	Scroll 1 = 90.7%	(Δ +5.5)		

At this record length the memory tiers do not merely match full context—they beat it decisively on both axes (Figure 1). Reading the entire ∼103K-token history scores 79.2%, below both tiers' 95.8%, and costs $0.5185 per query against $0.0141 (**Tablet 1**) and $0.0305 (**Scroll 1**): a 36.7× and 17.0× gap. A long input is not used uniformly [3], so handing the reader the whole record is both the most expensive option and, at this length, a less accurate one. Memory's job at large records is therefore not only to be cheaper but to deliver a smaller, better-organized context than the record itself.

5.3 Where the large-memory gain lives

The +5.5-point advantage at 140K is not spread evenly; it is concentrated in exactly the categories where evidence is scattered across sessions (Figure 2). On the two direct single-session categories, already near the ceiling, the tiers are within half a point. The gains appear in preference inference (+11.3), multi-session synthesis (+10.1), temporal reasoning (+6.3), and knowledge update (+2.8)—questions whose answer depends on assembling pieces laid down at different times. This is the mechanism behind the crossover: a single-pass delivery is enough when the answer sits in one

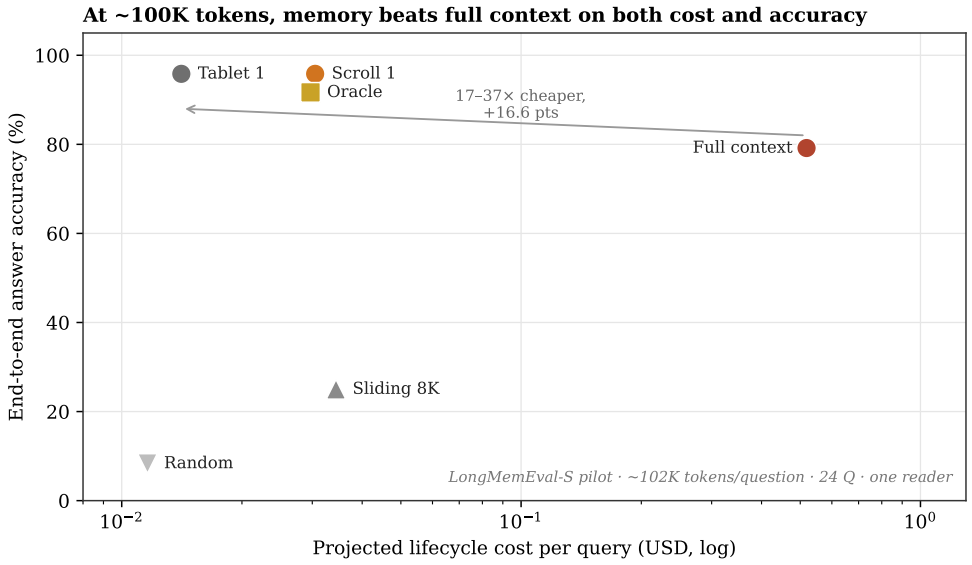

Figure 1: Cost–accuracy frontier at ∼102K tokens per question. Both memory tiers sit at the low-cost, high-accuracy corner; reading the full record is the most expensive condition and, at this length, less accurate than either tier. Oracle (gold evidence sessions) is a selection control, not a reader ceiling.

retrieved passage, and it is the expansion to session neighbors that recovers the answer when the required pieces are spread out. Because that spread grows with the size of the record, the categories that reward fuller delivery are also the ones that dominate at long record lengths.

5.4 The delivery-value crossover

Putting the two scales together gives the central result (Figure 3). The question is not which tier is more accurate in the abstract but *whether fuller delivery buys accuracy at a given memory size*. At 20K it does not (+0.15, well-powered, same reader). At 102K the pilot cannot tell (0.0, underpowered). At 140K it does (+5.5, but different readers). Since **Scroll 1** costs about 2.2× **Tablet 1** at every point, the product decision follows directly: where the accuracy gain is zero, pay less and choose **Tablet 1**; where the gain is real and the workload is integration-heavy, **Scroll 1**'s extra delivery is worth its extra cost.

We state the asymmetry plainly. The "small memory → Tablet" side is on firm ground: a same-reader, 1,986-question measurement showing fuller delivery adds nothing at 20K. The "large memory → Scroll" side is directionally supported—by the category structure (gains concentrated where evidence is scattered) and by the earlier delivery ablation, in which a bare GPT-5.5 reader scored 84.4%, close to **Tablet 1**'s 85.2%, so most of **Scroll 1**'s gain tracked delivery rather than the reader swap—but the single comparison that shows the +5.5 used different readers. It is a crossover we can see at two operating points, not yet a curve we have drawn.

5.5 Delivered context and cost composition

Whatever the record size, both tiers hand the reader a small, bounded context (Figure 4, left). At 102K, **Tablet 1** delivers about 1,625 tokens and **Scroll 1** about 2,380—1.6% and 2.3% of the full record. As stored memory grows, the model-side input stays flat while the record behind it can grow without bound; this is the structural reason memory scales where full context does not. The

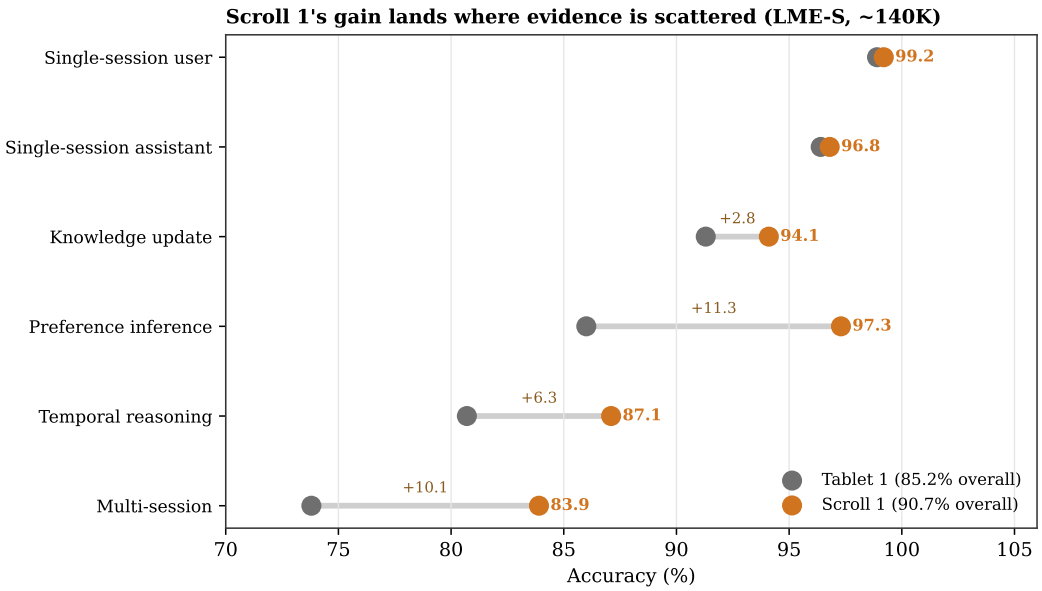

Figure 2: Per-category accuracy on the 500-question LongMemEval-S benchmark [9]. Scroll 1's gain over Tablet 1 is near zero on single-session recall and largest on preference, multi-session, and temporal questions—the categories where one scattered piece decides the answer.

two tiers also have different cost structures (Figure 4, right). For **Tablet 1**, reader inference is the larger share ($0.0096 of $0.0141); reducing delivered tokens is its main lever. For **Scroll 1**, metered retrieval is the larger share ($0.0173 of $0.0305), because expanding hits into session neighbors is work the engine does before the reader sees anything. Full context has no separate retrieval charge, but its reader cost alone is 36.7× the complete **Tablet 1** route.

5.6 Cost over repeated queries

A record is usually queried more than once, and cost accumulates over its life. Writing the observed per-query slopes with a one-time ingestion charge gives, on LoCoMo records,

$$C_{\text{full}}(N) = \$0.1032\,N, \qquad C_{\text{tablet}}(N) = \$0.040 + \$0.0147\,N, \qquad C_{\text{scroll}}(N) = \$0.040 + \$0.0322\,N,$$

where the $0.040 is the mean metered WOS ingestion per 20K-token record. Because full context re-reads the entire record on every query while memory ingests once and then retrieves a bounded context, the break-even point is immediate: **Tablet 1** is cheaper than full context from the first query ($N^* = 1$), and **Scroll 1** from the first query as well (Figure 5, Table 3). By ten queries against one record, **Tablet 1** has cost 14% of the full-context total and **Scroll 1** 30%; by fifty queries, 3% and 6%. These are projections from measured slopes and can be re-priced by applying a new table to the raw ledger.

5.7 Resource use and latency

Retrieval latency is bounded and low; reader latency dominates end-to-end time and is a property of the reader model, not the memory tier (Table 4). **Tablet 1** retrieval has a much lower median than **Scroll 1** retrieval (about 314–351 ms vs. 1,413–1,471 ms), consistent with **Scroll 1** doing more work to expand and merge passages. One-time ingestion, at real WOS billing, was about $0.20–

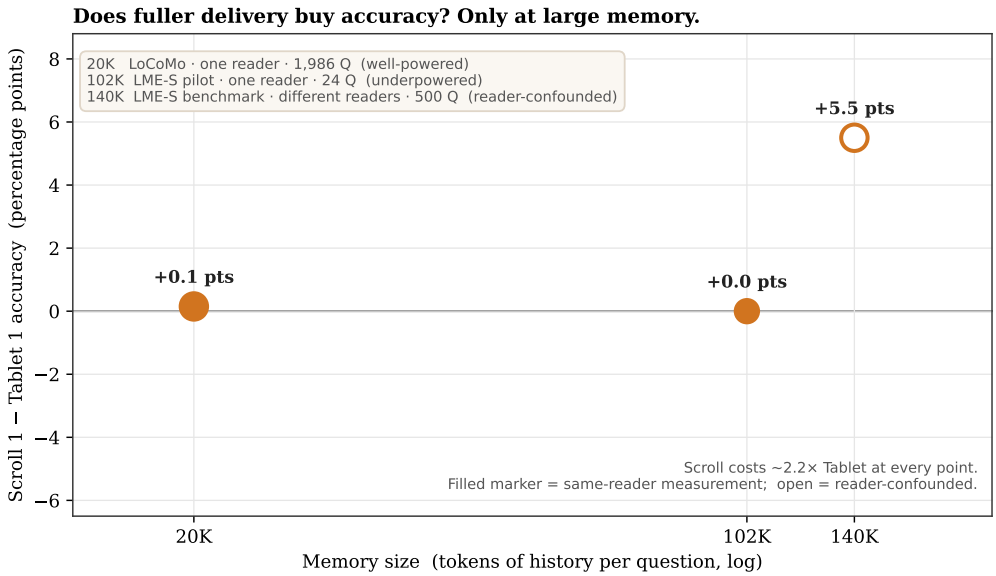

Figure 3: The crossover. Scroll 1 minus Tablet 1 accuracy at three memory sizes. Filled markers are same-reader measurements; the open marker at 140K is a comparison confounded by reader. Fuller delivery buys no accuracy at small memory and a measurable gain appears at large memory, while Scroll costs ∼2.2× Tablet throughout.

Table 3: Cumulative cost (USD) at N queries against one LoCoMo record, from the measured slopes.

N	Full context	Tablet 1	Tablet saving	Scroll 1	Scroll saving
1	$0.10	$0.05	47%	$0.07	30%
10	$1.03	$0.19	82%	$0.36	65%
50	$5.16	$0.78	85%	$1.65	68%
100	$10.32	$1.51	85%	$3.26	68%

0.21 per 100K-token record and about $0.03–0.05 per 20K-token LoCoMo record. Latencies here are measured on a subscription execution path and should not be read as a service-level guarantee.

Table 4: Latency percentiles (ms) over successful calls, LongMemEval-S pilot. Retrieval is WOS; reader is GPT-5.6 Sol.

Condition	Retrieval p50	Retrieval p95	Reader p50	Reader p95
Tablet 1	314	4,797	2,977	7,350
Scroll 1	1,471	2,764	3,945	8,898
Full context	—	—	4,629	7,877

6 Interpretation: deliver more only when it pays

The two findings compose into one rule. First, at long records memory beats reading the full history on both cost and accuracy—95.8% versus 79.2% at ∼102K, at a small fraction of the cost; this is why a memory tier exists at all. Second, *within* memory, the choice of how much to deliver is a function of memory size. When the record is small, retrieval alone captures what the reader needs and fuller delivery is redundant, so the minimal tier (**Tablet 1**) is the efficient choice. When the

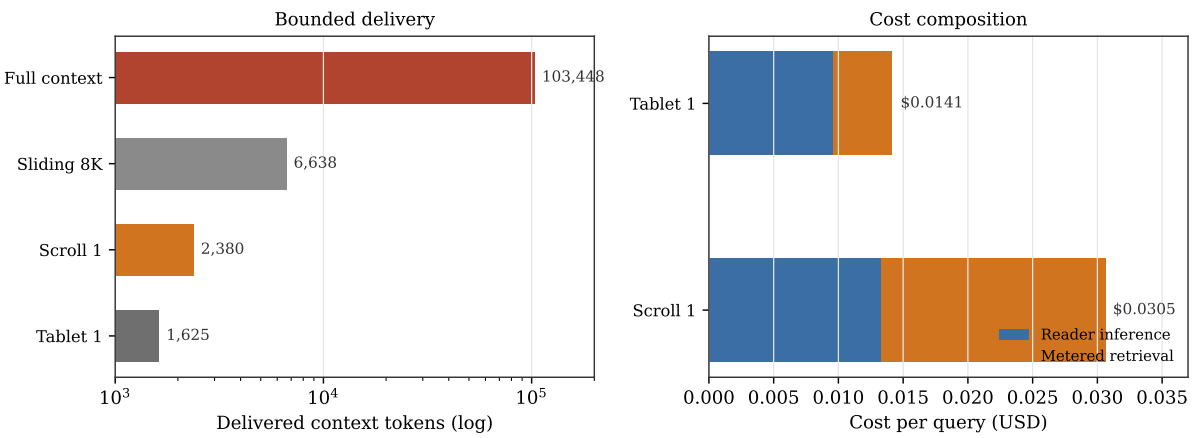

Figure 4: Left: delivered context tokens at ∼102K (log scale); both memory tiers are two orders of magnitude below the full record. Right: per-query cost split into reader inference and metered retrieval. Retrieval is Scroll 1's larger share; reader inference is Tablet 1's.

record is large, required evidence is spread across many sessions, and a single-pass delivery leaves neighboring pieces behind; the fuller tier (**Scroll 1**) brings them along and converts found memory into correct answers, paying for its extra cost on exactly the questions—preference, temporal, multi-session—where one missing piece decides the outcome.

The honest boundary of this claim is the reader confound and the two-benchmark design. What we have is two operating points at genuinely different memory scales, on two different benchmarks, whose ordering is consistent with a crossover and whose mechanism is visible in the category structure. What we do not have is a single benchmark measured at several controlled record lengths under one reader, which is what would prove the crossover is monotonic and locate the length at which the two tiers cross. We treat the crossover as a strong, mechanistically-grounded hypothesis on the large-memory side and a settled result on the small-memory side.

7 Limitations

This is an early-stage study, and its limitations are numerous and material; we state them with the same weight as the results. Several of the central claims are not yet established and require the further measurements noted below. Read the crossover as a preliminary observation, not a proven law.

Single-run measurements. The LoCoMo and LME-S pilot numbers are each from a single run, unlike the five-run first study [9]. Run-to-run reader nondeterminism (roughly ± 1 point in [9]) is therefore not quantified here, and the new figures should be read as single-sample estimates pending multi-run repetition.

Different readers on the large-memory quality comparison. The +5.5-point **Scroll 1** advantage comes from the 500-question benchmark [9], where **Tablet 1** used Claude Opus 4.8 and **Scroll 1** used GPT-5.5. Memory size, delivery policy, and reader all differ between that study's two arms. The same-reader runs we conducted (LoCoMo, and the 24-question LME-S pilot) show ties. A definitive crossover requires re-running both tiers at 140K under one reader and one prompt.

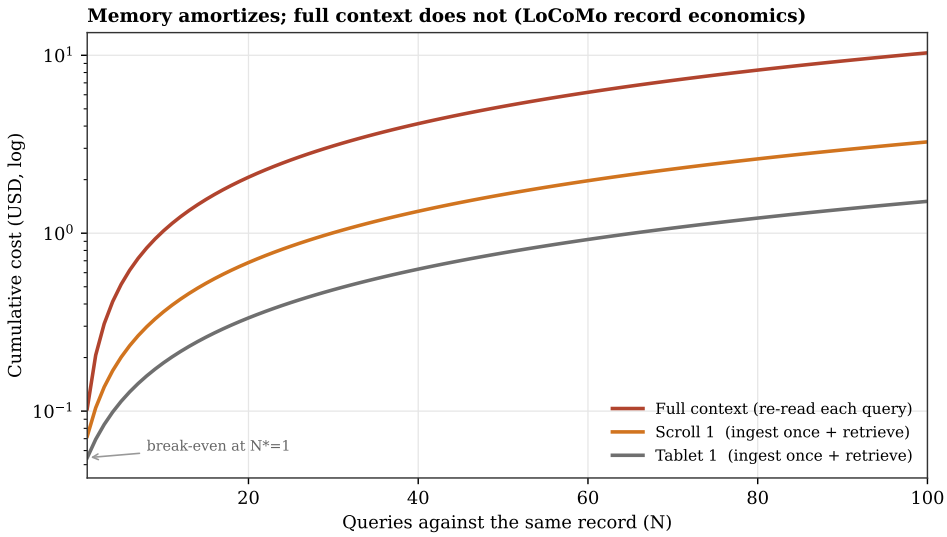

Figure 5: Cumulative cost against the number of queries per record (LoCoMo economics, log y). Memory ingests once and retrieves a bounded context; full context pays to re-read the whole record every query. Break-even is at the first query.

Two benchmarks, not one length axis. LoCoMo (∼20K) and LongMemEval-S (∼140K) differ in question distribution as well as size, so "memory size" and "task type" move together. A controlled length sweep on one benchmark (for example 10K, 30K, 60K, 100K, 140K) is needed to draw the curve and identify the crossing point.

Small primary pilot. The matched lifecycle sample at 102K is 24 questions; the +16.6-point memory-versus-full-context difference is descriptive, and a preregistered confidence interval requires the full 500-question sample.

Same-family judge; projected costs; in-distribution. Reader (GPT-5.6 Sol) and judge (GPT-5.6 Luna) share a provider family; agreement with an independent GPT-4o judge is follow-up work. Reader and judge costs are projected from measured tokens, not invoices, and the cache scenarios are counterfactuals. **Tablet 1** and **Scroll 1** were previously tuned on LongMemEval-S, so those numbers are in-distribution; held-out confirmation (LongMemEval-V2 [5]) is planned. Contamination (readers having seen either benchmark in pretraining) cannot be ruled out but applies to every condition equally.

Vendor evaluation. The author owns Wontopos and evaluated Wontopos products. The published ledger, prompts, and per-question scores support external rerun but not independent validation.

8 Reproducibility

Published: both benchmarks (public datasets), the per-condition and per-question scores (Appendix A), the cost ledger (Appendix B), the verbatim reader and judge prompts (Appendix C), and the report pages at wontopos.com/research. Not published: engine internals (embedding and reranking configuration). A third party can re-run the identical protocol at the black-box boundary with a WOS API key and their own LLM keys. **Scroll 1** is invoked with the `X-WOS-Model: scroll-1` header; LLM keys are passed per request and never stored. Per-question retrieval

dumps are available on request.

9 Conflict of Interest

The author is the founder and sole owner of Wontopos, and **Tablet 1** and **Scroll 1** are Wontopos products. Publication is tied to product launch. To offset this we publish every per-question score, every prompt, and the full ledger; we report no best-of numbers; and we keep a third-party re-verification path open (§8).

10 Conclusion

Under one reader and a published ledger, the tiering question has a clean answer that is not "one tier is better." At long records memory beats reading the full history on both cost and accuracy. And within memory, how much to deliver depends on memory size: at $\sim$20K tokens Tablet 1 and Scroll 1 tie on accuracy and Tablet 1 costs 2.2$\times$ less, so Tablet 1 wins; at $\sim$140K tokens Scroll 1 leads by 5.5 points on the categories where evidence is scattered, so Scroll 1 wins. The small-memory side is a well-powered, same-reader result; the large-memory side is directionally supported but reader-confounded, and the same-reader rerun plus a controlled length sweep are the next measurements. The rule for practitioners is to deliver more only when the record is large enough that fuller delivery buys accuracy—and to stop paying for it when it does not. We publish these numbers as a preliminary, honestly-limited snapshot: the crossover is a hypothesis we find compelling but have not yet proven, the new measurements are single-run, and the large-memory arm awaits a same-reader rerun. We invite the replication that would confirm or refute it, and will treat these results as provisional until it exists.

References

[1] Wu, D. et al. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. arXiv:2410.10813, ICLR 2025.

[2] Maharana, A. et al. Evaluating Very Long-Term Conversational Memory of LLM Agents (LoCoMo). arXiv:2402.17753.

[3] Liu, N. F. et al. Lost in the Middle: How Language Models Use Long Contexts. TACL 2024. arXiv:2307.03172.

[4] MemPalace benchmark methodology issue #29. (Criticism of mixing retrieval scores with end-to-end QA scores.)

[5] Wu, D. et al. LongMemEval-V2: Evaluating Long-Term Agent Memory Toward Experienced Colleagues. arXiv:2605.12493, 2026.

[6] Wontopos. Pricing and Scroll 1 Model. Public product documentation, accessed July 2026. wontopos.com/en/why.

[7] Packer, C. et al. MemGPT: Towards LLMs as Operating Systems. arXiv:2310.08560, 2023.

[8] Lewis, P. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. NeurIPS 2020. arXiv:2005.11401.

[9] Kim Sunwoo. What Remains After Retrieval Saturates: Measuring the Delivery Gap in AI Memory Systems for Tablet 1 and Scroll 1. Wontopos Technical Report, July 7, 2026.

A Per-condition accuracy

LoCoMo (1,986 Q): full 82.98%, oracle 84.74%, scroll 76.69%, tablet 76.54%, sliding 47.53%, random 24.22%. LME-S stratified pilot (24 Q): full 79.17%, oracle 91.67%, scroll 95.83%, tablet 95.83%, sliding 25.00%, random 8.33%. LME-S single-category slice (20 ss-user Q): full 100%, oracle 100%, scroll 100%, tablet 95%, sliding 15%, random 0%. Full 500-question benchmark [9]: tablet 85.2%, scroll 90.7%.

B Cost ledger (per-query aggregates, USD)

LoCoMo: reader/retrieve — full $0.1032/—, scroll $0.0138/$0.0185, tablet $0.0098/$0.0049; judge excluded; ingest $0.026–0.049/record. LME-S pilot: full $0.5185/—, scroll $0.0133/$0.0173, tablet $0.0096/$0.0045; ingest ∼$0.20–0.21/record. Codex (reader/judge) rows are projected at the official July 2026 API table; WOS rows are metered billing; provider-unreached failures are excluded from cost and latency aggregates.

C Verbatim evaluation prompts

Reader (all conditions): "Answer the question using ONLY the context below … If the context gives conflicting values for the same fact, prefer the most recent. If the answer is not in the context, say you don't know. Answer concisely, in the same language as the question." Judge (temperature 0): "I will give you a question, the correct answer, and a model's response. {rule} Respond with ONLY 'yes' or 'no'." The judge receives the official LongMemEval per-category rule for the item's category regardless of condition.

D Raw per-question verdicts (LME-S stratified pilot, 24 questions)

Per-question correctness (1 = correct, 0 = incorrect) for all six conditions, single run. Tablet 1 and Scroll 1 differ on no question (identical vectors; both miss only `57f827a0`). Columns: Tab = Tablet 1, Scr = Scroll 1, Full = full context, Ora = oracle, Sld = sliding 8K, Rnd = random.

qid	category	Tab	Scr	Full	Ora	Sld	Rnd
1568498a	ss-asst	1	1	1	1	0	0
18bc8abd	update	1	1	1	1	1	0
195a1a1b	pref	1	1	1	1	1	0
1cea1afa	update	1	1	1	1	1	0
311778f1	ss-user	1	1	1	1	0	0
4d6b87c8	update	1	1	1	1	0	0
57f827a0	pref	0	0	0	1	0	0
58bf7951	ss-user	1	1	1	1	0	0
60bf93ed	multi	1	1	1	1	0	0
75832dbd	pref	1	1	0	1	0	0
7a8d0b71	ss-asst	1	1	1	1	0	0
80ec1f4f_abs	multi	1	1	1	1	1	1
852ce960	update	1	1	0	0	1	0
89527b6b	ss-asst	1	1	1	1	0	0
95bcc1c8	ss-user	1	1	1	1	0	0
afdc33df	pref	1	1	1	1	0	0
b46e15ee	temporal	1	1	0	1	0	0
c4f10528	ss-asst	1	1	1	1	0	0
f4f1d8a4_abs	ss-user	1	1	1	1	1	1
gpt4_2ba83207	multi	1	1	1	1	0	0
gpt4_7fce9456	multi	1	1	0	0	0	0
gpt4_93f6379c	temporal	1	1	1	1	0	0
gpt4_b5700ca0	temporal	1	1	1	1	0	0
gpt4_f49edff3	temporal	1	1	1	1	0	0
correct / 24		**23**	**23**	19	22	6	2