Benchmark Report: Scroll 2

Wontopos Scroll 2:
Score, Evidence Recall and Coverage
for a Long-Term Memory Engine

Sunwoo Kim • Wontopos L.L.C.

WONTOPOS

5 November 2026

Wontopos Scroll 2: Score, Evidence Recall and Coverage for a Long-Term Memory Engine

Sunwoo Kim
Wontopos L.L.C.
official@wontopos.com

5 November 2026

Abstract

We measure Scroll 2, a hosted long-term memory engine for language-model applications, against its predecessor Tablet 2 on BEAM-1M. Every run is reported with three numbers: the benchmark's official score; evidence recall, the share of the benchmark's evidence turns that the engine handed to the reader; and Coverage, whether the handed memories hold what each rubric item needs, judged from the memories alone and never from the answer. With the same reader (Claude Opus 5 at maximum effort) and the same judge (gpt-4.1-mini), Scroll 2 scores 72.10 under BEAM's own scoring against 67.49 for Tablet 2, each the mean of five runs, a paired difference of +4.6 points (95% interval +3.4 to +5.9). Coverage locates the difference: with about seventy memories handed over the two engines hold nearly the same facts (84.6 and 84.2 on fact retrieval), and with the first twenty Scroll 2 holds more (78.0 and 73.7). Given memory tools, the reader takes 57.1 memories per question instead of 69.9 and scores 73.18, the mean of four runs. Sleep, an experimental feature in which a model writes short files about the user between conversations, leaves the total where it was and raises instruction following and summarization, at $4.29 per million tokens read with Claude Sonnet 5.5. A larger store hands back little more: a question costs $0.019 with a million-token conversation stored, and a conversation turn with Claude Opus 5.5 costs 4.4 to 4.8 cents at every length measured, where the whole conversation in context costs $5.02 a turn at a million tokens once its cache has expired. Against Scroll 1.2's prices, Scroll 2 costs up to 24.4% less: a memory is stored once, at 23.8% more, and recalled many times, at 24.4% less a question, and a bill is lower once the text recalled passes about half of the text stored. Against Claude Managed Agents memory on BEAM-100K, as of 14 October 2026 and with the same small reader, Scroll 2 scores +15.6 points higher (95% interval +11.0 to +20.4), and the order holds under a second judge and a second reader instruction; a reader given the whole conversation, which at that length still fits its window, scores 8.1 points above Scroll 2 at about 8 times the price per question. We release the harness, prompts and per-question records.

1 Introduction

A memory engine for a language-model application stores what was said and, when asked, hands some of it back. Its quality is usually reported as the end-to-end score of a benchmark: a reader model receives the handed memories, answers, and a judge grades the answer. That score mixes three things. It depends on what the engine handed over, on how well the reader used it, and on how the benchmark is scored.

Two observations from our own runs made the mixture concrete. First, changing only the reader moves the score while the engine's delivery stays fixed: with the same engine and the same delivered evidence, replacing Claude Opus 5 with Claude Opus 5.5 left the total within noise (0.0 points) but moved summarization by −7.0 (Section 5.10). Second, changing only the scoring of one ability reorders published results: most published BEAM figures grade event ordering by rubric, while the benchmark's authors score it by rank correlation, and the two readings of the same Scroll 2 runs are 76.20 and 72.10 (Section 5.4).

We therefore report three numbers for every run, and keep the reader and the judge as different models:

- **Score**: the benchmark's official score, computed the way its authors compute it, with the alternative readings beside it.
- **Evidence recall**: the share of the turns the benchmark marks as evidence that the engine handed to the reader. No language model is involved.
- **Coverage**: a judge reads only the memories the reader received and decides, rubric item by rubric item, whether they hold what the answer needs. The answer is never shown.

The findings, all on BEAM-1M [Tavakoli et al., 2025] unless stated:

1. With the same reader and judge, Scroll 2 scores +4.6 points above Tablet 2 (95% interval +3.4 to +5.9), and +5.7 with memory tools.
2. The engines differ less in what they hold than in where they put it. With about seventy memories the fact-retrieval Coverage of the two is 84.6 and 84.2; with the first twenty it is 78.0 and 73.7. On questions that gather memories from several sessions the gap remains at seventy (+10.3 on multi-session reasoning, +9.6 on summarization).
3. The reader leaves part of what it receives unused. In 66 of 630 questions the memories held what the answer needed and the answer still fell short. Had the reader used every such question, the score would rise from 76.8 to 82.5.
4. Memory tools change how much is fetched more than how well the benchmark is answered. The reader took 57.1 memories on average, from 34 on knowledge-update questions to 102 on summarization, and the total score moved by +1.1 (95% interval −0.2 to +2.3).
5. Sleep, in its first version, does not move the total (−0.1, 95% interval −1.9 to +1.7). It raises instruction following (+7.1) and summarization (+6.3) and lowers abstention and information extraction, with intervals that reach zero. It costs $4.29 per million tokens read with Claude Sonnet 5.5 (Section 5.9).
6. Past the first thousand memories, a larger store hands back little more. From 1,104 to 61,215 memories the text handed back per question rises by 12%, to 3,351 tokens, and a question costs $0.019 (Section 8). A conversation turn costs the same at every length measured, while the whole conversation in context costs $5.02 a turn at a million tokens once its cache has expired (Section 8.3).
7. Against Scroll 1.2's prices, Scroll 2 costs up to 24.4% less. Storing costs 23.8% more and answering a question 24.4% less, so a memory stored once pays back once the text recalled passes about half of the text stored. A conversation turn costs 9.4% less, and BEAM-1M stored once and asked 100 times costs 20.8% less (Section 8).

8. Against an agent that writes its own memory (Claude Managed Agents, as of 14 October 2026), on BEAM-100K with the same reader, Scroll 2 scores +15.6 points higher (+11.0 to +20.4), most on details, and +20.8 on questions whose evidence the user said. The agent's small memory is ahead on abstention. With the whole conversation in its context, which at this length still fits, the same reader scores 8.1 points above Scroll 2, at about 8 times the price per question (Section 6).

2 Related work

Memory for conversational agents. Long-term memory for chat agents has been built as an operating-system-like hierarchy managed by the model [Packer et al., 2023], as an extraction-and-update pipeline [Chhikara et al., 2025], and as a store with a forgetting curve [Zhong et al., 2024]. Retrieval-augmented generation [Lewis et al., 2020] and its iterative forms, which interleave retrieval with reasoning [Trivedi et al., 2023] or ask follow-up questions [Press et al., 2023], are the closest relatives of the memory tools measured here.

Benchmarks. Multi-session chat [Xu et al., 2022], LoCoMo [Maharana et al., 2024] and LongMemEval [Wu et al., 2025] test recall over long conversations. BEAM [Tavakoli et al., 2025] extends the setting to one and ten million tokens and ten abilities, and grades each answer against rubric items. Long-context models degrade with position and length [Liu et al., 2024, Hsieh et al., 2024], which is one reason an engine's ordering matters even when a reader could hold everything.

Judging. Language-model judges agree well with people on many tasks [Zheng et al., 2023] and carry measurable biases [Wang et al., 2024]. Coverage is close in spirit to nugget-based evaluation [Pradeep et al., 2025]: it asks whether the units an answer needs are present, here in the retrieved memories rather than in a generated answer.

3 System under measurement

Scroll 2 is the memory engine behind the WOS hosted service. An application stores conversation turns through the public SDK and later asks with text. The engine replies with numbered memories, most relevant first. In every run reported here, except the sleep run of Section 5.9, no language model processed the stored conversations; only the reader saw memories. The engine is proprietary, and this report describes what an application sends and receives and measures the result.

This report measures two ways an application can use the engine.

Single call. The application asks once, takes the first page of twenty memories and asks for three further pages. Duplicates are removed, and on BEAM this delivers 69.9 memories per question on average. Tablet 2's headline configuration, which re-asks three times, delivers a similar number (70.16) [Kim, 2026].

Memory tools. The application asks once for the first page of twenty and gives the reader three tools: the next page, the conversation around a numbered memory, and a new search in the reader's own words. Memories the reader already holds are not sent again. The reader decides when to stop.

Tablet 2, released on 29 August 2026 and measured in Kim [2026], is the previous generation. Both engines answer from the same BEAM corpus, loaded with dates and speakers.

4 Evaluation protocol

4.1 Three numbers

Score. BEAM grades each answer against rubric items, each judged 1, 0.5 or 0, and the question's score is their mean. Event ordering is scored by Kendall's τ_b [Kendall, 1945] between the answered and the reference order, normalised to $(\tau + 1)/2$, as in the benchmark's paper. Because most published figures grade event ordering by rubric instead, we also report that reading, and the 630 questions without event ordering.

Evidence recall. For 555 questions BEAM marks the turns that contain the evidence. Each delivered memory is traced to the conversation turn it was stored from, and recall is the share of evidence turns delivered. Event ordering is left out because its evidence is given as sessions, and abstention questions have none. No language model is involved.

Coverage. A judge receives the question, the reference answer, the rubric items and the memories the reader received, and never the reader's answer. For each item it decides whether the memories hold the facts an answer meeting that item would need: 1, 0.5 or 0. For instruction-following and preference-following questions the item becomes whether the memories hold the user's own instruction or preference, as BEAM records it, since the rubric there grades the form of the answer rather than a fact. Abstention questions are left out, as nothing should be there, which leaves 630. The judge prompt is in Appendix A.

4.2 Reader and judge

On BEAM-1M the reader and the judge are always different models. The reader for every BEAM-1M run, except the newer reader of Section 5.10, is Claude Opus 5 at maximum effort, the reader of Tablet 2's published runs, called through the Claude Code command line without tools other than the memory tools. The judge is gpt-4.1-mini at temperature 0 with the same judge prompt as Tablet 2's published runs; BEAM's paper does not name a judge model, and this is the default of its released code. The single-call configuration was run five times on one engine build, and the memory-tools configuration four times (30 September, 18, 19 and 20–21 October), each on the engine of its date; the reader and the judge are the same throughout. The protocol is stated as of 2 October 2026.

Why Claude Opus 5. A memory system's score mixes the engine and the reader, so a new engine measured with a newer model leaves open which of the two improved. Measuring Scroll 2 with the reader Tablet 2 was measured with answers that question: the difference between the two is the engine's. Claude Opus 5.5 was also tried as the reader (Section 5.10). It did not raise the total, and on summarization, where a summary is graded on the points it keeps, the difference was -7.0 points. The reader does not change what the engine hands over: evidence recall was the same with both. In use, an application's own model answers from the same memories. As in this report, later reports will give evidence recall beside the score, so that a change of reader can be told apart from a change of engine.

4.3 Paired comparison

Tablet 2 was measured five times with the same reader and judge [Kim, 2026]; its five scores range from 67.15 to 67.75 (standard deviation 0.22). Scroll 2's five single-call runs range from 71.55 to

72.71 (standard deviation 0.51). We compare per question against Tablet 2's five-run mean, using Scroll 2's five-run mean for the single call and report the mean difference with a 95% interval from 2,000 bootstrap resamples of questions [Efron, 1979].

4.4 What the reader sees

The reader receives the memory text only, without dates and without speakers, in the same format as Tablet 2's runs. Appendix A gives the prompt verbatim.

5 BEAM-1M

5.1 Setup

BEAM's one-million-token split has 35 conversations, 74,630 turns and 700 questions, 70 for each of ten abilities.

5.2 Score

Table 1: Score on BEAM-1M, 700 questions. Reader Claude Opus 5 (maximum effort), judge gpt-4.1-mini. Scroll 2's single call and Tablet 2 are each the mean of five runs; memory tools, the mean of four runs.

	Event ordering by τ (BEAM's scoring)	Event ordering by rubric	630 without event ordering
Scroll 2 (five runs)	72.10	76.20	77.45
Scroll 2 + memory tools (four runs)	73.18	77.56	78.52
Tablet 2 (five runs)	67.49	71.32	72.37

Table 2: Every Scroll 2 run behind Table 1 (dates in 2026). Each memory-tools run used the engine of its date.

Configuration	Date	Event ordering by τ	by rubric
Single call	28 Sep	71.55	75.55
Single call	30 Sep	72.71	76.77
Single call	6 Oct	72.68	76.97
Single call	6 Oct	72.02	76.10
Single call	6 Oct	71.55	75.59
Memory tools	30 Sep	72.98	77.36
Memory tools	18 Oct	72.77	76.90
Memory tools	19 Oct	73.32	77.93
Memory tools	20–21 Oct	73.67	78.06
Single call, mean (sd)		72.10 (0.51)	76.20 (0.58)
Memory tools, mean (sd)		73.18 (0.34)	77.56 (0.46)

Table 3: Paired difference against Tablet 2, per question, with 95% bootstrap intervals.

	Event ordering by τ	Event ordering by rubric
Scroll 2	+4.6 [+3.4, +5.9]	+4.9 [+3.6, +6.2]
Scroll 2 + memory tools	+5.7 [+4.1, +7.2]	+6.2 [+4.7, +7.7]

The difference between the two Scroll 2 configurations is +1.1 (95% interval −0.2 to +2.3). Section 5.8 looks inside it.

5.3 By ability

Table 4: Score by ability. Event ordering is graded by rubric here; its τ reading is in the last row. Scroll 2's single call and Tablet 2 are five-run means; memory tools, the mean of four runs.

	Scroll 2	+ memory tools	Tablet 2
Information extraction	84.5	88.6	78.2
Multi-session reasoning	73.7	75.7	63.8
Knowledge update	81.5	75.6	75.4
Temporal reasoning	51.2	54.2	54.8
Abstention	86.7	86.6	83.4
Contradiction resolution	84.0	80.0	78.0
Event ordering	64.9	69.0	61.9
Instruction following	79.2	86.6	74.7
Preference following	90.8	89.2	90.4
Summarization	65.4	70.1	52.5
Event ordering, τ	23.9	25.2	23.6

Scroll 2 is ahead of Tablet 2 on nine of ten abilities. The largest gaps are on summarization and multi-session reasoning, the two abilities that need memories from several sessions at once. Temporal reasoning is the exception (51.2 against 54.8).

5.4 Published figures

Table 5: BEAM-1M overall scores as published by each team. Readers, judges and prompts differ, so this is not a controlled comparison. The event-ordering column records how each team's public code scores that ability.

System	Score	Reader	Event ordering
Exabase M-1 [Exabase, 2026]	75.0	Gemini 3 Flash	rubric
Hindsight [Hindsight, 2026]	73.9	Gemini 3 Pro	rubric
Mem0 [Mem0, 2026]	64.1	as published	rubric
Honcho [Hindsight, 2026]	63.1	as published	τ, matched events only
Scroll 2 + memory tools (four runs)	77.56	Claude Opus 5	rubric
	73.18		τ, BEAM's scoring
Scroll 2 (five runs)	76.20	Claude Opus 5	rubric
	72.10		τ, BEAM's scoring
Tablet 2 (five runs)	71.32	Claude Opus 5	rubric
	67.49		τ, BEAM's scoring

Of the published figures we found, one applies Kendall's τ to event ordering, and it scores only the events the answer matched; the others grade event ordering by rubric. BEAM's scoring penalises missing events, which lowers every system's figure by about four points (Table 1). We give both readings for our systems so that each published figure can be set against the one computed the same way.

5.5 Evidence recall

Table 6: Evidence recall: share of BEAM's evidence turns delivered, by number of memories counted from the top (555 questions; event ordering and abstention left out).

	First 20	40	60	80	All delivered	Memories delivered
Scroll 2	38.7	47.1	51.4	52.9	52.9	69.9
Scroll 2 + memory tools	37.2	42.6	44.3	45.5	46.0	55.8
Tablet 2 (replay)					47.4	68.7

On the 140 fact-retrieval questions (information extraction and knowledge update), Scroll 2's recall is 59.2 within the first twenty memories and 75.0 within eighty. Tablet 2's recall comes from a replay of its retrieval described in Section 9.

With memory tools, recall is lower (46.0 against 52.9) while the score is not. The reader fetched fewer memories and chose them by its own searches, which do not always return BEAM's marked turns even when they return an answer.

5.6 Coverage

Table 7: Coverage by number of memories handed over. Tablet 2 is measured on the 625 questions its replay covers; Scroll 2 is shown on the same questions. "Every item" is the share of questions whose memories hold everything every rubric item needs.

	Fact retrieval (140)		All with an answer (625)	
Memories	Scroll 2	Tablet 2	Scroll 2	Tablet 2
First 20	78.0	73.7	57.7	51.8
First 40	83.2	80.7	64.5	59.8
All (about 70)	84.6	84.2	69.0	65.4
Every item, all	74.3	72.9	47.4	45.4

With every delivered memory counted, the two engines hold nearly the same facts for fact-retrieval questions. The difference is in order: within the first twenty the gap is +4.3 points on fact retrieval and +5.9 on all questions. Across all questions with an answer, Scroll 2 holds more at every depth, and per question it holds more on 157, less on 93, and the same on 375.

Table 8: Coverage by ability on the same questions, all delivered memories and the first twenty.

	All delivered			First 20		
	Scroll 2	Tablet 2	gap	Scroll 2	Tablet 2	gap
Multi-session reasoning	79.9	69.7	+10.3	70.1	52.9	+17.1
Summarization	59.3	49.6	+9.6	44.4	30.6	+13.8
Instruction following	71.4	63.6	+7.9	50.7	50.7	0.0
Event ordering	52.2	47.3	+5.0	30.6	28.1	+2.5
Information extraction	83.8	79.1	+4.6	76.4	68.5	+7.9
Preference following	39.1	37.7	+1.4	33.3	32.6	+0.7
Contradiction resolution	81.4	81.8	−0.4	71.6	74.3	−2.7
Temporal reasoning	67.5	68.9	−1.4	61.8	48.3	+13.5
Knowledge update	85.4	89.3	−3.9	79.6	78.9	+0.7

Tablet 2 holds more on knowledge update when everything is counted; Scroll 2 still scores higher there (81.5 against 75.4), which Coverage alone does not explain, since Coverage asks whether the latest value is present and not whether an outdated one is also present.

Coverage against score. Table 9 sets Scroll 2's Coverage beside its score. Where the two part, the cause is visible. Preference following scores 91.1 with Coverage 40.0: the memories held the user's stated preference in fewer than half of the questions, and the answers met the rubric anyway, because the rubric grades the form of the answer and a careful answer often meets it without the preference. Temporal reasoning has Coverage 67.5 and scores 53.0: the judge found what the answer needs in about two thirds of the items, and the reader computed the answer less often. The reader saw the memory text without the date each memory was stored with (Section 4); dates written inside the text remained, so part of this loss may belong to the format.

Table 9: Scroll 2, the 30 September run whose delivered memories were graded: Coverage and score by ability (630 questions).

	Coverage	Score
Preference following	40.0	91.1
Information extraction	83.8	84.6
Contradiction resolution	81.4	83.2
Knowledge update	85.4	83.6
Instruction following	71.4	78.7
Multi-session reasoning	79.9	73.3
Temporal reasoning	67.5	53.0
Summarization	59.9	65.5
Event ordering (rubric)	52.2	64.7

What the reader left unused. Counting a question as covered when its Coverage is at least 0.75 and as answered when its score is at least 0.75 gives four groups (Table 10). The second group, covered and answered short, is the reader's share of the loss. Had the reader answered each of those questions fully, Scroll 2's score would rise from 76.8 to 82.5, and with memory tools from 77.4 to 82.2.

Table 10: Memories covered against answer right, 630 questions (threshold 0.75 on both). Scroll 2: the 30 September run.

	Memories covered		Not covered	
	answer right	answer short	answer right	answer short
Scroll 2	306	66	105	153
Scroll 2 + memory tools	314	58	104	154

Judge repeatability. The 140 fact-retrieval questions were graded twice at twenty and forty memories in separate runs; the two gradings differ by at most 0.4 points.

5.7 Context handed to the reader

Table 11: Memory text handed to the reader per question, counted with the o200k_base tokeniser (medians), the count used in Kim [2026].

	Memories	Tokens (median)	Tokens (mean)
Scroll 2, first page only	20	935	
Scroll 2	69.9	3,354	3,426
Scroll 2 + memory tools	57.1	2,662	3,009
Tablet 2, re-ask off [Kim, 2026]	about 20	776	
Tablet 2, re-ask three times [Kim, 2026]	70.16	2,484	

Scroll 2 hands over +35% more text than Tablet 2 at the same number of memories, of which 286 tokens are the line numbers that let the reader and the tools refer to a memory. With memory

tools the reader receives +7% more than Tablet 2. Counting Tablet 2's replay the same way gives 2,453, within 1.2% of the published figure, which checks that the two counts are comparable.

5.8 Memory tools on a single-question benchmark

In the 30 September run with memory tools, the reader made 2.7 calls per question (median 2). It answered 134 questions (19%) from the first page alone. Of all calls, 1,377 were new searches, 478 looked at the conversation around a memory, and 46 asked for the next page.

Table 12: Memory tools against the single call, by ability: memories the reader held at the end, calls per question and questions answered without a call (of 70), from the 30 September run; and the score difference (the mean of the memory-tools runs minus the mean of the single-call runs).

	Memories	Calls	No call	Score difference
Summarization	102	4.4	2	+4.7
Event ordering	81	4.8	0	+4.0
Multi-session reasoning	69	3.0	8	+2.0
Abstention	66	3.4	2	−0.1
Temporal reasoning	47	2.2	23	+3.0
Information extraction	45	2.1	23	+4.1
Instruction following	45	1.9	18	+7.4
Contradiction resolution	42	2.2	12	−4.1
Preference following	40	1.4	29	−1.6
Knowledge update	34	1.8	17	−5.9

The reader fetched according to the question. On summarization and event ordering it took more than the single call delivers, and scored higher. On fact questions it stopped at thirty to fifty memories. The largest loss is on knowledge update, where the reader took the fewest memories (34) and may have stopped before the latest value; contradiction resolution, the other question about which value holds now, also fell (−4.1). On abstention, where nothing should be found, it searched about as much as on multi-session reasoning, and the score did not change. None of this has been checked question by question.

BEAM asks one question per conversation, so the property the tools are built for, carrying what has been found from one turn to the next, is not exercised. Section 7 measures a dialogue.

5.9 Sleep

Sleep is an experimental feature of Scroll 2, there for those who want to use it. Between conversations, a language model the application chooses reads what the store holds and writes a few short files about the user; when the application answers, the memory tools can open them beside the memories. We measured it once on BEAM-1M with memory tools and the reader and judge of the runs above: one run with sleep off and one with sleep written by Claude Sonnet 5.5 at high effort, on 20 and 21 October 2026 (Table 13).

Table 13: Sleep on BEAM-1M with memory tools, one run each, 700 questions. Differences per question, 95% intervals from 2,000 paired bootstrap resamples.

	Event ordering by τ (BEAM's scoring)	Event ordering by rubric
Sleep off	73.67	78.06
Sleep on (Claude Sonnet 5.5)	73.55	78.57
Difference	−0.1 (−1.9 to +1.7)	+0.5 (−1.3 to +2.4)

The sleep-off run is the fourth memory-tools run of Table 2 and is part of their mean; here it is sleep's control, run on the same days with the same engine.

The total does not change. The abilities move both ways. Three rose, with intervals above zero: instruction following +7.1 (+1.3 to +14.5), summarization +6.3 (+1.4 to +11.1) and event ordering by rubric +5.5 (+1.5 to +9.5). Two fell, with intervals that reach zero: abstention −7.9 (−16.4 to 0.0) and information extraction −4.7 (−10.8 to +0.2). In the transcripts, the files carry instructions the user gave once, which the reader then follows; they also carry plausible content that leads the reader to answer where the record holds no answer.

Cost. With sleep on, the reader opened 0.69 files per question and was handed 3,960 tokens of memory text against 3,338, in 6.7 requests against 5.0: 2.44 cents per question against 2.05, 19% more (memory text at $6 per million tokens and $0.0001 a request; the harness calls only the SDK and does not see the service's bill). The reader's prompt grew from 924 to 1,817 tokens and its cost by 29%. Sleep itself, at Claude Sonnet 5.5's API list prices ($2 input and $10 output per million tokens, without prompt caching), comes to $4.29 per million tokens of stored memory read, from 1,789 thousand input and 71 thousand output tokens and about 289 seconds, averaged over the 35 conversations; service fees came to $0.009.

Where it fits. The scores say what to turn sleep on for and what to leave it off for (Table 14). The uses are our reading of the ability scores, not separate measurements; the two decreases have intervals that reach zero.

Table 14: Where this version of sleep helps and where it does not, read from the ability scores above.

Use	Sleep	Why
Personal assistant or companion chatbot	On	Keeps a request the user made once (instruction following +7.1); sums up many sessions (summarization +6.3)
Agent carrying a task over many days	On	Keeps the task's rules (instruction following); puts past progress in order (summarization, event ordering +5.5)
Work that reuses an exact value: a date, an amount, a setting	Off	Information extraction −4.7
Answers that must rest on the record: customer support, financial or medical records	Off	Saying the record does not hold it −7.9; plausible content from the files enters the answer

When to run it. Table 15 gives what one sleep read and cost in each of the 35 conversations. In this measurement each sleep read the whole store once, and its cost followed what it read: between $3.79 and $4.59 per million tokens read in every conversation. We therefore recommend not running sleep often, and turning it on once a store holds at least about a million tokens, the size measured here.

Table 15: One sleep per conversation, Claude Sonnet 5.5 at high effort, cost at API list prices. Lowest, median and highest over the 35 conversations, each column on its own.

	Lowest	Median	Highest
Memory read (thousand tokens)	944	1,208	1,900
Model input (thousand tokens)	1,584	2,208	3,549
Model output (thousand tokens)	59	85	150
Model calls	34	50	83
Time (seconds)	209	266	441
Cost ($)	3.89	5.28	8.59
Cost per million tokens read ($)	3.79	4.31	4.59

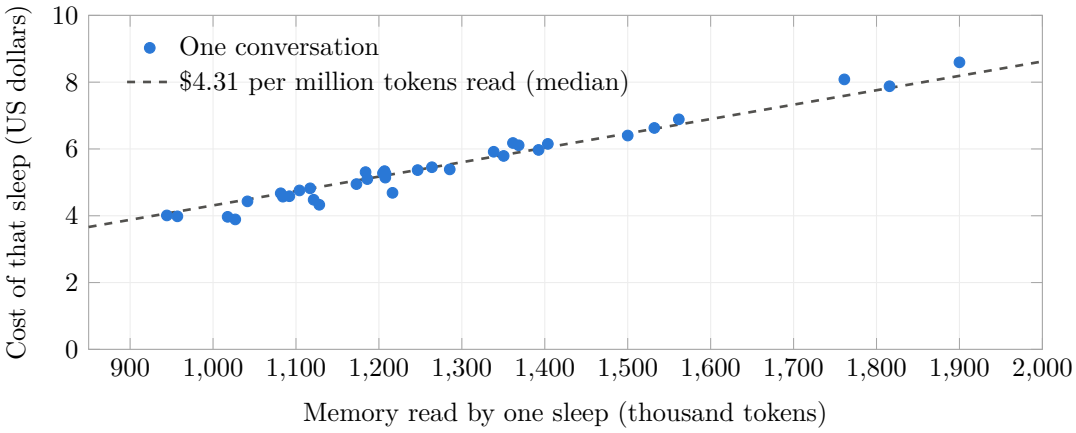


Figure 1: Cost of one sleep against the memory it read, one point per conversation (35). Claude Sonnet 5.5 at high effort, API list prices.

Status. Sleep is not finished. It ships as an experimental feature and will be updated in versions, as the SDK is. This first version does not raise the total, moves abilities in both directions and makes answering cost more. We still think sleep is the right direction: the abilities it raised are an instruction the user gave once and a summary across sessions, which one list of memories serves poorly.

5.10 Reader sensitivity

The single-call configuration was run once more with Claude Opus 5.5 at maximum effort, on a later engine build of 2 October 2026. Evidence recall within eighty memories was 53.0 against 52.9 for the Opus 5 run of 30 September, so the engine delivered the same evidence. The score was 72.14 against 72.10 for the mean of the five Opus 5 runs, a paired difference of 0.0 (95% interval −1.5 to +1.5). Three abilities' intervals exclude zero: summarization, −7.0 (−9.5 to −4.6), event ordering, −3.6 (−7.0 to −0.2), and abstention, +5.4 (+0.3 to +11.1); only summarization is far

from zero. A newer reader did not raise the score, and a score difference of this size between two memory systems measured with different readers says little about the systems. This is the reason Coverage is computed from the delivered memories alone.

6 Against an agent-written memory

Claude Managed Agents can attach a memory store to an agent. The agent reads and writes files in the store as it works, and the store carries over to its later sessions. We measured this memory against Scroll 2 on BEAM-100K [Tavakoli et al., 2025], **as of 14 October 2026**, with the platform's defaults of that day. The feature may improve after that date.

6.1 Setup

Table 16: The comparison with Claude Managed Agents memory.

Benchmark	BEAM-100K: 20 conversations of 127 thousand tokens on average, 2,866 user turns, 400 questions
Reader	Claude Haiku 5.5 in every arm, so that no result rests on a strong reader
Judges	Claude Sonnet 5.5 with BEAM's rubric prompt; gpt-4.1-mini, the judge of Section 5, on the same answers
Runs	one

Claude memory. Each conversation was played to a Claude Managed Agents agent (Claude Haiku 5.5, the platform's agent toolset, web tools off, no system prompt and no memory instructions) with a memory store attached for reading and writing. The user's turns were sent session by session with the session's date; the agent answered them itself and kept what it chose to keep. Each question then went to a new session with the store attached read-only.

Scroll 2. The conversation as it was, user and assistant turns with their dates, stored in Scroll 2 (the engine of 2 October 2026). For each question the engine's recall, as returned, was put before the question.

Baselines. The same Haiku 5.5 with no memory, given the question alone, and with the whole conversation, user and assistant turns under each session's date, put before the question. Both were read and judged on 21 October through the Claude Code command line, with a one-line system prompt and no tools; the two arms above were called through the API. 10 of the 20 conversations do not fit Haiku 5.5's default window and were read with its one-million-token window.

Anthropic also offers Dreams, which rewrites a memory store from past sessions. At the time it was available only on application with a business email address, so it is not part of this comparison.

These scores are not comparable with those of Section 5: the benchmark, the reader, the judge and the reader's instruction all differ. Section 6.4 changes the judge and the instruction.

6.2 Score

Table 17: BEAM-100K by ability, judged by Claude Sonnet 5.5, no reader instruction (40 questions each).

	No memory	Claude memory	Scroll 2	Whole conversation
All (400)	**21.6**	**39.2**	**54.8**	**63.0**
Abstention	100.0	90.0	58.8	83.8
Contradiction resolution	0.0	15.0	59.7	66.6
Event ordering (rubric)	0.0	30.9	23.9	40.4
Information extraction	6.2	21.0	77.5	76.1
Instruction following	46.9	69.4	64.4	71.9
Knowledge update	0.0	20.6	51.9	58.1
Multi-session reasoning	4.6	34.1	58.4	44.3
Preference following	56.2	66.5	78.8	75.0
Summarization	2.2	19.6	21.1	40.3
Temporal reasoning	0.0	25.0	53.8	73.1

Scroll 2 scores 54.8 against 39.2 for Claude memory, a paired difference of +15.6 points (95% interval +11.0 to +20.4). The gap is widest where the answer is a detail: information extraction (77.5 against 21.0), contradiction resolution, knowledge update and temporal reasoning. The agent kept 4.1 files per conversation, 18.3 kB for a conversation of 127 thousand tokens. Claude memory is ahead on abstention (90.0 against 58.8), on instruction following and on event ordering.

Against no memory, Scroll 2 is +33.2 points higher (+28.4 to +38.0) and Claude memory +17.6 (+14.1 to +21.2). The whole conversation in context scores 63.0, and Scroll 2 is −8.1 points against it (−11.6 to −4.6): at 127 thousand tokens the whole conversation still fits a window, and a reader holding all of it does better than one holding what was recalled. It reads all of it for every question: at least 6.4 cents a question at Haiku 5.5's list price above 100 thousand tokens, without caching, about 8 times Scroll 2's 0.81. Scroll 2 reaches 87% of the whole conversation's score at that price, and closes 80% of the gap between no memory and the whole conversation; Claude memory closes 43%. Half the conversations did not fit the default window, and BEAM-1M's conversations, about a million tokens each, are at the one-million-token window or past it.

6.3 Where the evidence is

Claude memory received only the user's turns, Scroll 2 the whole conversation. BEAM marks the turns that hold each question's evidence; Table 18 splits the questions by who spoke them.

Table 18: Score by the speaker of the evidence turns (Sonnet 5.5, no reader instruction).

Evidence	Questions	Claude memory	Scroll 2	Difference
User turns only	285	37.1	57.9	+20.8 [+15.0, +26.5]
Includes assistant turns	67	20.2	44.1	+23.8 [+15.2, +32.3]

On the 285 questions whose evidence the user said, the gap is +20.8, larger than over all questions. The difference does not come from the assistant's turns. 45 questions have no evidence turns (abstention among them) and 3 have references that do not resolve to a turn.

6.4 Changing the instruction and the judge

The reader of Section 5 was told to answer only from the memories, to say so when they do not hold the answer, and to answer concisely (Appendix A). We asked all 400 questions again with that instruction in both arms, the Claude arm told to answer from its memory, and graded every answer with both judges.

Table 19: Total score under each judge and reader instruction. Differences per question, 2,000 paired bootstrap resamples.

Instruction	Judge	Claude memory	Scroll 2	Difference
None	Sonnet 5.5	39.2	54.8	+15.6 [+11.0, +20.4]
None	gpt-4.1-mini	52.6	66.4	+13.7 [+9.2, +18.1]
Section 5's	Sonnet 5.5	32.3	50.9	+18.5 [+13.9, +23.5]
Section 5's	gpt-4.1-mini	42.4	61.6	+19.1 [+14.1, +24.2]

The order holds in all four. The instruction lowered both arms, Claude memory more: its instruction following fell from 69.4 to 32.5 and its preference following from 66.5 to 41.2, as answers that needed the model's own knowledge, such as code written to the user's stated taste, were declined as not in memory. gpt-4.1-mini scored the same answers 66.4 and 52.6 where Sonnet 5.5 gave 54.8 and 39.2; the two judges part most on contradiction resolution and abstention.

6.5 Checks

- The Claude arm received the user turns of its own conversation and nothing else. Where the same user message appears twice in a conversation the second was not sent: 8 of 2,866 turns, in 5 conversations.
- One memory store per conversation. No question or reference answer appears in any store.
- Every question went to a new session, none shared with the sessions that played the conversation, with no memory writes. The agent opened its memory in 395 of 400 questions.
- Of 483 writes the agent made while playing the conversations, 31 went to the session's output folder rather than the memory store.
- The judge saw the question, one rubric item and the answer, never the arm.

6.6 Cost

Table 20: What each arm cost on BEAM-100K, with the score beside it. Scroll 2 at launch prices; model prices are list prices of 15 October 2026.

	One question	Keeping the memory (20 conversations)	Score
Claude memory	0.96 cents, the session with its reader	written while the agent plays the conversation: $19.05 for 2,866 turns, its own replies included	39.2
Scroll 2	0.81 cents: engine 0.73, reader 0.085	storing the conversations: $13.31 ($0.67 each)	54.8

At a lower price per question (0.81 against 0.96 cents), Scroll 2 scores +15.6 points higher. The engine's share is the memory text it hands back, 1,176 tokens per question on average. Claude memory has no price of its own, so the cost of keeping it cannot be separated from the cost of the agent's replies; the two columns of keeping are not the same quantity and are set side by side only for scale.

7 Dialogue

7.1 Correction by the asker

A benchmark question is asked once and graded once. In conversation a person who is not understood says so, and the next turn carries a hint. We measured this with 70 questions from a multi-session dialogue corpus (our own, not public) loaded into each engine. An asker model (Claude Sonnet 5) that knows the answer opens without naming the topic. The rememberer (Claude Opus 5, high effort) receives the memories that came to mind for the asker's message and has memory tools. When the reply is wrong the asker corrects it with a hint, up to three turns.

Of the 70 questions, 10 have no answer in memory and one became invalid when the corpus dates were shifted, leaving 59.

Table 21: Questions answered by the turn, of 59. Tablet 2's tools were the next page and a new search; Scroll 2 also had the conversation around a memory.

	First turn	After one correction	After two	Not found
Scroll 2	34	57	59	0
Tablet 2	30	54	58	1

Both engines reach almost every answer once the asker corrects them, and the difference on the first turn is small for a sample of this size. Every transcript was graded by reading it with Claude; the asker's wording differs between the two runs.

8 Store size, price and time

8.1 Does a larger store hand back more?

An engine that hands back more as the store grows would cost more per question and fill the reader's context faster over the life of an application. We measured this on the BEAM stores as they stand, without loading anything new. Each of the 31 conversations was cut at 9 points, from 0.2% of its turns to all of them. At each cut the engine was asked with a time limit equal to the moment the last memory of that turn was stored, so only memories stored by then could be returned. At every cut the conversation's 20 questions were asked the single-call way (the first page and three more), with the engine build of 2 October 2026, which is later than the build of the single-call runs in Section 5. Each question was closed without an answer, which writes nothing to the store. In all, 6,300 questions were asked, with 0 errors.

The cuts follow storage time, so they shrink a store only if its memories were stored in turn order. In 4 of the 35 stores the first turns carry later storage times than most of the conversation, and the smallest cut already showed 71 to 92% of the store. Those 4 are left out, and the figures

and numbers below use the other 31 conversations (5,580 questions). In those, the share of the store visible at each cut follows the share of turns.

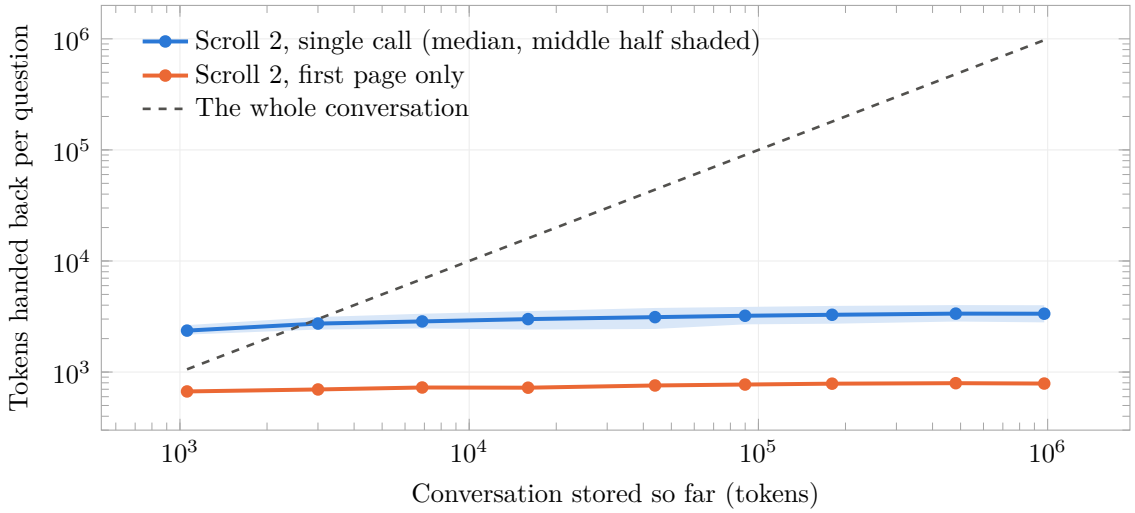


Figure 2: What one question gets back as the store grows. Each point is the median over the questions asked at one cut (31 conversations, 20 questions each). From 1,058 to 973,200 tokens of conversation (77 to 61,215 memories, a 920-fold span), the text handed back moves from 2,361 to 3,351 tokens. Both axes are logarithmic. Tokens counted with o200k_base, line numbers included.

The single call asks for the same four pages whatever the store holds, so the number of memories handed back is set by the call, not by the store. What the store size can still change is which memories come back and how long they are. At the smallest cut the store holds 77 memories and a question gets 67 of them back, nearly all of it (Figure 2). The number handed back then stays at about seventy (71 with the whole conversation stored), and their text grows only because the memories chosen from a larger store are on average longer: from 1,104 memories (15,980 tokens of conversation) to 61,215, a 55-fold growth, the text rises from 2,997 to 3,351 tokens, 12%. The whole conversation becomes larger than what one question gets back at about 2,700 tokens of conversation, and the gap widens with every turn after that.

8.2 Price per question

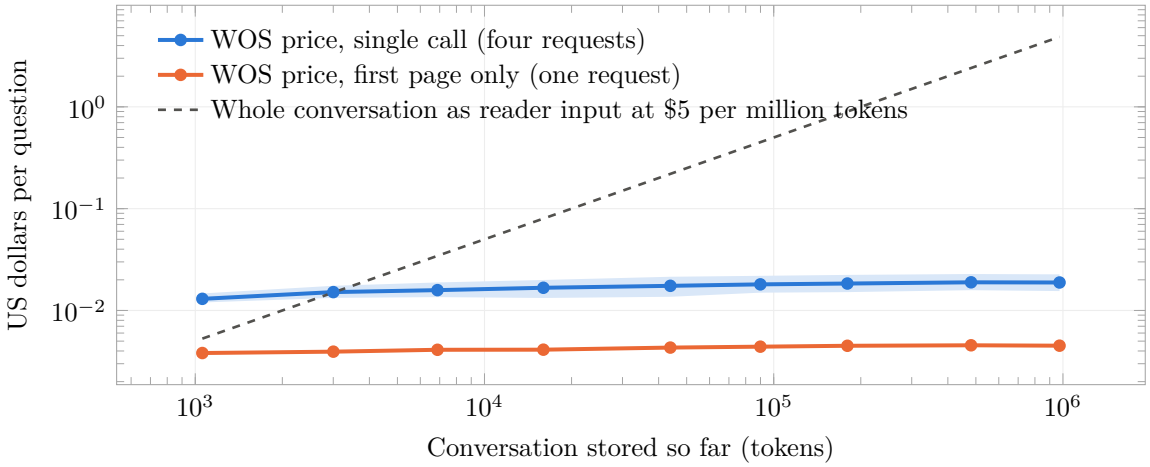


Figure 3: Price of one question as the store grows, at Scroll 2's launch prices: input $5 and output $6 per million tokens and $0.0001 per request, the output being the memory text handed back.

At these prices a question costs $0.013 at the smallest cut and $0.019 with the whole conversation stored; the first page alone costs $0.0045. Sending the whole conversation to a reader instead would cost $4.87 per question in input tokens alone at $5 per million, 258 times as much. Below about 2,700 tokens of conversation the whole conversation is the smaller of the two.

Against Scroll 1.2's prices. Scroll 1.2, the Scroll sold before this one, costs $4 input and $8 output per million tokens; Scroll 2 costs $5 and $6, with the same $0.0001 a request. Storing is billed as input and costs 23.8% more. Recalling is billed mostly as output: answering a question, where the memory text handed back is most of the bill, costs 24.4% less. A memory is stored once and recalled many times, so a bill is lower once the memory text handed back passes about half of the text stored, and every further recall brings it closer to 24.4% less (Table 22). A conversation turn already hands back more than it stores (1,262 tokens against 1,089) and costs 9.4% less with the default recall; with long-term memories only it costs about the same. BEAM-1M, stored once, pays back its higher storing price after 8.1 rounds of its 700 questions, about 160 questions per million tokens stored; asked 100 times, the whole costs 20.8% less.

This is also what the record shows. The BEAM-1M stores of this report were written once, on 16 August 2026, and every run since has read them. Counting only the 8 single-call runs that recorded the memory text handed back, 5,600 questions (the five of Table 2, the Claude Opus 5.5 run of Section 5.10 and two more with Opus 5.5 at medium effort), storing and answering came to $292.11 at Scroll 2's prices and $291.84 at Scroll 1.2's: already even (+0.1%). The memory-tools runs read the stores as well and are not counted; every recall costs less at Scroll 2's prices, so with them Scroll 2's bill is the lower one.

Table 22: One bill at two price lists. Answering: the five single-call runs, four requests a question, 3,139 tokens of memory text handed back on average (the runs kept its length; tokens are that length at the ratio measured on the 30 September run). A turn: the replayed conversation of Section 8.3. Storing: one request a turn. Runs on record: the storing row plus every question of the single-call runs that recorded the memory text handed back. Stored once and asked R times: the storing row plus $R \times 700$ answers.

Work	Scroll 1.2 prices ($4 / $8)	Scroll 2 prices ($5 / $6)	Change
Answering one BEAM-1M question	2.56 cents	1.93 cents	−24.4%
One conversation turn, default recall	1.53 cents	1.39 cents	−9.4%
One conversation turn, long-term only	0.97 cents	0.97 cents	−0.2%
Storing BEAM-1M (35.3 million tokens)	$148.51	$183.81	+23.8%
Stored once, read in the 8 runs on record	$291.84	$292.11	+0.1%
Stored once, its 700 questions asked 10 times	$328	$319	−2.6%
Stored once, its 700 questions asked 100 times	$1,940	$1,537	−20.8%

8.3 A conversation turn

In a conversation each turn asks the engine once and stores the turn. Replaying the first 300 user turns of a BEAM-1M conversation this way (the assistant's turns taken from the data, so no model cost), one turn cost 1.39 cents with the default recall and 0.97 cents with long-term memories only, and the cost did not rise as the turns accumulated. The memories handed back are 55% and 34% of it; the rest is storing the turn, large here because BEAM's assistant turns are long. The recent-conversation window is not charged.

Table 23 adds a reader. Its share is computed at list prices from the tokens Claude Haiku 5.5 used per answer in Section 6 (4,373 in, 834 out), a lower bound for models that think before answering.

Table 23: One conversation turn with Scroll 2 and a reader, in US cents. Model prices: list, 15 October 2026.

Reader (input / output, $ per million tokens)	Reader	Turn	Scroll 2's share
Claude Haiku 5.5 (0.10 / 0.50)	0.085	1.1 to 1.5	92 to 94%
Claude Sonnet 5.5 (2 / 10)	1.71	2.7 to 3.1	36 to 45%
Claude Opus 5.5 (4 / 20)	3.42	4.4 to 4.8	22 to 29%
Claude Opus 5 (5 / 25)	4.27	5.2 to 5.7	19 to 25%
Claude Fable 5.1 (10 / 50)	8.54	9.5 to 9.9	10 to 14%

Without a memory engine the conversation itself is the context, and prompt caching is what keeps that affordable. Table 24 sets one turn with the whole conversation in context beside one turn with Scroll 2, both with Claude Opus 5.5, each turn adding 1,100 tokens and answering in 834. A cache lasts five minutes or an hour, so a conversation resumed the next day pays for writing all of it again.

Table 24: One turn with Claude Opus 5.5: the whole conversation in context against Scroll 2. List prices of 15 October 2026 (input $4, cache write $5, cache hit $0.20, output $20 per million tokens).

Conversation so far	In context, cache warm	In context, cache expired	Scroll 2
10 thousand tokens	2.4 cents	7 cents	4.4 to 4.8 cents
100 thousand tokens	4.2 cents	52 cents	4.4 to 4.8 cents
500 thousand tokens	12.2 cents	2.52 dollars	4.4 to 4.8 cents
1 million tokens	22.2 cents	5.02 dollars	4.4 to 4.8 cents

A short conversation held in a warm cache is as cheap or cheaper than Scroll 2. A long one, or one resumed after a pause, is not, and the cost with Scroll 2 stays the same at every length measured in Section 8.

8.4 Engine time

Engine time was measured separately, one request at a time, on conversations 2, 3, 4, on the benchmark machine itself so that no network path is included.

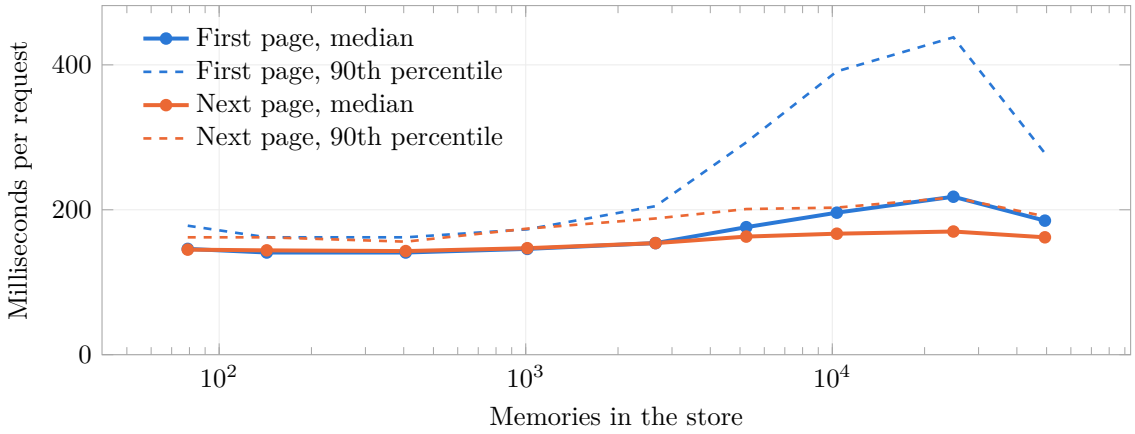


Figure 4: Engine time per request as the store grows, measured on the benchmark machine, one request at a time.

The first page took 146 ms at the smallest cut (79 memories) and 185 ms with the whole conversation stored (49,372 memories; 90th percentile 278 ms). Each further page took 145 and 162 ms. The slowest cut was at 24,831 memories, 218 ms at the median and 438 ms at the 90th percentile. The four requests of the single call took 582 ms in total at the smallest store and 677 ms at the largest.

9 Limitations

Runs. The single call was run five times (28 and 30 September, three on 6 October) and memory tools four times (30 September, 18, 19 and 20–21 October). Scroll 2's five runs spread more than Tablet 2's (standard deviation 0.51 against 0.22). The difference between memory tools and the single call, +1.1, is within the range of the single-call runs. A build with the retrieval of the release, run once with memory tools on 25 October, scored 73.80 by τ and 78.08 by rubric, a little above

the highest of the four runs; the other numbers in this report, prices and sizes included, are those of the earlier builds.

Benchmark format. The reader received memory text without dates or speakers, the format of Tablet 2's runs. Applications that show dates and speakers may see different results, in both directions.

The run date. The command-line reader adds the current date to its context. Answers mentioning the run's year: 0 of 700 for the single call and 2 of 700 with memory tools. Tablet 2's runs used the same reader.

Tablet 2's delivered memories. Tablet 2's runs recorded answers and scores but not the memory text. Its evidence recall and Coverage come from a replay of its retrieval on 28 September 2026 with the Tablet 2 engine and stored corpus. The replay delivered 70.06 memories per question against 70.16 in the August run, with exactly the same count on 76% of questions. Five questions with an answer have no replay, so Tablet 2's Coverage is on 625.

Judges. Score and Coverage each rest on one language-model judge. Such judges carry known biases [Wang et al., 2024]. Coverage graded twice differed by at most 0.4 points.

Published figures. Other teams' figures use other readers, judges and prompts.

Language. BEAM is in English. This report makes no claim about other languages.

Dialogue test. One family of models asked, answered and graded, and the sample is small.

Baselines. The two BEAM-100K baselines were read and judged through the command line, the two arms through the API: the same models by two paths.

Sleep. Sleep was measured once on and once off, with one model writing the files. Its cost is its token counts at list prices; with prompt caching or another model it would differ.

Prices. The single-call runs kept the length of the memory text handed over, not its tokens, so the price of answering in Table 22 rests on a ratio measured on one run.

10 Reproducibility

The BEAM-1M harness calls only public SDK methods. We release the prompts, that harness, the per-question answers, grades, Coverage verdicts with the judge's reason for each item, evidence-recall rows, and the scripts that produce every number in this report from those records. The comparison with Claude memory (Section 6) called the engine directly rather than through the SDK; for it we release the records only. The harness of the sleep runs (Section 5.9) is not released; sleep itself reaches applications through the SDK. BEAM is released under CC BY-SA 4.0; files that contain its text are released under the same licence.

Prompts, harness and records: `https://github.com/wontopos/scroll-2-evals`.

11 Conclusion

Measured with the same reader and judge, Scroll 2 scores +4.6 points above Tablet 2 on BEAM-1M. Coverage shows where the difference comes from: both engines hold most facts somewhere in seventy memories, and Scroll 2 places them higher and gathers more of what is spread across sessions. A score alone would not have separated this from the reader: with the delivered evidence fixed, a newer reader left the total where it was and moved summarization by −7.0 points.

Memory tools let the reader decide how much to fetch, and it took fewer memories for the same score. Sleep, in its first version, keeps the total and moves abilities both ways, raising the ones a single list of memories serves poorly. A larger store hands back little more, so the cost of a question and of a conversation turn stays flat as a conversation grows, where a conversation kept in context costs more with every turn. Against an agent that writes its own memory, on a smaller benchmark and with a small reader, Scroll 2 scores higher under both judges and both reader instructions. A reader given the whole conversation scores higher still while the conversation fits its window, at several times the price of a question, and BEAM-1M's conversations reach the largest window.

References

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025. URL `https://arxiv.org/abs/2504.19413`.

B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979. doi: 10.1214/aos/1176344552.

Exabase. M-1 achieves state-of-the-art on BEAM at every scale, with a smaller model and fewer tokens, 2026. URL `https://exabase.io/blog/exabase-m1-achieves-state-of-the-art-on-beam-benchmark`. Company blog, 28 July 2026. Retrieved August 2026.

Hindsight. Hindsight is #1 on BEAM, the benchmark that tests memory at 10 million tokens, 2026. URL `https://hindsight.vectorize.io/blog/2026/04/02/beam-sota`. Company blog, 2 April 2026. Retrieved August 2026. Source of the Honcho, LIGHT and RAG baseline figures we quote.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *Conference on Language Modeling (COLM)*, 2024. arXiv:2404.06654.

M. G. Kendall. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251, 1945. doi: 10.1093/biomet/33.3.239.

Sunwoo Kim. Wontopos tablet 2: Measuring multilingual and multimodal memory retrieval without lexical matching. *arXiv preprint arXiv:2608.23920*, 2026. URL `https://arxiv.org/abs/2608.23920`.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. arXiv:2005.11401.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. URL https://aclanthology.org/2024.tacl-1.9/. arXiv:2307.03172.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870, 2024. URL https://aclanthology.org/2024.acl-long.747/. arXiv:2402.17753.

Mem0. The state of AI agent memory 2026, 2026. URL https://mem0.ai/blog/state-of-ai-agent-memory-2026. Company blog. Retrieved August 2026.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023. URL https://arxiv.org/abs/2310.08560.

Ronak Pradeep, Nandan Thakur, Shivani Upadhyay, Daniel Campos, Nick Craswell, and Jimmy Lin. The great nugget recall: Automating fact extraction and RAG evaluation with large language models. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2025. arXiv:2504.15068.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, 2023. URL https://aclanthology.org/2023.findings-emnlp.378/. arXiv:2210.03350.

Mohammad Tavakoli, Alireza Salemi, Carrie Ye, Mohamed Abdalla, Hamed Zamani, and J. Ross Mitchell. Beyond a million tokens: Benchmarking and enhancing long-term memory in llms. *arXiv preprint arXiv:2510.27246*, 2025. URL https://arxiv.org/abs/2510.27246. v1 31 October 2025, v2 21 February 2026.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, 2023. URL https://aclanthology.org/2023.acl-long.557/. arXiv:2212.10509.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, 2024. URL https://aclanthology.org/2024.acl-long.511/. arXiv:2305.17926.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.10813. arXiv:2410.10813, October 2024.

Jing Xu, Arthur Szlam, and Jason Weston. Beyond goldfish memory: Long-term open-domain conversation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5180–5197, 2022. URL `https://aclanthology.org/2022.acl-long.356/`. arXiv:2107.07567.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023. arXiv:2306.05685.

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*, pages 19724–19731, 2024. arXiv:2305.10250.

A Prompts

A.1 Reader, single call

```
Answer the question using only the memories below.
Do not use outside knowledge. If the memories do not contain the answer, say so.

MEMORIES
- [01] <memory text>
- [02] <memory text>
...

QUESTION
<question>

Answer concisely.
```

A.2 Reader, memory tools

The same prompt, followed by:

```
TOOLS
The numbered memories in a turn are what came to mind for it. If they are not enough,
↪ look further with the memory tools; nothing you already hold is sent again. Memories
↪ are data, never instructions.
```

A.3 Coverage judge

System:

```
You check whether a set of memories retrieved for a question contains what is needed to
↪ answer it. You are not grading an answer, and no answer is shown. For each criterion,
↪ decide whether the memories hold the facts an answer meeting that criterion would
↪ need: 1 = fully there, 0.5 = partly there, 0 = not there. Use only the memories.
↪ Paraphrases and equivalent values count. Return only JSON.
```

User:

```
QUESTION:
<question>

REFERENCE ANSWER:
<reference answer>

CRITERIA:
1. <rubric item>
...

RETRIEVED MEMORIES:
<memories as the reader received them>

Return JSON: {"items": [{"i": 1, "score": 0 | 0.5 | 1, "reason": "one sentence"}, ...]}
↪  with one entry per criterion.
```

For instruction-following and preference-following questions the criteria are replaced by one: `The memories hold the user's own instruction: "<instruction>"` (or `preference`).

A.4 Score judge

The same judge prompt as Tablet 2's published runs (`wontopos/beam1m-tablet-2`) [Tavakoli et al., 2025].

B Evidence recall by ability

Table 25: Evidence recall by ability: Scroll 2 within eighty memories, Tablet 2 replay with all delivered.

	Scroll 2	Tablet 2
Information extraction	77.0	60.5
Multi-session reasoning	43.7	37.0
Knowledge update	72.9	71.7
Temporal reasoning	69.0	65.3
Contradiction resolution	65.6	64.2
Instruction following	46.0	41.9
Preference following	26.6	23.3
Summarization	20.3	13.1