DETERMINISTIC REPLAY · 1,314 SESSIONS · COMMIT 528023E

What Kemory saves, and when it doesn't

Token cost savings are not a single number. They depend almost entirely on how long the conversation runs — negative on one-off prompts, above 80% on long projects.

72.2%
READER INPUT AVOIDED

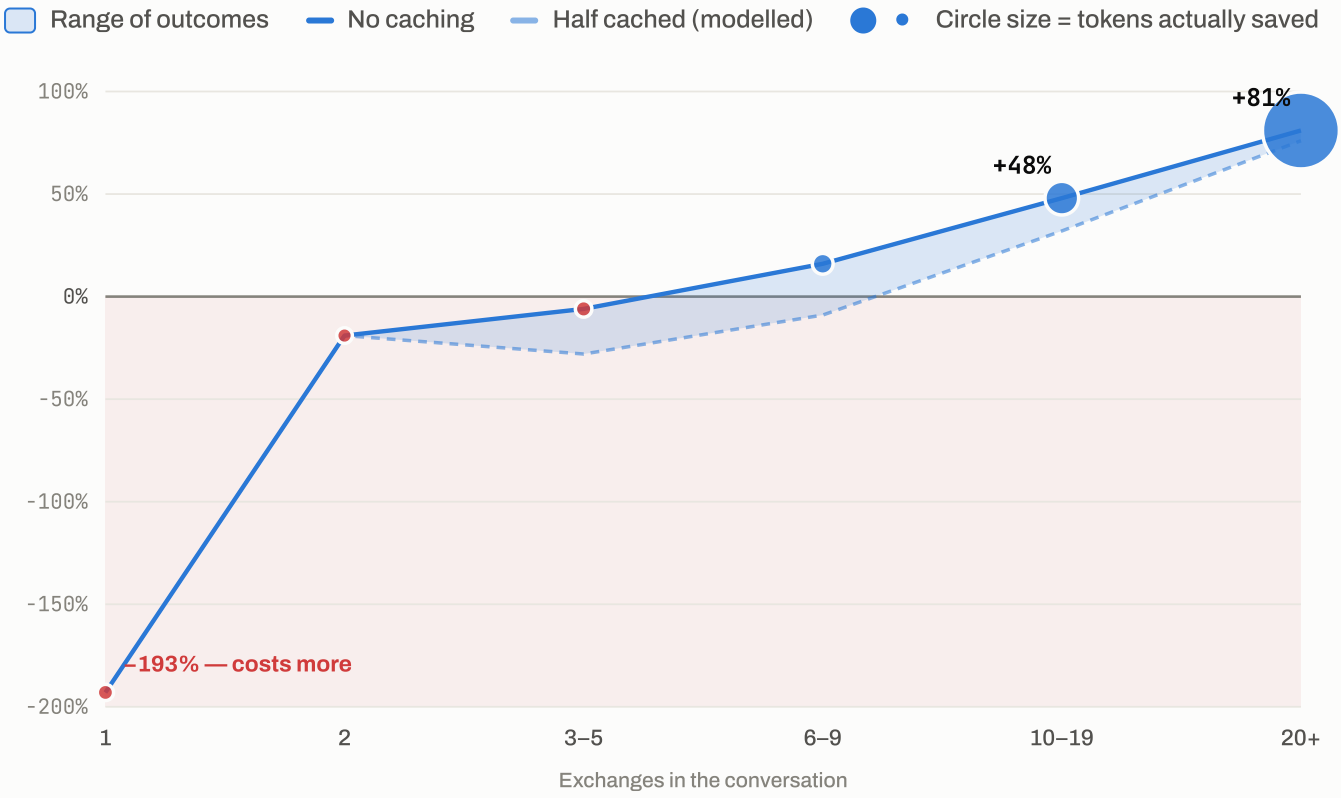
61.3%
NET OF KEMORY'S OVERHEAD

~5
EXCHANGES TO BREAK EVEN

8,369
DECISION STEPS MEASURED

Saving by conversation length

The shaded band is the range of outcomes. Its upper edge assumes the customer uses no provider prompt caching; the lower edge assumes half their requests hit a warm cache.



Negative values mean Kemory costs more than replaying the full history. Circle area is the tokens actually saved in each band, so it shows where the money is rather than only the rate.

Where the savings actually are

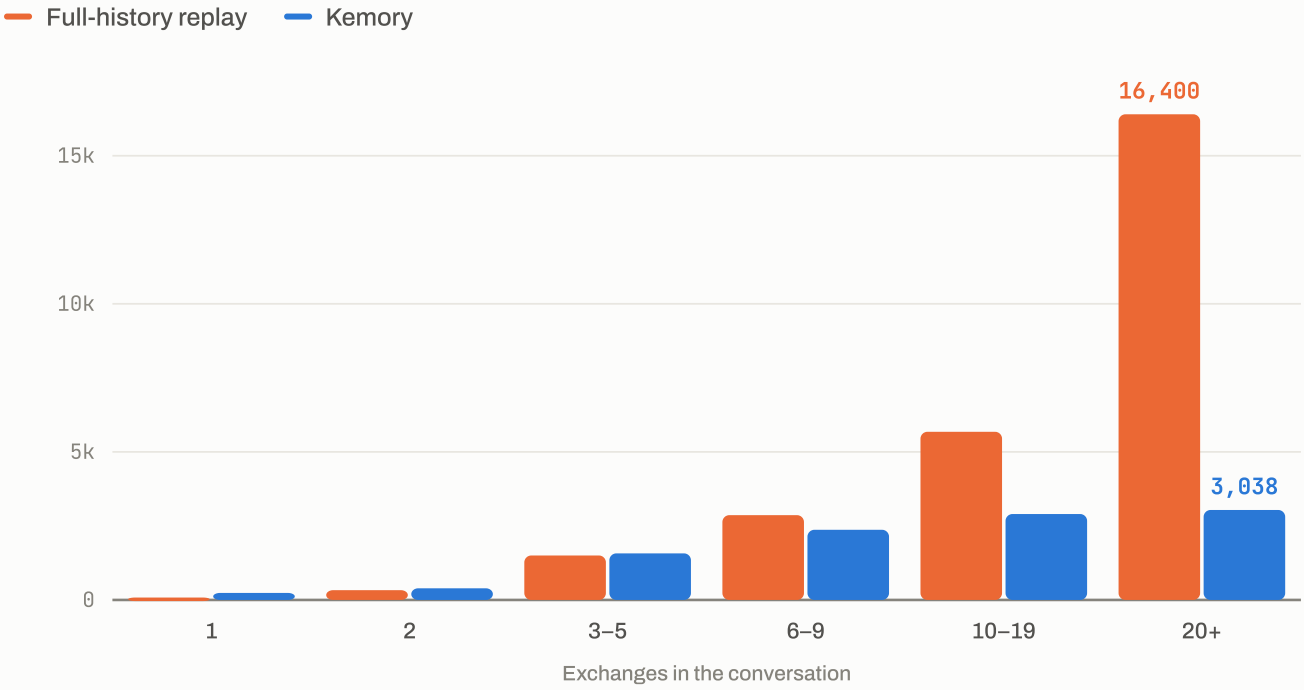
Rate and impact are different questions. The 97 sessions of 20 exchanges or more are 7% of the sample and produce **90.5% of all tokens saved.** Meanwhile the three loss-making bands — 928 sessions, 71% of them — cost **$0.85 in total**, against $264.66 saved.



So the losses on short conversations are real but economically trivial, and the case rests on whether a customer's users hold long conversations at all.

Why: what each turn has to read

Without memory the model re-reads the whole conversation on every turn, so per-turn cost climbs steeply. Kemory substitutes a bounded summary plus the last few exchanges, which also grows — but far more slowly.



Mean tokens read per decision step. Full-history grows about 200× across the range; Kemory about 13×.

The numbers

EXCHANGES	SESSIONS	NO CACHING	HALF CACHED	TOKENS SAVED	SHARE	FULL-HISTORY / TURN	KEMORY / TURN
1	396	-193%	-193%	-61,838	-0.1%	81	237
2	254	-19%	-19%	-32,166	-0.1%	330	393
3–5	278	-6%	-28%	-76,681	-0.1%	1,501	1,572
6–9	159	+16%	-9%	558,472	1.1%	2,863	2,369
10–19	130	+48%	+32%	4,665,499	8.8%	5,678	2,901
20+	97	+81%	+76%	47,878,945	90.5%	16,400	3,038

What everyone else publishes

> **These numbers cannot be compared with ours**
>
> Every vendor measures its own traffic on its own benchmark. Nothing below was run against the workload used on this page, so none of it is a ranking and none of it should be quoted as one — in either direction.
>
> The clearest evidence of why: **our own figure moves from −193% to +81% on conversation length alone.** The workload drives the result as much as the technology does, and nobody publishes their workload.

They are listed because they are the claims in the field, and because being able to say precisely why they don't transfer is stronger than pretending they don't exist.

TOKEN REDUCTION AGAINST FULL HISTORY — different workloads — not a ranking

SYSTEM	CLAIM	EVIDENCE
Kemory	72% (61% net)	Replay of 1,314 real sessions, 8,369 steps, reproducible from a pinned commit
mem0	40%	One customer's traffic. Measured per prompt, not per session. No token counts, sample size, or method stated
mem0 (LoCoMo)	~3–4×	~6,956 tokens per retrieval vs 25,000+ full-context, on a benchmark corpus

Zep and Letta publish no token-reduction figure at all.

ANSWER ACCURACY — different benchmarks — not a ranking

SYSTEM	SCORE	BENCHMARK AND READER
mem0	92.5	LoCoMo. Open evaluation framework
Kemory	85.0%	LongMemEval-500, full set, 425/500. Measured 25 Sep 2026
Zep	71.2%	LongMemEval, gpt-4o, against 60.2% full-context. Peer-reviewed
Letta	—	No published score on either benchmark

KEMORY ON LONGMEMEVAL-500, BY QUESTION TYPE



The two weakest types, multi-session and temporal-reasoning, are also the two where a separate retrieval sweep showed the largest regression under the v2 read profile. Worth treating as one open question rather than two.

These are **not** a ranking and the rows are ordered by score only to make that obvious: LoCoMo and LongMemEval are different corpora, run with different readers and different graders. mem0's 92.5 and Kemory's 87.8% were never measured against each other, and mem0 and Zep are currently disputing in public exactly how far such numbers travel. The one way to settle it is to run every system through the same questions with the same reader — about a week of work, since each needs its own ingest path.

What this does and doesn't show

- **The accuracy figure is a fresh run, not the previously circulated one.** An earlier internal note quoted 87.8% on LongMemEval-500. The 85.0% here is the 25 Sep 2026 run of the official harness over the full 500 questions. If both are cited anywhere, the older one should be retired rather than the two averaged.
- **The savings figure is an upper bound.** Recalled-memory payloads are absent from the replay — recall is topic-driven and there is no query topic when replaying — so Kemory's side of the comparison is a lower bound on what it really reads. Real-world saving is somewhat below the number on this page.
- **Input tokens only.** Output is identical under both approaches, so the absolute gap holds — but the percentage overstates the saving as a share of total inference cost.
- **The tokenizer is a proxy.** Counts use `o200k_base`, which undercounts Anthropic-billed tokens by roughly 15–20% on prose and more on code. Every dollar figure here is a floor, on both sides.
- **Digest sizes are reconstructed, not observed.** Persisted digest records were empty at measurement time, and the replay builds the digest through the deterministic extractive path rather than a live model — reproducible, but the wording a production digest would produce differs.
- **The headline rests on a long tail.** 97 sessions — 7% of the sample — carry 58.8M of the 73.3M baseline tokens. A customer whose conversations are mostly short will not see 72%.
- **The lower band edge is modelled, not measured.** Provider caching charges roughly 10× less to re-read the same text quickly, which favours full-history replay. The upper edge is measured; the lower edge is a sensitivity estimate.
- **This is a replay at published rates, not billing data.** Reader cost is priced at the `claude-opus-4-8` reference rate; a different reader rate moves the absolute dollars but not the percentage. Reproducible from commit `528023e`, which currently sits on an unmerged analysis branch rather than on `main`.
- **Competitors are not plotted on the chart, deliberately.** Nothing published by another vendor is measured on the same workload as ours. The token-reduction claims at least answer the same question against the same

baseline; the accuracy scores sit on different corpora with different readers and graders. Both are listed for completeness, neither is a ranking, and the only way to produce one is to run every system through the same questions with the same reader.