Knowing What is Missing: Efficient Conversational Memory via Explicit Evidence-Gap Tracking

Xingbo Du[1] Loka Li[1] Duzhen Zhang[1] Leonard Song[2] Le Song[1]

[1]Mohamed bin Zayed University of Artificial Intelligence [2]The Westminster Schools

Mohamed bin Zayed University of Artificial Intelligence

github.com/mbzuai-ai4bio/memr3

Problem: memory over long conversations

LLM agents must answer questions about weeks of past dialogue.

- **Full context**: about 115k tokens per query on LongMemEval$_s$; costly, and the answer can get lost in the history.
- **One-shot RAG**: a single query misses multi-hop and temporal evidence.
- **Implicit multi-step retrievers**: keep appending raw chunks, so the prompt grows and noise accumulates.

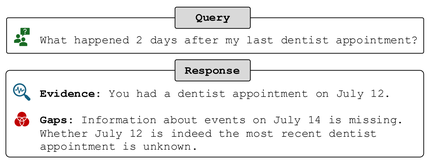
Q: *How many months passed between Andrew adopting Toby and Buddy?*

Full-Context	reads every session → "six months" ✗
RAG	retrieves once, misses Buddy's date → "about 4 months" ✗
MemR^3	Evidence: Toby adopted Jul 11, 2023; Buddy named Oct 19, 2023. Gap: Buddy's exact adoption date → retrieve, reflect → "3 months" ✓

Key idea: track the known and the missing

MemR^3 keeps an explicit **evidence-gap state** $\mathcal{S}_k = (\mathcal{E}_k, \mathcal{G}_k)$:

- **Evidence** $\mathcal{E}_k$: grounded facts found so far.
- **Gaps** $\mathcal{G}_k$: requirements still unresolved.

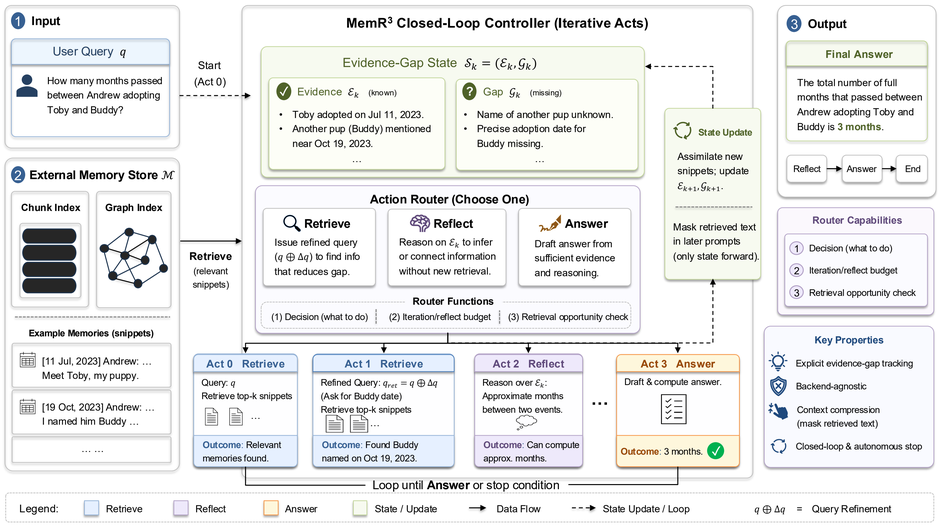


Absorb and discard: new snippets are absorbed into the state, then masked from later prompts. The context scales with the state, not with the retrieval history.

Takeaway

Explicitly tracking *what is missing* turns multi-step memory retrieval into a bounded, inspectable control loop. Strong long-context backends (GPT-5-mini, GPT-5.6 Terra) still favor Full-Context; MemR^3 targets regimes where full-history inference is costly or unreliable.

Method: a closed-loop memory controller



Retrieve

A refined query $q \oplus \Delta q$ targets the open gaps; already-seen snippets are excluded.

Reflect

Reason over $\mathcal{E}_k$ (e.g., date arithmetic) without new retrieval.

Answer

Stop when the gaps close or the budget is used, and answer from the state.

Plug-and-play: runs on top of chunk-based RAG or graph-based Zep memory, with no training.

Is the state faithful? Human audit and ablation

Audit (152 LoCoMo questions, GPT-4o-mini)

81.4%
of retrieved answer-critical evidence is preserved in the state (105/129)

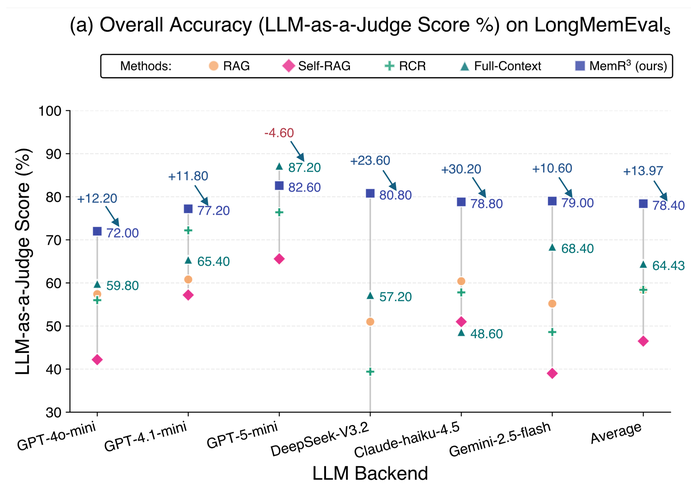
63.2% → 84.2%
human accuracy with a human-verified oracle state: state estimation is the bottleneck

Ablation on LoCoMo (J score, GPT-4o-mini)

Variant	Overall
RAG	75.54
MemR^3	**81.55**
w/o mask	68.54
w/o Δq_k	77.11
w/o reflect	76.65

Removing the masking costs 13 points: retaining raw history adds interference.

Results on LongMemEval$_s$ (6 backends)



- Beats all retrieval baselines on **6/6** backends and Full-Context on **5/6**.
- Uses 1%-5% of Full-Context's retrieved-context tokens.
- Largest gains where Full-Context struggles: +23.6 on DeepSeek-V3.2 and +30.2 on Claude-haiku-4.5.

Cost per query (GPT-4o-mini)

72.0	**5.9%**	**3.9 s**
J score vs 59.8 Full-Context	of Full-Context's total tokens	latency vs 11.0 s

LoCoMo: gains over the memory backbone

Overall J score (%)	GPT-4o-mini	GPT-4.1-mini
RAG	75.54	79.46
MemR^3 (RAG)	**81.55** (+6.01)	**86.75** (+7.29)
Zep	74.62	78.94
MemR^3 (Zep)	**76.26** (+1.64)	**80.88** (+1.94)
Full-Context	76.32	89.00