MemGPT: Towards LLMs as Operating Systems

Virtual context management for fixed-window LLMs

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang,
Shishir G. Patil, Ion Stoica, Joseph E. Gonzalez

arXiv:2310.08560v2 | 10 November 2026

Publication: no formal venue found as of 10 Nov 2026; Berkeley lists CoRR (source).
Official code (now Letta, formerly MemGPT)

The core move is to manage context, not enlarge it

Main idea

Treat the prompt as a scarce working set. Keep recent and salient information in the LLM window; keep full history and documents outside it; let the model page information in and out through function calls.

- The model's per-call context limit stays fixed.
- An event and function loop lets one task span multiple retrieval steps.
- Evidence: better deep-memory recall and multi-step retrieval on the paper's tasks.

Paper, Abstract and Sec. 5, PDF pp. 1, 8.

Longer windows do not remove the whole problem

- Conversations and document collections can outgrow one prompt.
- Transformer context scaling increases attention cost; the paper also cites evidence that models use long contexts unevenly.
- MemGPT asks whether external storage plus iterative retrieval can extend task-time access without changing the base model.

Historical context

The paper's context-window comparison was collected in January 2024. Its reported model limits are experimental context, not current model specifications.

Paper, Sec. 1 and Table 1, PDF pp. 1, 4.

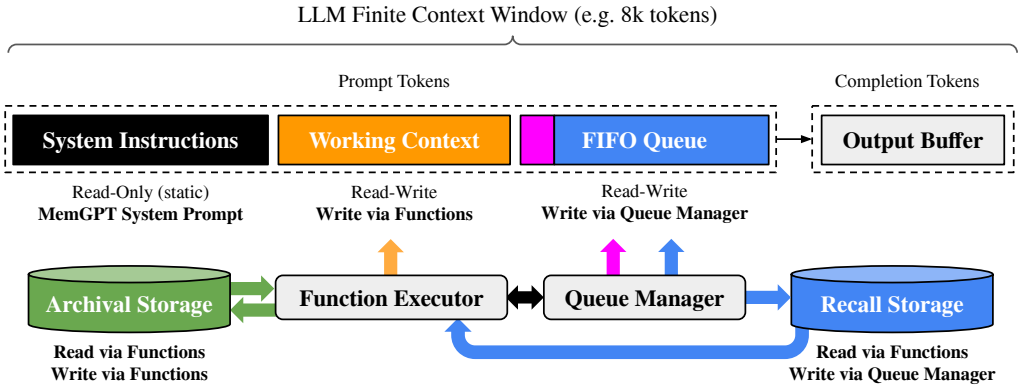
MemGPT maps operating-system ideas onto LLM context

Operating-system idea	MemGPT counterpart
Physical memory	The fixed prompt window: instructions, working context, FIFO queue
Disk / persistent store	Recall logs; archival documents
Paging	Function calls move selected information into or out of the prompt
Memory pressure	A system message warns the model before older queue entries are flushed
Interrupt / event	User input, capacity warning, upload event, or timer can trigger inference

- "Virtual context" is the model's ability to access stored material over multiple calls.

Paper, Sec. 2, PDF pp. 2–4.

A queue manager and function executor surround the LLM



- The queue manager persists messages and enforces capacity; the executor performs memory operations.

Paper, Fig. 3, PDF p. 3. Original author figure.

Three prompt regions and two external stores have different jobs

Main context: visible to the model	External context: fetched as needed
System instructions Static control policy and function descriptions.	**Recall storage** Persistent conversation messages; searchable and paginated.
Working context Fixed-size, editable facts and persona state.	**Archival storage** Arbitrary-length text or documents; accessed with memory functions.
FIFO queue Recent messages, system events, and tool results.	

Paper, Sec. 2.1–2.2, PDF pp. 2–3.

Capacity pressure triggers save, then flush

1. **Before overflow:** a warning message prompts the LLM to preserve important queue content in working context or archival storage.
2. **At the flush threshold:** the queue manager evicts older messages and creates a recursive summary to keep recent context available.
3. **After eviction:** full messages remain in recall storage and can be searched back into the prompt later.

Thresholds in the paper

About 70% for the warning, 100% for flush, and evicting about 50% of the window are illustrative examples, not universal constants.

Paper, Sec. 2.2, PDF p. 3.

Heartbeats let the model chain retrieval calls

- A function call is parsed and validated; results or runtime errors return to context.
- With `request_heartbeat=true`, MemGPT immediately runs another inference, so the model can search another page or follow a retrieved key.
- Without a heartbeat, the agent yields and waits for the next external event.

Why this matters

The LLM chooses whether to save, search, revise, or continue. The system provides the storage and control loop, but successful retrieval still depends on the model making useful tool calls.

Paper, Sec. 2.3–2.4, PDF pp. 3–4.

The experiments test memory across two kinds of long task

Domain	Task	Main comparison
Multi-session chat	Recall facts from earlier sessions; write personalized openers	MemGPT vs fixed-context summaries; GPT-3.5 / GPT-4 / GPT-4 Turbo
Document analysis	NaturalQuestions document QA	Paginated archival search vs one-shot top-K context
Document analysis	Nested UUID key-value lookup	Repeated search calls vs fixed prompt

- Tasks evaluate historical recall, document grounding, and multi-hop retrieval.

Paper, Sec. 3, PDF pp. 4–8.

Searchable history improves deep-memory answers

Base model	Base accuracy	+ MemGPT	Base ROUGE-L R	+ MemGPT
GPT-3.5 Turbo	38.7%	66.9%	0.394	0.629
GPT-4	32.1%	92.5%	0.296	0.814
GPT-4 Turbo	35.3%	93.4%	0.359	0.827

- The base sees a lossy summary of five prior sessions; MemGPT can search the full logs.
- GPT-4 and GPT-4 Turbo gain 60.4 and 58.1 percentage points in accuracy.
- The task uses a generated sixth-session question; answers are assessed with ROUGE-L recall and an LLM judge.

Paper, Sec. 3.1, Table 2 and Appendix 6.1, PDF pp. 5, 11.

Openers include more persona details by similarity scores

Method	SIM-1	SIM-3	SIM-H
Human opener	0.800	0.800	1.000
MemGPT + GPT-3.5 Turbo	0.830	0.812	0.817
MemGPT + GPT-4	0.868	0.843	0.773
MemGPT + GPT-4 Turbo	0.857	0.828	0.767

- SIM-1 / SIM-3 compare with gold persona labels; SIM-H compares with the human opener.
- The scores support greater persona coverage, while SIM-H stays below the human reference.

Paper, Sec. 3.1 and Table 3, PDF p. 5.

Document QA gives both methods the same retriever

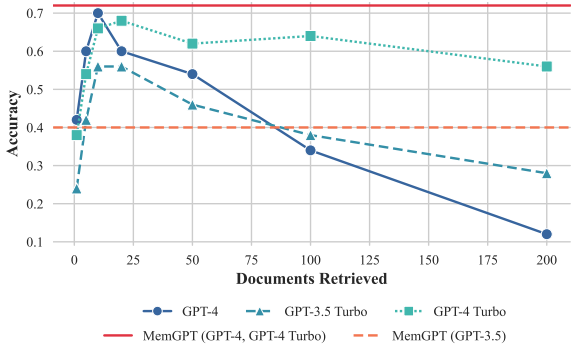
- 50 NaturalQuestions-Open questions with a late-2018 Wikipedia dump.
- Both use cosine search over text-embedding-ada-002 vectors.
- Fixed-context baselines receive top-K documents in one prompt. MemGPT stores documents in PostgreSQL / pgvector (HNSW) and can request further pages.
- An LLM judge checks both answer correctness and support from retrieved text.

What is being tested?

Whether iterative search can recover useful evidence beyond the first batch that fits in a single context window.

Paper, Sec. 3.2.1, PDF pp. 6–7.

Iterative archive search is stable as more documents are considered



- MemGPT stays near 0.72 with GPT-4 / Turbo and 0.40 with GPT-3.5.
- One-shot baselines lose accuracy when larger K forces document truncation.

Paper, Fig. 5, PDF p. 6. Values read from the plotted curves.

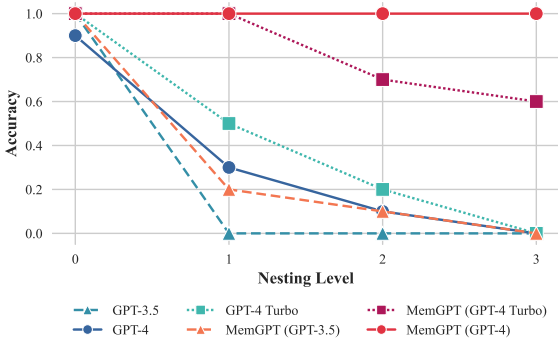
Nested KV isolates the need to follow several retrieval hops

- 140 key-value pairs (about 8K tokens); each key and value is a UUID.
- A value may itself be a key, so the agent repeats lookup until it reaches a terminal value.
- The task varies nesting depth from 0 to 4 and samples 30 orderings.

$$k_0 \to v_0 (= k_1) \to v_1 (= k_2) \to \cdots \to v_{\text{final}}$$

Paper, Sec. 3.2.2 and Fig. 8, PDF pp. 7–8.

GPT-4 MemGPT sustains nested lookup where baselines collapse



- GPT-3.5 baseline reaches zero at depth 1; GPT-4 / Turbo baselines by depth 3.
- MemGPT + GPT-4 stays at 1.0 through depth 3; other MemGPT bases fall.

Paper, Sec. 3.2.2 and Fig. 7, PDF p. 7.

The evidence supports a control loop, not automatic memory quality

What the tasks support	Where results depend on the setup
• Full-history search beats a lossy summary on DMR. • Multi-call retrieval helps with document QA and nested lookups. • Working context can retain evolving persona facts.	• GPT-3.5 tool calling is weaker; MemGPT can stop paging early. • DMR questions are generated; document QA uses 50 questions. • Similarity metrics and LLM judges are proxies for quality.

Paper, Sec. 3 and Appendix 6.1, PDF pp. 5–8, 11–13.

Takeaways

1. MemGPT treats a fixed prompt window as a working set managed by an LLM operating-system loop.
2. Persistent recall, archival search, queue eviction, and heartbeats enable multi-step use of external context.
3. The reported gains are strongest when the base model can call functions reliably and keep searching.

Paper: Packer et al., arXiv:2310.08560v2.
Project: MemGPT research page; official repository, now Letta.
Publication lookup: no formal venue found as of 10 Nov 2026; Berkeley lists CoRR.