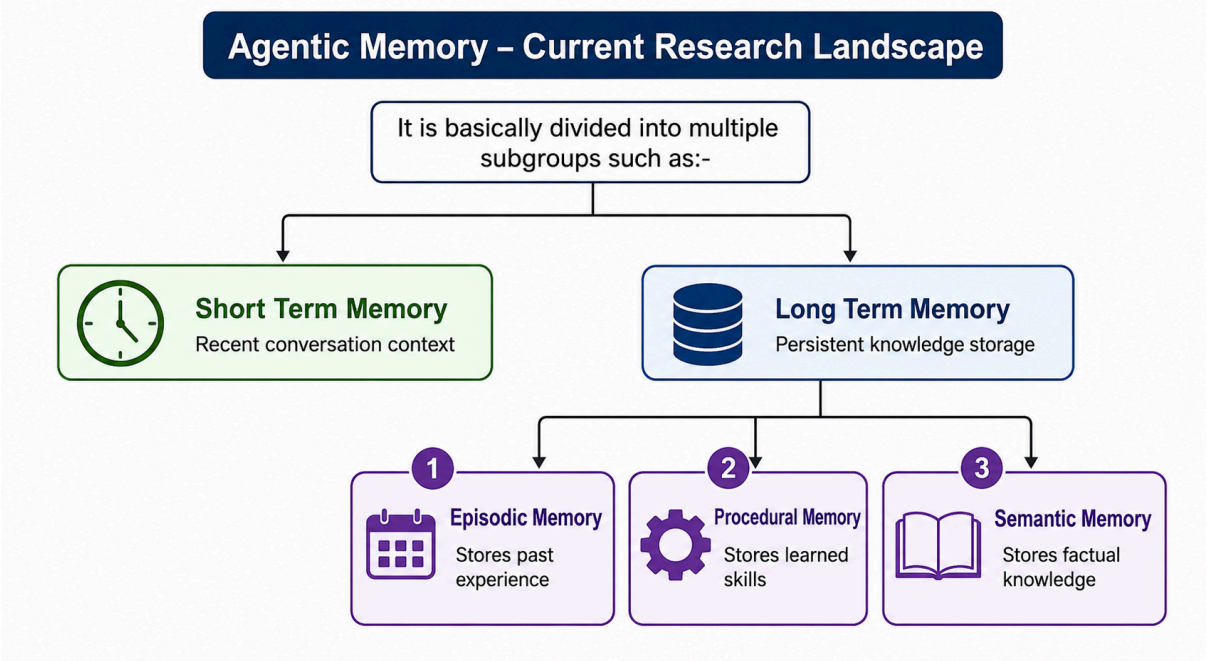
Agentic Memory-Current Research Landscape

At inference time, an LLM is essentially a parameterized math function that is completely **stateless.**
This means its output depends solely on the current input and its pre-trained parameters. It has no intrinsic memory to remember what you discussed in a previous prompt .

$$y = f_\theta(x)$$

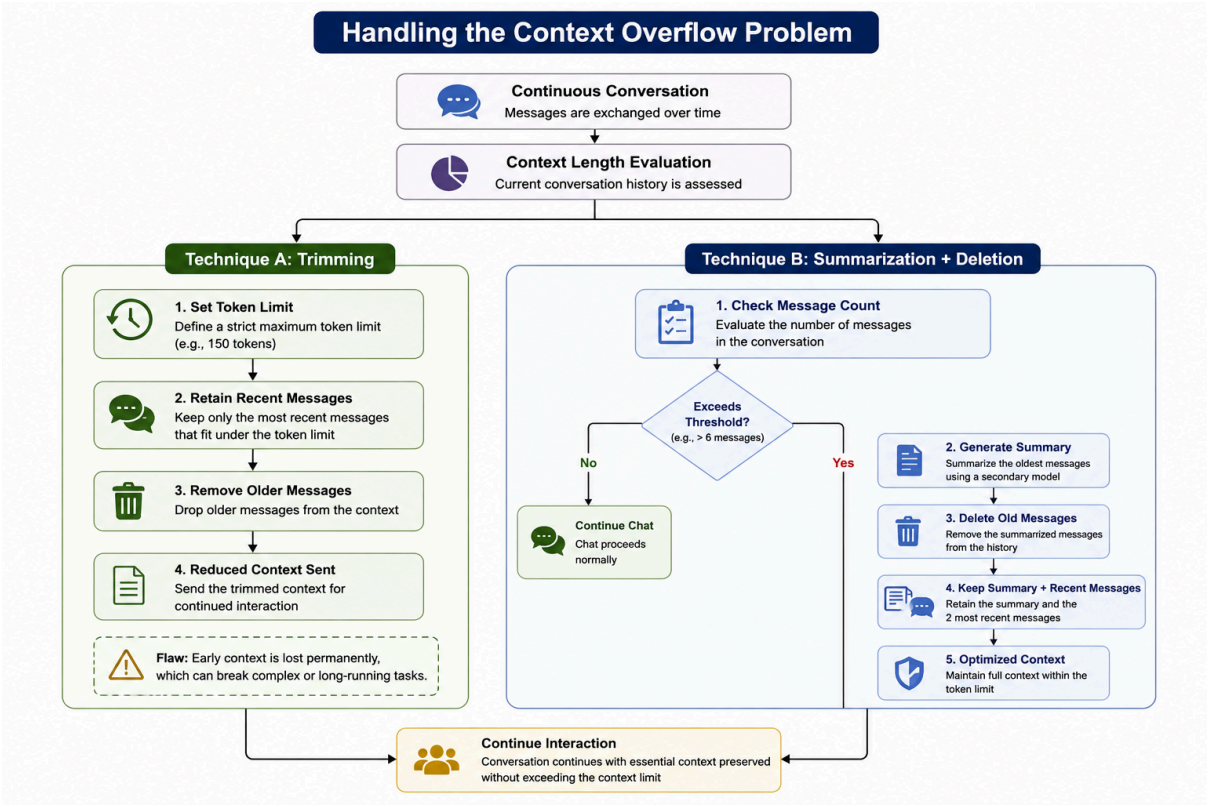
This is the reason why we need to invent new techniques to give llm's a memory.



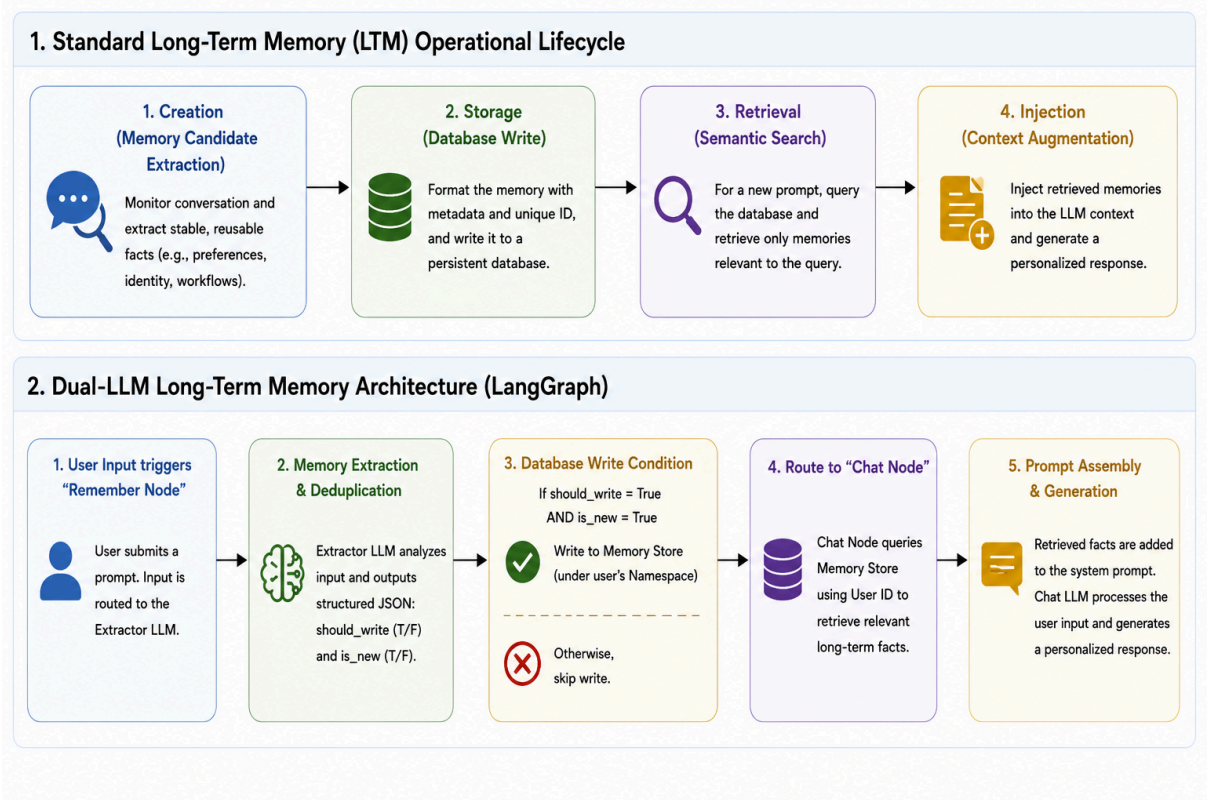
Short-term memory = In-context learning, thread-scoped

Traditional Methodology

1. Short-term Memory



2. Long-term Memory



Shortcomings:

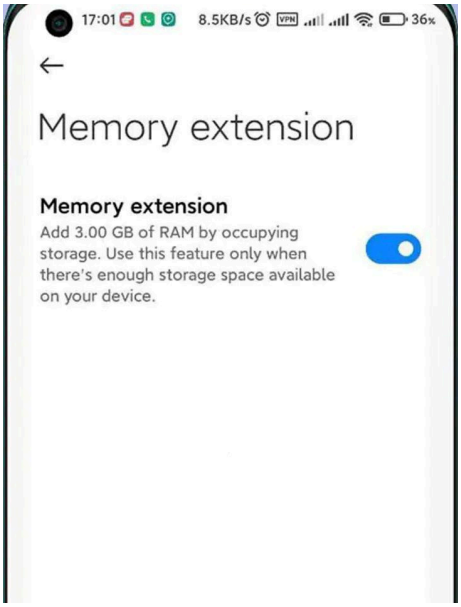
1. Loss of Information Over Time
2. Flat Memory Architecture (lack of hierarchical memory architecture)

The current pioneers in this research, in my view, are MemGPTresearchers, as they show significantly higher performance on Multi-Session Chats and Document Analysis.

MemGPT's Advantages

MemGPT is a system devised to enhance the performance of LLMs by introducing a more advanced memory management scheme. Below are some of the key features of MemGPT:

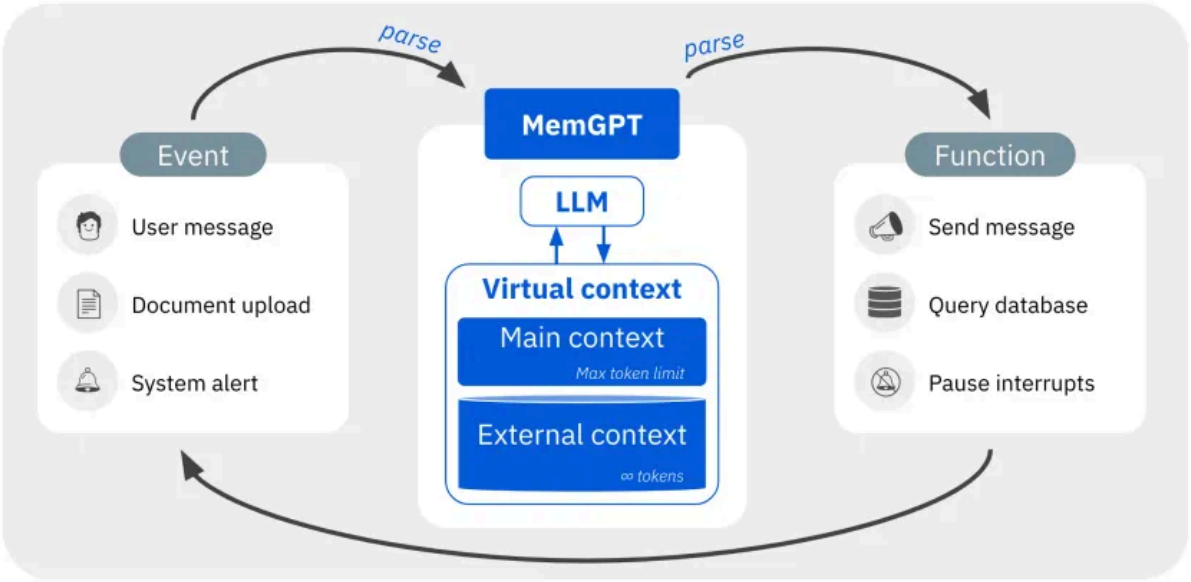
1. **Memory Management**:MemGPT uses a tiered memory system within a fixed-context LLM, enabling the processor to manage its own memory. This intelligent handling of different tiers extends the context beyond standard limits, solving the problem of restricted context windows in large language models.
2. **Virtual Context Management:**



Just like VRAM in modern OSes like Android etc, MemGPT follows a similar architecture.

3. **Operating System-Inspired**: The architecture of MemGPT draws inspiration from traditional operating systems, especially their hierarchical memory systems that facilitate data movement between fast and slow memory.

MemGPT's Architecture



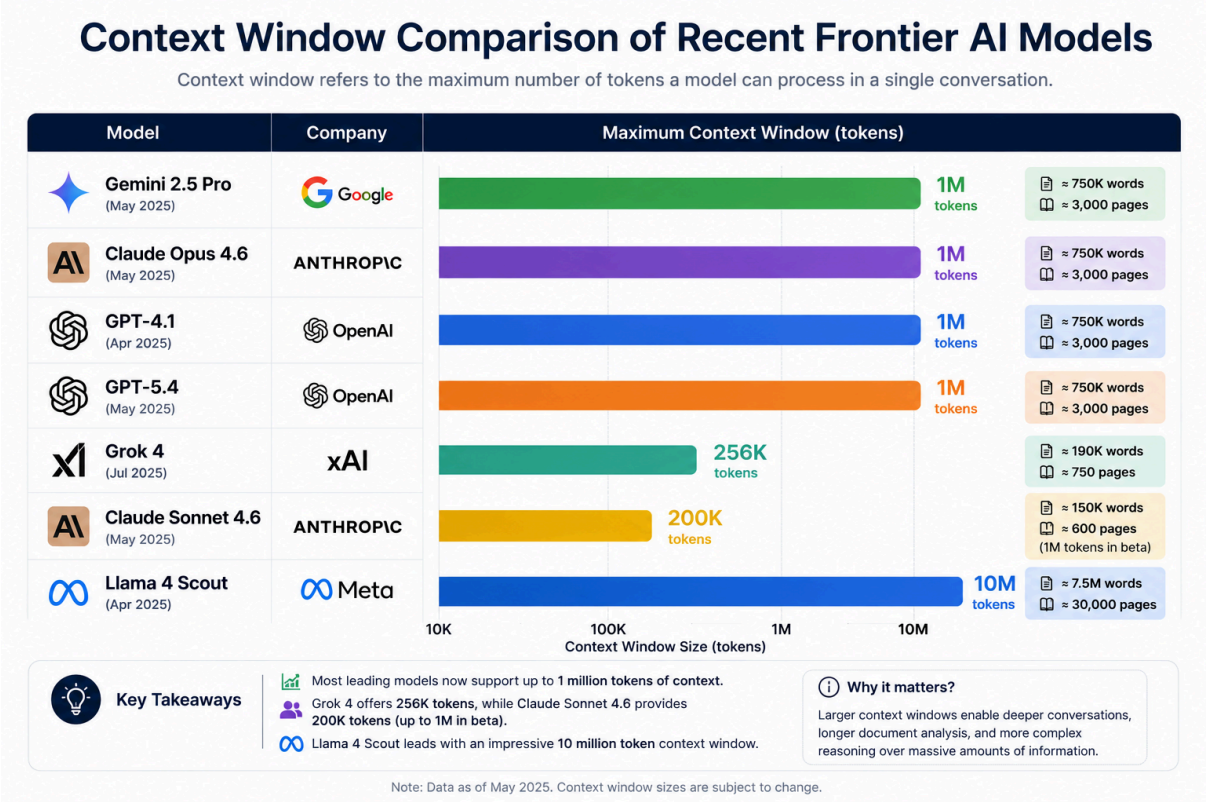
Similar to how a computer OS manages RAM and disk storage to overcome physical limits, MemGPT utilizes hierarchical memory tiers within an LLM, including:

1. **Main Context**: Immediate inference workspace, like RAM.
2. **External Context**: On-demand data storage, like a hard drive.

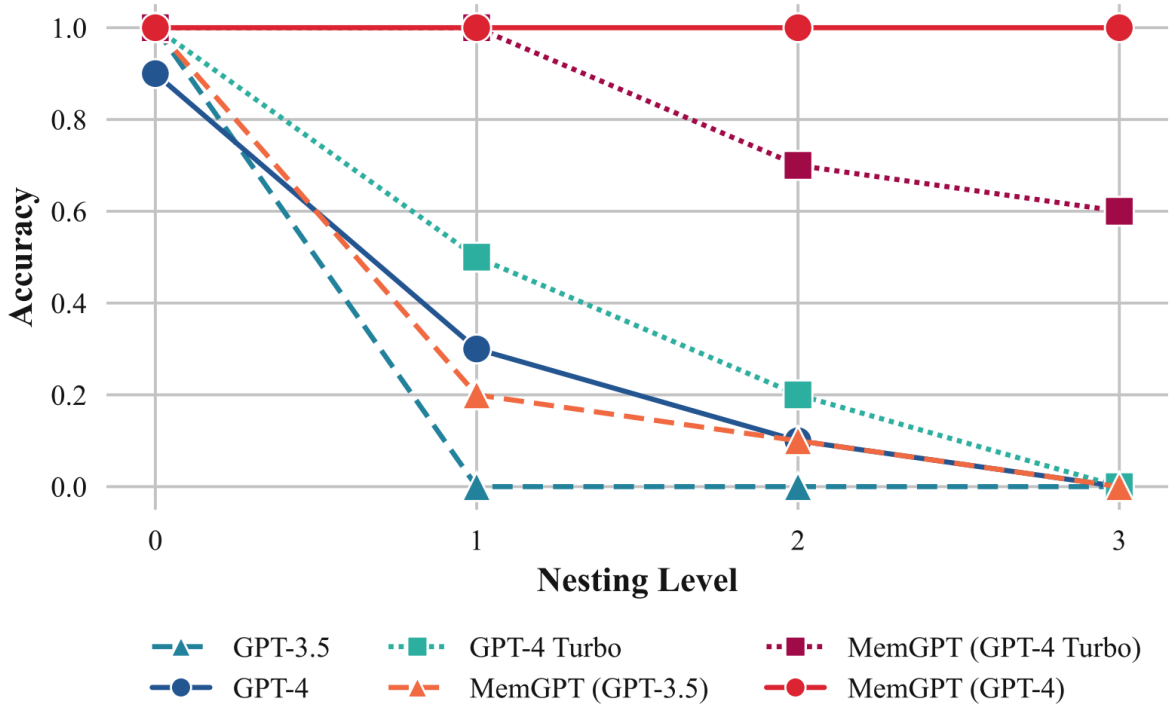
3. **Interrupts**: OS-style pausing to manage user control flow. ↳ Page faulting

This architecture allows for dynamic context management, enabling the LLM to retrieve relevant historical data akin to how an OS handles page faults.

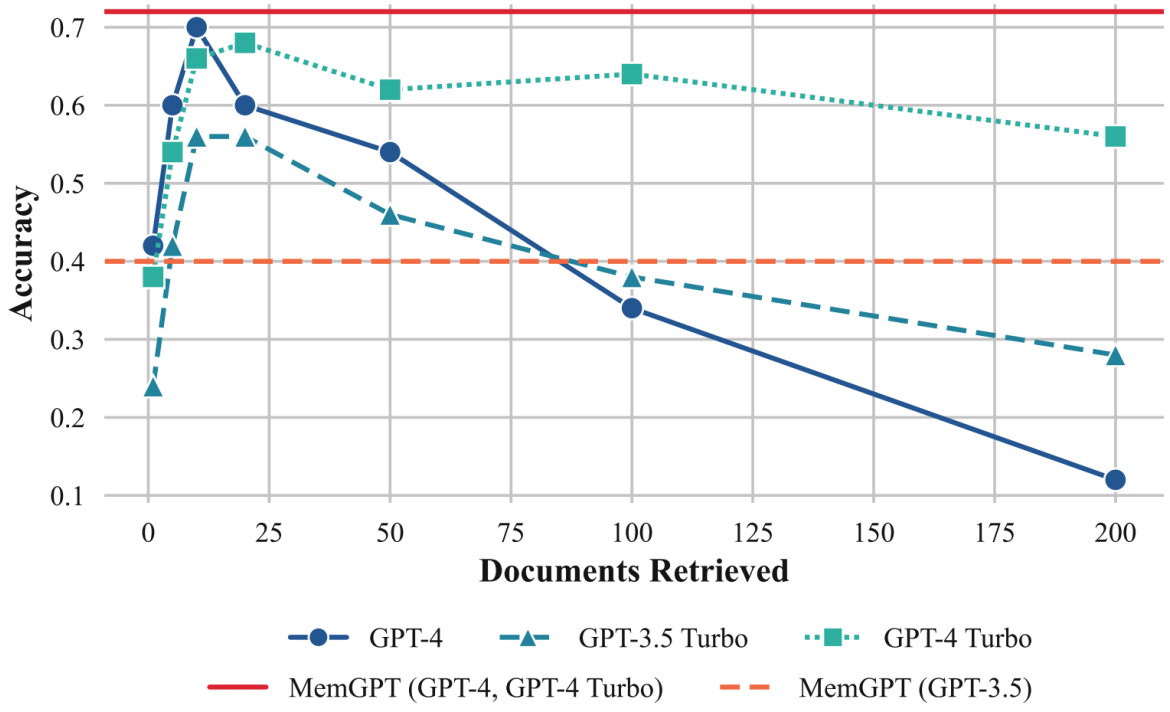
Context length of different frontier LLMs:



Multi-Session Chat Results



Document Analysis



Evaluation

The biggest unsolved problems are not “how to store text” but “what to keep, when to update it, how to forget it, and how to prove it helped.” This is benchmarked by the research paper namely MemoryAgentBench.

The paper proposes 4 core competencies that are still not solved together completely in even the most heavily funded LLMs as well which are:-

1. Accurate Retrieval
2. Test-time learning
3. Long-range understanding
4. Selective forgetting

Currently, even the most top-tier models score around 2-4%, while it peaks at just around 7% for modern AI models like GPT-4o, Sonnet 3.7 etc.

	AR					TTL			LRU			SF			Overall
Agent Type	SH-QA	MH-QA	LME(S*)	EventQA	**Avg.**	MCC	Recom.	**Avg.**	Summ.	DetQA	**Avg.**	FC-SH	FC-MH	**Avg.**	Scores
Long-Context Agents															
GPT-4o	72.0	51.0	32.0	77.2	58.1	87.6	12.3	50.0	32.2	77.5	54.9	**60.0**	**5.0**	**32.5**	48.8
GPT-4o-mini	64.0	43.0	30.7	59.0	49.2	82.0	15.1	48.6	28.9	63.4	46.2	45.0	5.0	25.0	42.2
GPT-4.1-mini	83.0	**66.0**	**55.7**	**82.6**	**71.8**	75.6	16.7	46.2	41.9	56.3	49.1	36.0	5.0	20.5	46.9
Gemini-2.0-Flash	**87.0**	59.0	47.0	67.2	65.1	84.0	8.7	46.4	23.9	59.2	41.6	30.0	3.0	16.5	42.4
Claude-3.7-Sonnet	77.0	53.0	34.0	74.6	59.7	**89.4**	**18.3**	**53.9**	**52.5**	71.8	62.2	43.0	2.0	22.5	49.6
GPT-4o-mini	64.0	43.0	30.7	59.0	49.2	82.0	15.1	48.6	28.9	63.4	46.2	45.0	5.0	25.0	42.3
Simple RAG Agents															
BM25	66.0	56.0	45.3	**74.6**	60.5	75.4	13.6	44.5	**19.0**	52.1	35.6	48.0	3.0	25.5	41.5
Embedding RAG Agents															
Contriever	22.0	31.0	15.7	66.8	33.9	70.6	15.2	42.9	17.2	42.3	29.8	18.0	**7.0**	12.5	29.8
Text-Embed-3-Small	60.0	44.0	48.3	63.0	53.8	70.0	15.3	42.7	17.7	54.9	36.3	28.0	3.0	15.5	37.1
Text-Embed-3-Large	54.0	44.0	50.3	70.0	54.6	72.4	**16.2**	44.3	18.2	56.3	37.3	28.0	4.0	16.0	38.0
Qwen3-Embedding-4B	57.0	47.0	43.3	71.4	54.7	**78.0**	12.2	**45.1**	14.8	59.2	37.0	29.0	3.0	16.0	38.2
Structure-Augmented RAG Agents															
RAPTOR	29.0	38.0	34.3	45.8	36.8	59.4	12.3	35.9	13.4	42.3	27.9	14.0	1.0	7.5	27.0
GraphRAG	47.0	47.0	35.0	34.4	40.9	39.8	9.8	24.8	0.4	39.4	19.9	14.0	2.0	8.0	23.4
MemoRAG	29.0	33.0	20.0	56.0	34.5	77.0	13.1	**45.1**	9.2	50.7	30.0	21.0	**7.0**	14.0	30.9
HippoRAG-v2	**76.0**	66.0	50.7	67.6	**65.1**	61.4	10.2	35.8	14.6	57.7	36.2	**54.0**	5.0	**29.5**	**41.6**
Mem0	25.0	32.0	36.0	37.5	32.6	32.4	10.0	21.2	4.8	36.6	20.7	18.0	2.0	10.0	21.1
Cognee	31.0	26.0	29.3	26.8	28.3	35.4	10.1	22.8	2.3	29.6	16.0	28.0	3.0	15.5	20.6
Zep	44.0	25.0	38.3	42.5	37.5	62.8	12.1	37.5	4.2	28.2	16.2	7.0	3.0	5.0	24.0
Agentic Memory Agents															
Self-RAG	35.0	42.0	25.7	31.8	33.6	11.6	12.8	12.2	0.9	35.2	18.1	19.0	3.0	11.0	18.7
MemGPT	41.0	38.0	32.0	26.2	34.3	67.6	14.0	40.8	2.5	42.3	22.4	28.0	3.0	15.5	28.3
MIRIX	62.0	61.0	37.3	29.8	47.5	38.4	9.8	24.1	9.9	40.8	25.4	14.0	2.0	8.0	26.2
MIRIX (4.1-mini)	73.0	**75.0**	**51.0**	53.0	63.0	61.0	10.3	35.7	18.9	**62.0**	**40.5**	20.0	3.0	11.5	37.7

Solving the problem of Selective forgetting can alone contribute significantly in memory research.

Selective Forgetting:-
Any deliberate mechanism by which an AI system reduces, removes, decays, compresses, or suppresses specific stored information — whether parametric (weights), structural (external memory), or operational (context) — while preserving the system's ability to perform downstream tasks.

Forgetting is necessary due to following reasons:-

1. Retrieval Quality
2. Personalisation
3. Stability
4. Storage Efficiency

A basic code demo for selective forgetting:- Code

A novel approach of MemGPT architecture:- Sparse Priming Representation

Indepth discussion on MemGPT's architecture:- Notes