Hippocampus: An Efficient and Scalable Memory Module for Agentic AI

Yi Li[1,2], Lianjie Cao[1], Faraz Ahmed[1], Puneet Sharma[1], Bingzhe Li[2]
[1] HPE Labs, [2] University of Texas at Dallas

May 19, 2026

Agenda

01	What is Agentic Memory
02	Existing Memory Solutions
03	The Bottleneck
04	Hippocampus – Core Ideas
05	Evaluation and Conclusion

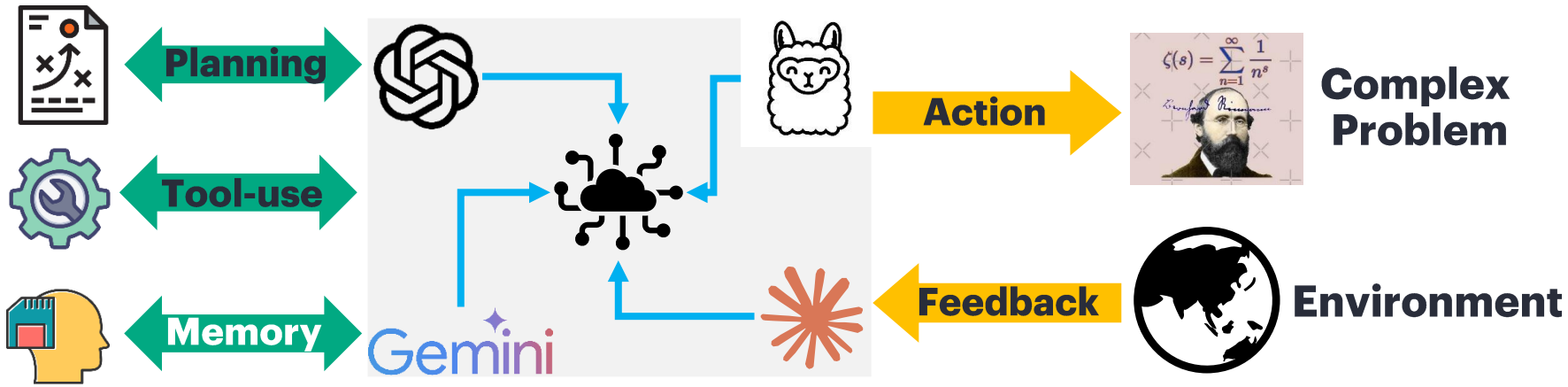
Memory for Agentic AI

What is agentic memory?

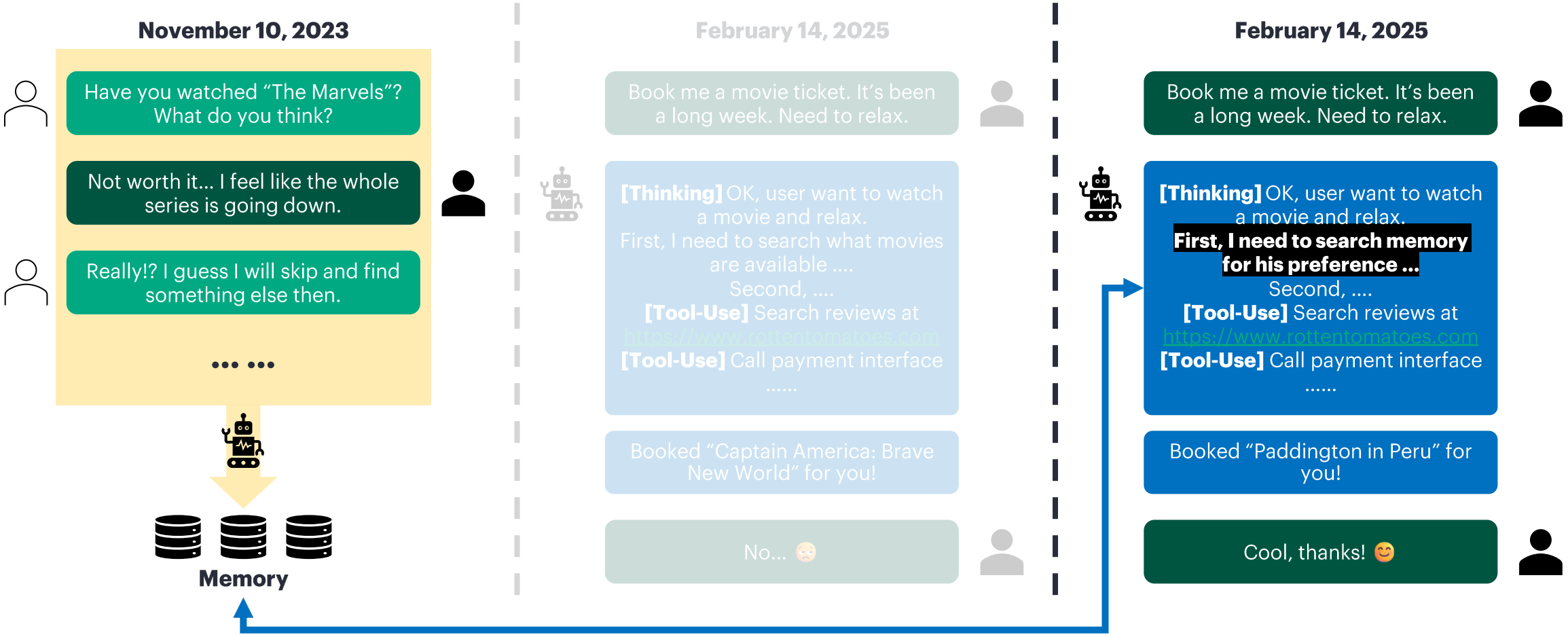
- A persistent mechanism that lets an AI agent store, update, and retrieve information over time.
- Internal memory (model parameters) and external memory (contextual information).

Why does it matter?

- Agents increasingly operate over long-horizon, multi-session tasks.
- LLMs still struggle to use information effectively in long contexts.
- Memory enables continuity, personalization, adaptation, and better decisions over time.



Agentic Memory – A Personal Assistant Example

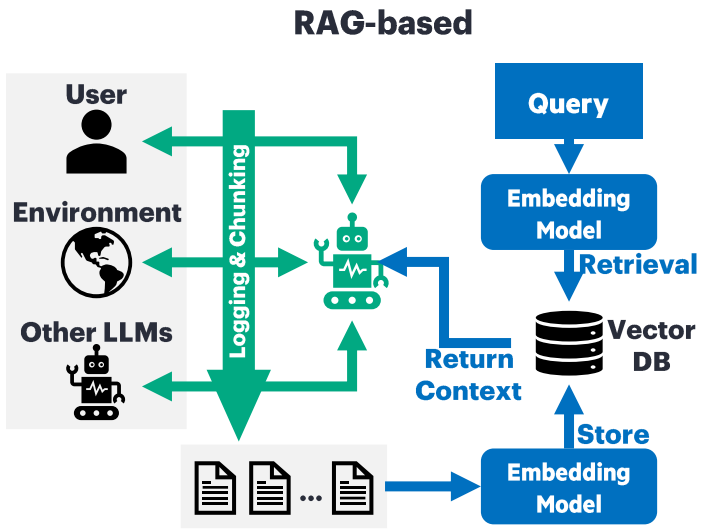


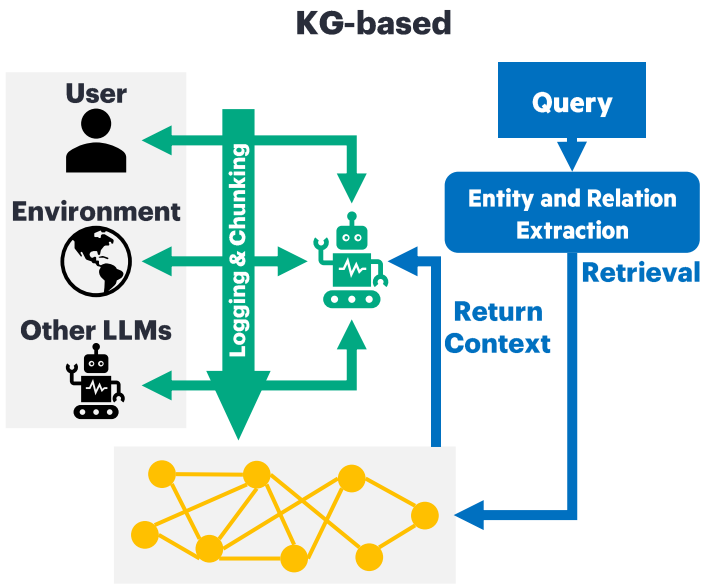
Existing Solutions

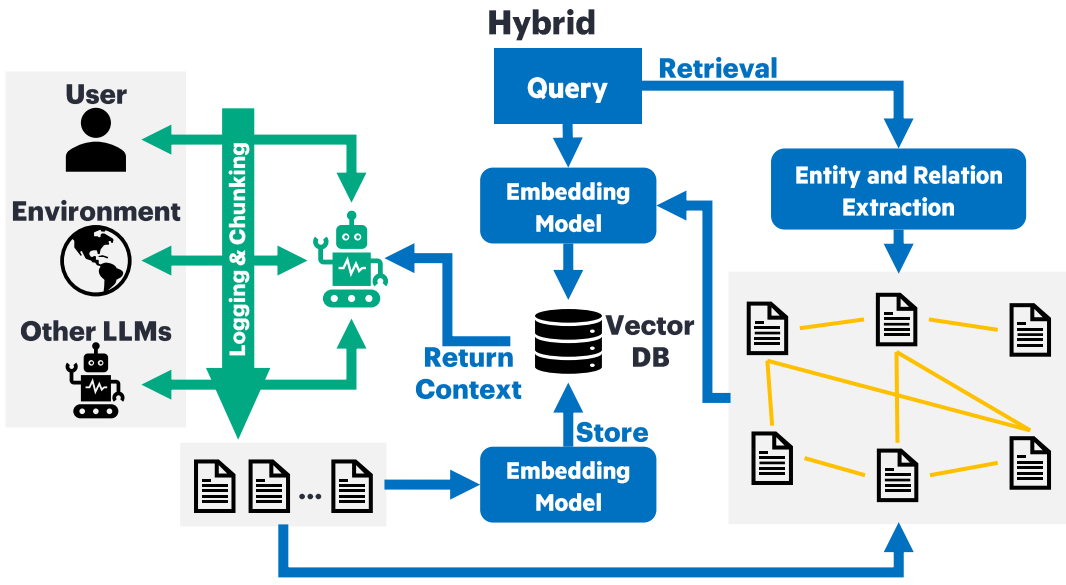
Source: https://github.com/AgentMemoryWorld/Awesome-Agent-Memory

Current Designs

- Three mainstream solutions
  - **Retrieval-Augmented Generation (RAG)-based:** MemoryOS, Mem0, ...
  - **Knowledge graph (KG)-based:** Zep, Mem0-Graph, ...
  - **Hybrid (RAG + KG)-based:** A-Mem, ...



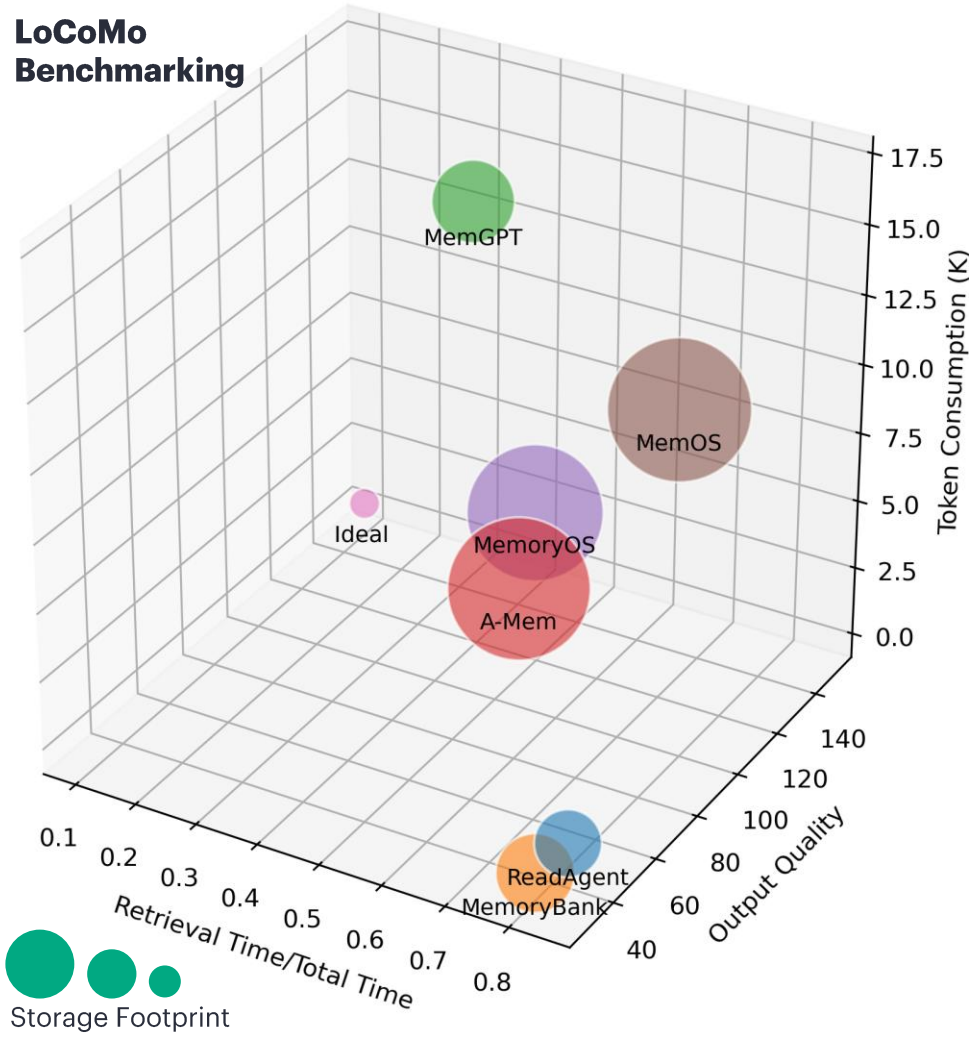




Identifying the Bottleneck

- Dataset/Benchmark: LoCoMo
  - 4 different task types: single-hop, multi-hop, temporal, open-domain
- Solutions
  - Commercial (API-only): Zep, Mem0 (-Graph)
  - Open-source: ReadAgent, MemoryBank, MemGPT, A-Mem, MemoryOS
  - Baseline: full-context, RAG, knowledge graph (Neo4J)
- Base model (as agents): GPT 4o-mini
- Metric
  - Output quality: F1 score
  - Efficiency: token consumption, retrieval time, and storage footprint

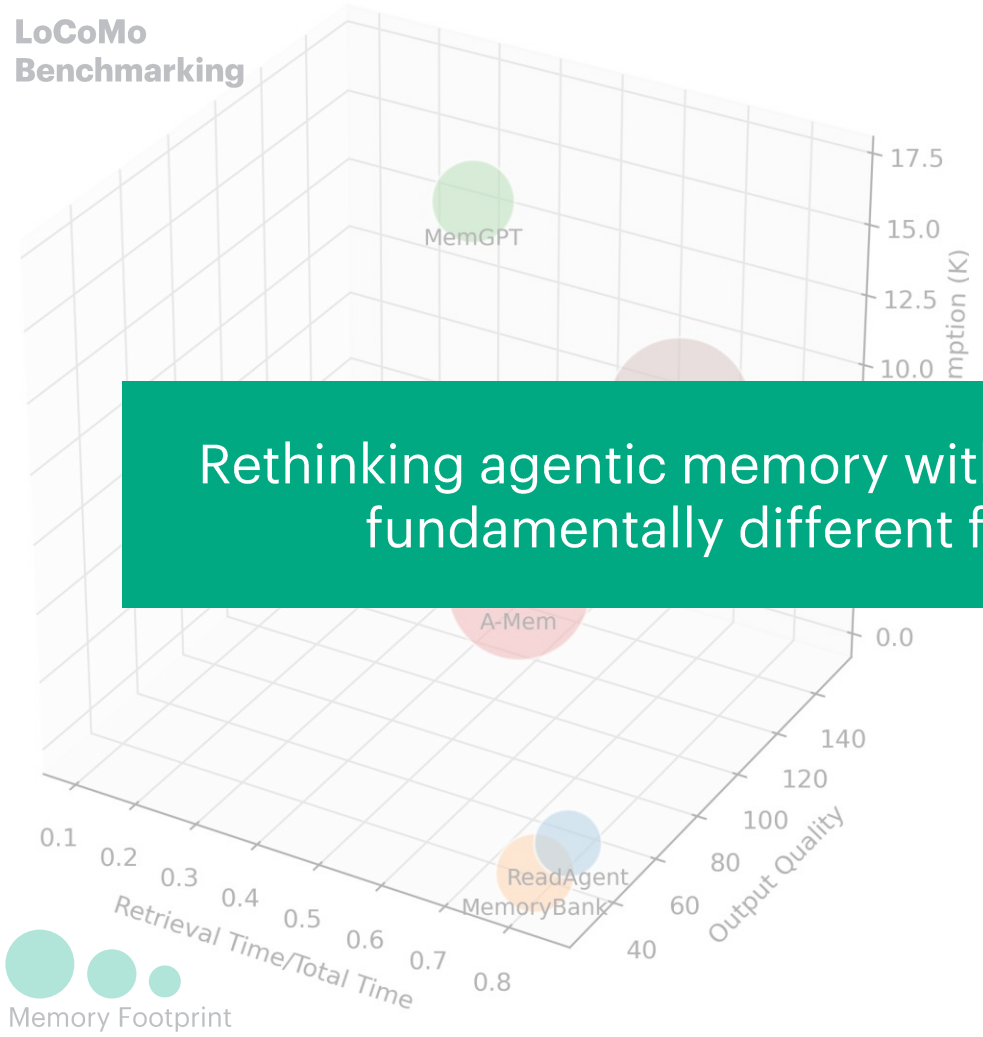
Benchmarking Results



- High-accuracy designs (MemGPT and A-Mem) achieve strong F1 scores but incur significant latency and token overhead.
- Lightweight systems (MemoryBank) reduce latency and cost but suffer from degraded output quality.
- Insertion overhead:
  - RAG: chunking, embedding, and index updates
  - KG: fact insertions and graph index maintenance
  - Hybrid: additional abstraction creation, cross-linking

None of the evaluated systems simultaneously optimize both quality and efficiency.

Benchmarking Results



- High-accuracy designs (MemGPT and A-Mem) achieve strong F1 scores but incur significant latency and token overhead.
- Lightweight systems (MemoryBank) reduce latency and cost but suffer from degraded output quality.
- Insertion overhead:

ng

Rethinking agentic memory with a lightweight, compression-native substrate fundamentally different from vector embedding-heavy designs.

None of the evaluated systems simultaneously optimize all four dimensions.

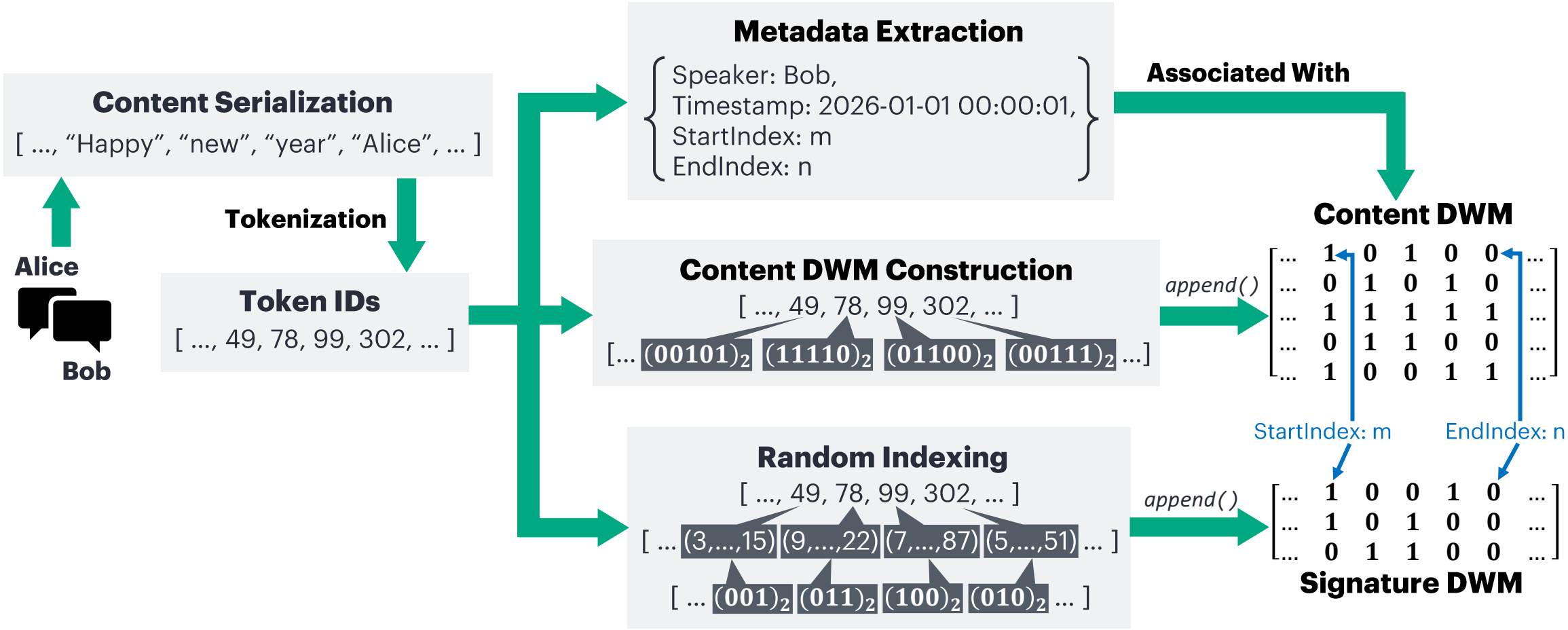
Hippocampus – Core Ideas

- Lightweight and efficient memory substrate
  - Token ID ⇒ Vector embeddings

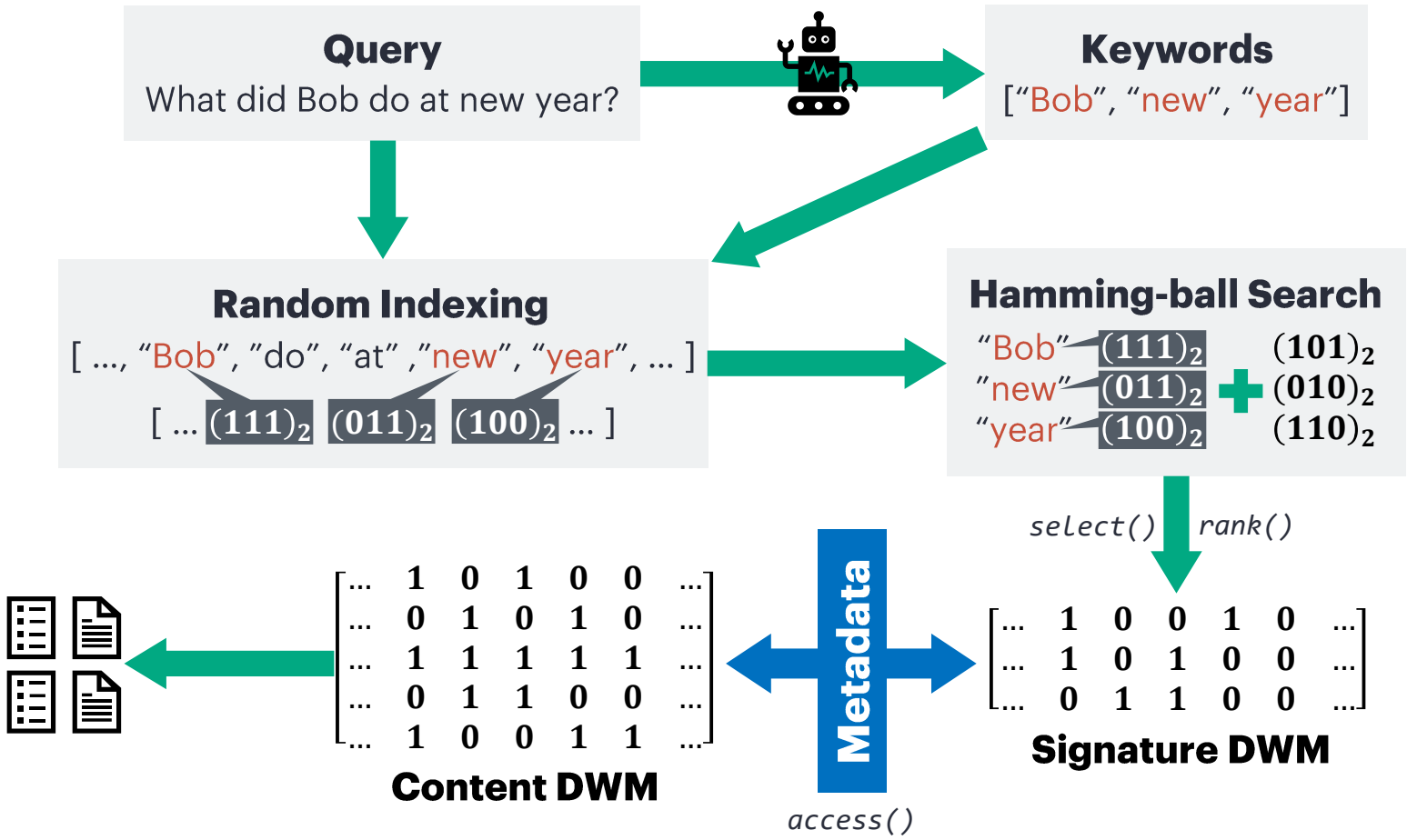
Hippocampus – Core Ideas

- Lightweight and efficient memory substrate
  - ~~Token ID ⇒ Vector embeddings~~ Token ID ⇒ Wavelet Matrix (WM), a succinct data structure
  - Token IDs are LLM-native and WM is compression-native
- Key Challenges with WM
  - Canonical WM is a statically constructed over a fixed sequence
  - WM supports efficient exact symbol queries, but not semantic retrieval
- Dynamic Wavelet Matrix (DWM)
  - Extend WM to support incremental, append-friendly construction
  - Store token-ID sequences in a **Content DWM** for exact, lossless reconstruction
- Semantic Indexing
  - Modify lightweight embedding – **random indexing**
  - Build a parallel **Signature DWM** as the semantic representation
- Augmented Retrieval
  - Adopt **Hamming-ball search** on Signature DWM for efficient approximate matching

Memory Construction



Memory Retrieval



Evaluation

- Data sets: LoCoMo benchmark and LongMemEval-S
- Comparison: 6 different agentic memory systems
- Metrics
  - Output quality: F1, BLEU-1, and LLM-as-a-Judge
  - Efficiency: token consumption, retrieval time, and storage footprint,

Accuracy	Single-Hop			Multi-Hop		
	F1	BLEU-1	LLM Judge	F1	BLEU-1	LLM Judge
ReadAgent	8.78	5.93	1.03	5.44	5.03	1.01
MemoryBank	5.05	3.97	2	6.02	5.89	1.12
MemGPT	25.43	17.68	1.91	9.11	8.82	1.06
A-Mem	19.82	19.86	2.66	12.97	12.81	1.85
MemoryOS	32.5	30.13	2.76	28.61	26.81	1.79
MemOS	39.24	40.76	2.75	30.11	30.91	2.56
Hippocampus	34.36	30.04	**3.08**	31.97	31.85	**3.22**

Retrieval Time	Search Time (s)	Total Time (s)	Search Ratio
ReadAgent	28.45	33.61	0.84
MemoryBank	0.87	1.08	0.81
MemGPT	3.24	6.95	0.46
A-Mem	1.12	2.34	0.51
MemoryOS	1.07	2.3	0.47
MemOS	0.8	1.21	0.66
Hippocampus	**0.13**	**1.08**	**0.12**

Conclusion

- Identified the prevailing vector-embedding substrate as a key efficiency bottleneck, motivating a **lightweight, compression-native** memory substrate for agentic AI systems.
- Extended the Wavelet Matrix into a **Dynamic Wavelet Matrix (DWM)** to support incremental, append-friendly construction for streaming memory workloads.
- Designed a **dual-representation memory layout** with **Content DWM** for exact token-ID reconstruction and Signature DWM for semantic retrieval via modified random indexing.
- Enabled efficient approximate matching with **Hamming-ball search**, providing low-cost semantic matching during retrieval.
- Evaluated Hippocampus on LoCoMo and LongMemEval-S benchmarks, demonstrating substantially **more efficient memory construction and retrieval** while maintaining strong retrieval quality.

MAGMA: A Multi-Graph based Agentic Memory Architecture

HAGE: Harnessing Agentic Memory via RL-Driven Weighted Graph Evolution

Thank you!

Q&A