HINDSIGHT: Structured Agent Memory that Retains, Recalls, and Reflects

Chris Latimer[♣], **Nicoló Boschi**[♣], **Andrew Neeser**[♢], **Chris Bartholomew**[♣], **Gaurav Srivastava**[♡], **Xuan Wang**[♡], **Naren Ramakrishnan**[♡]

[♣]Vectorize.io, USA
[♢]The Washington Post, USA
[♡]Virginia Tech, USA

GitHub: https://github.com/vectorize-io/hindsight
Website: https://hindsight.vectorize.io/
PyPI: https://pypi.org/project/hindsight-all/

Abstract

We demonstrate HINDSIGHT, a working memory system for AI agents that organizes long-term memory into four logical networks and exposes three core operations. The world, experience, observation, and opinion networks separate objective facts from subjective beliefs, giving developers visibility into what an agent knows versus what it believes. The retain, recall, and reflect operations handle ingestion, retrieval, and reasoning respectively, with a parallel pipeline that combines vector search, keyword matching, graph traversal, and temporal filtering, backed by PostgreSQL with pgvector. Unlike existing systems such as MemGPT, Zep, and Mem0, HINDSIGHT is the only one that jointly provides fact-belief separation, temporal entity graphs, evolving opinions with confidence scores, and configurable behavioral profiles. On LongMemEval and LoCoMo, HINDSIGHT with a 20B open-source model reaches 83.6% and 83.2% accuracy, outperforming full-context GPT-4o and all prior memory systems; with Gemini-3 Pro, LongMemEval accuracy reaches 91.4%. Our interactive demo lets users build memory graphs through multi-session conversations, inspect how memories are classified, and watch opinions form and change. The system is **open-source under the MIT license**, available as a Python package (`pip install hindsight-all`) and Docker image, with **13.3K GitHub stars** and 763 forks to date, and in production use at Fortune 500 enterprises. Video demo: https://youtu.be/4M2wS-yEmVA.

1 Introduction

Large language models have unlocked a new generation of AI agents that can plan, reason, and take actions in open-ended environments (Yao et al., 2023; Wei et al., 2022; Shinn et al., 2023). Early work on generative agents showed that equipping LLMs with persistent memory produces believable, long-running character behavior (Park et al., 2023), and cognitive architecture research has since laid out principled designs for perception, memory, and action modules in language agents (Sumers et al., 2024). Recent surveys on LLM-based agent memory identify three core challenges that remain open: how to retain the right information from growing interaction histories, how to recall it efficiently at query time, and how to reason over it coherently (Zhang et al., 2025b; Wu et al., 2025; Zhang et al., 2025a).

Most agent memory systems today operate as thin retrieval layers around stateless language models. They extract snippets from conversations, store them in vector or graph-based stores, and retrieve top-k items into the prompt when needed, following the retrieval-augmented generation (RAG) paradigm (Lewis et al., 2020; Mialon et al., 2023). While this improves context carry-over, it struggles with three problems: it cannot selectively access information across many sessions (Tavakoli et al., 2025), it conflates what the agent observed with what it believes (Shan et al., 2025), and it produces responses that are locally plausible but globally inconsistent (Huang et al., 2025). Systems like MemGPT (Packer et al., 2023) introduce tiered memory management, Zep (Rasmussen et al., 2025) adds temporal knowledge graphs, and MemoryBank (Zhong et al., 2024) proposes forgetting mechanisms inspired by human cognition, but none of them jointly address all three challenges.

We present HINDSIGHT, a working memory system that addresses these challenges through two ideas. First, all agent memory is organized into four logical networks that separate world facts, agent experiences, entity summaries, and evolving opinions.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 275–285

Second, three explicit operations, retain, recall, and reflect, govern how information enters the system, how it is retrieved, and how the agent reasons.

HINDSIGHT is a compound AI system with two main components. TEMPR (Temporal Entity Memory Priming Retrieval) builds a temporal, entity-aware memory graph and provides multi-strategy retrieval with configurable token budgets. CARA (Coherent Adaptive Reasoning Agents) produces preference-conditioned responses and maintains opinions that evolve as new evidence arrives. Both components use configurable LLM backbones and are designed for production deployment, with drop-in wrappers for existing OpenAI and Anthropic clients (Appendix E).

On LongMemEval (Wu et al., 2024) and LoCoMo (Maharana et al., 2024), HINDSIGHT with an open-source 20B model achieves 83.6% and 83.2% overall accuracy, outperforming full-context GPT-4o and prior memory systems such as Zep (Rasmussen et al., 2025), Mem0 (Chhikara et al., 2025), and A-Mem (Xu et al., 2025); scaling to Gemini-3 Pro pushes LongMemEval accuracy to 91.4%. HINDSIGHT is designed for **developers and researchers building AI agents** that need persistent, structured memory across sessions, such as chatbots, personal assistants, and autonomous task agents. The system is released under the **MIT license** and can be installed via PyPI (`pip install hindsight-all`) or Docker. The full architecture is in (Latimer et al., 2025); here we focus on system design and demonstration.

2 Related Work

Three lines of work are most relevant to HINDSIGHT: memory-augmented language agents, retrieval for conversational memory, and cognitive foundations for structuring agent knowledge.

Memory-augmented agents. MemGPT (Packer et al., 2023) pioneered tiered memory management for LLMs by treating the context window as a virtual memory hierarchy. Zep (Rasmussen et al., 2025) builds temporal knowledge graphs from conversations and supports time-aware retrieval. Mem0 (Chhikara et al., 2025) focuses on production-ready scalable memory with graph-based storage. A-Mem (Xu et al., 2025) introduces an agentic approach where the LLM itself decides how to organize its memories. MemoryBank (Zhong et al., 2024) incorporates an Ebbinghaus forgetting curve to model memory decay. KARMA (Wang et al., 2025) applies long-and-short term memory to embodied agents, while MemVerse (Liu et al., 2025) extends memory to multimodal lifelong learning settings. Memory-R1 (Yan et al., 2025) uses reinforcement learning to train agents to manage their own memory operations. HINDSIGHT differs from all of these by jointly providing four-network epistemic separation, temporal entity graphs, opinion evolution with confidence tracking, and configurable behavioral profiles within a single system.

Retrieval techniques. Our multi-strategy recall pipeline combines HNSW-based vector search (Malkov and Yashunin, 2020), BM25 keyword retrieval (Robertson and Zaragoza, 2009), graph traversal, and temporal filtering. Reciprocal Rank Fusion (Cormack et al., 2009) merges these heterogeneous ranked lists, and a cross-encoder (Nogueira and Cho, 2019) provides final reranking. This hybrid approach draws on the broader RAG literature (Lewis et al., 2020; Mialon et al., 2023) but extends it with graph and temporal channels specific to conversational memory.

Cognitive foundations. Our four-network design is inspired by cognitive science models of human memory (Sumers et al., 2024; Shan et al., 2025). Recent surveys (Zhang et al., 2025b; Wu et al., 2025; Zhang et al., 2025a) map these cognitive constructs to LLM agent architectures and identify the separation of factual and belief-like memory as a key open problem, which HINDSIGHT addresses directly. We next describe how HINDSIGHT's architecture builds on these foundations.

3 System Architecture

HINDSIGHT organizes all agent memory into four networks and exposes three operations over them. Figure 1 shows the end-to-end data flow.

3.1 Four-Network Memory Organization

Each memory bank maintains four networks, each storing a different kind of information. This design is informed by research on cognitive memory in LLMs (Shan et al., 2025) and surveys mapping human memory structures to agent architectures (Wu et al., 2025):

- **World network:** objective facts about the external environment (e.g., "Alice works at Google in Mountain View on the AI team").

Feature	MemGPT	Zep	A-Mem	Mem0	HINDSIGHT
Fact/opinion separation	✗	✗	✗	✗	✓
Temporal reasoning	✗	✓	✗	✗	✓
Entity-aware graph	✗	✓	✓	✓	✓
Opinion evolution	✗	✗	✗	✗	✓
Behavioral profiles	✗	✗	✗	✗	✓
Multi-strategy retrieval	✗	✗	✗	P	✓

Table 1: Comparison of memory architectures. ✓ = present, ✗ = absent, P = partial.

- **Experience network:** first-person records of the agent's own actions (e.g., "I recommended Yosemite to Alice for hiking").
- **Observation network:** preference-neutral entity summaries synthesized from underlying facts (e.g., "Alice is a software engineer at Google specializing in ML").
- **Opinion network:** subjective beliefs with confidence scores that evolve over time (e.g., "Python is better for data science because of pandas"; confidence: 0.85).

This separation provides epistemic clarity: developers can inspect what the agent knows versus what it believes, and trace opinions back to the facts that support them.

Table 1 compares HINDSIGHT with existing memory systems across key architectural features. Unlike MemGPT (Packer et al., 2023), which focuses on tiered context management, or A-Mem (Xu et al., 2025), which builds agentic memory without temporal reasoning, HINDSIGHT is the only system that combines all of these capabilities: separating facts from beliefs, supporting temporal reasoning and entity-aware graphs, maintaining evolving opinions with confidence scores, and offering configurable behavioral profiles. KARMA (Wang et al., 2025) and MemVerse (Liu et al., 2025) explore related ideas in embodied and multimodal settings respectively, but neither provides the structured four-network separation with opinion evolution.

3.2 Three Core Operations

Retain ingests input data (conversations, documents) and converts them into structured narrative facts. Each fact is classified into one of the four networks, timestamped with occurrence intervals, linked to resolved entities, and connected to related memories through temporal, semantic, entity, and causal graph edges. The system extracts 2–5 coarse-grained narrative facts per conversation, where each fact covers an entire exchange rather than individual utterances. This makes downstream retrieval less sensitive to segmentation choices.

Recall takes a query and a token budget and returns a ranked set of memories that fit within the budget. It runs four retrieval channels in parallel: (1) vector similarity search using HNSW-indexed embeddings (Malkov and Yashunin, 2020), (2) BM25 keyword matching using GIN-indexed full-text search (Robertson and Zaragoza, 2009), (3) graph-based spreading activation that traverses entity, temporal, and causal links, and (4) temporal filtering that matches occurrence intervals against query date ranges. Results are merged using Reciprocal Rank Fusion (RRF) (Cormack et al., 2009) and refined with a cross-encoder reranker (Nogueira and Cho, 2019). The token budget lets callers control how much context to retrieve, keeping inference costs predictable regardless of memory bank size.

Reflect takes a query along with a behavioral profile and generates a response conditioned on retrieved memories. The behavioral profile is defined by three disposition traits (skepticism, literalism, empathy) on an integer scale of 1–5, which together control how strongly the bank's accumulated beliefs shape the response. During reflection, the system may form new opinions or update existing ones through a reinforcement mechanism: supporting evidence increases confidence, contradicting evidence decreases it. This is conceptually related to verbal reinforcement in Reflexion (Shinn et al., 2023), but applied to persistent opinion state rather than episodic task feedback.

3.3 Component Design

TEMPR implements retain and recall. During retain, an LLM extracts narrative facts from conversational transcripts, performs entity recognition and resolution using string similarity and co-occurrence patterns, and constructs four types of graph links (temporal, semantic, entity, causal). Entity links connect memories about the same person or thing across distant sessions, enabling multi-hop discovery that pure vector search would miss (Johnson et al., 2021). Each memory unit carries rich temporal metadata: an occurrence interval (τ_s, τ_e) that records when the event happened, and a mention timestamp τ_m that records when the agent learned

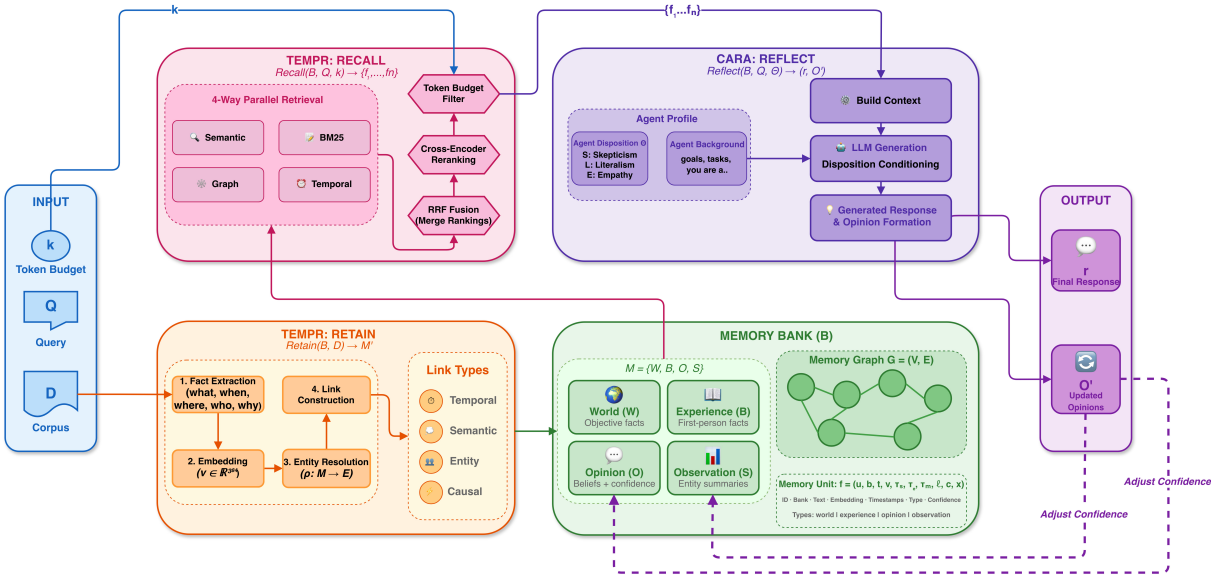

Figure 1: **HINDSIGHT architecture.** Input flows through TEMPR's retain pipeline (fact extraction, embedding, entity resolution, link construction) into a four-network bank. Given a query and token budget, TEMPR's recall runs four-way retrieval (semantic, BM25, graph, temporal) with RRF fusion and reranking. CARA's reflect generates preference-conditioned responses and updates opinions.

about it. This supports both point events and extended periods, and enables recency-aware ranking.

CARA implements reflect. Each memory bank has a configurable behavioral profile that modulates response generation. Opinions carry confidence scores in $[0, 1]$ and are stored separately from facts. When new facts arrive through retain, CARA identifies related opinions using entity overlap and embedding similarity, assesses whether the new evidence supports or contradicts each opinion, and adjusts confidence scores accordingly. Recent work on reinforcement-learned memory management (Yan et al., 2025) explores a complementary approach where the agent learns when to store and retrieve; HINDSIGHT instead uses explicit operations with structured schemas to give developers full control. This produces agents whose beliefs evolve with evidence rather than remaining fixed or getting overwritten.

A background *consolidation engine* refines observations from accumulated facts. Each observation tracks a proof count, supporting quotes, and a freshness trend (*new*, *strengthening*, *stable*, *weakening*, *stale*) so developers can tell whether a belief is still well supported. New facts refine existing observations rather than overwriting them, keeping the evidence trail intact. Reflect runs as an agentic loop: the agent decides which tools to call (`search_observations`, `search_mental_models`, `recall`, `expand`) and iterates until it has enough evidence. Hierarchical retrieval queries mental models first, then observations, then raw facts. Banks also support *directives*: prioritized hard rules injected into every response, giving developers explicit control on top of the disposition traits.

4 Implementation

Having described the architecture, we now cover the storage backend, retrieval pipeline, and deployment details that make HINDSIGHT practical for production use.

4.1 Storage and Retrieval

HINDSIGHT is deployed as a REST service backed by PostgreSQL with the pgvector extension. Each memory unit is stored as a database row containing the narrative text, embedding vector, temporal metadata (occurrence interval and mention timestamp), fact type, confidence score (for opinions), and auxiliary fields including access counts and full-text search vectors. HNSW indexes (Malkov and Yashunin, 2020) support sub-linear vector search, while GIN indexes enable BM25 (Robertson and Zaragoza, 2009) full-text retrieval, both within a single database without requiring external search infrastructure like Elasticsearch or dedicated vector stores such as FAISS (Johnson et al., 2021). This single-database design simplifies deployment and avoids the consistency issues that arise when

memory state is split across multiple systems.

The four recall channels run in parallel. Vector and BM25 search use PostgreSQL indexes directly. Graph traversal performs bounded breadth-first search (typically 2–3 hops) with activation decay and configurable link-type multipliers that upweight causal and entity edges over weaker semantic or long-range temporal edges. Temporal retrieval uses a hybrid parser: a rule-based analyzer handles common date expressions (e.g., "last week", "in March"), with a lightweight T5 model (flan-t5-small) as fallback for expressions the rules cannot resolve. After RRF (Cormack et al., 2009) merges the four ranked lists, a cross-encoder (ms-marco-MiniLM-L-6-v2) (Nogueira and Cho, 2019) reranks the top candidates for precision.

4.2 LLM Integration and Latency

All LLM calls, including fact extraction, opinion formation, and answer generation, use a configurable backbone. Structured output is enforced through Pydantic schemas that validate fact types, temporal ranges, entity annotations, and causal relationships. This lets the system swap between open-source and commercial models without changing the pipeline logic.

Observation generation (entity summary updates) runs asynchronously in the background, so retain latency is dominated by the fact extraction LLM call rather than summary recomputation. This keeps write-path latency low while gradually improving entity profiles as new facts arrive. Recall latency is bounded by the parallel retrieval channels and reranker, both of which operate over indexed data and a small candidate set (typically 20–50 candidates before reranking). Reflect adds one LLM generation call on top of recall, so end-to-end query latency is driven primarily by the backbone model's inference speed. For a bank with 10,000 memory units, recall completes in under 200ms excluding the backbone LLM call, and the token budget mechanism ensures this remains stable as the bank grows.

4.3 Production Scalability

The single-database design is deliberate in a space where Elasticsearch and FAISS are common. PostgreSQL with HNSW (pgvector) and GIN indexes scales well for production memory banks, and recall stays bounded by the token budget rather than by corpus size. For larger workloads, HINDSIGHT supports schema-per-tenant isolation, connection pooling with PgBouncer, and read-replica routing. An Oracle AI Database backend with full feature parity is also supported. Avoiding a separate vector store removes a class of consistency bugs and keeps deployment to a single connection string, backup, and monitoring target. Blob attachments can be offloaded to S3, Azure, or GCS through pluggable adapters.

4.4 Multilingual Support

HINDSIGHT detects the input language at retain time and preserves it through the pipeline. Facts are extracted in the original language, entities are stored in their native script, and reflect responses match the query language. No per-language configuration is needed.

4.5 Scalability and API

The token budget mechanism is the primary scalability control. Regardless of how large a memory bank grows, recall returns a bounded amount of context, preventing the downstream LLM's window from overflowing and keeping per-query cost stable. Each memory bank is isolated with its own four-network structure and behavioral profile, so the system serves multiple agents concurrently without cross-contamination.

The system exposes a REST API with three endpoints corresponding to the core operations: `retain` for ingesting data, `recall` for retrieving memories given a query and token budget, and `reflect` for generating preference-conditioned responses. Banks are created with a name, background description, and the three disposition traits (skepticism, literalism, empathy), and can be reconfigured at any time. Beyond the REST API, HINDSIGHT ships an **MCP (Model Context Protocol) server** so that AI coding assistants and any MCP-compatible client can read and write memory directly. An **extensions framework** provides pluggable hooks for multi-tenant authentication (a Supabase JWT extension is bundled and gives each user a separate PostgreSQL schema), custom HTTP endpoints, and per-operation validation for rate limiting and audit logging. Webhook subscriptions notify external systems when memories are retained or when observations change, which enables event-driven downstream pipelines. Out-of-the-box **integrations** are provided for LangGraph, CrewAI, AutoGen, Pydantic-AI, OpenAI Agents, LlamaIndex, Smolagents, Strands, AG2, AgentCore, n8n, Dify, Pipecat, and the Vercel AI SDK, in addition

System	LongMemEval	LoCoMo
Full-context (GPT-4o)	60.2	–
Full-context (OSS-20B)	39.0	–
Zep (GPT-4o)	71.2	75.1
Memobase	–	75.8
Supermemory (Gemini-3)	85.2	–
Backboard	–	**90.0**
HINDSIGHT (OSS-20B)	83.6	83.2
HINDSIGHT (OSS-120B)	89.0	85.7
HINDSIGHT (Gemini-3)	**91.4**	89.6

Table 2: Accuracy (%) on LongMemEval (S, 500 Qs) and LoCoMo. HINDSIGHT with a 20B open backbone beats full-context GPT-4o and prior systems. Best per column in bold.

to drop-in LLM-client wrappers for OpenAI, Anthropic, and 100+ providers through LiteLLM (Appendix E). Appendix A covers installation options, Appendix B provides code examples, Appendix C lists structured output schemas, and Appendix D gives per-category benchmark breakdowns.

5 Demonstration

HINDSIGHT is available as an installable package via `pip install hindsight-all` and as a Docker image at `https://github.com/vectorize-io/hindsight`. A hosted cloud version is accessible at `https://ui.hindsight.vectorize.io/signup`, and full documentation is at `https://hindsight.vectorize.io`. Our demonstration walks attendees through the full lifecycle of HINDSIGHT's memory system. The demo consists of five parts.

Multi-session conversation. Attendees interact with a HINDSIGHT-backed agent over several conversation sessions, providing information about people, events, and preferences. The system incrementally builds a structured memory graph that attendees can watch grow in real time.

Memory inspection. Attendees browse the four memory networks to see how incoming conversations are decomposed into classified facts. They can inspect entity resolution results, graph link types, and how opinions form with confidence scores. This shows the epistemic separation that the four-network design provides.

Retrieval walkthrough. Attendees issue queries, including temporal questions ("What did we discuss last month?") and multi-hop questions that span sessions. The interface shows which retrieval channels contributed to each result, along with RRF fusion scores and reranker outputs, making the multi-strategy recall pipeline transparent.

Opinion evolution. Attendees observe how opinions form and change as new evidence is retained. For example, after telling the agent "I tried React for my project and it worked great," the agent forms the opinion "React is a good framework choice" with moderate confidence. In a later session, if the attendee says "Actually, the React project became hard to maintain," the opinion's confidence drops. We show two agents with different behavioral profiles (e.g., high skepticism vs. high empathy) accessing the same facts but forming different opinions, illustrating how disposition parameters shape belief formation.

Benchmark explorer. Attendees explore per-question results from LongMemEval (Wu et al., 2024) and LoCoMo (Maharana et al., 2024) through our interactive results viewer, drilling into retrieved memories and model configurations for each response. This component also supports MemoryBench (Ai et al., 2025) questions, providing additional coverage of memory and continual learning evaluation scenarios (Tavakoli et al., 2025).

6 Evaluation

We summarize key results from the full evaluation in (Latimer et al., 2025). We test on two established benchmarks for long-term conversational memory: LongMemEval (Wu et al., 2024), which contains 500 questions across conversations spanning up to 1.5M tokens, and LoCoMo (Maharana et al., 2024), which uses multi-session human conversations with up to 35 sessions each. Both benchmarks are widely used to evaluate memory-augmented agents and conversational assistants (Zhang et al., 2025a; Ai et al., 2025).

6.1 Overall Results

Table 2 reports overall accuracy across both benchmarks. On LongMemEval, HINDSIGHT with an open-source 20B backbone achieves 83.6%, a 44.6-point gain over a full-context baseline with the same model and 23 points above full-context GPT-4o. Scaling the backbone to 120B reaches 89.0%, and using Gemini-3 Pro for answer generation pushes accuracy to 91.4%, the best result across all systems including those backed by frontier models.

On LoCoMo, HINDSIGHT consistently outperforms prior open memory systems. With the 20B

backbone, overall accuracy is 83.2%, compared to 75.8% for Memobase and 75.1% for Zep (Rasmussen et al., 2025). With Gemini-3, HINDSIGHT reaches 89.6%, competitive with Backboard's reported 90.0%.

6.2 Where the Architecture Helps Most

The largest gains on LongMemEval appear in exactly the categories that stress long-horizon memory. Multi-session questions, which require connecting information across different conversations, jump from 21.1% to 79.7% with the 20B backbone and reach 87.2% with Gemini-3. Temporal reasoning questions improve from 31.6% to 79.7% (20B) and 91.0% (Gemini-3), showing that TEMPR's temporal filtering and time-aware graph retrieval directly mitigate context dilution. Preference questions, which test whether the agent remembers user preferences stated in earlier sessions, improve from 20.0% to 66.7% (20B) and 80.0% (Gemini-3).

On LoCoMo, HINDSIGHT shows particularly strong open-domain performance: 91.0% with the 20B backbone and 95.1% with Gemini-3, the highest score across all systems on this category. Multi-hop questions, which require reasoning across multiple facts to produce an answer, reach 64.6% (20B) and 70.8% (Gemini-3). These results illustrate that the memory architecture matters more than model size. HINDSIGHT with a 20B model outperforms full-context GPT-4o and matches or exceeds systems backed by frontier models, because the structured four-network organization and multi-strategy retrieval prevent information from being lost as conversation history grows. This aligns with findings from recent benchmarking studies that show structured memory consistently outperforms brute-force long-context approaches when conversations exceed tens of thousands of tokens (Tavakoli et al., 2025; Ai et al., 2025).

6.3 Where Each Component Helps

The per-category breakdown in Table 4 isolates the components targeted at long-horizon memory. Temporal-reasoning and multi-session questions, which depend on the temporal channel and entity-aware graph traversal, rise from 45.1% and 44.3% (full-context GPT-4o) to 79.7% with the 20B backbone. Single-session categories, which a cross-encoder alone can address, are already strong in the full-context baseline (81.4% and 94.6%) and gain little. This is consistent with the cross-encoder acting as a precision step on top of structured retrieval rather than as the main source of accuracy. Preference questions, which depend on four-network separation and observation-level consolidation, improve from 20.0% to 66.7% with the same backbone, the largest relative gain. The full breakdown is in (Latimer et al., 2025), and shows that the structured memory design, not the reranker, accounts for most of the lift.

6.4 Qualitative Case Study

Consider a three-session interaction. In session 1 the user says "I started learning Rust last March for my team's payments rewrite"; in session 2 they ask "what should I focus on next quarter?"; in session 3 they say "Rust has been frustrating, I miss Go's compile times." Retain in session 1 creates a *world* fact (team rewrite in Rust), an *experience* fact (user began learning Rust), and updates the *observation* for the user-entity. At session 2, recall combines all four channels: the temporal channel binds "last March" to the Rust fact, graph traversal expands to the payments-service entity, and the observation supplies a consolidated profile. Reflect grounds its recommendation in the user's stated motivation. After session 3, the consolidation engine refines the existing observation: the new evidence weakens the earlier belief and the freshness trend flips from *stable* to *weakening*, while the world fact is unchanged. Both supporting and contradicting evidence link to the same observation, so a later response can acknowledge the frustration without losing context.

7 Conclusion

We presented HINDSIGHT, a working memory system for AI agents that organizes memory into four networks and exposes retain, recall, and reflect as explicit operations. The system separates evidence from beliefs, supports temporal and entity-aware retrieval through a parallel pipeline, and maintains evolving opinions with confidence tracking. On two standard benchmarks, the structured memory architecture lifts a 20B open-source model above full-context GPT-4o, and with Gemini-3 Pro reaches 91.4% on LongMemEval, the highest reported accuracy across all systems. HINDSIGHT is open-source under the MIT license, with code, documentation, Python/TypeScript/Go/Rust SDKs, and an interactive benchmark viewer available at `https://github.com/vectorize-io/hindsight`.

Acknowledgements

We thank the Vectorize.io engineering team for their work on the open-source release, and the early adopters and contributors who have helped shape HINDSIGHT through GitHub issues, pull requests, and Slack discussions. We are grateful to research collaborators at the Virginia Tech Sanghani Center for Artificial Intelligence and Data Analytics and at The Washington Post for independently reproducing the benchmark results.

Limitations

1) HINDSIGHT relies on LLM calls for fact extraction, entity resolution, and opinion formation. The quality of these steps depends on the backbone model, and extraction errors can propagate through the memory graph. **2)** All evaluation was conducted on English-language benchmarks (LongMemEval and LoCoMo). Performance on other languages has not been tested. **3)** The opinion evolution mechanism has not been validated through formal user studies. While benchmark results show strong retrieval accuracy, the subjective quality of opinion formation and confidence calibration would benefit from human evaluation. **4)** The system requires PostgreSQL with pgvector, which adds deployment complexity for teams that do not already use PostgreSQL (though the embedded Python package bundles its own PostgreSQL instance to lower this barrier). **5)** Temporal parsing uses a combination of rules and a lightweight T5 model, which handles common date expressions well but may miss highly ambiguous or culturally specific temporal references.

Ethics Statement

HINDSIGHT stores and organizes information from user conversations in persistent memory banks. This raises privacy considerations, as the system retains personal details, preferences, and opinions over time. Developers using HINDSIGHT should implement appropriate access controls, data retention policies, and user consent mechanisms. The system does not currently include built-in PII detection or automatic redaction. The opinion formation mechanism reflects patterns in the data it processes and may amplify biases present in the input. We encourage developers to monitor opinion networks for harmful or biased beliefs. All evaluation was conducted on publicly available benchmarks, and no personal data was collected. The system is released under the MIT license to support open research and responsible development.

Broader Impact Statement

HINDSIGHT provides long-term memory for AI agents, enabling them to learn from past interactions and maintain consistent beliefs over time. This can improve user experience in applications like personal assistants, customer support, and educational tools. However, persistent memory also introduces risks: long-term storage of user interactions could enable surveillance-like behavior if misused, and the ability to form and maintain opinions could lead agents to develop fixed views that resist correction. We address some of these risks through transparent memory organization: the four-network design makes all stored information inspectable, and configurable behavioral profiles give developers control over how opinions form. The open-source release allows the community to audit, extend, and improve safety mechanisms.

References

Qingyao Ai, Yichen Tang, Changyue Wang, Jianming Long, Weihang Su, and Yiqun Liu. 2025. MemoryBench: A benchmark for memory and continual learning in LLM systems. arXiv preprint. ArXiv:2510.17281.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready AI agents with scalable long-term memory. arXiv preprint. ArXiv:2504.19413.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *SIGIR*, pages 758–759, New York, NY, USA. ACM.

Jen-tse Huang, Kaiser Sun, Wenxuan Wang, and Mark Dredze. 2025. LLMs do not have human-like working memory. arXiv preprint. ArXiv:2505.10571.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Chris Latimer, Nicoló Boschi, Andrew Neeser, Chris Bartholomew, Gaurav Srivastava, Xuan Wang, and Naren Ramakrishnan. 2025. Hindsight is 20/20: Building agent memory that retains, recalls, and reflects. arXiv preprint. ArXiv:2505.XXXXX.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020.

Retrieval-augmented generation for knowledge-intensive NLP tasks. NeurIPS. ArXiv:2005.11401.

Junming Liu, Yifei Sun, Weihua Cheng, Haodong Lei, Yirong Chen, Licheng Wen, Xuemeng Yang, Daocheng Fu, Pinlong Cai, Nianchen Deng, and 1 others. 2025. MemVerse: Multimodal memory for lifelong learning agents. arXiv preprint. ArXiv:2512.03627.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of LLM agents. arXiv preprint. ArXiv:2402.17753.

Yury A. Malkov and Dmitry A. Yashunin. 2020. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented language models: a survey. Transactions on Machine Learning Research. ArXiv:2302.07842.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. arXiv preprint. ArXiv:1901.04085.

Charles Packer, Vivian Fang, Shishir G. Patil, Kevin Lin, Sarah Wooders, and Joseph E. Gonzalez. 2023. MemGPT: Towards LLMs as operating systems. arXiv preprint. ArXiv:2310.08560.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. UIST. ArXiv:2304.03442.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. 2025. Zep: A temporal knowledge graph architecture for agent memory. arXiv preprint. ArXiv:2501.13956.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Lianlei Shan, Shixian Luo, Zezhou Zhu, Yu Yuan, and Yong Wu. 2025. Cognitive memory in large language models. arXiv preprint. ArXiv:2504.02441.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. NeurIPS. ArXiv:2303.11366.

Theodore R. Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. 2024. Cognitive architectures for language agents. Transactions on Machine Learning Research. ArXiv:2309.02427.

Mohammad Tavakoli, Alireza Salemi, Carrie Ye, Mohamed Abdalla, Hamed Zamani, and J Ross Mitchell. 2025. Beyond a million tokens: Benchmarking and enhancing long-term memory in LLMs. arXiv preprint. ArXiv:2510.27246.

Zixuan Wang, Bo Yu, Junzhe Zhao, Wenhao Sun, Sai Hou, Shuai Liang, Xing Hu, Yinhe Han, and Yiming Gan. 2025. KARMA: Augmenting embodied AI agents with long-and-short term memory systems. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8, Piscataway, NJ, USA. IEEE.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. NeurIPS. ArXiv:2201.11903.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2024. LongMemEval: Benchmarking chat assistants on long-term interactive memory. arXiv preprint. ArXiv:2410.10813.

Yaxiong Wu, Sheng Liang, Chen Zhang, Yichao Wang, Yongyue Zhang, Huifeng Guo, Ruiming Tang, and Yong Liu. 2025. From human memory to AI memory: A survey on memory mechanisms in the era of LLMs. arXiv preprint. ArXiv:2504.15965.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025. A-Mem: Agentic memory for LLM agents. arXiv preprint. ArXiv:2502.12110.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Kristian Kersting, Jeff Z Pan, Hinrich Schütze, and 1 others. 2025. Memory-R1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. arXiv preprint. ArXiv:2508.19828.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. ICLR. ArXiv:2210.03629.

Dianxing Zhang, Wendong Li, Kani Song, Jiaye Lu, Gang Li, Liuchun Yang, and Sheng Li. 2025a. Memory in large language models: Mechanisms, evaluation and evolution. arXiv preprint. ArXiv:2509.18868.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2025b. A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, 43(6):1–47.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. MemoryBank: Enhancing large language models with long-term memory. AAAI. ArXiv:2305.10250.

A Installation and Setup

HINDSIGHT can be installed in two ways: as a **Docker container** (recommended for production) or as a **Python package** (for development and experimentation).

Docker. The Docker image bundles the API server, PostgreSQL with pgvector, and a web UI:

```
export OPENAI_API_KEY=sk-xxx
docker run --rm -it --pull always \
  -p 8888:8888 -p 9999:9999 \
  -e HINDSIGHT_API_LLM_API_KEY=
      $OPENAI_API_KEY \
  ghcr.io/vectorize-io/hindsight:latest
```

The API is available at `localhost:8888` and the UI at `localhost:9999`. The LLM provider can be configured via the `HINDSIGHT_API_LLM_PROVIDER` environment variable, with support for OpenAI, Anthropic, Gemini, Groq, MiniMax, Ollama, LM Studio, and any LiteLLM-compatible endpoint.

Python embedded. For development or notebook use, the `hindsight-all` package bundles an embedded PostgreSQL instance so no external database is needed:

```
pip install hindsight-all
```

Client SDKs. Client libraries are available for Python (`pip install hindsight-client`), Node.js/TypeScript (`npm install @vectorize-io/hindsight-client`), Go, and Rust. A CLI tool (`hindsight-cli`) is also provided. For local development, the embedded mode uses `pg0`, a single-binary PostgreSQL distribution with pgvector pre-installed, so no external database is required.

Supported LLM providers. Table 3 lists the LLM backends that HINDSIGHT supports out of the box. The provider is selected via the `HINDSIGHT_API_LLM_PROVIDER` environment variable.

B API Usage Examples

The Python client exposes the three core operations directly:

```
from hindsight_client import Hindsight
client = Hindsight(base_url="http://
    localhost:8888")

# Retain: store a conversation
client.retain(
    bank_id="my-bank",
    content="Alice works at Google as an
        ML engineer")

# Recall: retrieve relevant memories
results = client.recall(
    bank_id="my-bank",
    query="What does Alice do?")

# Reflect: generate a grounded response
response = client.reflect(
    bank_id="my-bank",
    query="Tell me about Alice")
```

Provider	Example models
OpenAI	GPT-4o, GPT-5-mini
Anthropic	Claude Sonnet, Claude Haiku
Google	Gemini 2.0 Flash, Gemini Pro
Groq	Llama 3, Mixtral
MiniMax	MiniMax-Text, abab series
Ollama	Any local GGUF model
LM Studio	Any local model
LiteLLM	100+ providers via routing

Table 3: Supported LLM providers. All providers use the same pipeline; switching requires only changing an environment variable.

Banks are configured with **disposition** traits that tune how opinionated or cautious the agent is:

```
# Create a bank with a behavioral
    profile
client.create_bank(
    bank_id="analyst",
    background="A data analyst assistant
        ",
    skepticism=4, literalism=3, empathy
        =2)  # 1-5
```

The embedded Python mode requires no server setup:

```
import os
from hindsight import HindsightServer,
    HindsightClient

with HindsightServer(
    llm_provider="openai",
    llm_model="gpt-4o-mini",
    llm_api_key=os.environ["
        OPENAI_API_KEY"]
) as server:
    client = HindsightClient(base_url=
        server.url)
    client.retain(bank_id="demo",
        content="Bob prefers hiking over
            cycling")
    results = client.recall(bank_id="
        demo",
        query="What are Bob's hobbies?")
```

C Structured Output Schemas

HINDSIGHT enforces structured output from all LLM calls using Pydantic models, so parsing is

reliable regardless of the backbone model. The key schemas follow.

Fact extraction. Each extracted fact carries five dimensions (what, when, where, who, why), a network classification, temporal metadata, entity annotations, and optional causal links:

```
class ExtractedFact(BaseModel):
    what: str        # Core fact - concise but
        complete
    when: str        # When it happened. 'N/A' if
        unknown
    where: str       # Location if relevant. 'N/A' if
         none
    who: str         # People involved. 'N/A' if
        general
    why: str         # Context/significance if
        important
    fact_type: Literal["world", "assistant"]
    fact_kind: str = "conversation"  # 'event' or '
        conversation'
    occurred_start: Optional[str] = None
    occurred_end: Optional[str] = None
    entities: Optional[List[Entity]] = None
    causal_relations: Optional[List[
        FactCausalRelation]] = None
```

The internal `fact_type` is `world` or `assistant` (first-person actions and experiences by the speaker). The public REST and MCP APIs expose three external types, `world`, `experience`, and `observation`, where `observation` corresponds to consolidated entity summaries described next.

Observations and trends. Observations are evidence-grounded beliefs consolidated from multiple facts. Each observation carries a list of evidence items with exact quotes, and a trend that is computed from evidence timestamps rather than supplied by the LLM:

```
class ObservationEvidence(BaseModel):
    memory_id: str
    quote: str       # Exact quote from the source
        memory
    relevance: str = ""
    timestamp: datetime

class Observation(BaseModel):
    title: str       # Short summary (5-10 words)
    content: str     # What we believe to be true
    evidence: List[ObservationEvidence] = []
    created_at: datetime
    # Computed: trend in {new, strengthening, stable
        ,
    #                     weakening, stale}
    # Computed: evidence_span, evidence_count
```

Directives. Directives are user-defined hard rules that are injected into prompts at reflect time, with a priority order and an active flag:

```
class Directive(BaseModel):
    bank_id: str
    content: str     # Rule injected into prompts
    priority: int = 0 # Higher priority injected
        first
    is_active: bool = True
```

These schemas are validated at the LLM boundary so downstream code always operates on typed data.

D Per-Category Benchmark Results

Table 4 breaks down LongMemEval accuracy by question category, complementing the per-category numbers used in §6.3.

Category	Full-ctx (GPT-4o)	Zep (GPT-4o)	Ours (20B)	Ours (120B)	Ours (Gem3)
Single-sess. (user)	81.4	92.9	95.7	**100.0**	97.1
Single-sess. (asst)	94.6	80.4	94.6	98.2	96.4
Preference	20.0	56.7	66.7	**86.7**	80.0
Knowledge update	78.2	83.3	84.6	92.3	**94.9**
Temporal reasoning	45.1	62.4	79.7	85.7	**91.0**
Multi-session	44.3	57.9	79.7	81.2	**87.2**
Overall	60.2	71.2	83.6	89.0	**91.4**

Table 4: Per-category accuracy (%) on LongMemEval (S setting). HINDSIGHT with a 20B open-source backbone outperforms full-context GPT-4o on every category. Best result per row in bold.

E Framework Integration

HINDSIGHT provides drop-in wrappers for existing LLM clients, so developers can add memory to an agent without changing their application code. The `hindsight-litellm` package wraps OpenAI, Anthropic, and 100+ other providers via LiteLLM:

```
from openai import OpenAI
from hindsight_litellm import wrap_openai

# Wrap existing client -- memories are now automatic
client = wrap_openai(
    OpenAI(),
    bank_id="my-agent",
    hindsight_api_url="http://localhost:8888")

# Use exactly as before; memory stored and recalled
response = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[{"role": "user",
               "content": "What did we discuss last
                   week?"}])
```

With this wrapper, conversations are automatically stored via retain after each response, and relevant memories are injected into the prompt via recall before each LLM call. No changes to the rest of the application are needed. HINDSIGHT also exposes an **MCP server** (Model Context Protocol), allowing AI coding assistants and other MCP-compatible tools to access agent memory directly. Full integration documentation is at https://hindsight.vectorize.io.