PolyMemDB: A Polyglot Database System for AI Memory Management

Yu Wang
University of Helsinki
yu.wang@helsinki.fi

Jiaheng Lu
University of Helsinki
jiaheng.lu@helsinki.fi

***Abstract*—With the widespread adoption of personal intelligent agents, users generate massive, heterogeneous data during long-term interactions. Leveraging this data as long-term memory helps reduce token overhead and deliver personalized experiences. However, existing memory systems face two primary limitations: they rely on single-storage paradigms that fragment multi-dimensional data, and they lack fine-grained data provenance to resolve long-term factual conflicts, thereby worsening large language model (LLM) hallucinations.**

In this demonstration, we introduce PolyMemDB, a novel system tailored for managing agent memory. PolyMemDB has a polyglot storage architecture designed to track and manage various memory types, including graph, vector, probability and spatial-temporal data. To ensure factual consistency and reduce hallucinations, it features a probabilistic inference engine that integrates temporal decay with semiring aggregation, resolving long-term factual conflicts, providing detailed data provenance, and enabling users to trace reasoning chains transparently.

I. INTRODUCTION

A. Motivation and Contributions

The rapid advancement of agent technology based on LLMs has enabled the creation of personal assistants capable of managing long-term interactive tasks. In this context, the memory mechanism has emerged as a key factor influencing both system performance and user experience [1]. By structuring and persistently storing the historical context generated during an agent's operation, it is possible not only to reduce inference latency and token costs through the reuse of semantically similar responses but also to accurately capture users' personalized preferences and enable the agent to abstract higher-order generalized knowledge from them [1].

However, in real-world scenarios, the management and maintenance of long-term memory face two core challenges: 1) Bottlenecks in the unified storage and multi-dimensional association of heterogeneous data: Long-term interactions generate massive amounts of heterogeneous context with diverse attributes (such as complex entity networks, logical relationships, and spatio-temporal constraints). Existing systems generally rely on a single storage paradigm, making it difficult to support efficient storage and retrieval of multimodal heterogeneous information; 2) Lack of data provenance tracking when dealing with long-term factual conflicts: During long-term memory evolution, systems typically employ coarse-grained state overwriting and lack factual maintenance mechanisms that integrate temporal sequences and probability, resulting in an urgent need to improve data provenance capabilities and the ability to address LLM hallucinations.

To tackle these challenges, we developed and implemented **PolyMemDB**, which combines multi-modal storage, graph databases, and spatio-temporal indexing technologies. It also features graph updates that integrate data provenance with temporal decay, facilitating conflict resolution for multi-source facts. In this demonstration, our core contributions are summarized as follows:

- **Polyglot storage layer with multiple databases:** We integrated various heterogeneous storage components, including multi-modal object storage, graph databases, spatio-temporal databases, and probabilistic databases. This architecture addresses the limitations of traditional text-only and vector-based retrieval.
- **Probabilistic memory graph maintenance:** To address the challenges of event integration and factual conflicts in long-term interactions, we introduce a dynamic graph update mechanism that combines temporal decay and data provenance. By extending semiring inference in probabilistic databases to the evidence chain, the system can resolve conflicts and align entities, thereby mitigating LLM hallucinations while preserving interpretability.

B. Comparison to Existing Systems

Recent research has explored various aspects of structured representations and retrieval mechanisms for agent memory. For example, at the level of data organization, MemForest [2] approaches memory as a problem focused on efficient time-based writes, creating a tree structure that organizes time-related information and allows updates to occur only along specific paths within the tree. This method avoids the need to rewrite the entire global memory state. HyperMem [3] improves traditional graph databases by introducing a three-tier hypergraph model of topics, fragments, and facts, using hyperedges to capture complex dependencies among discrete memory fragments. Unlike MemForest, which focuses on text-time indexing, and HyperMem, which focuses on homogeneous hypergraph structures, PolyMemDB addresses the scheduling problem of heterogeneous multi-dimensional data. PolyMemDB abandons the single-storage paradigm and introduces a polyglot storage architecture at the lower level. It not only utilizes a graph database (Neo4j) to maintain entity

Source code: https://github.com/wangyu-1999/PolyMemDB

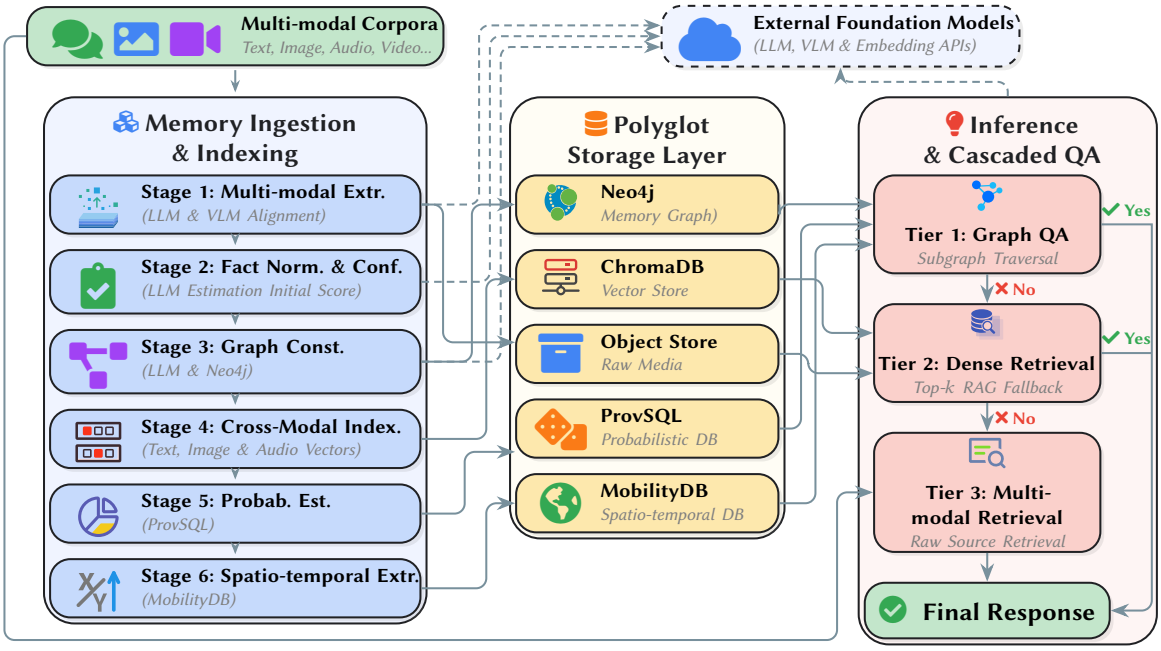


Fig. 1. PolyMemDB architecture: multi-modal ingestion, polyglot storage, and cascaded QA routing.

topology but also pushes complex spatio-temporal constraints directly down to a spatio-temporal database (MobilityDB) for joint filtering of dynamic time windows and geographic polygons, overcoming the limitation of existing systems that can only process pure text relationships.

At the level of conflict resolution mechanisms, MRAgent [4] constructs a "Cue-Tag-Content" graph and relies on an active reconstruction mechanism, enabling the LLM to iteratively explore and prune paths based on intermediate states during the retrieval phase. PolyMemDB extends the semiring inference in probabilistic databases. For memory triplets, PolyMemDB appends a historical observation sequence to the edge attributes and, through probabilistic inference and temporal decay, computes distribution vectors for positive drive, negative inhibition, and evidence conflict at the database level. This enables PolyMemDB to provide fine-grained data provenance while resolving cognitive conflicts through underlying mathematical inference, thereby enhancing the system's reliability.

II. System Overview

As shown in Fig. 1, the workflow of PolyMemDB comprises three core stages:

First, in the **Memory Ingestion & Indexing**, the system decomposes the input data into atomic facts. During this process, PolyMemDB leverages LLMs and vision language models (VLMs) to perform entity alignment, execute coreference resolution and ellipsis recovery, and complete named entity extraction, fact normalization, and confidence estimation.

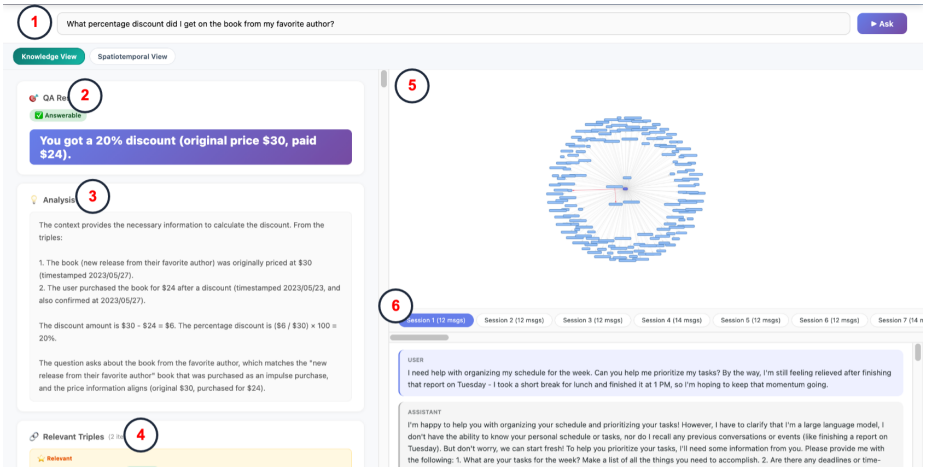
Subsequently, it enters the **Polyglot Storage Layer**. To fully capture the complex associations in the data, PolyMemDB performs targeted extraction and routing of storage for normalized facts based on their multi-dimensional features. Specifically, the system constructs entities, attributes, and their evolutionary trajectories across long periods into a connected relationship graph (relying on Neo4j), providing structural support for subsequent complex chain inference; it stores event confidence in a probabilistic database (relying on ProvSQL); and parses complex spatio-temporal constraints for management by a spatio-temporal database (relying on MobilityDB). Meanwhile, embedding vectors and raw context content are persisted into a vector database (relying on ChromaDB) and object storage, respectively.

Finally, when responding to user interactions, the system triggers the **Inference & Cascaded QA**. To balance inference overhead and recall accuracy, PolyMemDB designs a top-down, three-tier federated retrieval mechanism. The system first executes *Tier 1 (Graph QA)*, prioritizing low-latency subgraph traversal and jointly utilizing the probabilistic database and spatio-temporal index for rule filtering and logical inference; if this tier yields no answer, it triggers a fallback mechanism to enter *Tier 2 (Dense Retrieval)*, utilizing the query's embedding vectors to perform Top-k semantic retrieval of structured facts within the vector database; if sufficient evidence is still not found, the system ultimately drills down to *Tier 3 (multi-modal Retrieval)* to directly retrieve raw data snippets, thereby generating the final response.

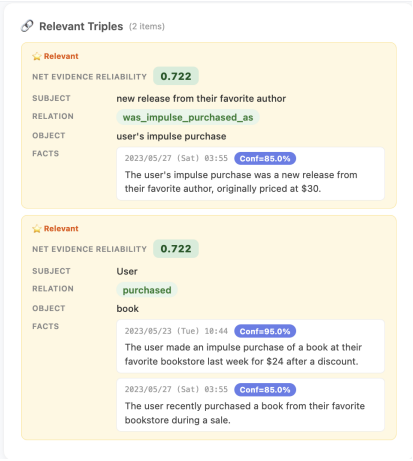
III. Probabilistic Memory Graph Maintenance

Event consolidation and conflict resolution are core challenges in long-term memory management for agents. To this end, PolyMemDB designs a dynamic graph update mechanism that combines temporal decay and probabilistic inference.

Memory Graph Formalization and Dynamic Update: PolyMemDB formalizes the agent's memory as a memory



(a) Main dashboard.



(b) Evidence panel.

Fig. 2. PolyMemDB UI: (a) Main dashboard for query and graph visualization; (b) Evidence panel showing probabilistic facts.

graph $\mathcal{G} = (V, E, \alpha)$. Here, V is the set of heterogeneous entities (e.g., PERSON, CONCEPT, etc.); $E \subseteq V \times \mathcal{L} \times V$ is the set of directed relationship edges; and α is the attribute labeling function. We define a single fact as a tuple $f_i = \langle \rho_i, c_i, t_i \rangle$. Here, $\rho_i \in \{+, -\}$ denotes the sentiment polarity of the evidence, $c_i \in [0, 1]$ is the initial confidence assigned by the LLM, and t_i is the timestamp of the fact observation. Before inserting a fact into the memory graph, the system retrieves the Top-k local subgraphs via semantic similarity retrieval as context, which the agent uses to autonomously decide whether to create or update nodes or edge relationships, thereby maximizing the accuracy of entity alignment.

Data Provenance and Evidence Chain Construction: To effectively mitigate LLM hallucinations and preserve the interpretability of raw materials, PolyMemDB abandons the simple state overwriting mechanism. For any triplet $e = (u, l, v) \in E$ in the graph, the system appends and maintains a historical observation sequence $F = (f_1, f_2, \ldots, f_T)$ (satisfying $t_1 \leq t_2 \leq \cdots \leq t_T$) in its edge attributes. This mechanism fully preserves the evidence observed at different time points.

Probabilistic Inference Based on Temporal Decay: When calculating the confidence of a specific triplet, we extend the semiring framework found in probabilistic databases (e.g., ProvSQL [5]). Since the influence of real-world events decays over time, we introduce an exponential decay factor $\lambda \in (0, 1]$. For a fact f_i in the sequence with a temporal distance index of $\Delta_i = T - i$, its dynamic weight is calculated as $w_i = c_i \cdot \lambda^{T-i}$. Based on the inclusion-exclusion principle, we partition the evidence sequence F into a positive set F^+ and a negative set F^-, and calculate their cumulative joint confidences, respectively, as defined in (1):

$$P^+ = 1 - \prod_{f_i \in F^+} (1 - w_i), \quad P^- = 1 - \prod_{f_j \in F^-} (1 - w_j) \quad (1)$$

Based on P^+ and P^- derived from (1), PolyMemDB computes four intermediate cognitive states to capture the interactions between conflicting evidence: positive drive $S_{action} = P^+(1 - P^-)$, negative inhibition $S_{inaction} = P^-(1 - P^+)$, evidence conflict $S_{conflict} = P^+P^-$, and absence of evidence $S_{ignorance} = (1 - P^+)(1 - P^-)$.

To comprehensively evaluate the factual reliability, PolyMemDB aggregates these intermediate states into a *Net Evidence Reliability* score, denoted as $\mathcal{R} \in [-1, 1]$:

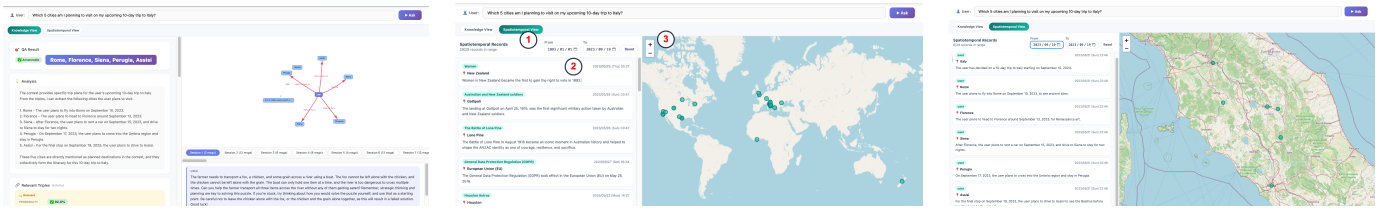
$$\mathcal{R} = (S_{action} - S_{inaction}) \cdot (1 - S_{conflict} - S_{ignorance}) \quad (2)$$

Here, $\mathcal{R}$ compactly integrates all evidence dimensions. The base term $(S_{action} - S_{inaction})$ establishes the directional truthfulness, yielding a positive score for supported facts and a negative score for refuted ones. The multiplier $(1 - S_{conflict} - S_{ignorance})$ serves as a dynamic penalty, reducing the net reliability $\mathcal{R}$ when there are significant factual disputes or insufficient evidence. When $\mathcal{R} \approx 0$, identifying the maximum intermediate state ($\arg\max$) clearly distinguishes whether the low score stems from contradictory facts ($S_{conflict}$) or an evidence void ($S_{ignorance}$). In the subsequent QA stage, providing $\mathcal{R}$ and these decomposed distributions as quantitative references empowers the LLM to avoid binary hallucinations and generate nuanced, provenance-grounded answers.

IV. DEMONSTRATION

PolyMemDB's backend relies on FastAPI and Pydantic-AI, while its frontend provides an interactive visual dashboard. At the conference demonstration, attendees will experience how PolyMemDB extracts and infers information from conversations spanning long periods.

Scenario 1: Long-Session QA with Fine-Grained Provenance. As shown in Fig. 2a, we introduce a long-session case (ID: d905b33f) from the LongMemEval [6] benchmark. This conversation spans 48 sessions; following ingestion by PolyMemDB, a memory graph containing 229 entities and 221 relationships was successfully constructed. Through the interactive analysis interface, attendees can track the system's complete process of answering a question: First is the ①



(a) Knowledge view. (b) Global map. (c) Filtered map.

Fig. 3. spatio-temporal UI: (a) Itinerary extraction; (b) Global historical map; (c) Map filtering by dynamic time window.

Query Input (e.g., entering a question involving long-range dependencies: *"What percentage discount did I get on the book by my favorite author?"*), followed by the completion of ② Answer Generation (the retrieval outputs *20%*). Subsequently, through ③ Reasoning Chain Parsing, the interface shows attendees how the LLM arrived at the final answer. To demonstrate that the answer is not an LLM hallucination, the system lists all relevant triplets supporting this answer, the observational facts attached to the edges, and the positive drive probability calculated based on temporal decay within the ④ Evidence panel. Finally, the interface ⑤ shows the core relational subgraph of the matched entities to reduce visual clutter, while ⑥ presents the raw corpus.

Scenario 2: Spatio-Temporal Memory Evolution. To demonstrate PolyMemDB's capabilities in handling complex spatio-temporal constraints and long-term context summarization, we designed a second demonstration scenario (based on the extended LongMemEval [6] case ID: e47becba). When the user asks the system to summarize the "graduation trip" itinerary from a lengthy historical dialogue, PolyMemDB not only automatically extracts and generates a structured itinerary response (as shown in Fig. 3a), but also provides an interactive analysis dashboard to visualize the spatio-temporal evolution of heterogeneous memories intuitively: First, through the Global Spatio-Temporal Map, the system panel lists all geographical locations and associated facts mentioned in the long conversation (Fig. 3b); subsequently, attendees can operate the interface's ① Dynamic Time Selector to set a specific time window. When limited to the "graduation trip planning period," the system automatically filters out irrelevant locations (Noise Reduction), precisely highlights the 5 Italian cities the user plans to visit on the ③ Geospatial Map (Fig. 3c), and synchronously renders a ② Fine-grained Fact List in the panel to trace the underlying evidence sources of these spatio-temporal memories.

Scenario 3: Dynamic Probabilistic Inference for Factual Conflicts. We asks the question "Did Alice enjoy running?" across 10 months of history (simplified in Fig. 4). Using temporal decay ($\lambda = 0.8$) and semiring aggregation. The engine computes a 4D cognitive distribution. Although Alice's recent marathon completion drives a strong positive state ($S_{action} = 47.2\%$), historical setbacks like knee pain and heat exhaustion trigger a dominant conflict state ($S_{conflict} = 52.3\%$), penalizing the overall net reliability ($\mathcal{R} \approx 0.22$).

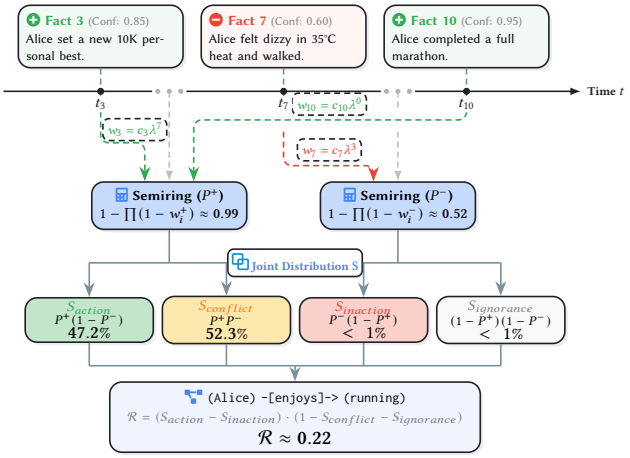


Fig. 4. Probabilistic inference pipeline. Decayed historical facts are aggregated into a 4D distribution and synthesized into a net reliability score to resolve factual conflicts.

Empowered by these explicit metrics and the evidence chain, the LLM avoids binary hallucinations. Instead of a naive "yes," it generates a strictly provenance-grounded response: concluding that while Alice is highly committed to running, her relationship with the sport is complex and tempered by specific physical and motivational hurdles. This demonstrates PolyMemDB's capability to guide agents through contradictory memories without needing to brute-force overwrite their states.

REFERENCES

[1] Z. Zhang, Q. Dai, X. Bo, C. Ma, R. Li, X. Chen, J. Zhu, Z. Dong, and J.-R. Wen, "A Survey on the Memory Mechanism of Large Language Model-based Agents," *ACM Transactions on Information Systems*, vol. 43, no. 6, pp. 1–47, Nov. 2025.

[2] H. Chen, Z. Zhang, W. Pei, B. He, M. Wu, J. Zeng, M. Heinrich, W. Wu, and H. Zhang, "MemForest: An Efficient Agent Memory System with Hierarchical Temporal Indexing," May 2026.

[3] J. Yue, C. Hu, J. Sheng, Z. Zhou, W. Zhang, T. Liu, L. Guo, and Y. Deng, "HyperMem: Hypergraph memory for long-term conversations," 2026.

[4] S. Ji, Y. Li, and B. Hooi, "Memory is reconstructed, not retrieved: Graph memory for LLM agents," 2026.

[5] P. Senellart, L. Jachiet, S. Maniu, and Y. Ramusat, "ProvSQL: Provenance and probability management in PostgreSQL," *Proceedings of the VLDB Endowment (PVLDB)*, vol. 11, no. 12, pp. 2034–2037, 2018.

[6] D. Wu, H. Wang, W. Yu, Y. Zhang, K.-W. Chang, and D. Yu, "LongMemEval: Benchmarking chat assistants on long-term interactive memory," in *The Thirteenth International Conference on Learning Representations*, Oct. 2024.