MemLens: A Value-Aware Memory Management System with Interactive Analytics for LLM-based Agents

Shuyue Wei
C-FAIR & School of Software,
Shandong University
weishuyue@sdu.edu.cn

Chang Liu
SKLCCSE Lab,
Beihang University
bjliuchang@buaa.edu.cn

Zimu Zhou
Department of Data Science,
City University of Hong Kong
zimuzhou@cityu.edu.hk

Yongxin Tong
SKLCCSE Lab,
Beihang University
yxtong@buaa.edu.cn

Lizhen Cui
C-FAIR & School of Software,
Shandong University
clz@sdu.edu.cn

ABSTRACT

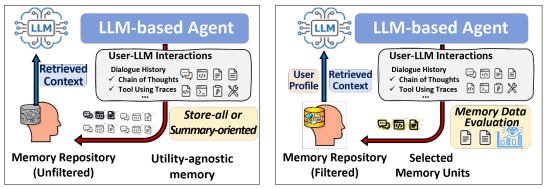
Recently, memory management has become a key infrastructure for LLM-based agents, as it directly affects long-horizon reasoning, personalized responses, and knowledge reuse. However, existing LLM memory systems typically adopt a coarse-grained (utility-agnostic or heuristic utility) manner that treats heterogeneous user-LLM interaction records uniformly, leading to redundant and low-impact records persisting in the memory repository. To address this challenge, we present MemLens, a value-aware memory management system that takes memory records as first-class data objects. MemLens provides an end-to-end interactive analytics dashboard that exposes the complete memory lifecycle, including Shapley-style memory evaluation, value-aware storage, and memory-assisted response. Through a study-copilot application, the system enables users to inspect memory values, visualize hierarchical memory structures, and compare various memory management strategies in terms of response quality, retrieval latency, and token consumption. Therefore, our MemLens can serve as an efficient, interpretable, and personalized long-term memory management system for agents.

PVLDB Reference Format:
Shuyue Wei, Chang Liu, Zimu Zhou, Yongxin Tong, Lizhen Cui. MemLens: A Value-Aware Memory Management System
with Interactive Analytics for LLM-based Agents. PVLDB, 19(12): XXX-XXX, 2026.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/LIUHA1ZHU/MemLens.

1 INTRODUCTION

Memory data management has emerged as a key infrastructure layer for LLM-based agents, enabling long-horizon task execution, personalized interactions, and cross-session knowledge reuse. From a data management perspective, agent memory is inherently *heterogeneous*. Beyond dialogue histories, reusable memory may also arise from intermediate reasoning traces, retrieved documents, and tool execution outputs. As agents operate over longer sessions, records accumulate into a growing repository of reusable context, making effective memory management increasingly important.



(a) Utility-agnostic memory management (b) Utility-aware memory management

Figure 1: Comparison between (a) *utility-agnostic* memory management and (b) *value-aware* memory management.

However, existing LLM memory systems [1, 3, 4, 10] typically manage memory in a coarse-grained manner. They either *retain nearly all* candidate records or compress them using *summary-oriented* heuristics, without explicitly modeling which memory units are truly useful for future tasks. As memory accumulates, the utility-agnostic or heuristic scoring strategy allows redundant and low-impact records to persist in the repository, which can dilute task-relevant context, enlarge the retrieval space, and consume unnecessary token overhead during response generation (see Fig. 1).

The core challenge is that memory usefulness is *neither uniform nor static*. A memory unit that is beneficial for one future query may be irrelevant for another, and its contribution may also depend on which other memory units are already available. Thus, effective memory management requires more than generic summarization. It requires a principled way to *estimate the downstream utility* of heterogeneous memory units, while keeping such estimation *efficient* enough to support practical storage and retrieval decisions.

To address this challenge, we present MemLens, a value-aware memory management system with interactive analytics for LLM-based agents. MemLens manages memory based on its *value*, *i.e.*, contribution of a memory unit to downstream response quality. Specifically, it decomposes archived interactions into fine-grained memory units, estimates their utility for downstream reasoning

Proceedings of the VLDB Endowment, Vol. 19, No. 12 ISSN XXXX-XXXX.
doi:XX.XX/XXX.XX

using a lightweight proxy model together with LLM-based scoring, selectively retains high-value memory, and injects concise yet effective memory into LLM context at response time. As the usefulness of a memory unit may depend on which other memory units are already available, MemLens models memory value through Shapley-style marginal contribution analysis, and makes this valuation practical via an efficient sampling-based approximation. These designs yield a unified memory lifecycle spanning *memory evaluation*, *value-aware storage*, and *memory-assisted response*.

We demonstrate MemLens through a study-copilot application, where users can interactively explore the full memory lifecycle. Specifically, users can: *(i)* inspect how archived user-LLM interactions are decomposed into memory units and assigned value-aware scores; *(ii)* visualize how the memory evolves as retention is guided by value thresholds and hierarchical organization; and *(iii)* compare various memory management strategies, including *store-all*, agent-based summarization, and proposed *value-aware* strategy, in terms of response quality, retrieval latency, and token consumption.

Contributions. The main contributions are summarized as follows: *(i)* We introduce a value-aware memory management paradigm for LLM-based agents that explicitly models memory value as downstream utility, enabling selective retention of high-impact memory units. *(ii)* We design an end-to-end memory lifecycle framework that integrates memory valuation, value-aware storage, and memory-assisted response, providing a unified and interpretable mechanism for long-term memory management in LLM-based agents. *(iii)* We develop an interactive demonstration system that visualizes memory value, storage evolution, and response behavior, allowing users to directly observe how different memory strategies affect response quality, retrieval latency, and token consumption.

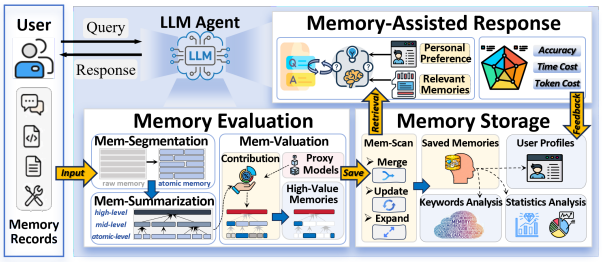


Figure 2: System Overview of MemLens

2 MEMLENS OVERVIEW

Overview. MemLens is a value-aware memory management system that shifts how LLM-based agents store and utilize long-term memory. As illustrated in Fig. 2, it integrates three core components, including memory data evaluation, value-aware memory storage, and memory-assisted response. The system treats heterogeneous records originating from the LLM-based agent's reasoning process (such as dialogue histories, retrieved knowledge, and chain-of-thought traces) as candidate memory records. Instead of indiscriminately storing all records, MemLens explicitly quantifies their utility and selectively retains only the high-value ones. During query response, the system retrieves and injects the concise but effective memories to support complex reasoning, enabling more accurate responses with significantly lower token overhead.

Furthermore, MemLens also features an interactive analytics dashboard that makes value-aware memory management transparent and visible. Users can inspect memory values on MemLens , where each record is assigned a utility score and aggregated into global views such as value distributions and retention ratios, enabling identification of the critical versus redundant memories. The system also presents an interpretable storage view that highlights salient keywords and evolving user preference profiles, revealing how long-term memory structures are continuously shaped. In addition, MemLens enables side-by-side comparison of different memory management strategies (*e.g.*, store-all, agent-based summarization, and value-aware storage), reporting response quality, retrieval latency, and token consumption on testing datasets, which allows users to directly observe how value-aware memory reduces unnecessary context while enhancing the response quality.

Workflow. MemLens implements a value-aware memory lifecycle that enables LLM agents to retain and utilize the long-term memory. Once the user archives a session, the system can decompose raw records into fine-grained memory units and organize them into a hierarchical structure with multiple levels of abstraction. Then, each memory unit is assigned a value using a widely-adopted Shapley-value-based framework [6, 7] that captures its contribution to downstream reasoning performance. Guided by these values, MemLens selectively preserves high-value memories and continuously updates a user preference profile. Finally, the system responds to the user query by retrieving relevant memory units and using the user profile, then injecting them into the LLM context to enhance response quality while avoiding unnecessary context expansion. The above closed-loop lifecycle allows MemLens to maintain compact yet high-impact and personalized memory records adaptively.

3 MEMLENS SYSTEM DESIGN

3.1 Memory Data Evaluation

MemLens treats memory records as first-class data objects and quantifies their value via a principled Shapley-value-based approach in two steps: *memory unit construction* and *value estimation*.

Memory Unit Construction. MemLens converts raw user-LLM interaction records into structured memory units. Firstly, for fine-grained analysis, the raw record is segmented into sentence-level atomic units. The system then groups the memory units into multi-level summaries via LLMs, providing a hierarchical memory structure that captures both detailed semantics and high-level knowledge. We maintain provenance links from summarized nodes to the atomic memory units as well to avoid the information loss. This design allows the MemLens system to replace multiple low-utility fragments into a single high-value alternative, which potentially improves both storage efficiency and memory quality.

Memory Value Estimation. MemLens assesses the utility of each memory unit through an LLM-as-Judge utility score, which is a different judge model from the one employed during memory valuatio and measures unit contribution to the downstream reasoning task. Intuitively, a memory unit is valuable if it can significantly improve the output quality of a lightweight proxy model $\mathcal{M}_p$ with

limited reasoning capability. By default, MemLens uses Qwen2.5-7B as the proxy model in EduMemBench, while the dashboard allows users to switch among different lightweight models. Specifically, given the original user query q and its related memory units subset $S \subseteq \mathsf{M} = \{m_1, m_2, \cdots, m_M\}$, we can formalize the utility function $\mathcal{V}(\cdot)$ of a unit set S as follows,

$$\mathcal{V}_q(S) = \textbf{Score}\big(\mathcal{M}_p(q \oplus S)\big) - \textbf{Score}\big(\mathcal{M}_p(q)\big), \tag{1}$$

where $\mathcal{M}_p$ is the proxy model, $\oplus$ denotes the prompts injection operation, and $\textbf{Score}(\cdot)$ can be implemented via LLM-as-Judge approaches. As Eq. (1) enables MemLens to qualify the utility of a set of memory units S, we can define the value of a memory unit $m_i \in \mathsf{M}$ by extending the widely-adopted Shapley value as follows.

Definition 1 (Memory Shapley Value). *Given a user query task q and its related memory units $M = \{m_1, \ldots, m_M\}$ from the user-LLM interactions, the memory value of m_i can be formally defined as,*

$$\phi(m_i) = \sum_{S \subseteq M \setminus \{m_i\}} \frac{|S|!(M - |S| - 1)!}{M!} \left[\mathcal{V}_q(S \cup \{m_i\}) - \mathcal{V}_q(S)\right], \tag{2}$$

where M is the total number of memory units. The Memory Shapley Value (MS-value) extends the Shapley value-based data valuation concepts and inherits some required fairness properties, such as null player (or no-free-riders), symmetric fairness, and linear-additivity.

To support in-time interactive memory data analytics and feedback, we further extend a stratified sampling strategy [8] to approximate memory values efficiently, where subsets S are grouped by size, and samples are drawn proportionally to reduce approximation errors. Then, the evaluation time complexity can be reduced to $O(\rho M)$, where ρ is the sampling budget and ρ is usually set to $20 \sim 100$ for efficient interactive latency. In summary, this module enables us to save high-utility memories selectively and provides transparent and fine-grained insight into the value of memory units.

3.2 Value-Aware Memory Storage

Building upon the computed Memory Shapley values, this module implements a value-aware memory storage that handles memory retention, organization, and evolution under explicit utility constraints, providing a compact yet high-utility memory warehouse that balances storage cost and reasoning performance.

Value-Aware Storage Strategy. For memory units M and their associated with values $\{\phi(m_i)\}$, MemLens can make selectively storage by a value threshold τ, $\mathsf{M}^* = \{m_i \in \mathsf{M} \mid \phi(m_i) \geq \tau\}$, allowing users to flexibly adjust τ and explore a trade-off between storage cost and downstream reasoning performance. We also achieve interactive threshold tuning to provide immediate feedback on memory size, compression rate, and expected utility in MemLens system.

Memory Organization and Consolidation. In MemLens , the saved memory units are organized into a hierarchical tree derived from previous multi-level abstractions. As sentence-level atomic memory units have been grouped into higher-level semantic nodes, they can form a memory tree in which leaf nodes are atomic memory units and internal nodes represent their high-level abstractions. To reduce memory redundancy, MemLens will incrementally add a new memory unit to the existing memory tree. Specifically, the system applies an LLM-aided consolidation operator to decide how to handle m': *(i)* merge m' into an existing node, *(ii)* update an existing node in memory tree, or *(iii)* insert m' as a new node. The operator first computes semantic similarity between the incoming memory unit and existing nodes. Highly similar nodes are merged, partially overlapping nodes are updated, and unrelated units are inserted as new nodes. The memory tree is also visualized in MemLens , allowing users to inspect memory organization and semantic relationships directly.

Interactive Memory Analytics. Our demonstration system also provides an interactive analytics interface to improve transparency and interpretability of the memory lifecycle. Specifically, the user profiles are constructed as a structured summary of retained high-value memory units and injected as system prompts, which provides a *word-cloud-based exploration* of saved memory content, and presents *key memory statistics*, including value distributions, memory size, and compression rates. These allow users to observe how a value-aware storage strategy affects the saved LLM memory units.

3.3 Memory-Assisted Response

This module uses value-aware memories and user profiles to generate personalized responses, then allows users to archive completed sessions for continuous memory evolution and management.

Response Generation. MemLens first uses the user preference profile, constructed from historical high-value memories, as the *system prompts* to maintain consistent personalization across sessions. Then, the system retrieves relevant memories from the integrated vector database and re-ranks the retrieved units based on their MS-value $\phi(m)$, prioritizing units that are both semantically relevant and empirically high-utility. These selected memory units are injected into the LLM prompts to enhance response quality.

Memory Archiving. Finally, the user can archive the interaction records and feed them into the value-aware memory pipeline for memory unit construction, value estimation, and selective storage, which establishes a closed-loop LLM memory lifecycle that continuously adapts to evolving user behavior and query requirements.

4 DEMONSTRATION SCENARIOS

We demonstrate MemLens through an interactive study-copilot scenario, where it assists students in understanding key concepts, creating profiles, and maintaining memory across sessions.

Simulation Environment. To emulate realistic user behavior, we construct a synthetic benchmark dataset, EduMemBench, containing over 2,000 multi-turn dialogue sessions generated by an LLM-based student agent, covering core database topics such as transactions, indexing, and concurrency control. Based on these interactions, we further derive a testing set of 500 query–answer pairs with reference answers, enabling systematic evaluation of memory-assisted response quality, retrieval latency and token consumption overhead under different memory management strategies.

Interactive Demonstration Workflow. The MemLens provides three coordinated interfaces corresponding to the LLM memory lifecycle: *(i) memory evaluation*, *(ii) value-aware memory storage*, and *(iii) memory-assisted response.* Through MemLens users can interactively explore how LLM memory units are valued, organized, and utilized, which creates a complete end-to-end experience.

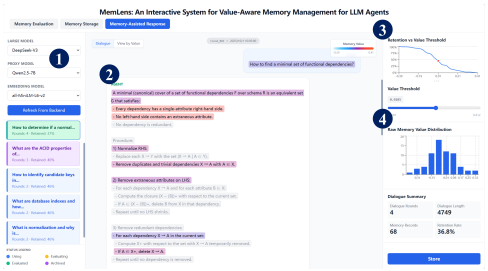


Figure 3: Memory Evaluation Interface

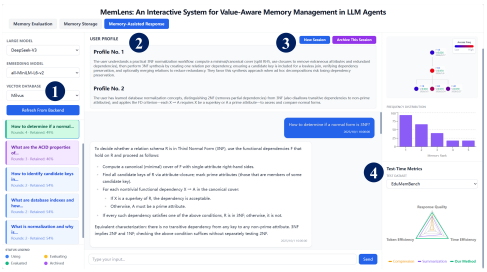


Figure 5: Memory-Assisted Response Interface

(i) Memory Evaluation Interface (in Fig. 3). Users can first explore how memory values are assigned. ① The system supports configurable evaluation settings, allowing users to select different LLMs as scoring models and lightweight proxy models for utility estimation. ② Each archived session is decomposed into fine-grained memory units, whose values are visualized via color-coded highlights, enabling immediate identification of high- and low-value memory fragments. ③ An interactive threshold controller allows users to dynamically adjust the retention threshold and observe corresponding changes in memory compression. ④ The memory evaluation interface further presents value distributions, revealing the empirical patterns of memory across multiple sessions.

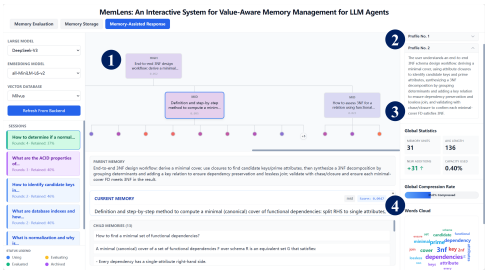


Figure 4: Value-Aware Memory Storage Interface

(ii) Value-Aware Memory Storage Interface (in Fig. 4). Users then inspect how memory is organized and stored. ① The system visualizes the hierarchical memory structure as an interactive tree, where node colors encode memory utility. Users can explore the tree by clicking nodes in the memory tree to examine their contents and semantic relationships. ② This panel continuously updates a user preference profile derived from retained memories, reflecting long-term learning behavior. ③ This panel presents memory statistics (*e.g.*, memory unit size, compression rate, and storage capacity), enabling users to observe how value-aware storage reshapes the memory repository. ④ MemLens also provides a word-cloud view that highlights the key concepts in stored memory based on their frequency, enabling users to quickly grasp the dominant memory patterns during their personalized knowledge learning process.

(iii) Memory-Assisted Response Interface. (in Fig. 5). Finally, users interact with the system in real time to observe the impact of memory on responses, which helps them to recall previous interactions. ①Users can select different vector backends (*e.g.*, FAISS [2]) to set system configurations. ②The system generates responses using both retrieved memory and user profiles, allowing users to observe improvements in answer relevance and consistency directly. ③Users can archive completed sessions, triggering the full memory pipeline, thereby closing the memory lifecycle. ④This panel supports benchmarking across different memory strategies and also presents scores of response quality, retrieval time, and token consumption in a radar plot, enabling users to observe the trade-offs and advantages of various memory strategies. Experiments on EduMemBench shows that the value-aware strategy consistently improves the trade-off among response quality, latency, and token consumption. Though EduMemBench is used as the primary demonstration workload, MemLens is benchmark-agnostic and can be readily extended to public long-term memory benchmarks such as LoCoMo[5] and LongMemEval[9].

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their suggestions. Shuyue Wei's research is supported Shandong Provincial Natural Science Foundation (No.ZR2026QC1215). Zimu Zhou's research is supported by CityU APRC grant (No. 9610633) Yongxin Tong's research is supported by NSFC Grant (Nos. 62425202 and 62336003), the BJNSF Z230001, the Fundamental Research Funds for the Central Universities (No.JKF-2026007844235), and the SKLCCSE Lab. Lizhen Cui's research is supported by NSFC Grant (No.92367202). Yongxin Tong and Lizhen Cui are the corresponding authors.

REFERENCES

[1] Prateek Chhikara, Dev Khant, Saket Aryan, et al. 2025. Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. In *ECAI*, Vol. 413. IOS Press, 2993–3000.
[2] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, et al. 2025. The faiss library. *IEEE Transactions on Big Data* (2025).
[3] Jiajie Fu, Junwen Chen, et al. 2026. VikingMem: A Memory Base Management System for Stateful LLM-based Applications. *arXiv* abs/2605.29640 (2026).
[4] Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, et al. 2025. Memory in the age of ai agents. *arXiv* abs/2512.13564 (2025).
[5] Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, et al. 2024. Evaluating Very Long-Term Conversational Memory of LLM Agents. In *ACL*. Association for Computational Linguistics, 13851–13870.
[6] Yuxiang Wang, Shuyuan Li, et al. 2026. Efficient shapley-based data valuation for federated trajectories. *Frontiers Comput. Sci.* 20, 9 (2026), 2009621.
[7] Shuyue Wei, Yongxin Tong, Zimu Zhou, et al. 2025. Federated reasoning LLMs: a survey. *Frontiers Comput. Sci.* 19, 12 (2025), 1912613.
[8] Shuyue Wei, Yongxin Tong, Zimu Zhou, Tianran He, and Yi Xu. 2025. Efficient Data Valuation Approximation in Federated Learning: A Sampling-Based Approach. In *ICDE*. IEEE, 2922–2934.
[9] Di Wu, Hongwei Wang, Wenhao Yu, et al. 2025. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. In *ICLR*. OpenReview.net.
[10] Zeyu Zhang, Quanyu Dai, et al. 2025. A Survey on the Memory Mechanism of Large Language Model-based Agents. *ACM TOIS* 43, 6 (2025), 155:1–155:47.