VALMEM: VALUE-AWARE MEMORY MANAGEMENT FOR LONG-TERM LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon LLM agents rely on diverse memories to support multi-session interactions, but effective memory management requires more than retrieving semantically similar history. The agent must decide which memories to retain, compress, or discard, so that storage and reasoning resources are not spent on outdated or redundant content. Although retrieval-augmented memory systems have been developed for long-term memory management, existing methods often rely on single-level memory segmentation and retrieval. Reliance on single-level memory segmentation and retrieval makes it difficult to capture complex relationships among memories. As a result, useful evidence may be overlooked, while irrelevant content may be introduced. To overcome these limitations, we propose ValMem, a value-aware memory management framework that combines a Multi-Granularity Hierarchical Memory (MGHM) structure with Value-Driven Training (VDT). MGHM organizes interaction history along temporal and granularity dimensions, providing memory units ranging from raw evidence to compact abstractions. To better identify the latent value of memory, VDT assigns credit to memory operations by optimizing downstream task utility jointly with memory cost. Experiments on three long-horizon memory benchmarks demonstrate that ValMem outperforms current state-of-the-art methods on multi-session question answering and personalized response tasks, with consistent gains across factual recall, multi-hop reasoning, temporal reasoning, and preference inference.

1 INTRODUCTION

Large language models (LLMs) are increasingly operating as agents for multi-turn and cross-session interactions (Yao et al., 2023; Zhao et al., 2024). Memory serves as a crucial foundation for the long-term interaction of such LLM agents in scenarios such as personalized assistants, recommendation systems, and task automation (Chhikara et al., 2025; Shinn et al., 2023). However, long-term interactions can lead to changes in user states and evidence, making previously stored memory incomplete, redundant, or outdated (Fang et al., 2026; Gutiérrez et al., 2024; Lewis et al., 2020). Existing memory systems struggle to address these issues, resulting in difficulties in balancing storage efficiency and task utility in long-term interaction environments. The core difficulty is that memory management is still largely governed by fixed heuristics and semantic similarity, rather than by the downstream utility of individual memory operations. Therefore, this study addresses the problem of value-aware memory management by comprehensively considering the construction, retrieval, and operation of memory. Specifically, the system must construct memory representations that support diverse future queries, retrieve useful evidence, and determine whether each memory item should be retained, compressed, or discarded, with the goal of balancing task utility and storage cost. However, achieving such a system requires overcoming two key challenges.

First, memory construction must support diverse information needs across future interactions. Different queries may require different types of user memory, making fixed memory representations difficult to generalize across future interactions (Schick et al., 2023; Park et al., 2023). Recent agent systems have developed richer memory storage structure by designing finer-grained memory categories, incorporating vector, graph, hybrid, or hierarchical storage structures, and using agentic methods to manage memory items (Park et al., 2023; Packer et al., 2024; Gutiérrez et al., 2024; Sarthi et al., 2024). However, most of these methods still treat memory access as a system-level process, leaving agents with limited control over how memory is utilized (Fang et al., 2026; Yan et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2026b). The update of memories is often predetermined by system-level heuristic algorithms rather than truly reflecting the actual contribution of each memory to downstream tasks. As a result, agents have difficulty dynamically balancing memory retention and removal during reasoning, leading to a decline in task utility (Yan et al., 2026b; Zhong et al., 2024).

Second, memory retrieval must adapt to the information needs of each query. Prior studies have introduced adaptive mechanisms such as retrieval-time reflection, query personalization, and result re-ranking (Chhikara et al., 2025; Wu et al., 2025; Trivedi et al., 2023; Shi et al., 2024). However, these methods still rely primarily on semantic similarity between the current query and memory items (Lewis et al., 2020; Karpukhin et al., 2020; Izacard et al., 2022; Borgeaud et al., 2022). Such similarity reflects only surface-level relevance and cannot determine whether a memory is still valid, reliable, or truly useful for the final answer (Asai et al., 2024; Jiang et al., 2023). A memory item that is highly similar to the query may merely repeat existing information or even conflict with the user's latest state, while a seemingly dissimilar memory may be precisely the key evidence needed for the current task. Therefore, memory access should not be a static query-memory match, but a dynamic value judgment based on the model's internal latent state (Packer et al., 2024; Wang et al., 2023; Modarressi et al., 2025; Lu et al., 2023). This latent state view suggests that memory should be represented, accessed, and updated according to how it affects the agent's reasoning trajectory, rather than solely according to external embedding similarity (Xu et al., 2026a).

To address these two challenges, we propose ValMem, a value-aware memory management framework that treats memory maintenance as a learned decision problem. ValMem combines Multi-Granularity Hierarchical Memory (MGHM) with Value-Driven Training (VDT). MGHM organizes memory along two dimensions: temporal layers (short-term, episodic, and long-term) and representation granularities (raw, record, topic, and abstract). This two-dimensional organization does not predetermine which representation is best for future tasks; instead, it provides explicit memory units over which retrieval and update decisions can be made. VDT addresses credit assignment for these retrieval and update decisions. For each training instance, VDT samples factual and counterfactual trajectories from the same checkpoint and derives an operation-specific credit signal from the resulting differences in downstream task utility and memory cost.

Extensive experiments on various benchmarks demonstrate that ValMem achieves superior success rates and exhibits strong generalization across different scenarios. The contributions are summarized as follows:

- We formulate long-horizon memory management as a fine-grained credit assignment problem. Building on this formulation, we propose ValMem, a value-aware framework that couples structured memory organization with a learned memory-operation policy.
- We introduce MGHM, which organizes interaction history across temporal and granularity dimensions. Beyond hierarchical storage, MGHM exposes memories at different resolutions as explicit decision units, allowing the agent to retain valuable evidence.
- We propose VDT for operation-specific credit assignment, avoiding diffusing sequence-level rewards across memory operations with different long-term effects.
- Experiments on three long-horizon memory benchmarks show that ValMem improves overall task performance while reducing memory and inference costs.

2 Related Work

2.1 Memory Construction and Organization

Early memory systems store experiences, reflections, and contextual information through largely predefined mechanisms (Park et al., 2023; Zhong et al., 2024; Packer et al., 2024). Recent work introduces more structured representations: Mem0 extracts key facts and supports graph-based relational memory (Chhikara et al., 2025), LightMem performs staged filtering and topic-aware updates (Fang et al., 2026), and A-MEM maintains a dynamically evolving linked memory network (Xu et al., 2025). These methods improve memory organization and storage efficiency, but their representation spaces and consolidation procedures remain largely fixed. In contrast, ValMem organizes memory jointly across temporal layers and representation granularities, and learns from downstream utility whether each item should be retained, compressed, or discarded.

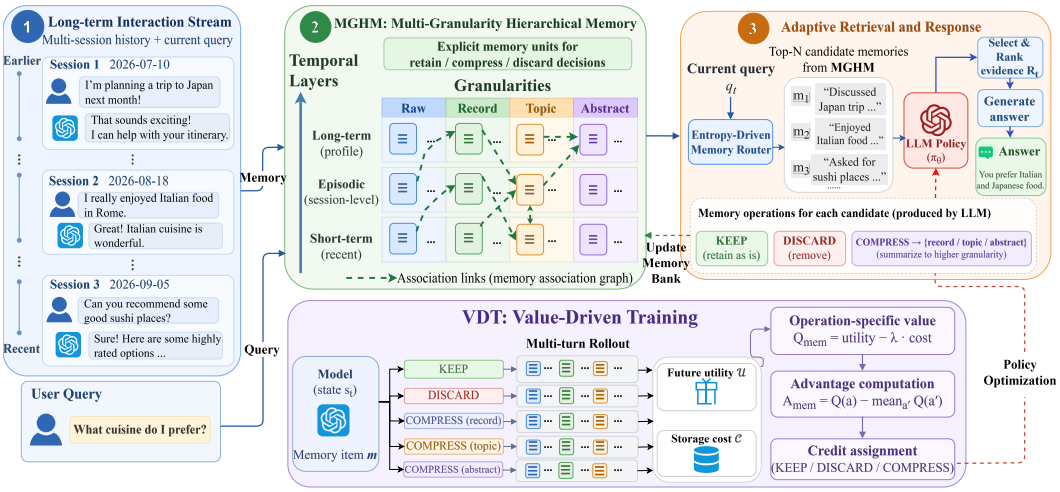


Figure 1: Overall framework of ValMem. MGHM organizes long-term memories across temporal and granularity dimensions, while VDT learns value-aware keep, discard, and compression decisions through operation-specific credit assignment, enabling effective retrieval, memory updating, and efficient responses.

2.2 Adaptive Retrieval for User Memory

Long-term retrieval requires more than semantic similarity: selected memories should remain temporally valid, preserve sufficient context, and expose the granularity required by the current query. MemoryBank and Mem0 rely substantially on semantic relevance (Zhong et al., 2024; Chhikara et al., 2025), HippoRAG augments retrieval with graph-based associations (Gutiérrez et al., 2024), and RaMem restores contextual information from compressed memories before generation (Yang et al., 2026). RMM and NapMem further introduce multi-granularity representations with learned access policies (Tan et al., 2025; Xu et al., 2026b). However, these methods do not explicitly assign value to individual memory operations according to their downstream marginal contribution. ValMem addresses this gap by using MGHM as a structured memory space over which VDT estimates operation-specific value.

2.3 RL-Based and Agentic Memory Systems

Recent work increasingly treats memory management as part of the agent's action space. MEM1 learns long-horizon reasoning under constrained memory consumption (Zhou et al., 2026), AgeMem learns operations such as storing, retrieving, updating, summarizing, and discarding information (Yu et al., 2026b), and NapMem learns to navigate a multi-level memory through reinforcement learning (Xu et al., 2026b). A central challenge is credit assignment: early memory operations alter the state inherited by later interactions, making trajectories with different memory decisions difficult to compare directly. Memory-R2 mitigates this issue through local rerollouts from the same intermediate memory state (Yan et al., 2026a). ValMem further estimates the utility of each memory item–operation pair by evaluating alternative operations from the same checkpoint, combining downstream task utility with storage cost, and assigning the resulting advantage directly to the corresponding memory-decision tokens. This design enables finer-grained optimization of memory decisions by linking downstream utility directly to the specific operations that produced them.

3 Problem Formulation

Our goal is to build a personalized assistant from long-term conversational memory. Given multi-session user-agent interactions, the system constructs an external memory bank M with a multi-granularity hierarchical structure, where memories are organized along temporal layers $\mathcal{L} = \{\text{long-term}, \text{episodic}, \text{short-term}\}$ and granularity levels $\mathcal{G} = \{\text{raw}, \text{record}, \text{topic}, \text{abstract}\}$. When a query q arrives, the assistant retrieves a relevant subset $M_{\text{rel}} \subseteq M$ through cross-granularity

associations, and generates a response $a = \text{LLM}(q, M_{\text{rel}})$. The memory bank is then updated based on the interaction. Formally, this study aims to learn a retrieval and update policy θ:

$$M_{\text{rel}} = \text{Retrieve}(q, M|\theta), \quad a = \text{LLM}(q, M_{\text{rel}}), \quad M \leftarrow \text{Update}(M, q, a|\theta), \tag{1}$$

where the core challenges are as follows: (i) establishing cross-granularity memory associations so that evidence at different temporal layers and granularities can be jointly retrieved, and (ii) learning an adaptive memory operation policy that balances information completeness against retrieval noise, thereby maximizing downstream task performance under a memory budget.

4 METHOD

ValMem consists of Multi-Granularity Hierarchical Memory and Value-Driven Training. MGHM defines how historical information is represented, retrieved, and updated across temporal layers and granularity levels. VDT estimates the downstream credit of each memory operation and uses it to train the LLM agent.

4.1 MULTI-GRANULARITY HIERARCHICAL MEMORY

4.1.1 MEMORY ORGANIZATION

MGHM maintains three temporal layers: short-term memory ST for the current interaction, episodic memory EP for specific events and their circumstances, and long-term memory LT for persistent cross-session knowledge such as preferences and facts. Formally, a memory item is defined as

$$m_i = \langle c_i, \ell_i, g_i, \mu_i \rangle, \qquad \ell_i \in \mathcal{L} = \{\text{ST}, \text{EP}, \text{LT}\}, \qquad g_i \in \mathcal{G}, \tag{2}$$

where c_i represents the content, ℓ_i represents the temporal layer, g_i represents the available granularity, and μ_i is the metadata (item identifier, source interaction, and available time information). The memory bank at step t is

$$M_t = \bigcup_{(\ell,g)\in\mathcal{L}\times\mathcal{G}} M_t^{\ell,g} \tag{3}$$

where $M_t^{\ell,g}$ is the set of memory items at temporal layer ℓ and granularity g. The metadata is returned along with the retrieved content. When the compressed information is insufficient to complete the task, the original context can be retrieved using the metadata for supplementation, and the memory can be updated accordingly.

4.1.2 GRANULARITY-AWARE REPRESENTATION

Within each temporal layer, a memory item can take four granularities, which can be formally expressed as $\mathcal{G} = \{\text{raw}, \text{record}, \text{topic}, \text{abstract}\}$, where *raw* retains the original content, *record* extracts a compact account of a fact or event, *topic* summarizes information concerning a shared subject, and *abstract* retains preferences, patterns, or conclusions.

To capture cross-granularity dependencies, we maintain a dynamic association graph over the memory bank. For each memory item (m) and granularity g, we encode its granularity-specific representation as $e_g(m) = \phi_g(z_g(m))$, where $z_g(m)$ is the granularity-specific content of m. For m_{new} and each existing memory $m_i \in M_t$, the pairwise similarity across all granularities is

$$s_{i,g} = \text{sim}\left(\phi_g(z_g(m_{\text{new}})), \phi_g(z_g(m_i))\right), \tag{4}$$

The resulting similarity vector $s_{\text{sim}} = [s_{i,g}]$ is clustered by a Gaussian Mixture Model (GMM) (Rasmussen, 1999) into an accept set and a reject set. The concentrated memories are contextually related to m_{new} and form a direct association, while the rejected memories are excluded. We then update the association graph $\mathcal{K}_t \leftarrow \mathcal{K}_t \cup \mathcal{K}_{\text{new}}$, where $\mathcal{K}_{\text{new}}$ contains edges between m_{new} and its accept set. This process selectively strengthens relevant memories at different granularities, simulating human memory consolidation and providing a structural basis for cross-granularity retrieval and adaptive granularity selection.

4.1.3 MEMORY RETRIEVAL AND UPDATE

Given a query q_t and the structured memory $\{M, \mathcal{K}\}$, ValMem adaptively assigns retrieval weights to different memories through an entropy-driven memory router. Then ValMem selects and ranks the final evidence set and generates the response.

Entropy-Driven Memory Routing. For each memory item $m_i \in M_t$, let $\mathcal{N}(i) \subseteq M_t$ denote its local neighborhood in the association graph $\mathcal{K}_t$. We compute the relevance distribution of q_t over this neighborhood:

$$p_i^j = \frac{\exp\big(\phi(q_t)^\top \phi(z_j)/\tau\big)}{\sum_{m_{j'} \in \mathcal{N}(i)} \exp\big(\phi(q_t)^\top \phi(z_{j'})/\tau\big)}, \qquad m_j \in \mathcal{N}(i), \tag{5}$$

where $\phi(\cdot)$ is the embedding encoder, z_j is the content of m_j, and $\tau > 0$ is a temperature parameter. A lower entropy indicates that the query admits a concentrated, reliable match within the local neighborhood of m_i, while a higher entropy suggests ambiguous or distributed relevance. The router weight for memory item m_i is obtained by inverse-entropy normalization. The entropy of this distribution $H_i > 0$ and the router weight w_i are

$$H_i = -\sum_{m_j \in \mathcal{N}(i)} p_i^j \log p_i^j, \qquad w_i = \frac{1/H_i}{\sum_{m_{i'} \in M_t} 1/H_{i'}}. \tag{6}$$

Candidate Retrieval and Evidence Selection. The router weights are used to produce a unified retrieval score for each memory item, $\mathrm{score}_i = w_i \cdot \mathrm{sim}(q_t, m_i)$, and the top-$N$ items form the candidate set C_t:

$$C_t = \mathrm{TopN}_{m_i \in M_t}\, \mathrm{score}_i, \qquad |C_t| = N, \tag{7}$$

where $\mathrm{sim}(q_t, m_i)$ is the cosine similarity between the query embedding and the memory embedding of m_i. The LLM agent then receives C_t together with q_t and H_t, selects and ranks up to $k < N$ items to form the evidence set R_t, and generates the response y_t conditioned on R_t:

$$R_t = \mathrm{Select}_\theta(C_t, q_t, H_i), \qquad y_t \sim \pi_\theta(\cdot \mid q_t, R_t). \tag{8}$$

This design separates memory routing from evidence selection: the entropy-driven router determines which memory items are reliable for the current query, while the learned policy determines which specific memories support the answer.

Memory Update. After generating y_t, the same model predicts an operation for each candidate $m_i \in C_t$. For an item with granularity g_i, the operation space is

$$\mathcal{A}(m_i) = \{\texttt{keep}, \texttt{discard}\} \cup \{\texttt{compress}(\alpha) \mid \alpha \in \mathcal{G},\ g_i \prec \alpha\}, \tag{9}$$

where α specifies the target granularity. The transformation induced by an operation $a_{t,i}$ is

$$\mathcal{T}(m_i, a_{t,i}) = \begin{cases} \{m_i\}, & a_{t,i} = \texttt{keep}, \\ \{\widetilde{m}_{t,i}^{\alpha}\}, & a_{t,i} = \texttt{compress}(\alpha), \\ \emptyset, & a_{t,i} = \texttt{discard}, \end{cases} \tag{10}$$

4.2 VALUE-DRIVEN TRAINING

In order to better identify the latent value of memory, we use reinforcement learning to train an LLM agent. Standard group-relative reinforcement learning (Guo et al., 2025) assigns the same sequence-level advantage to all decisions within a sampled trajectory. Assigning the same sequence-level advantage to all decisions is problematic because a trajectory may contain several memory operations whose downstream effects differ substantially. To address this credit-assignment mismatch, VDT separates the task-level learning signals from the memory signals specific to items and operations, and combines them at the token-level.

Reward Design. Let $B(M)$ denote the persistent storage occupied by memory bank M, and let B_{ref} denote the reference storage cost. This study defines the task reward, excluding the counterfactual memory-value term, as follows:

$$r(\tau) = F(\tau) + C(\tau) - S(\tau), \quad S(\tau) = 1 - \exp\Big(-\frac{B(M_{t+h})}{B_{\text{ref}}}\Big), \tag{11}$$

where F measures format validity, C measures answer correctness, and S measures the memory cost. Memory operations affect the state available for subsequent interactions, rendering the environment non-stationary. To capture the delayed effects of memory operations, we additionally consider the long-term effect of each memory action in addition to the immediate task reward.

Latent Value Assessment. Consider a memory-management checkpoint $s_t = (q_t, M_t)$ immediately before applying an operation to candidate memory m. The feasible operation set is $\mathcal{A}(m)$ as defined in Eq. 9. Importantly, each compression target is treated as a distinct action. For every $a \in \mathcal{A}(m)$, we construct the post-operation memory state:

$$M_t^{(a)} = \big(M_t \setminus \{m\}\big) \cup \mathcal{T}(m, a). \tag{12}$$

We use the same checkpoint, the same subsequent request sequence, and the same frozen policy snapshot for all alternative actions to sample K continuation trajectories:

$$\tau_k^{(a)} \sim \pi_{\theta_{\text{old}}}\big(\cdot \mid s_t, M_t^{(a)}\big), \qquad k = 1, \ldots, K, \tag{13}$$

The action-conditioned utility is estimated by

$$\widehat{Q}^{\text{mem}}(m, a; s_t) = \frac{1}{K}\sum_{k=1}^{K} r\big(\tau_k^{(a)}\big). \tag{14}$$

Rather than using an ambiguous "apply versus not apply" comparison, VDT evaluates all feasible actions explicitly. For policy optimization, we center the value of each operation against the other feasible operations for the same memory item:

$$\widehat{A}^{\text{mem}}(m, a; s_t) = \frac{\widehat{Q}^{\text{mem}}(m, a; s_t) - \mu_Q(m; s_t)}{\sigma_Q(m; s_t) + \epsilon}, \quad \mu_Q(m; s_t) = \frac{1}{|\mathcal{A}(m)|}\sum_{a' \in \mathcal{A}(m)} \widehat{Q}^{\text{mem}}(m, a'; s_t), \tag{15}$$

where $\sigma_Q(m; s_t)$ is the corresponding standard deviation. This construction provides credit to all feasible decisions, including KEEP, DISCARD, and each COMPRESS(α) action.

Memory-Aware Optimization Objective. For each context x, this study samples a standard GRPO group $\{y_i\}_{i=1}^{G}$ from $\pi_{\theta_{\text{old}}}$. The task-level advantage is

$$A_i^{\text{task}} = \frac{r(\tau_i) - \text{mean}_j\, r(\tau_j)}{\text{std}_j\, r(\tau_j) + \epsilon}. \tag{16}$$

Let $\mathcal{D}_i$ denote the memory decisions appearing in trajectory i. We assign the counterfactual memory advantage only to the decision tokens corresponding to each operation. For a decision $d \in \mathcal{D}_i$, let $\mathcal{T}_d$ be the set of token positions that represent this decision. The token-level advantage is defined as

$$\widetilde{A}_{i,t} = A_i^{\text{task}} + \sum_{d \in \mathcal{D}_i} \frac{\mathbf{1}[t \in \mathcal{T}_d]}{|\mathcal{T}_d|} \cdot \widehat{A}_d^{\text{mem}}, \tag{17}$$

where $\mathbf{1}[\cdot]$ is the indicator function. Non-decision tokens receive only the task advantage A_i^{task}. In this way, answer tokens retain the ordinary task-level advantage, whereas tokens corresponding to a memory operation additionally receive the counterfactual credit associated with that operation. The resulting optimization objective is

$$\mathcal{J}(\theta) = \mathbb{E}_x\left[\frac{1}{\sum_{i=1}^{G}|y_i|}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|}\Big(\min\big(\rho_{i,t}\widetilde{A}_{i,t}, \text{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)\widetilde{A}_{i,t}\big) - \beta D_{\text{KL}}\big(\pi_\theta \| \pi_{\text{ref}}\big)\Big)\right], \tag{18}$$

where

$$\rho_{i,t} = \frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid x, y_{i,<t})}. \tag{19}$$

5 EXPERIMENTS

In this section, this study evaluated ValMem on a diverse set of benchmarks designed to assess the capabilities of LLM agents. We adopt Qwen3.5-9B as the base model. Sec. 5.1 provides a comprehensive overview of the evaluation benchmarks and metrics. Implementation details are provided in Sec. 5.2. Experimental results on each benchmark are reported in Sec. 5.3. Ablation studies are provided in Sec. 5.4 to validate the effectiveness of each component. Additional validation experiments regarding memory operations are included in Appendix C.

5.1 EVALUATION DETAILS

Benchmarks. We evaluate on LoCoMo, LongMemEval, and PersonaMem-v2. LoCoMo contains conversations spanning multiple sessions and supports the assessment of factual recall as well as reasoning about previous events. LongMemEval tests information extraction, cross-session reasoning, temporal reasoning, knowledge update, and the ability to abstain from answering. PersonaMem-v2 evaluates personalized responses based on implicit user preferences, requiring the agent to select responses that are consistent with the information revealed in previous interactions. The experimental setup and data division were based on the official settings. The detailed information about the datasets is provided in Appendix A.

Metrics. For LoCoMo and LongMemEval, we report the token-level F1 and LLM-judge accuracy, denoted as L-J. The F1 measures the token-level overlap between the answer and the reference answer, while L-J measures the proportion of answers judged as correct against the reference answers. For PersonaMem-v2, we report the accuracy of multiple-choice questions. All scores are on a scale of 0 to 100, with higher values indicating better performance.

Baselines. We compare against RAG (Lewis et al., 2020), A-MEM (Xu et al., 2025), Mem0 (Chhikara et al., 2025), MEM1 (Zhou et al., 2026), Zep (Rasmussen et al., 2025), MemOS (Li et al., 2025), MemoryOS (Kang et al., 2025), MemAgent (Yu et al., 2026a), AgeMem (Yu et al., 2026b), Memory-R2 (Yan et al., 2026a), MemGAS (Xu et al., 2026a) and NapMem (Xu et al., 2026b) under the official settings. These systems include fact-based memory, temporal knowledge graphs, hierarchical memory organization, and learned memory management strategies. The detailed information about the baselines is provided in Appendix B.

5.2 IMPLEMENTATION DETAILS

We train the LLM agent on LoCoMo (Maharana et al., 2024), a benchmark consisting of long-term multi-session conversations. The division is carried out at the level of complete dialogues, so all sessions and annotations related to the same dialogue narrative will be retained in the same partition. LongMemEval (Wu et al., 2025) and PersonaMem-v2 (Jiang et al., 2025) are only used to evaluate the generalization ability across benchmarks, and their examples are not used to update the model parameters. All experiments are conducted using 4 NVIDIA A800 GPUs. In the training stage, the learning rate is set to 2×10^{-6} with a global batch size of 8. The detailed information about the training and deploy is provided in Appendix B.1.

5.3 MAIN RESULTS

LoCoMo. As shown in Table 1, ValMem achieved the best results in both overall F1 and BLEU-1 metrics, with scores of 48.14 and 39.13 respectively, which were 5.62 and 8.3 points higher than the strongest baseline. In multi-hop and open-domain tasks, the improvement effect was particularly significant, with F1 scores increasing by 7.51 and 10.59 points respectively. These results are consistent with the intended roles of MGHM and VDT. Although ValMem performed the best in terms of overall F1 and BLEU-1 metrics, its overall L-J score was slightly lower than that of MemAgent. One possible explanation is a difference in response verbosity. ValMem sometimes produces answers containing additional contextual information, which may affect judge-based evaluation.

LongMemEval. As shown in Table 2, ValMem achieves F1 and L-J scores of 65.79 and 86.28, exceeding NapMem by 9.76 and 6.74 points, respectively. This indicates that ValMem has an advan-

Table 1: Main results on LoCoMo. We report token-level F1, B1, and LLM-as-a-Judge. The highest score in each column is in **bold**, and the second-highest is underlined.

	Multi-hop			Open Domain			Single Hop			Temporal			Overall		
Model	**F1**	**B1**	**L-J**	**F1**	**B1**	**L-J**	**F1**	**B1**	**L-J**	**F1**	**B1**	**L-J**	**F1**	**B1**	**L-J**
Baselines															
RAG (Lewis et al., 2020)	11.93	5.12	17.87	9.35	11.98	16.52	10.84	4.72	15.25	5.39	10.44	3.13	11.56	5.25	15.09
A-MEM (Xu et al., 2025)	21.26	10.73	43.65	12.95	14.95	28.65	32.44	23.69	49.52	21.74	22.85	25.93	27.76	19.36	43.62
Mem0 (Chhikara et al., 2025)	27.57	15.69	64.86	18.32	18.36	45.36	34.75	22.57	67.98	30.99	28.72	35.73	32.51	20.98	56.20
Memory-R2 (Yan et al., 2026a)	22.37	20.52	59.17	22.62	13.47	50.84	30.55	27.76	62.37	35.29	23.72	41.13	38.30	26.25	60.87
MemoryOS (Kang et al., 2025)	31.95	20.37	50.96	19.09	20.74	35.91	42.71	33.72	65.94	24.16	22.14	22.02	36.43	26.73	53.94
MEM1 (Zhou et al., 2026)	19.78	10.34	44.26	20.65	17.18	40.33	30.00	20.35	60.06	29.49	29.01	41.78	28.42	18.87	53.55
MemAgent (Yu et al., 2026a)	38.50	22.20	**70.50**	27.39	27.82	**59.41**	52.48	41.42	**76.40**	20.88	18.96	54.13	42.52	30.83	**67.23**
Our Method															
ValMem	**46.01**	**32.84**	68.58	**37.98**	**32.65**	58.27	**59.27**	**51.16**	71.12	**47.31**	**38.36**	**64.69**	**48.14**	**39.13**	65.01

Table 2: Memory-intensive task performance on LongMemEval and PersonaMem-v2. The highest score in each column is in **bold**, and the second-highest is underlined.

Method	**LongMemEval**		**PersonaMem-v2**
	F1	L-J	Acc.
Mem0 (Chhikara et al., 2025)	52.84	76.12	41.23
Zep (Rasmussen et al., 2025)	50.97	72.45	38.02
MemOS (Li et al., 2025)	41.26	55.83	35.74
MemoryOS (Kang et al., 2025)	23.51	22.36	33.18
AgeMem (Yu et al., 2026b)	36.42	49.78	26.53
Memory-R2 (Yan et al., 2026a)	47.09	64.87	40.86
MemGAS (Xu et al., 2026a)	19.25	58.73	35.11
NapMem (Xu et al., 2026b)	56.03	79.54	45.92
ValMem	**65.79**	**86.28**	**50.55**

tage in organizing recent, event-specific, and persistent information at multiple granularities, which helps retain precise evidence while reducing the interference from outdated or redundant memories.

PersonaMem-v2. On PersonaMem-v2, ValMem achieved the highest accuracy rate of 50.55%, which was 4.63 points higher than NapMem. Questions related to individuals usually require integrating multiple observations to infer stable user preferences. ValMem supports this process by using MGHM to retain fine-grained interaction information and integrating repetitive evidence into higher-level representations, while VDT reduces the influence of transient or weakly informative memories on the results. These mechanisms work together to improve the accuracy of preference inference while retaining relevant contextual details.

5.4 Ablation Study

Table 3 shows the results of the ablation study. MGHM brings a greater independent improvement in all three benchmarks, indicating that structured and multi-granularity memory is a key source of performance improvement, especially in PersonaMem-v2, where it is necessary to infer stable user characteristics from sparse and distributed evidence. Even without MGHM, VDT is still effective, indicating that it improves memory utilization efficiency rather than just improving representation quality. On LongMemEval and PersonaMem-v2, when VDT is combined with MGHM, the gain from VDT becomes larger, suggesting that structured memory provides a clearer unit for downstream credit assignment and helps distinguish useful information from outdated or redundant information. On LoCoMo, the additional improvement of VDT is relatively small, indicating that its the gains from VDT may partially overlap with those provided by improved memory organization. Overall, MGHM builds an effective memory space, while VDT learns how to fully utilize this space based on downstream task utility.

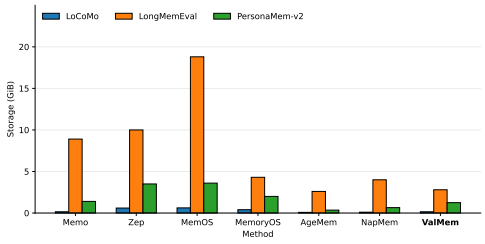


Figure 2: Memory-bank storage size in GiB.

Table 3: Ablation studies of ValMem on LoCoMo, LongMemEval, and PersonaMem-v2.

Variant	L.C.M. F1	L.M.E. F1	P.M.-v2 Acc.
ValMem	**48.1**	**65.7**	**50.5**
w/o VDT	45.8	59.9	45.3
w/o MGHM	41.7	56.9	39.5
w/o all	36.3	52.6	36.4

5.5 More Analysis: Consumption and Efficiency

Memory Storage Efficiency. Figure 2 shows that ValMem requires only 4.25 GiB of storage across the three benchmarks. Compared with NapMem, MemoryOS, Mem0, Zep, and MemOS, it reduces storage by 12.0%, 36.4%, 59.3%, 69.9%, and 81.6%, respectively. The advantage is most evident on LongMemEval, where ValMem uses 2.80 GiB, substantially less than most baselines. We attribute this efficiency to VDT, which suppresses low-utility and redundant memories, together with MGHM, which consolidates information at appropriate granularities. Although AgeMem uses 2.56 GiB of storage, its performance significantly declined on LongMemEval and PersonaMem-v2, indicating that there may be a trade-off between aggressive compression and evidence preservation. In contrast, ValMem achieves a more favorable trade-off between memory cost and answer quality by preserving detailed information when necessary while compacting less informative content.

Inference Efficiency. As shown in Figure 3, ValMem achieves the best efficiency trade-off, requiring 4.2 seconds and 0.62k completion tokens per successful run. It also achieves low latency and completion-token usage while maintaining competitive task performance. This indicates that ValMem not only reduces the average inference cost per run but also maintains stable generation behavior, avoiding the high variance observed in several baselines. The combination of MGHM and VDT allows the agent to avoid retrieving and generating redundant context, thereby achieving a favorable balance between answer quality and inference efficiency.

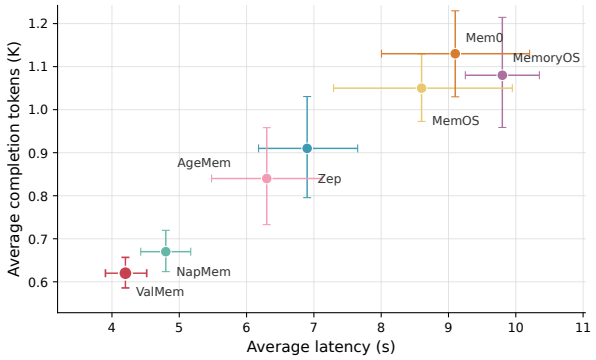


Figure 3: Statistics of delay and completion tokens for 100 PersonaMem-v2 samples.

6 Conclusion

This study presents ValMem, a unified value-aware memory framework designed to improve long-term memory management for LLM agents. We first introduce MGHM, which organizes historical interactions across temporal layers and representation granularities, enabling the agent to preserve task-relevant details while compacting redundant information. Furthermore, this study proposes VDT, which estimates memory utility through counterfactual marginal contribution and trains the agent to adaptively retain, compress, or discard memories under a storage budget. By combining structured memory organization with utility-driven policy optimization, ValMem enables memory decisions to be directly integrated into the reasoning process without auxiliary scorers during inference. Experiments on LoCoMo, LongMemEval, and PersonaMem-v2 demonstrate consistent improvements over strong baselines, while ablation and efficiency analyses confirm the complementary contributions of MGHM and VDT and their favorable trade-off between answer quality and memory consumption. These results highlight the potential of value-aware memory management for building more capable and efficient long-horizon LLM agents.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing, LaTeX formatting assistance, and generating draft code for data preprocessing and evaluation scripts. We have not used generative AI tools for generating research ideas, designing the method, fabricating experimental results, or writing the core technical contributions. Additionally, we used generative AI tools to assist with literature search and summarization, which was subsequently verified by the authors through manual inspection of the original papers. We have reviewed all AI-assisted work. Specifically, we checked all LLM-generated text for factual accuracy and plagiarism through manual literature comparison; LLM-generated code was tested and verified for correctness by authors on a subset of the datasets. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work focuses on memory management for long-horizon LLM agents. We evaluate our method exclusively on publicly available benchmark datasets (LoCoMo, LongMemEval, and PersonaMem-v2), which do not contain personally identifiable information or sensitive user data. No human subjects were involved in this study, and no new data was collected. The datasets used are governed by their respective licenses, and we adhered to their terms of use. While memory systems for personalized assistants may raise privacy concerns in real-world deployments, our research is conducted in a simulated environment and does not deploy any system to real users. We do not foresee any direct harmful applications of this work. The authors declare no conflicts of interest.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, we provide the following resources. For the novel method, we have released the anonymous source code as supplementary material, including the implementation of Multi-Granularity Hierarchical Memory (MGHM), Value-Driven Training (VDT), and all training and evaluation scripts. For the datasets used, we describe the preprocessing steps, data splits, and statistics in Appendix A. All datasets are publicly available and we follow the same experimental settings as prior work (Chhikara et al., 2025). For the experiments, we specify the base model (Qwen3.5-9B), hardware (4 NVIDIA A800 GPUs), and all hyperparameters (learning rates, batch sizes, group size) in Appendix B. We also provide the prompt templates used for memory operations and LLM-as-a-Judge evaluation in Appendix D. Random seeds and evaluation scripts are included in the supplementary code. We believe these details are sufficient for independent reproduction of our results.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 9112–9141, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/25f7be9694d7b32d5cc670927b8091e1-Paper-Conference.pdf.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 2206–2240. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/borgeaud22a.html.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory, 2025. URL https://arxiv.org/abs/2504.19413.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. Lightmem: Lightweight and efficient memory-augmented generation, 2026. URL https://arxiv.org/abs/2510.18866.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 59532–59569. Curran Associates, Inc., 2024. doi: 10.52202/079017-1902. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/6ddc001d07ca4f319af96a3024f6dbd1-Paper-Conference.pdf.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning, 2022. URL https://arxiv.org/abs/2112.09118.

Bowen Jiang, Yuan Yuan, Maohao Shen, Zhuoqun Hao, Zhangchen Xu, Zichen Chen, Ziyi Liu, Anvesh Rao Vijjini, Jiashu He, Hanchao Yu, Radha Poovendran, Gregory Wornell, Lyle Ungar, Dan Roth, Sihao Chen, and Camillo Jose Taylor. Personamem-v2: Towards personalized intelligence via learning implicit user personas and agentic memory, 2025. URL https://arxiv.org/abs/2512.06688.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7969–7992, Singapore, December 2023. Associa-

tion for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.495. URL https://aclanthology.org/2023.emnlp-main.495/.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI agent. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25961–25970, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1318. URL https://aclanthology.org/2025.emnlp-main.1318/.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

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. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Zhiyu Li, Shichao Song, Hanyu Wang, Simin Niu, Ding Chen, Jiawei Yang, Chenyang Xi, Huayi Lai, Jihao Zhao, Yezhaohui Wang, Junpeng Ren, Zehao Lin, Jiahao Huo, Tianyi Chen, Kai Chen, Kehang Li, Zhiqiang Yin, Qingchen Yu, Bo Tang, Hongkang Yang, Zhi-Qin John Xu, and Feiyu Xiong. Memos: An operating system for memory-augmented generation (mag) in large language models, 2025. URL https://arxiv.org/abs/2505.22101.

Junru Lu, Siyu An, Mingbao Lin, Gabriele Pergola, Yulan He, Di Yin, Xing Sun, and Yunsheng Wu. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation, 2023. URL https://arxiv.org/abs/2308.08239.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

Ali Modarressi, Abdullatif Köksal, Ayyoob Imani, Mohsen Fayyaz, and Hinrich Schütze. Memllm: Finetuning llms to use an explicit read-write memory, 2025. URL https://arxiv.org/abs/2404.11672.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems, 2024. URL https://arxiv.org/abs/2310.08560.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701320. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Carl Edward Rasmussen. The infinite gaussian mixture model. In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, NIPS'99, pp. 554–560, Cambridge, MA, USA, 1999. MIT Press.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory, 2025. URL https://arxiv.org/abs/2501.13956.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 32628–32649, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/8a2acd174940dbca361a6398a4f9df91-Paper-Conference.pdf.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551. Curran Associates, Inc., 2023. doi: 10.52202/075280-2997. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/d842425e4bf79ba039352da0f658a906-Paper-Conference.pdf.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: Retrieval-augmented black-box language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8371–8384, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.463. URL https://aclanthology.org/2024.naacl-long.463/.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652. Curran Associates, Inc., 2023. doi: 10.52202/075280-0377. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1b44b878bb782e6954cd888628510e90-Paper-Conference.pdf.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, Anand Rajan Iyer, Tianlong Chen, Huan Liu, Chen-Yu Lee, and Tomas Pfister. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8416–8439, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.413. URL https://aclanthology.org/2025.acl-long.413/.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10014–10037, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.557. URL https://aclanthology.org/2023.acl-long.557/.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 74530–74543. Curran Associates, Inc., 2023. doi: 10.52202/075280-3259. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/ebd82705f44793b6f9ade5a669d0f0bf-Paper-Conference.pdf.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 86809–86836, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/d813d324dbf0598bbdc9c8e79740ed01-Paper-Conference.pdf.

Derong Xu, Yi Wen, Pengyue Jia, Yingyi Zhang, Wenlin Zhang, Yichao Wang, Huifeng Guo, Ruiming Tang, Xiangyu Zhao, Enhong Chen, and Tong Xu. From single to

multi-granularity: Toward long-term memory association and selection of conversational agents. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 91581–91617, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/93e444b4a1a64815631c3c74977ee200-Paper-Conference.pdf.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents, 2025. URL https://arxiv.org/abs/2502.12110.

Yue Xu, Yutao Sun, Yihao Liu, Mengyu Zhou, Jiayi Qiao, Lu Ma, Kai Tang, Wenjie Wang, Xiaoxi Jiang, and Guanjun Jiang. From passive retrieval to active memory navigation: Learning to use memory as a structured action space, 2026b. URL https://arxiv.org/abs/2607.05794.

Sikuan Yan, Ahmed Bahloul, Ercong Nie, Susanna Schwarzmann, Riccardo Trivisonno, Volker Tresp, and Yunpu Ma. Memory-r2: Fair credit assignment for long-horizon memory-augmented llm agents, 2026a. URL https://arxiv.org/abs/2605.21768.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schuetze, Volker Tresp, and Yunpu Ma. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12805–12825, San Diego, California, United States, July 2026b. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.583. URL https://aclanthology.org/2026.acl-long.583/.

Wei Yang, Bryce Kan, Shixuan Li, Li Li, Yuehan Qin, Jiate Li, Paul Bogdan, and Jesse Thomason. Ramem: Contextual reinstatement for long-term agentic memory, 2026. URL https://arxiv.org/abs/2606.22844.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 39458–39486, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/4264ee4376776907c0b87ed70b959585-Paper-Conference.pdf.

Yi Yu, Liuyi Yao, Yuexiang Xie, Qingquan Tan, Jiaqi Feng, Yaliang Li, and Libing Wu. Agentic memory: Learning unified long-term and short-term memory management for large language model agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21457–21483, San Diego, California, United States, July 2026b. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.981. URL https://aclanthology.org/2026.acl-long.981/.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pp. 19632–19642. AAAI Press, 2024. doi: 10.1609/aaai.v38i17.29936. URL https://ojs.aaai.org/index.php/AAAI/article/view/29936.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: enhancing large language models with long-term memory. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications*

of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i17.29946. URL https://doi.org/10.1609/aaai.v38i17.29946.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Bryan Kian Hsiang Low, and Paul Liang. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 58413–58438, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/5fc8b3bdfbb9167b5144df5d3fae4616-Paper-Conference.pdf.

Table 4: Dataset statistics. The term "Avg." (e.g., Avg. Sessions) represents the average number corresponding to each conversation/question.

	LoCoMo	**LongMemEval**	**PersonaMem-v2**
Total conversations / personas	50	–	1,000
Avg. sessions per conversation	19	–	–
QA questions	7,512	500	5,000
Interaction topics/scenarios	–	–	300+
Session dates	✓	✓	–
Retrieval ground-truth	✓	✓	–
QA ground-truth	✓	✓	✓
Conversation subject	User–User	User–AI	Persona-grounded user–AI
Usage in ValMem	2:1:7 conversation-level train/validation/test split	Evaluation only	Evaluation only

A Dataset Details

We use three public benchmarks with substantially different data organizations. To avoid conflating conversations, sessions, and evaluation questions, Table 4 reports the native scale used by each benchmark rather than forcing all datasets into a common dialogue/session schema.

We evaluate the proposed method with the following benchmarks:

- **LoCoMo** contains 50 long conversations, with 19 sessions per conversation on average, and 7,512 QA questions in the released QA benchmark. We split LoCoMo by conversation identifier into a 2:1:7 train/validation/test partition, so that no session or annotation associated with the same conversation appears in multiple partitions. The exact conversation IDs and the resulting number of evaluation questions in each partition are provided in the supplementary split file.
- **LongMemEval** contains 500 curated long-term-memory questions. It is used for evaluation only and is not used to update model parameters; it measures transfer beyond the LoCoMo training distribution.
- **PersonaMem-v2** contains 1,000 user personas, 300+ interaction topics/scenarios, and 5,000 benchmark QA pairs. It is used for evaluation only and is not used to update model parameters; it measures transfer beyond the LoCoMo training distribution.

For LoCoMo, splitting is performed exclusively by conversation identifier so that no session or annotation associated with the same conversation appears in multiple partitions. LongMemEval and PersonaMem-v2 are not used to update model parameters and are used only to measure transfer beyond the LoCoMo training distribution.

B Baseline Details

We compare against the following baselines. Table 5 summarizes their memory organization, retrieval mechanism, and whether training is required. Detailed descriptions are provided below.

- **RAG** (Lewis et al., 2020): Standard retrieval-augmented generation using a dense retriever.
- **A-MEM** (Xu et al., 2025): A linked memory note network with autonomous memory generation and connection.
- **Mem0** (Chhikara et al., 2025): A fact-based memory system with graph-based relational memory.

Table 5: Overview of baseline methods. "Training" indicates whether the method requires additional model training or reinforcement learning beyond standard inference.

Baseline	Memory Organization	Retrieval Mechanism	Training
RAG	Dense vector store	Dense similarity search	No
A-MEM	Linked memory note network	Autonomous memory generation and connection	No
Mem0	Fact-based graph memory	Graph-based relational retrieval	No
MEM1	Constant memory	RL-based long-term reasoning	RL
Zep	Temporal knowledge graph	Temporal graph traversal	No
MemOS	Hierarchical memory organization	OS-like memory management	No
MemoryOS	Memory-centric operating system	Hierarchical memory access	No
MemAgent	Agentic memory system	Learned memory management	RL
AgeMem	Age-aware compressed memory	Aggressive compression	No
MemGAS	Multi-granularity memory units	Granularity-aware association and retrieval	No
NapMem	Linked memory pyramid	Tool-based navigation	No
Memory-R2	RL-based memory management	Reflective retrieval	RL

- **MEM1** (Zhou et al., 2026): An RL-based memory system for long-term reasoning with constant memory consumption.
- **Zep** (Rasmussen et al., 2025): A temporal knowledge graph memory system.
- **MemOS** (Li et al., 2025): A memory operating system with hierarchical memory organization.
- **MemoryOS** (Kang et al., 2025): A memory-centric operating system for LLM agents.
- **MemAgent** (Yu et al., 2026a): An agentic memory system with learned memory management.
- **AgeMem** (Yu et al., 2026b): An age-aware memory system with aggressive compression.
- **MemGAS** (Xu et al., 2026a): A granularity-aware memory system that organizes and retrieves memories at multiple levels of granularity, from fine-grained details to coarse-grained summaries.
- **NapMem** (Xu et al., 2026b): A linked memory pyramid with tool-based navigation.
- **Memory-R2** (Yan et al., 2026a): An RL-based memory management system with reflective retrieval.

Each baseline was evaluated under the official settings.

B.1 Implementation Details

All experiments are conducted using 4 NVIDIA A800 GPUs. In the training stage, the learning rate is set to 2×10^{-6} with a GRPO group size of 8, counterfactual rollout count of 8, rollout horizon of 3, selected-evidence limit of 5, router temperature of 1.0, PPO clipping parameter of 0.2, KL coefficient of 0.005, the optimizer is AdamW, training steps of 100, maximum generation length of 1024, gradient accumulation of 16. Regarding the retrieval part, for the Eq. 7, we follow prior work (Xu et al., 2026b) and set N = 5 for baselines and our method.

C Additional Experimental Results

Effect of Storage-Cost Reference Scale. All results are averaged over three training seeds (42, 43, 44). We evaluate ValMem on LoCoMo with $B_{\text{ref}} \in \{2, 4, 8\}$ GiB while keeping all other settings fixed. As shown in Table 6, F1 increases from 45.2 to 48.1 and 49.5 as the available budget increases,

with a smaller improvement between 4 and 8 GiB than between 2 and 4 GiB. This experiment establishes that performance is relatively stable once additional memory capacity is available, but it does not by itself demonstrate that VDT preferentially allocates capacity to high-value memories. Testing that mechanism requires direct comparison with a policy trained without value-based credit assignment.

Table 6: Effect of memory budget on LoCoMo.

B_{ref} (GiB)	F1	BLEU-1	L-J
2	45.2	37.1	62.8
4	48.1	39.1	65.0
8	49.5	42.0	68.9

Case Study. We provide a qualitative example in Table 7 to illustrate how ValMem selects and compresses memories. Given a query about the user's preferred cuisine, the agent retrieves a raw memory about a recent restaurant visit and an abstract memory summarizing the user's dietary preferences. The agent then compresses the raw memory into a record granularity, retaining the key fact while reducing storage.

Table 7: Case study of memory selection and compression.

Query	What kind of food does the user like?
Retrieved memories	Raw: "Yesterday I went to an Italian restaurant and had pasta." Abstract: "User prefers Italian and Japanese cuisine."
Operation	Compress raw memory to record: "User visited Italian restaurant."
Final answer	The user likes Italian and Japanese food.

C.1 Computational Cost

Training ValMem takes approximately 120 hours on 4 NVIDIA A800 GPUs. Inference on 100 samples takes 4.2 seconds per successful run on average, with 0.62K completion tokens. The memory bank storage is 4.25 GiB across all three benchmarks. These results demonstrate that ValMem achieves a favorable trade-off between performance and efficiency.

C.2 Controlled Comparison of Credit Assignment Strategies

To isolate the effect of the credit signal itself, we compare four strategies under the same backbone (Qwen3.5-9B), the same LoCoMo 2:1:7 split. B_{ref} is used to normalize the storage penalty in Eq. 11 and is a storage-penalty reference scale; realized storage may therefore exceed this value: (i) Standard GRPO with sequence-level advantage only; (ii) Memory-R2-style local/session rerollout, reimplemented under the common backbone and action space; (iii) item-operation rerollout without decision-token-localized credit; (iv) VDT, which is identical to (iii) except that the operation-specific advantage is assigned only to the corresponding decision tokens. Table 8 reports performance, storage, and operation quality. Operation accuracy is defined as the proportion of memory operations that match the action with the highest downstream utility in the rerollout estimate.

C.3 Ablation of VDT Components

We ablate three design choices in VDT: (a) **same-checkpoint**: all alternative actions are evaluated from the same memory state; (b) **explicit action enumeration**: every feasible operation, including each compression target, is evaluated as a distinct action; (c) **decision-token assignment**: the counterfactual memory advantage is assigned only to the tokens that represent the corresponding decision. Table 9 reports performance, storage, and operation accuracy.

Table 8: Controlled comparison of credit assignment strategies under identical backbone, split, and budget.

Method	F1	BLEU-1	L-J	Storage (GiB)	Op. Acc. (%)
Standard GRPO	42.53	34.81	60.24	0.64	55.04
Local/Session Rerollout	44.21	36.05	62.13	0.40	62.48
Item-Operation Rerollout	45.86	38.02	63.92	0.14	70.35
VDT (ours)	48.14	39.13	65.01	0.09	78.62

Table 9: Ablation of VDT components on LoCoMo, LongMemEval, and PersonaMem-v2.

Variant	LoCoMo F1	LongMemEval F1	PersonaMem-v2 Acc.	Storage (GiB)
VDT (full)	48.14	65.79	50.55	4.25
w/o same-checkpoint	46.23	62.41	47.82	4.81
w/o explicit action enumeration	45.52	60.93	46.17	4.94
w/o decision-token assignment	44.86	59.32	45.08	5.12

C.4 Ablation of MGHM Components

We ablate the main components of MGHM: (a) **GMM association**: replace the Gaussian Mixture Model accept/reject clustering with a fixed similarity threshold; (b) **entropy router**: replace inverse-entropy weights with uniform weights over the neighborhood; (c) **temporal hierarchy**: remove the short-term / episodic / long-term separation and store all items in a single layer; (d) **multi-granularity**: restrict all memories to the raw granularity. We also include a MemGAS-style controlled baseline (Xu et al., 2026a) that uses multi-granularity retrieval without value-aware operations. Table 10 reports F1 and storage.

Table 10: Ablation of MGHM components and comparison with a MemGAS-style baseline.

Variant	LoCoMo F1	LongMemEval F1	PersonaMem-v2 Acc.	Storage (GiB)
MGHM (full)	48.14	65.79	50.55	4.25
w/o GMM association	45.31	61.24	46.83	4.53
w/o entropy router	44.62	60.15	45.91	4.67
w/o temporal hierarchy	43.24	58.47	44.12	4.79
w/o multi-granularity	42.73	57.92	41.54	5.18
MemGAS-style baseline	26.41	19.25	35.11	4.82

C.5 Mechanistic Validation of Value-Aware Memory Operations

To test whether VDT actually learns to discard or compress low-value memories, we inject four types of adversarial memories into the bank: **outdated** (superseded user states), **duplicated** (repeated facts), **conflicting** (contradictory statements), and **irrelevant** (off-topic content). Table 11 reports the operation distribution and the selected-memory precision for each type. Table 12 reports the correlation between the estimated memory value and the true downstream delta measured by an independent rerollout.

D Prompt Templates

We provide the complete prompt templates used for memory operations, multi-granularity information generation, memory filtering, question answering, and LLM-as-a-Judge evaluation. All templates are designed for instruction-following LLMs and have been manually optimized to achieve strong performance on long-term conversational memory benchmarks. Placeholders are enclosed in curly braces.

Table 11: Operation distribution and selected-memory precision on injected adversarial memories.

Memory type	Discard rate (%)	Compress rate (%)	Keep rate (%)	Selected-memory precision (%)
Outdated	72.5	18.3	9.2	88.4
Duplicated	65.8	24.6	9.6	86.1
Conflicting	58.3	20.5	21.2	82.7
Irrelevant	81.6	12.4	6.0	90.2

Table 12: Correlation between estimated memory value and true downstream delta.

Metric	Pearson r	Spearman ρ
Value estimate vs. downstream delta	0.78	0.74

D.1 Memory Operation Prompt

This prompt is used to decide how a newly retrieved candidate memory should be handled given the current query.

Memory Operation Prompt

```
You are a memory management assistant for a long-term conversational
agent. Given a user query and a candidate memory retrieved from the
memory bank, choose the most appropriate operation for this memory.
Operations: - KEEP: The memory is directly relevant to the query
and should be retained without modification. - DISCARD: The memory
is irrelevant, redundant, or outdated and should be removed. -
COMPRESS: The memory contains useful information but is too verbose.
Compress it to a target granularity: record (a single fact or
event), topic (a coherent theme spanning multiple turns), or
abstract (a high-level summary).
Query: {query}
Candidate Memory: {memory}
Current Memory Bank Summary (optional): {memory_summary}
Instructions: 1. Analyze the relevance between the query and the
candidate memory. 2. If the memory is directly useful, output
KEEP. 3. If the memory is irrelevant or redundant, output DISCARD.
4. If the memory is partially useful but verbose, output COMPRESS
and specify the target granularity. 5. Provide a brief reason for
your choice.
Output format: Operation: [KEEP | DISCARD | COMPRESS] Target
Granularity (if COMPRESS): [record | topic | abstract] Reason:
[one-sentence explanation]
```

Example: Memory Operation

```
Query: What did I say about my sister's wedding?
Candidate Memory: User mentioned attending a wedding in July 2023
and discussed gift ideas.
Output: Operation: KEEP
Reason: The memory directly relates to the user's query about the
sister's wedding.
```

D.2 Multi-granularity Information Generation Prompts

These prompts generate structured multi-granularity metadata from raw dialogue sessions.

Summary Generation Prompt

```
Below is a user--AI assistant dialogue memory. Please summarize the
following dialogue as concisely as possible in a short paragraph,
extracting the main themes and key information. Preserve important
entities, dates, user preferences, and events. Do not include
irrelevant chit-chat.
Dialogue: {dialogue}
Abstract:
```

Keyword Generation Prompt

```
Below is a user--AI assistant dialogue memory. Please extract the
most relevant keywords, separated by semicolons. Focus on entities,
topics, actions, and attributes that capture the core content.
Provide 5--10 keywords, ordered by importance.
Dialogue: {dialogue}
Topic:
```

Record Extraction Prompt

```
Below is a user--AI assistant dialogue memory. Extract all
factual records about the user, such as personal details,
preferences, plans, and past events. Each record should be a
single, self-contained sentence. Output as a bullet list.
Dialogue: {dialogue}
Records:
```

D.3 Memory Filtering Prompt

This prompt refines retrieved memories by removing irrelevant or redundant content before answer generation.

Memory Filtering Prompt

```
You are an intelligent dialogue assistant. You will be shown
history dialogues and corresponding multi-granularity information
(session text, summaries, and keywords). Your task is to filter
the retrieved content to extract only the parts that are directly
relevant to the question.
Instructions: - Read the question and the retrieved history
carefully. - Preserve original tokens; do not paraphrase or
rewrite. - Remove irrelevant turns, redundant information, and
non-essential details. - Keep all information that is necessary
to answer the question, including temporal anchors, entities, and
preferences. - If no relevant information exists, output: ``No
relevant information found.''
History Dialogues: {retrieved_texts}
Question Date: {question_date}
Question: {question}
Filtered Context:
```

Example: Memory Filtering

```
Question: What company did I mention for sustainability?
Retrieved: [Session 1] ... Patagonia is known for sustainability
... [Session 2] ... eco-friendly laundry detergent ...
Filtered Context: Patagonia is known for its commitment to
sustainability and environmental responsibility throughout its
supply chain.
```

D.4 Question Answering Prompt

This prompt generates the final answer using the filtered context.

Question Answering Prompt

```
You are an intelligent dialogue assistant. You will be shown
history dialogues. Please read, memorize, and understand the given
dialogues, then generate one concise, coherent, and helpful response
for the question.
Instructions: - Base your answer strictly on the provided history
dialogues. - If the answer is not present, respond with ``The
information is not mentioned in the conversation.'' - Do not
fabricate details. - Keep the answer concise and directly address
the question.
History Dialogues: {retrieved_texts}
Question Date: {question_date}
Question: {question}
Answer:
```

D.5 LLM-as-a-Judge Prompt

This prompt is used for automatic evaluation of answer correctness.

LLM-as-a-Judge Prompt

```
You are an impartial judge. Evaluate the following answer based on
the reference answer.
Instructions: - Compare the model answer with the reference
answer. - Consider whether the model answer contains the same
key information as the reference answer. - Ignore differences in
phrasing, style, or additional correct details. - Output YES if
the answer is correct, otherwise output NO. - Do not provide any
explanation.
Question: {question}
Reference Answer: {reference}
Model Answer: {answer}
Correct? (YES/NO):
```

Example: LLM-as-a-Judge

```
Question: How many online courses have I completed?
Reference: 5
Model Answer: You have completed five online courses.
Output: YES
```

E CASE STUDY

We present representative cases from our experiments to illustrate how ValMem performs memory operations, multi-granularity information generation, filtering, and question answering. We compare ValMem with several strong baselines, including RAG (Lewis et al., 2020), A-MEM (Xu et al., 2025), Mem0 (Chhikara et al., 2025), and NapMem (Xu et al., 2026b). Correct and incorrect responses are marked with ✓ and ×, respectively.

E.1 CASE STUDY ON QA COMPARISON

Case 1: Multi-session Question – Online Course Completion

```
Query: How many online courses have I completed in total?
Ground-Truth: 5
ValMem: You have completed a total of five online courses. ✓
NapMem: You have completed five online courses. ✓
Mem0: Based on the dialogues, you have completed two courses on
edX. ×
A-MEM: You have completed a total of 2 online courses on edX. ×
RAG: Please provide the number of courses completed from each
platform. No Answer
```

Case 2: Multi-session Question – Tomato and Cucumber Plants

```
Query: How many plants did I initially plant for tomatoes and
cucumbers?
Ground-Truth: 8
ValMem: You initially planted 5 tomato plants and 3 cucumber
plants. ✓
NapMem: You initially planted 5 tomato plants and 3 cucumber
plants. ✓
Mem0: You initially planted 5 tomato plants. The conversation does
not specify the number of cucumber plants. ×
A-MEM: You did not specify the number. Please clarify. No Answer
RAG: A good starting point is 1 tomato per 18--24 inches. No Answer
```

Case 3: Temporal Reasoning – Meeting Emma

```
Query: How many days ago did I meet Emma?
Ground-Truth: 9 days ago (10 days incl. the last day is
acceptable)
ValMem: You met Emma 9 days ago, on April 11, 2023. ✓
NapMem: You met Emma 9 days ago, on April 11, 2023. ✓
Mem0: Please provide the date you met Emma. No Answer
A-MEM: The exact date isn't available. No Answer
RAG: There is no specific mention of meeting Emma. No Answer
```

These results indicate that ValMem is better equipped to support multi-turn factual consistency and basic temporal inference within multi-session contexts, largely due to its value-aware memory management and multi-granularity hierarchical memory.

E.2 Case Study on Memory Operation

Memory Operation Example

```
Query: ''I was looking back at our previous conversation about
environmentally responsible supply chain practices, and I was
wondering if you could remind me of the company you mentioned that's
doing a great job with sustainability?''
Candidate Memory 1: ''Patagonia, an outdoor clothing and gear
company, is known for its commitment to sustainability and
environmental responsibility throughout its supply chain.''
Operation: KEEP
Reason: Directly answers the query by naming the company and its
sustainability practices.
Candidate Memory 2: ''The user is considering trying a new grocery
delivery service that partners with local farmers. They also asked
about eco-friendly laundry detergent.''
Operation: DISCARD
Reason: Irrelevant to the supply chain sustainability question.
Final Answer: ''The company you mentioned is Patagonia, known for
its commitment to environmentally responsible practices throughout
its supply chain.''
```

This demonstrates how the memory operation module reduces noise and improves answer precision.

E.3 Case Study on Multi-granularity Information Generation and Filtering

Multi-granularity Information Generation

```
Original Session (abridged):
[user] I'm thinking of taking an antique restoration class to learn
how to fix up some of the old pieces I've inherited, like that
beautiful vase I got from my grandmother last month. Do you know
of any good classes in the city?
[assistant] I'll definitely look into those options. By the way, do
you know how to appraise the value of antique items?
[user] I'm thinking of taking an antique restoration class to learn
how to fix up some of the old pieces I've inherited...
[assistant] Appraising the value of antique items can be a
fascinating and complex process...
Generated Summary: ''The user plans to retrieve their antique
tea set, which they lent to their sister for her wedding. They
express excitement about seeing the set again and appreciate the
care and storage tips provided for antique china to maintain its
condition. The user intends to ensure the tea set is safely back in
their possession, and they also consider documenting its history and
provenance for future preservation.''
Generated Keywords: antique tea set; storage tips; caring for
china; keep in good condition; avoid stacking; sturdy box; line
with tissue; dry cool place.
```

Multi-granularity Filtering

```
Query: What did I lend to my sister for her wedding?
Before Filter:
Memory 1: [Session] ... user lent antique tea set to sister for
wedding ... [Summary] ... [Keywords] antique tea set; wedding;
sister ...
Memory 2: [Session] ... user asked about eco-friendly laundry
detergent ... [Summary] ... [Keywords] laundry; eco-friendly ...
After Filter:
The user lent their antique tea set to their sister for her wedding.
Final Answer: ''You lent your antique tea set to your sister for
her wedding.''
```

This illustrates how multi-granularity generation and filtering work together to produce a concise and accurate response.

E.4 Case Study on the Effect of Value-Driven Training

Effect of Value-Driven Training

```
Query: ''How many days did I spend in total on my trip to Japan?''
Gold Answer: 12 days.
Before VDT: Retrieves only ''5 days in Tokyo'' → Answer: ''5
days'' ×
After VDT: Retrieves ''3 days in Kyoto'', ''5 days in Tokyo'', ''4
days in Osaka'' → Answer: ''12 days'' ✓
Reason: VDT teaches the router to aggregate complementary sessions
and the policy to retain high-value memories.
```

This improvement is attributed to VDT, which estimates the counterfactual advantage of each memory operation and assigns credit to the corresponding decision tokens. The agent learns to retain complementary memories and compress redundant ones, enabling it to discover and combine relevant memories across multiple sessions—a capability that static retrieval methods often lack.