SAMem: State-Aware Memory as a Fine-Grained Memory for LLM Agents in Decision-Making

Tong Wang[1,2], Pei Xu[2*], Shiyue Cao[1,2], Likun Yang[1,2], Daipeng Li[1,2], Jianbin Jiao[1], Kaiqi Huang[1,2*]

[1]University of Chinese Academy of Sciences,

[2]National Key Laboratory of Cognition and Decision Intelligence for Complex Systems, Institute of Automation, Chinese Academy of Sciences

wangtong181@mails.ucas.ac.cn, pei.xu@ia.ac.cn, kqhuang@nlpr.ia.ac.cn

Abstract

Existing LLM-based agents primarily utilize coarse-grained experiential memory, where experiences are retrieved based on global task or scene context. While effective in simple settings, such coarse-grained memory lacks the situational alignment required for complex multi-step decision-making. As a result, recalled experiences often fail to match the agent's current state, blurring reasoning focus and leading to inaccurate decisions at critical steps. To this end, we propose **S**tate-**A**ware **memory**(**SAMem**), a new fine-grained memory paradigm for LLM agents that explicitly aligns memory retrieval with the current state. Instead of storing and reusing globally shared experiences, SAMem organizes memory at the level of state-specific reasoning thoughts, enabling the agent to retrieve only the most relevant experience for the current decision context. This state-conditioned memory allows the agent to focus on the most informative reasoning cues at each step, rather than being distracted by task-level but state-misaligned guidance. Extensive experiments on complex decision-making benchmarks demonstrate that SAMem outperforms existing experiential memory approaches, achieving superior performance and substantially improved task-solving efficiency. These results indicate that state-aware, fine-grained memory enhances the decision-making capabilities of LLM agents.

1 Introduction

Memory mechanisms enable large language model (LLM) agents to learn from experiences(Anderson et al., 2018; Dong et al., 2024; Hu et al., 2026; Hatalis et al., 2023), which is key to improving their performance on decision-making tasks(Shridhar et al., 2021; Chang et al., 2024).

Experiential memory(Hu et al., 2026; Zhao et al., 2024) enables LLM agents to store and retrieve past experiences, enhancing decision-making capabilities without the need for costly parameter training(Zhang et al., 2025c; Ouyang et al., 2025). Most existing approaches rely on task-level memory(Wang et al., 2024b; Zhao et al., 2024; Zhou et al., 2025), where experiences are stored as holistic records, retrieved based on task similarity, and reused throughout the entire task execution. While effective in simple settings, such methods become ineffective for complex multi-step decision-making(Zhong et al., 2024; Gao et al., 2025). The main issue is that agents rely on coarse-grained memory, which provides globally shared guidance that fails to align with the current state,

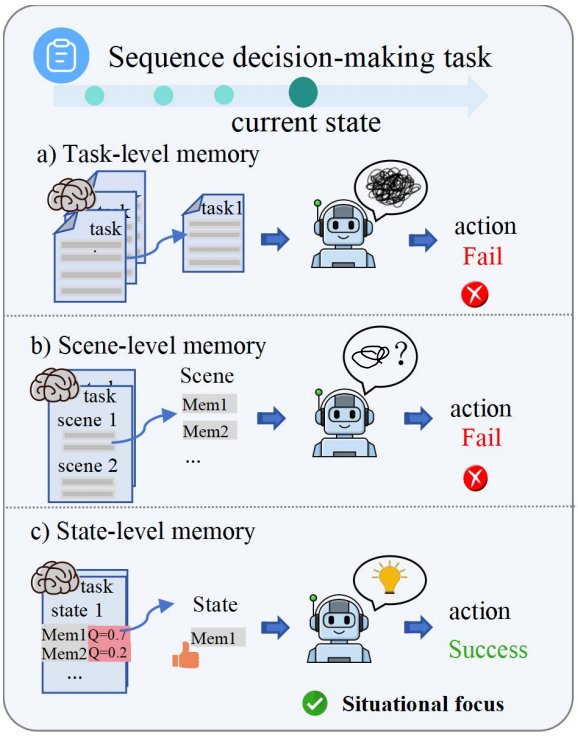

Figure 1: Motivation of SAMem. (a) and (b) Task-level and scene-level memory lack situational granularity, leading to inaccurate decision-making. (c) Our state-level memory features comparatively finer granularity, providing focused and better guidance for reasoning.

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 14691–14710

leading to inaccurate decision-making. While several methods introduce scene-level experience granularity(Gao et al., 2025; Fu et al., 2024) to alleviate the aforementioned problems, they may fail in tasks where clear scene boundaries are not defined or or in scenarios with changing states. This is primarily due to the insufficient granularity of scene-level memory, which can obscure critical decision-making steps.

To address this, we introduce **S**tate-**A**ware **mem**ory(**SAMem**), a new fine-grained memory paradigm for LLM agents that explicitly aligns memory retrieval with the agent's current state. SAMem stores memory as state–thought pairs with associated Q-values that captures the long-term benefit of each thought under a given state. Specifically, the agent explores the environment to generate thoughts and actions, forming state-aware experience trajectories with rewards. These trajectories are then used to update the Q-value of each thought, resulting in a structured Q-table memory composed of value-oriented state–thought pairs. This structured memory enables the agent to retrieve and apply high-value thoughts tailored to the state for decision-making. In addition, we add a forgetting mechanism that actively deletes low value and logically inconsistent entries, thereby streamlining SAMem store.

We evaluate SAMem on three complex multi-step decision-making benchmarks including ALFWorld(Shridhar et al., 2021) , ScienceWorld(Wang et al., 2022), and Jericho(Hausknecht et al., 2020), demonstrating that it outperforms existing baselines. Furthermore, we analyze the task-solving efficiency of our method.

The contribution of this article can be summarized as follows:

1) We propose **SAMem**, a state-aware memory framework that enables fine-grained memory tailored to the current state. It allows the agent to focus on the most informative reasoning direction at each step, rather than being distracted by task-level but state-misaligned guidance.

2) We introduce a state-aware memory update method that refines the Q-value estimates of state-conditioned thoughts in memory. It uses environmental feedback to learn explicit Q-values that evaluate the expected utility of each thought, optimizing memory over time.

3) Experimental results on complex decision-making benchmarks demonstrate the superior performance of SAMem over existing baselines.

2 Preliminaries

Markov Decision Process The interaction process of an LLM agent in decision-making tasks can be formulated as a Markov decision process (MDP). We use MDP to solve the problem in this article. It is defined by the tuple(S, A, P, R, γ), where S and A denote the state and action spaces, $P(s'|s, a)$ is the state transition probability, $R(s, a)$ is the reward function(Sutton and Barto, 1999). The objective is to find a policy $\pi(a|s)$, which defines a probability distribution over actions for each state and maximizes the expected discounted cumulative return $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$.

Q-Learning Our memory update process uses Q-learning. Q-Learning is a reinforcement learning algorithm. It directly estimates the optimal action-value function $Q^*(s, a)$, which represents the maximum expected cumulative reward achievable by following any policy after taking action a in state s. The core of Q-Learning is the iterative update rule:

$$Q\left(s_t, a_t\right) \leftarrow Q\left(s_t, a_t\right) + \alpha\left[r_{t+1} + \gamma \max_{a'} Q\left(s_{t+1}, a'\right) - Q\left(s_t, a_t\right)\right], \quad (1)$$

Where $\alpha \in (0, 1]$ is the learning rate.

3 Method

3.1 Overview

We propose a state-aware structured memory framework that provides LLM agents with situationally focused guidance under the current state. As shown in Figure 2, our framework comprises two key components: SAMem construction and SAMem utilization.

In the SAMem construction stage, a summarization module integrates the accumulated raw observation history into a concise and accurate state. Conditioned on this state, the LLM agent generates a high-level reasoning thought and the corresponding action to interact with the environment, forming coherent state-thought trajectories. From these trajectories, state-thought pairs are incrementally clustered online, and their Q-values are updated based on interaction feedback. This process produces a structured memory as a Q-table, where each entry corresponds to a state-thought pair associated with its learned Q-value. In the SAMem utilization stage, the structured memory is leveraged to select high-value reasoning thoughts conditioned on the current state, guiding the agent's decision-making.

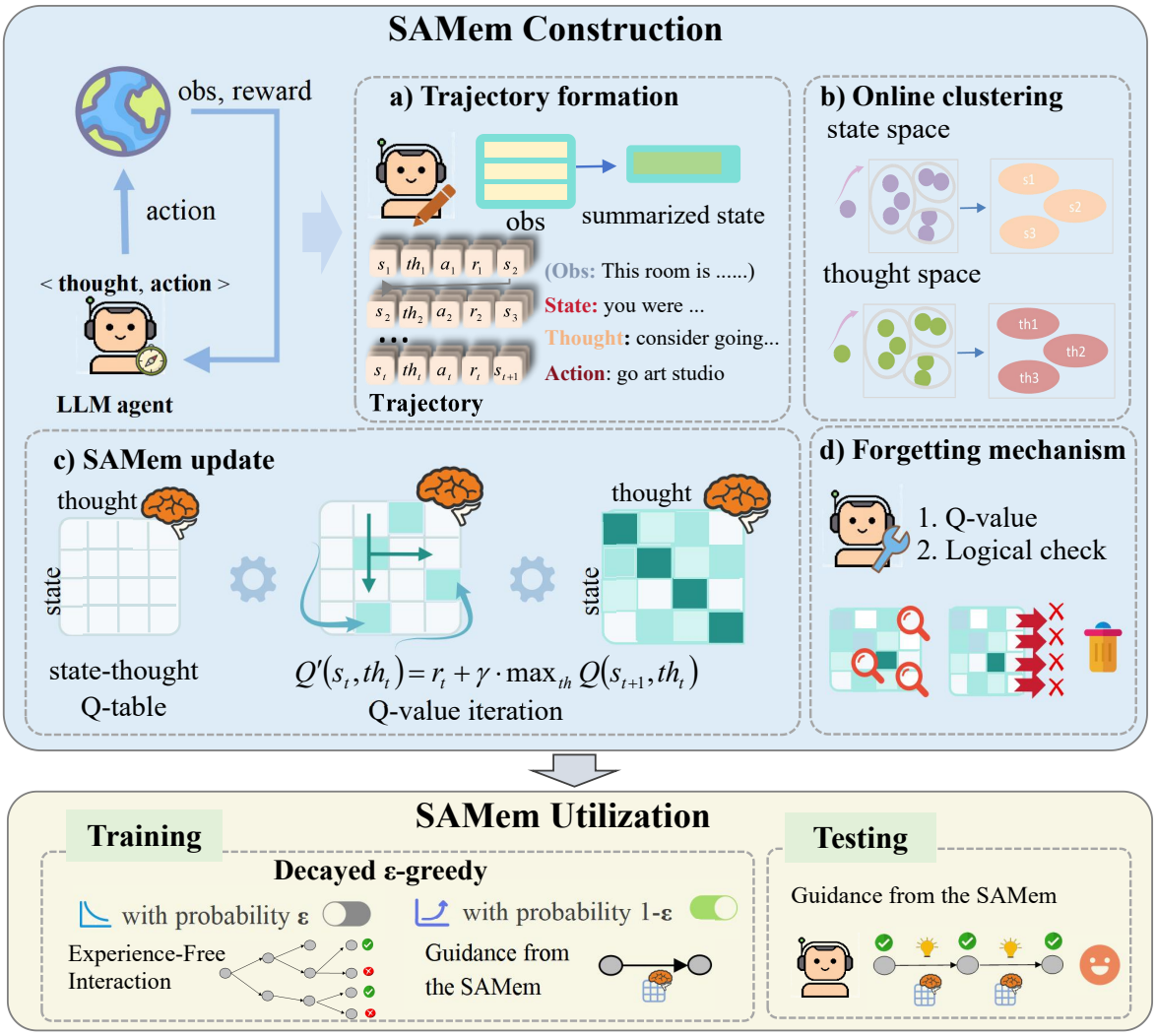

Figure 2: The SAMem framework comprises two parts. The system first constructs a state-aware, value-oriented memory from interaction experiences, and then utilizes this memory to retrieve high-value reasoning thoughts to guide decision-making under the current state.

3.2 SAMem Construction

We construct the SAMem as a table of state–thought pairs, where each entry stores the corresponding Q-value. A state is defined as a situational summary that encapsulates historical and current raw observation at a given step, while a thought represents the reasoning process that leads to a decision within that state. Building the SAMem during training involves three main stages: trajectory formation, online clustering and SAMem update. In addition, we add a forgetting mechanism to remove meaningless units.

Trajectory Formation. Since a single raw observation is often insufficient to fully characterize the agent's current situation, we introduce a situation summarization module that generates concise and accurate state representations. For a given timestep t, given raw observation sequence $Obs_{0:t} = (obs_0, obs_1, obs_2, \ldots, obs_t)$, we prompt an LLM to produce a summary as follows:

$$sum_t \leftarrow LLM_{\text{summay}}\ (Obs_{0:t}) \tag{2}$$

We define sum_t as the state s_t, representing the current state at step t. Our prompt templates are shown in Appendix D.1. Conditioned on the current state s_t, the LLM agent generates a high-level reasoning thought th_t, which serves as the reasoning process behind the corresponding action:

$$th_t \leftarrow LLM\ (s_t) \tag{3}$$

The LLM agent then generates the corresponding action a_t based on reasoning thought th_t. Through interaction with the environment, the agent obtains the reward r_t and the next state

s_{t+1}, which is processed by the summarization module. This cycle forms a continuous trajectory $\tau = \{s_0, th_0, a_0, r_0, s_1, \ldots, s_t, th_t, a_t, r_t, s_{t+1}\}$, which serves as the foundation for subsequent memory update and value estimation.

Online clustering. Each experience tuple (s_t, th_t, a_t, r_t, s_{t+1}) is associated with state and thought entries in the Q-table via incremental online clustering. Specifically, states are encoded into semantic vectors using an embedding model (text-embedding-v3) and matched to existing clusters via cosine similarity. If the maximum similarity exceeds a predefined threshold θ_{sim}, the state is merged into the nearest state cluster; otherwise, a new state entry is created. The same procedure is applied to reasoning thoughts. This dual-level clustering organizes the Q-table around semantically meaningful and reusable units, enabling efficient indexing and improved generalization.

SAMem update. Following online clustering, the mapped experience is then leveraged to iteratively update the SAMem. Instead of optimizing over low-level actions that lack generalizability, our framework operates on high-level reasoning thoughts, facilitating the reuse of higher-order experiential knowledge.

Our framework reframes the decision-making process by treating the reasoning thought th of LLM agent as a high-level action within a semantically enriched action space. We assume that each action is uniquely determined by its corresponding reasoning thought, allowing the underlying MDP to be abstracted as (S, Th, P', R', γ) where $P'(s'|s, th) = P(s'|s, a)$ and $R'(s, th) = R(s, a)$. Under this assumption, the Q-function can be simplified to $Q(s, th)$. The optimal thought th in state s satisfies: $th^* = \arg\max_{th \in Th} Q(s, th)$.

For the mapped state cluster s_t and thought cluster th_t, the Q-value associated with the pair (s_t, th_t) is updated by analogy to the Bellman optimality equation(Bellman, 1952), incorporating the immediate reward and the discounted future value estimated from the next state cluster:

$$Q'(s_t, th_t) = r_t + \gamma \cdot \max_{th} Q(s_{t+1}, th_t), \quad (4)$$

Then, the temporal differential error updates the estimated Q value:

$$Q(s_t, th_t) \leftarrow (1-\alpha) Q(s_t, th_t) + \alpha Q'(s_t, th_t), \quad (5)$$

where α is the learning rate and γ is the discount factor. These Q-values are continuously updated, thereby gradually improving the memory and enabling the agent to select memory content based on the current state over time.

Forgetting mechanism. To prevent the SAMem from becoming bloated with invalid entries, we introduce a forgetting mechanism that performs dual validation based on both estimated Q-value and LLM-guided logical verification. Specifically, SAMem assesses the richness of the thought space for each state recorded in the Q-table. When a state has more than five thoughts with non-zero Q-values, the exploration for that state is considered relatively sufficient. The system then selects the lowest-valued thought and uses the LLM to assess its logical validity, invalid thoughts are softly pruned by setting their Q-values to zero, and empty state rows or thought columns are subsequently removed. This module structurally compresses the memory, reducing storage overhead.

3.3 SAMem Utilization

During training, our agent uses a decayed ϵ-greedy strategy to improve its learning. This mechanism dynamically adjusts the exploration rate ϵ, prompting the agent to explore at the beginning of training and gradually rely on high-quality memory learned over time. This process is expressed as:

$$(th, a) \leftarrow \begin{cases} \text{LLM}(s) & \text{if } \varepsilon_t \\ \text{LLM}\left(s, \arg\max Q(s, th)\right) & \text{if } 1 - \varepsilon_t \end{cases} \quad (6)$$

where the exploration rate ε_t decays with training steps, following an exponential decay strategy:

$$\varepsilon_t = \varepsilon_0 \cdot \exp\left(-\frac{\beta \cdot t}{T}\right), \quad (7)$$

where $\varepsilon_0 = 0.95$, $\beta = 6$. During training, the LLM agent explores with probability ϵ, generating novel trajectories without memory guidance. As ϵ decays, the agent increasingly exploits high-value thoughts from the SAMem. This shift leads to the generation of higher-quality trajectories, which allows the agent to progressively leverage its learned memory to generate more informative trajectories and improve the efficiency and stability of memory updates.

During testing, at each decision-making step, the LLM agent retrieves the thought with the highest Q-value corresponding to the current state from SAMem, and uses it to guide its final output.

	Model	ALFWorld	ScienceWorld		Jericho		Average
		SR	AR	SR	AR	SR	
GPT-4o	baseline	62.7	35.5	22.2	32.9	10.0	32.7
	Reflexion	78.8	43.7	25.6	35.6	10.0	38.7(+18.3%)
	ExpeL	81.3	45.4	32.2	43.6	15.0	43.5(+33.0%)
	AWM	82.3	46.0	28.9	44.1	15.0	43.3(+32.4%)
	AutoGuide	83.3	49.3	26.7	47.8	15.0	44.4(+35.8%)
	CDMem	90.5	52.1	30.0	54.0	20.0	49.3(+50.8%)
	SAMem	98.5	65.4	40.0	63.8	35.0	60.5(+85.0%)
GPT-4o-mini	baseline	37.3	29.7	15.6	26.2	10.0	23.8
	Reflexion	64.9	35.4	13.3	33.1	10.0	31.3(+31.5%)
	ExpeL	70.1	38.4	18.9	37.6	15.0	36.0(+51.3%)
	AWM	69.4	37.3	18.9	39.2	15.0	36.0(+51.3%)
	AutoGuide	74.1	40.8	20.0	40.6	15.0	38.1(+60.0%)
	CDMem	79.6	46.0	23.3	44.8	20.0	42.7(+79.4%)
	SAMem	82.6	59.7	34.4	50.4	30.0	51.4(+116.0%)
Qwen-2.5-72b -instruct	baseline	51.5	31.6	12.2	29.6	10.0	27.0
	Reflexion	67.2	40.5	14.4	34.3	10.0	33.3(+23.3%)
	ExpeL	72.1	42.8	17.8	42.2	15.0	38.0(+40.7%)
	AWM	73.1	44.7	17.8	41.1	15.0	38.3(+41.9%)
	AutoGuide	76.9	47.3	21.1	42.4	15.0	40.5(+50.0%)
	CDMem	80.6	49.9	22.2	46.7	20.0	43.9(+62.6%)
	SAMem	85.1	61.0	28.9	52.5	30.0	51.5(+90.7%)
Llama-3.1-70b -instruct	baseline	52.2	26.9	10.0	27.6	10.0	25.3
	Reflexion	68.7	35.1	13.3	33.3	10.0	32.1(+26.9%)
	ExpeL	73.9	39.8	17.8	38.6	15.0	37.0(+46.2%)
	AWM	74.6	40.3	16.7	38.5	15.0	37.0(+46.2%)
	AutoGuide	78.1	42.6	17.8	39.2	15.0	38.5(+52.2%)
	CDMem	81.3	47.8	21.1	41.0	20.0	42.2(+66.8%)
	SAMem	88.8	57.7	23.3	48.3	25.0	48.6(+92.1%)

Table 1: Performance comparison of different models on ALFWorld, ScienceWorld and Jericho. SR(%) and AR denote success ratio(%) and average reward(score), respectively. Results are averaged from three runs. We run experiments on GPT-4o, GPT-4o-mini, Qwen-2.5-72b and Llama-3.1-70b.

4 Experiment

4.1 Setting

We conduct experiments in complex decision-making benchmarks: 1) ALFWorld(Shridhar et al., 2021)[1] focus on embodied reasoning for daily household tasks. We select five tasks from each task type in the training set of ALFWorld during training. We evaluate our method on an unseen test set comprising 134 tasks across all six types. 2) ScienceWorld(Sciworld)(Wang et al., 2022)[2] encompasses a diverse set of tasks, ranging from short-term to long-term. Our experiment covers all 30 types of tasks. We select three tasks per type for training. The evaluation uses a unseen test set of 90 tasks, formed by selecting three variants for each task type. 3) Jericho(Hausknecht et al., 2020) evaluates LLM agents' sequential decisions in classic interactive fiction games. The test set for Jericho are based on AgentBoard(Chang et al., 2024) [3]. Jericho is used as an additional environment to test the online evolving learning capability of our memory framework. Details can be found in Appendix B.1.

4.1.1 Baselines

To fully demonstrate the effectiveness of our method, we compared SAMem with the following methods: Reflexion(Shinn et al., 2023); ExpeL(Zhao et al., 2024); AWM(Wang et al., 2024b); AutoGuide(Fu et al., 2024); CDMem(Gao et al., 2025). Details of baselines can be found in Appendix B.2.

4.1.2 Implementation

When used for indexing, we employ text-embedding-v3 with the default embedding dimension of 1024. For the situation summarization module, we use GPT-4o. The memory construction process for all algorithms is performed over 9 iterations on the training set. For evaluation, ALFWorld evaluates performance based on success rate, while ScienceWorld and Jerich based on both success rate and average reward (score). In addition to compare the efficiency of different methods, we add a Suc-

[1]Datas available at https://alfworld.github.io/
[2]https://github.com/allenai/ScienceWorld
[3]https://github.com/hkust-nlp/AgentBoard

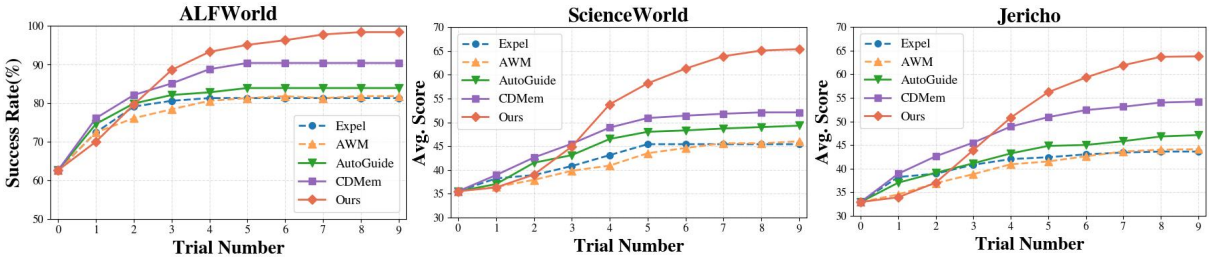

Figure 3: Visualization of agent performance over training iterations on ALFWorld, ScienceWorld, and Jericho.

cess weighted by Path Length (SPL) (Anderson et al., 2018; Duan et al., 2022) to assess the efficiency of task-solving. SPL is calculated using the following formula:

$$R_{SPL} = \frac{1}{N}\sum_{i=1}^{N} Success_i \frac{L_i}{\max(P_i, L_i)} \quad (8)$$

where N is the number of tasks; $Success_i$ is the success or failure of task i (1 represents success); P_i represents the actual path length of task i; An SPL value closer to 1 indicates that the sequence of decision steps is nearer to the optimal path. Details can be found in Appendix B.3.

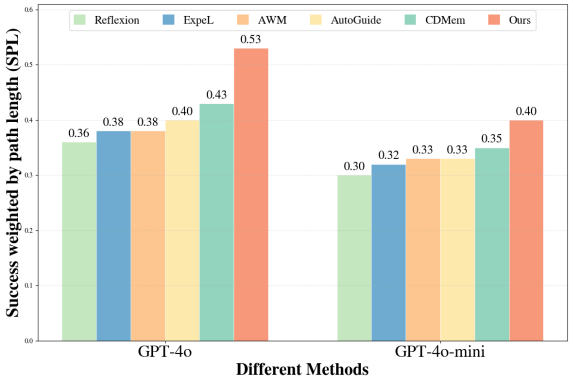

Figure 4: SPL Performance of different methods. This figure shows the SPL values of different methods under different LLMs evaluated on ALFWorld. Our method consistently outperforms the baselines in task efficiency.

4.2 Results

Main Results with Different LLMs. As shown in Table 1, SAMem significantly outperforms the baseline methods across ALFWorld, ScienceWorld, and Jericho. ExpeL(Zhao et al., 2024) and AWM(Wang et al., 2024b) retrieve workflows or past experiences based on task but lack contextual guidance, which can result in irrelevant or confusing suggestions. AutoGuide(Fu et al., 2024) and CDMem(Gao et al., 2025) provide scene-level guidance, which yields better performance than previous methods. However, the granularity remains relatively coarse, leading to incorrect decision-making. As demonstrated by the case study presented in Appendix C.2, our method provides the agent with state-level guidance derived from the high-value thought, allowing the agent to focus on the most informative reasoning cues at each step. This experiment demonstrates that transitioning from task-level to scene-level, and then to state-level memory, enhances decision-making with finer-grained memory.

Analysis on Efficiency. The agent's efficiency is measured by its ability to maximize rewards while minimizing action steps. We compare the SPL of different methods on ALFWorld, as depicted in Figure 4. Compared with other methods, the SPL value of our method is the highest, with GPT-4o and GPT-4o-mini reaching 0.53 and 0.40, respectively. Details of the SPL values for different tasks can be found in Appendix B.3. On ScienceWorld, following similar analysis (Song et al., 2024), we compare the score trajectories of different methods, as shown in the Figure 5. Compared to AutoGuide and CDMem, SAMem can achieve higher scores with fewer action steps. This suggests that this finer-grained memory can enhance the task-solving efficiency of agents.

Analysis of the Iterative Process. Figure 3 illustrates the progression of training process. In the initial stage, SAMem explores diverse trajectories, and its success rate is slightly lower than those of other methods. As the memory is progressively optimized, the performance of SAMem improves accordingly. The baseline method improves quickly initially, but as iterations continue, its experience becomes entrenched, hindering further progress. This indicates that coarse-grained memory tends to become rigid after a certain point, limiting further improvement despite continued updates.

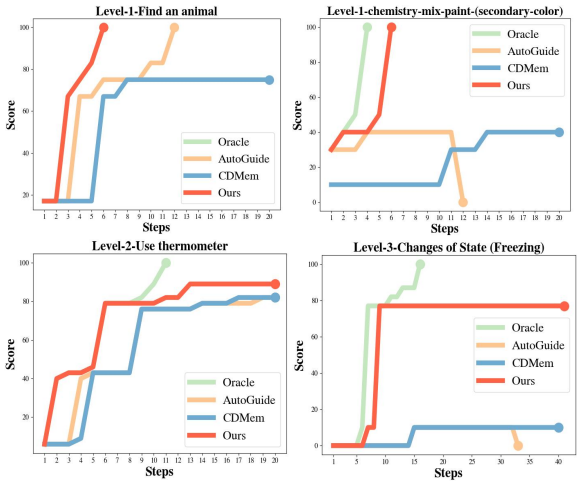

Figure 5: Cases of ScienceWorld score trajectory for different methods. X-axis: number of steps; Y-axis: score. Compared to AutoGuide and CDMem, Our method can achieve higher scores with fewer action steps. Our method outperforms the baselines in task efficiency.

Table 2: Ablation on key components on GPT-4o. **w/o Q-learning** denotes removing Q-value iteration. **w/o Situation processing** denotes removing summarization and clustering. **w/o Decayed ϵ-greedy** means that memory is not utilized during training. ALFWorld uses success rate(%) whereas the others use average reward.

Model	ALFWorld	SciWorld	Jericho
SAMem(Full)	98.5	65.4	63.8
w/o Q-learning	84.3	47.4	44.6
w/o Situation processing	82.8	48.5	50.4
w/o Decayed ϵ-greedy	94.0	62.4	57.8

4.3 Ablation Studies

Component Ablations. We evaluate the impact of several modules on SAMem, as shown in the Table 2. The removal of the Q-learning module leads to a substantial decline in performance. This directly demonstrates that introducing Q-value evaluation mechanism is essential for our memory update. Removing the situation processing, which reduces the state to the raw current observation without summarization and clustering, also causes noticeable performance degradation, highlighting the crucial role of situation processing in the system. Removing the decayed ϵ-greedy strategy causes a performance decline, which indicates that it serves as a contributing factor. The ablation study highlights the critical importance of these modules for effective state-level memory.

Forgetting Ablation. We analyze the effect of ablating the forgetting mechanism on the size of SAMem. We conducted same iterations of training using GPT-4o. As shown in the Figure 6, the memory size of the model with the forgetting mechanism is significantly smaller than that of the model without it, which intuitively demonstrates the positive impact of the forgetting mechanism in controlling memory inflation.

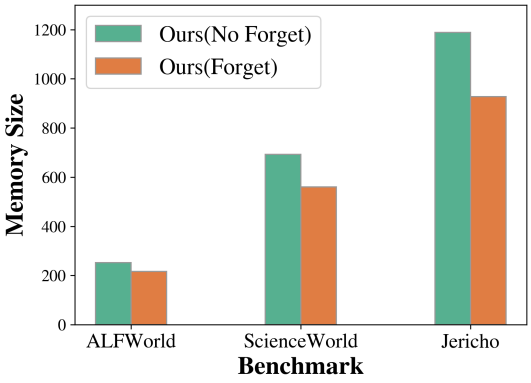

Figure 6: The effect of the forgetting mechanism on SAMem size.

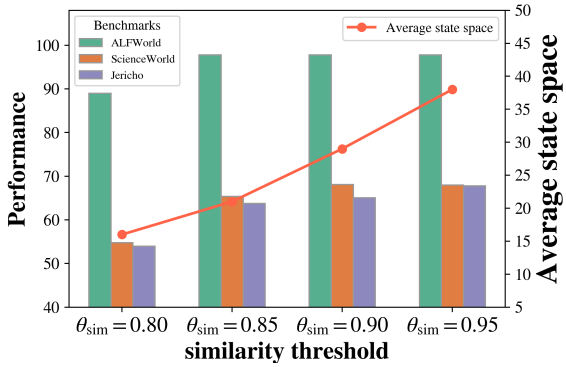

Figure 7: The effect of similarity threshold θ_{sim} on performance and state space size. The bar chart represents the performance across different benchmarks, while the red line indicates the average state space.

4.4 More Analysis

Similarity Threshold Analysis. To study the impact of similarity threshold θ_{sim}, we test different θ_{sim} and analyze how they affect the experimental results and the size of the state space in the SAMem. Average state space refers to the mean state space size averaged across the three environments. Figure 7 shows that the similarity threshold strikes a trade-off: higher values improve clustering accuracy and performance, but also expand the state space and may reduce retrieval efficiency. To balance these factors, we typically set the similarity threshold to an intermediate value of 0.85. The analysis on thoughts is in Appendix C.1.

Action-Level Optimization vs. Thought-Level

Table 3: Success rate(%) of action-level and thought-level methods on ALFWorld unseen test set.

Model	$n = 3$	$n = 5$	$n = 7$	$n = 9$
Action-level	70.1	72.4	76.1	79.9
Thought-level	**89.6**	**95.5**	**97.7**	**97.7**

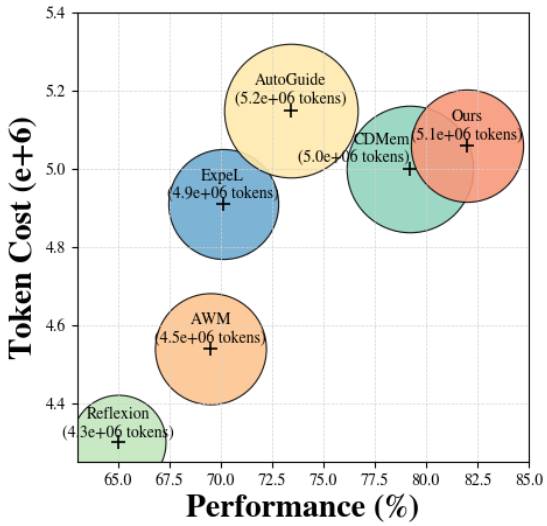

Figure 8: Cost analysis of SAMem. SAMem achieves a decisive performance advantage in exchange for an acceptable increase in cost.

Optimization. We compare SAMem's action-level optimization paradigm with the thought-level method using GPT-4o on ALFWorld, training on n tasks per type for five iterations each. The results are shown in the Table 3. The action-level approach exhibits limited generalizability, as merely providing action instructions fails to convey the underlying rationale, hindering adaptation to similar state. In contrast, the thought-level method emphasizes reasoning patterns, which more effectively conveys knowledge and achieves strong performance with minimal training.

Computational Cost Analysis. We conduct a cost analysis of token consumption for different methods on ALFWorld based on the GPT-4o-mini, as shown in the Figure 8. Each method is run for 10 trials. Although our method introduces additional token overhead due to per-step situation summarization, the overall token consumption remains at a manageable level. In contrast, scene-level methods like AutoGuide rely on the LLM for repeated comparisons and reflections during experience updates, resulting in higher token consumption. Our method maintains an acceptable token cost while achieving superior performance.

5 Related Work

Memory for LLM agent. Memory mechanisms are crucial for enhancing the capabilities of LLM agents(Zhang et al., 2025c; Hu et al., 2026; Wang et al., 2023; Yin et al., 2024; Chen et al., 2023; Liu et al., 2023). Some works focus on the context window limitations and memory management in conversational LLM systems(Hu et al., 2023), such as Mem0(Chhikara et al., 2025), MemoryBank(Zhong et al., 2024), MemoChat(Lu et al., 2023). Some memory mechanisms are designed to address the needs of task-solving for decision-making LLM agents(Zheng et al., 2023; Zhao et al., 2024; Wang et al., 2024a), primarily addressing experience extraction and reuse, such as Memento(Zhou et al., 2025), G-Memory(Zhang et al., 2025a). Our memory work focuses on supporting LLM agents in multi-step decision-making tasks.

Experiential Memory for decison-making. Experiential memory for LLM agents allows the integration of past interactions and experiences, enhancing their decision-making capabilities(Feng et al., 2025; Tan et al., 2025; Zhang et al., 2025b; Yang et al., 2025). Some memory systems have focused on task-level memory, where experiences are retrieved based on task similarity, such as AWM(Wang et al., 2024b), ExpeL(Zhao et al., 2024). Some studies focus on scene-level methods(Fu et al., 2024). Methods like ReMe(Cao et al., 2025) and CDMem(Gao et al., 2025) have demonstrated the utility of scene-level guidance, where memories are indexed according to broader situational cues. However, these approaches still suffer from limitations in granularity, which can lead to less relevant or overly general guidance in complex multi-step decision-making. In contrast, our work focuses on a significantly finer-grained memory.

6 Conclusion

In this research, we propose SAMem, a state-aware memory framework that provides fine-grained guidance tailored to the agent's current state. SAMem stores state-thought pairs alongside their corresponding Q-values, which represent the expected long-term utility of each reasoning thought in a given state. The agent can refine its memory, ensuring that high-value thoughts are prioritized while low-value and unreasonable thought is pruned. This structured memory allows the agent to retrieve and apply the high-value thoughts for decision-making, leading to improved accuracy and effec-

tiveness over time. Experimental results on three complex reasoning environments demonstrate the superior performance of SAMem over existing baselines. Additionally, it achieves higher task-solving efficiency compared to the baselines.

Limitations

Despite its effectiveness in decision-making tasks, SAMem has several limitations. First, our method currently relies on tabular Q-learning. Although situation summarization and online clustering enable effective abstraction and keep the state/thought spaces tractable in our experiments, the Q-table may not scale to domains with higher state diversity and more complex reasoning patterns. Future work will explore function approximation methods, such as neural networks, to replace the Q-table, which may have additional data-related challenges. Second, in environments with sparse or zero rewards, Q-value iteration may becomes more challenging and unstable. Addressing these limitations and extending our framework to broader application domains remain important directions for future exploration.

Ethical Statement

All datasets used in this work are publicly available. Furthermore, we adhere to the core tenets of transparency in LLM agents' decision logic. Their design prioritizes harm avoidance, algorithmic bias mitigation, and safety. Ultimately, all decisions must align with human values and societal well-being.

Acknowledgments

This work was supported by the National Science and Technology Major Project (Grant No.2022ZD0116403) and Beijing Natural Science Foundation (Grant No.4264131).

References

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. 2018. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*.

Richard Bellman. 1952. On the theory of dynamic programming. *Proceedings of the national Academy of Sciences*, 38(8):716–719.

Zouying Cao, Jiaji Deng, Li Yu, Weikang Zhou, Zhaoyang Liu, Bolin Ding, and Hai Zhao. 2025. Remember me, refine me: A dynamic procedural memory framework for experience-driven agent evolution. *arXiv preprint arXiv:2512.10696*.

Ma Chang, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. 2024. Agentboard: An analytical evaluation board of multi-turn llm agents. *Advances in neural information processing systems*, 37:74325–74362.

Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. 2023. Fireact: Toward language agent fine-tuning. *arXiv preprint arXiv:2310.05915*.

Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. 2021. History aware multimodal transformer for vision-and-language navigation. *Advances in neural information processing systems*, 34:5834–5847.

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

Xiaofei Dong, Xueqiang Zhang, Weixin Bu, Dan Zhang, and Feng Cao. 2024. A survey of llm-based agents: Theories, technologies, applications and suggestions. In *2024 3rd International Conference on Artificial Intelligence, Internet of Things and Cloud Computing Technology (AIoTC)*, pages 407–413. IEEE.

Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. 2022. A survey of embodied ai: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(2):230–244.

Erhu Feng, Wenbo Zhou, Zibin Liu, Le Chen, Yunpeng Dong, Cheng Zhang, Yisheng Zhao, Dong Du, Zhichao Hua, Yubin Xia, and Haibo Chen. 2025. Get experience from practice: Llm agents with record & replay. *arXiv preprint arXiv:2505.17716*.

Yao Fu, Dong-Ki Kim, Jaekyeom Kim, Sungryull Sohn, Lajanugen Logeswaran, Kyunghoon Bae, and Honglak Lee. 2024. Autoguide: Automated generation and selection of context-aware guidelines for large language model agents. *Advances in Neural Information Processing Systems*, 37:119919–119948.

Pengyu Gao, Jinming Zhao, Xinyue Chen, and Long Yilin. 2025. An efficient context-dependent memory framework for llm-centric agents. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pages 1055–1069.

Theophile Gervet, Soumith Chintala, Dhruv Batra, Jitendra Malik, and Devendra Singh Chaplot. 2023. Navigating to objects in the real world. *Science Robotics*, 8(79):eadf6991.

Kostas Hatalis, Despina Christou, Joshua Myers, Steven Jones, Keith Lambert, Adam Amos-Binks, Zohreh Dannenhauer, and Dustin Dannenhauer. 2023. Memory matters: The need to improve long-term memory in llm-agents. In *Proceedings of the AAAI Symposium Series*, volume 2, pages 277–280.

Matthew Hausknecht, Prithviraj Ammanabrolu, Marc-Alexandre Côté, and Xingdi Yuan. 2020. Interactive fiction games: A colossal adventure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7903–7910.

Chenxu Hu, Jie Fu, Chenzhuang Du, Simian Luo, Junbo Zhao, and Hang Zhao. 2023. Chatdb: Augmenting llms with databases as their symbolic memory. *arXiv preprint arXiv:2306.03901*.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, Senjie Jin, Jiejun Tan, Yanbin Yin, Jiongnan Liu, Zeyu Zhang, Zhongxiang Sun, Yutao Zhu, Hao Sun, Boci Peng, and 28 others. 2026. Memory in the age of ai agents. *arXiv preprint arXiv:2512.13564*.

Lei Liu, Xiaoyan Yang, Yue Shen, Binbin Hu, Zhiqiang Zhang, Jinjie Gu, and Guannan Zhang. 2023. Think-in-memory: Recalling and post-thinking enable llms with long-term memory. *arXiv preprint arXiv:2311.08719*.

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

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. 2025. Reasoningbank: Scaling agent self-evolving with reasoning memory. *arXiv preprint arXiv:2509.25140*.

Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. 2024. Trial and error: Exploration-based trajectory optimization of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7584–7600.

Richard S Sutton and Andrew G Barto. 1999. Reinforcement learning. *Journal of Cognitive Neuroscience*, 11(1):126–134.

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. 2025. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8416–8439, Vienna, Austria. Association for Computational Linguistics.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. ScienceWorld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*, pages 11279–11298, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. 2023. Augmenting language models with long-term memory. *Advances in Neural Information Processing Systems*, 36:74530–74543.

Zihao Wang, Shaofei Cai, Anji Liu, Yonggang Jin, Jinbing Hou, Bowei Zhang, Haowei Lin, Zhaofeng He, Zilong Zheng, Yaodong Yang, Xiaojian Ma, and Yitao Liang. 2024a. Jarvis-1: Open-world multi-task agents with memory-augmented multimodal language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3):1894–1907.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. 2024b. Agent workflow memory. *arXiv preprint arXiv:2409.07429*.

Cheng Yang, Xuemeng Yang, Licheng Wen, Daocheng Fu, Jianbiao Mei, Rong Wu, Pinlong Cai, Yufan Shen, Nianchen Deng, Botian Shi, Yu Qiao, and Haifeng Li. 2025. Learning on the job: An experience-driven self-evolving agent for long-horizon tasks. *arXiv preprint arXiv:2510.08002*.

Da Yin, Faeze Brahman, Abhilasha Ravichander, Khyathi Chandu, Kai-Wei Chang, Yejin Choi, and Bill Yuchen Lin. 2024. Agent lumos: Unified and modular training for open-source language agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12380–12403.

Guibin Zhang, Muxin Fu, Guancheng Wan, Miao Yu, Kun Wang, and Shuicheng Yan. 2025a. G-memory: Tracing hierarchical memory for multi-agent systems. *arXiv preprint arXiv:2506.07398*.

Kai Zhang, Xiangchao Chen, Bo Liu, Tianci Xue, Zeyi Liao, Zhihan Liu, Xiyao Wang, Yuting Ning, Zhaorun Chen, Xiaohan Fu, Jian Xie, Yuxuan Sun, Boyu Gou, Qi Qi, Zihang Meng, Jianwei Yang, Ning Zhang, Xian Li, Ashish Shah, and 11 others. 2025b.

Agent learning via early experience. *arXiv preprint arXiv:2510.08558*.

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

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19632–19642.

Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. 2024. Towards learning a generalist model for embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13624–13634.

Longtao Zheng, Rundong Wang, Xinrun Wang, and Bo An. 2023. Synapse: Trajectory-as-exemplar prompting with memory for computer control. *arXiv preprint arXiv:2306.07863*.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731.

Huichi Zhou, Yihang Chen, Siyuan Guo, Xue Yan, Kin Hei Lee, Zihan Wang, Ka Yiu Lee, Guchun Zhang, Kun Shao, Linyi Yang, and Jun Wang. 2025. Memento: Fine-tuning llm agents without fine-tuning llms. *arXiv preprint arXiv:2508.16153*.

A Detailed Descriptions of SAMem

State Details State is a situational summary that encapsulates both historical and current context. It distills a sequence of raw observations into a coherent, succinct representation. For example, in ALFWorld, the state at a given step might be: *You have found the target object. You have picked up the target object*. In ScienceWorld, the state might be: *You move the thermometer to the inventory. You move the metal pot to the inventory. You move the metal pot to the sink.* In Jericho, the state might be: *The ocean water is a bit deep, but with the snorkel on you can breathe. The shore is to the south. The island looms ahead to the north.*

It is important to note that Jericho, being a story-driven adventure game, often generates lengthy raw historical observations. The raw current observation typically provides sufficient information to represent the present situation. Therefore, in the Jericho, our summarization is focused exclusively on summarizing the raw current observation.

Thought Details Thought represents the reasoning process underlying decision-making in a situation. It is worth noting that in order to prevent over specialization of thoughts, we impose semantic constraints in the system prompts: all expressions are de-indexicalized, and any reference to specific entities should be avoided, thereby ensuring the transferability of thought templates. SAMem stores state–thought pairs together with their corresponding Q-values, which estimate the long-term value of generated thoughts.

Clustering Details We implemented an online similarity-based clustering mechanism. Taking state clustering as an example, we use a FAISS vector store to maintain state. When a new state is encountered, we compute the cosine similarity between its vector representation and all existing state vectors in the Q-table. If the maximum similarity exceeds a predefined threshold (0.85), the state is assigned to the most similar existing state cluster; otherwise, it is initialized as a new entry in the Q-table. The same clustering procedure is applied to the thought space.

The pseudocode of SAMem update process is shown in Algorithm 1.

B Evaluation Details

B.1 Environment Details

ALFWorld(Shridhar et al., 2021): ALFWorld is a text-based interactive simulator designed for embodied reasoning (licensed under the MIT License). To demonstrate the generalization of our method, we train on the training task set and evaluate on an unseen testing set. During training, we selected five tasks from each task type in the ALFWorld training set. During testing, we use a publicly available test set, which consists of 134 tasks covering 6 types.

ScienceWorld(Wang et al., 2022): ScienceWorld is a text-based interactive simulation environment for scientific knowledge and procedural reasoning (licensed under the Apache-2.0 license). Due to the variety of task types, we divide the 30 task types into three different difficulty levels based on the average number of steps(Gao et al., 2025), as shown in Figure 9. Tasks that can be completed within 20 steps are defined as Level-1, with a total of 10 types of tasks and the max steps set to 20. Those requiring between 20 and 70 steps are classified as Level-2, also with a total of 10 types of tasks and a maximum of 20 steps. Tasks needing 70 or more steps are designated as Level-3, for which the

Algorithm 1 SAMem Update

Initialize Q-table $Q(\cdot, \cdot) = 0$
Summarization module LLM_{summary}
Trial number H
Number of training tasks N
Initialize exploration rate ϵ
for $h \leftarrow 0$ **to** H **do**
  for task $n \leftarrow 0$ **to** N **do**
    $s \leftarrow LLM_{\text{summary}}(Obs_{0:t})$
    Sample $u \sim \text{Uniform}(0, 1)$
    if $u < \epsilon$ **then**
      $th, a \leftarrow LLM(s)$
    else
      $th^* \leftarrow \arg\max_{th} Q(s, th)$
      $th, a \leftarrow LLM(s, th^*)$
    end if
    $obs', r, done \leftarrow \text{env.step}(a)$
    $s' \leftarrow LLM_{\text{summary}}(Obs_{0:t+1})$
    Collect transition tuple (s, th, a, r, s')
    Associate the experience with existing memory entries via clustering
    $s \leftarrow \text{ClusterMatch}(s)$
    $th \leftarrow \text{ClusterMatch}(th)$
    $s' \leftarrow \text{ClusterMatch}(s')$
    Perform the following Q-learning update rule for this transition:
    $Q' \leftarrow r + \gamma \max_{th} Q(s', th)$
    $Q(s, th) \leftarrow (1 - \alpha)\, Q(s, th) + \alpha\, Q'$
    Decay ϵ
    if done **then**
      break
    end if
  end for
end for
return $Q(\cdot, \cdot)$

max steps is set to 40. During traing, we randomly select three tasks per task type. The evaluation is then performed on a unseen test set of 90 tasks that consists of the top 3 variants of each task type.

Jericho(Hausknecht et al., 2020): Jericho is a benchmark for evaluating sequential decision-making by LLM agents in classic interactive fiction games. We follow the AgentBoard(Chang et al., 2024) task set. The task set comprises 20 tasks, each mapped to a unique game, including 905, Acorncourt, Afflicted, Balances, Dragon, Jewel, Library, Omniquest, Reverb, Snacktime, Zenon, Zork1–3, Detective, Night, Pentari, Weapon, Darkhunt, and Loose. Jericho is used as an additional environment to test the online evolving learning capability of our memory framework. We train and evaluate our method on this task set.

B.2 Baseline

1) **Reflexion**(Shinn et al., 2023) is one of the earliest memory mechanisms, which enhances working memory by transforming failure cases into structured reflections. 2) **ExpeL**(Zhao et al., 2024) autonomously gathers success/failure trajectories from training tasks, extracts general insights and retrieves similar successful experiences via task similarity, then leverages them to enhance decision-making in unseen tasks during inference with a single attempt. ExpeL is a typical task-level method, where experiences are retrieved and utilized based on the overall task context. 3) **AWM**(Wang et al., 2024b) induces reusable, abstract workflows for the task from trajectories and integrates them into agent memory to guide decision-making. 4) **AutoGuide**(Fu et al., 2024) is a framework that automatically extracts trajectory snippet-aware guidelines from contrastive trajectories. This approach identifies the divergence point between successful and suboptimal trajectories, encodes the shared prior context into natural language, and extracts actionable decision rules from their contrasting actions. AutoGuide is a scene-level method. 5) **CDMem**(Gao et al., 2025) is an efficient online memory framework designed for LLM-centric agents. It employs a multi-stage encoding process (expert, short-term, and long-term) to extract structured knowledge, while integrating awareness of both environmental and task contexts for effective memory storage and retrieval. CDMem is also a scene-level method, relying on the environment for experience retrieval and reuse.

B.3 Implementation Details of SPL Value Calculation

To compare the efficiency of different methods, we evaluated the SPL values of different methods on ALFWorld. SPL is a widely recognized and authoritative metric for evaluating the efficiency of robot navigation algorithms (Anderson et al., 2018; Gervet et al., 2023), commonly applied in open environments such as home service robotics (Chen et al., 2021; Zheng et al., 2024; Gervet et al., 2023) and rescue operations. It accounts not only for the success rate but also for the path length. In the process of calculating SPL, it is necessary to know the oracle step of the task, which is the shortest

number of action steps required to complete the task. We set the optimal number of steps based on the actual task on ALFWorld, as shown in Table 4.

Table 4: Optimal number of steps on ALFWorld

Task type	Optimal Number of Steps
Put	4
Clean	6
Heat	7
Cool	6
Examine	4
Puttwo	8

Whereas the prior section illustrate the data with a bar chart, this section provides a more detailed, value-specific breakdown for each task type. We calculated the SPL values of different methods in different tasks, as shown in Table 5.

Table 5: Comparison of SPL values for different methods on ALFWorld.

Method	Put	Clean	Heat	Cool	Exa.	Puttwo	ALL
GPT-4o							
Reflexion	0.47	0.36	0.33	0.32	0.27	0.38	0.36
ExpeL	0.40	0.37	0.36	0.36	0.40	0.39	0.38
AWM	0.40	0.39	0.37	0.34	0.39	0.40	0.38
AutoGuide	0.36	0.41	0.38	0.38	0.42	0.45	0.40
CDMem	0.52	0.43	0.38	0.39	0.45	0.42	0.43
SAMem	0.54	0.50	0.43	0.45	0.69	0.57	0.53
GPT-4o-mini							
Reflexion	0.35	0.31	0.34	0.20	0.34	0.28	0.30
ExpeL	0.37	0.30	0.35	0.26	0.32	0.30	0.32
AWM	0.37	0.30	0.36	0.28	0.32	0.32	0.33
AutoGuide	0.38	0.30	0.36	0.29	0.33	0.31	0.33
CDMem	0.39	0.32	0.38	0.34	0.34	0.33	0.35
SAMem	0.41	0.36	0.44	0.46	0.37	0.36	0.40

C Further Anaylsis

C.1 Thought Similarity Threshold Analysis

We evaluate different similarity thresholds θ_{sim} and analyze their impact on both performance and the size of the thought space in SAMem. The average thought space size is computed across the three environments. As shown in Figure 9, low similarity thresholds lead to coarse thought representations, resulting in inaccurate clustering and degraded performance. Increasing the threshold improves clustering precision, but at the cost of an expanded thought space, which indirectly reduces retrieval efficiency. To balance performance and thought space size, we adopt an intermediate similarity threshold of 0.85 in all experiments.

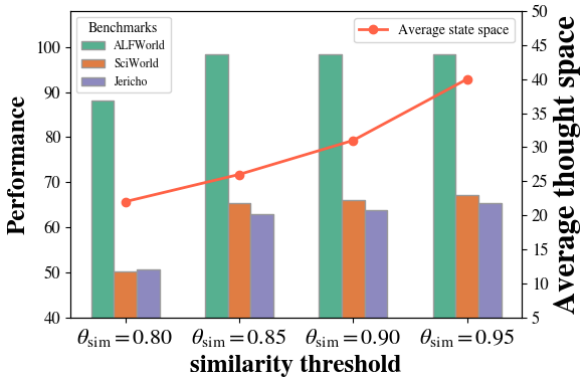

Figure 9: The effect of similarity threshold θ_{sim} on performance and thought space size.

C.2 Case study

To order to provide deeper insights into the effectiveness of SAMem, we conduct a case study of its decision-making process. As illustrated in Figure 10, we present an example from a randomly selected cool task in the ALFWorld test set. Expel(Zhao et al., 2024) uses task-level experience, with all guidelines available at each timestep. This abundance of information, which may not always align with the current situation, can lead to incorrect decisions. CDMem(Gao et al., 2025) uses scene-level memory, retrieving information based on the environment. Its coarse granularity still limits decision-making effectiveness. In contrast, SAMem use state-level memory, obtaining high-value guidelines that match the current state, which leads to superior decision-making performance.

D Implement Prompts

D.1 Summarization pormpt

Figure 12 illustrates the prompts employed in our summarization module. These prompts are designed to extract and condense key information from historical observations and current raw observation.

D.2 Interaction prompt for agent

Figure 13, 14, 15 show the interaction prompts employed by the agent across three distinct environments: ALFWorld, ScienceWorld, and Jericho.

D.3 logical check prompt

Figure 16 presents the prompt used for the logical check within the forgetting module. The core function of this prompt is to verify the logical consistency between the thought and the state.

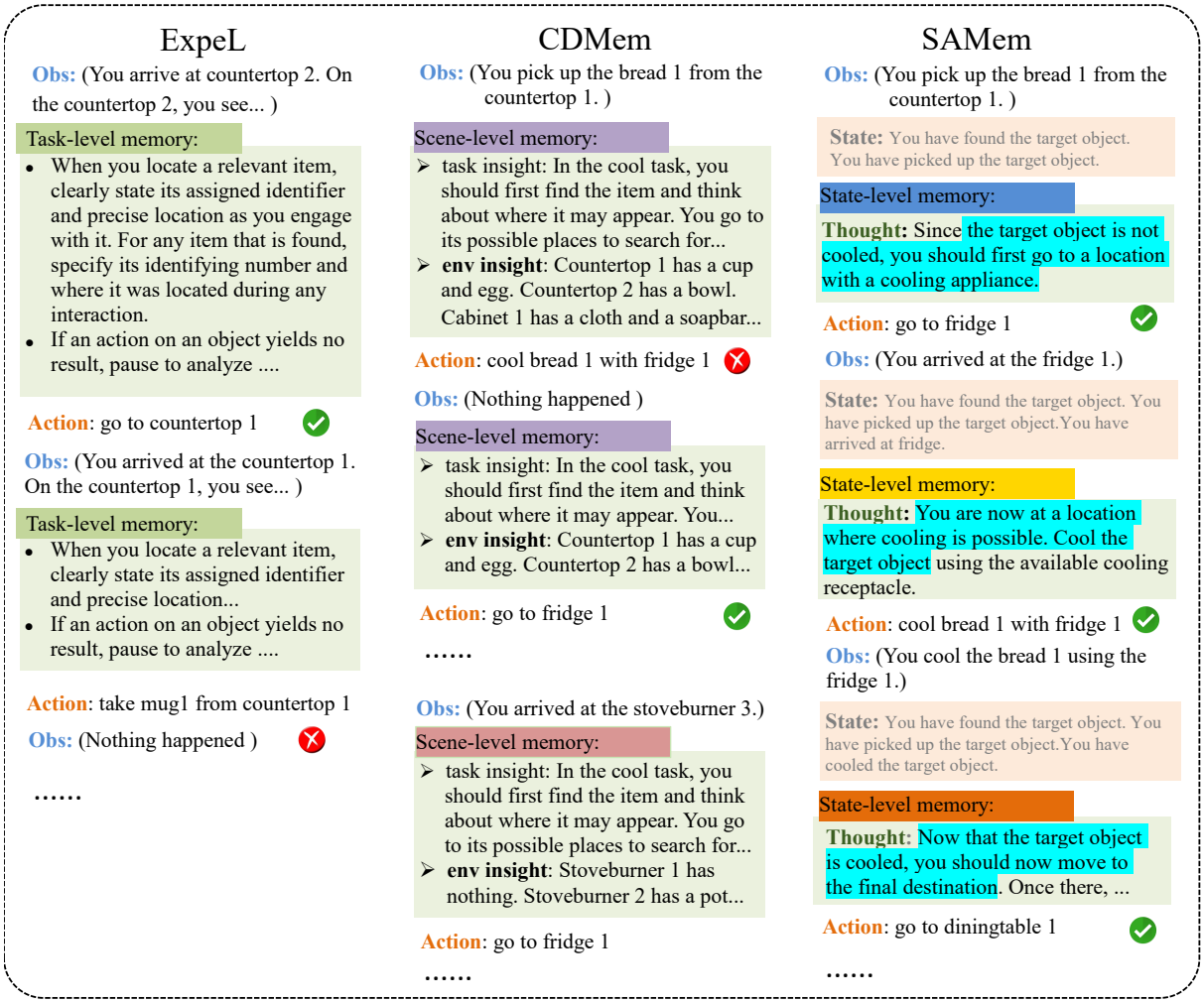

Figure 10: Case study.

Task Type	Topic	Name	Lens	Vars	Level
1-1	Matter	Changes of State (Boiling)	107.7	30	★★★
1-1	Matter	Changes of State (Melting)	78.6	30	★★★
1-1	Matter	Changes of State (Freezing)	88.9	30	★★★
1-1	Matter	Changes of State (Any)	75.2	30	★★★
2-1	Measurement	Use Thermometer	21.4	540	★★
2-2	Measurement	Measuring Boiling Point (known)	35.2	436	★★
2-3	Measurement	Measuring Boiling Point (unknown)	65	300	★★
3-1	Electricity	Create a circuit	13.6	20	★
3-2	Electricity	Renewable vs Non-renewable Energy	20.8	20	★★
3-3	Electricity	Test Conductivity (known)	25.6	900	★★
3-4	Electricity	Test Conductivity (unknown)	29	600	★★
4-1	Classification	Find a living thing	14.6	300	★
4-2	Classification	Find a non-living thing	8.8	300	★
4-3	Classification	Find a plant	12.6	300	★
4-4	Classification	Find an animal	14.6	300	★
5-1	Biology	Grow a plant	69.5	126	★★
5-2	Biology	Grow a fruit	79.6	126	★★★
6-1	Chemistry	Mixing (generic)	33.6	32	★★
6-2	Chemistry	Mixing paints(secondary colours)	15.1	32	★
6-3	Chemistry	Mixing paints(tertiary colours)	23	36	★★
7-1	Biology	Identify longest-lived animal	7	125	★
7-2	Biology	Identify shortest-lived animal	7	125	★
7-3	Biology	Identify longest-then-shortest-lived animal	8	125	★
8-1	Biology	Identify life stages (plant)	40	14	★★
8-2	Biology	Identify life stages (animal)	16.3	10	★
9-1	Forces	Inclined Planes (determine angle)	97	168	★★★
9-2	Forces	Friction (known surfaces)	84.9	1386	★★★
9-3	Forces	Friction (unknown surfaces)	123.1	162	★★★
10-1	Biology	Mendelian Genetics (known plants)	130.1	120	★★★
10-2	Biology	Mendelian Genetics (unknown plants)	132.1	480	★★★

Figure 11: Chosen different difficulty levels tasks of ScienceWorld benchmark. Lens is the average length of the standard agent's trajectories. Vars is the total number of variants in this environment. A single star represents Level-1 in our article, two stars represent Level-2, and three stars represent Level-3.

ALFWorld	You are now a situation summarizer. Please summarize the information related to the environmental tasks and interactive historical observations given to you. Note that you should first carefully consider your task type, target object, and destination. To ensure universality, please don't specify the specific item. Instead, refer to the target items as "target objects" and the locations in the task as "final destinations". Request: Extract and synthesize key information related to the tasks and historical context. For example, You have discovered original target object ! You have picked up the original target object! You should provide a clear and concise description of the situation.
ScienceWorld	You are now a situation summarizer. Please summarize the information related to the environmental tasks and interactive historical observations given to you. Note that you should first carefully consider your task. Request: Extract and synthesize key information related to the tasks and historical context. For example, You focus on the thermometer. You focus on the unknown substance B. You move the thermometer to the inventory. You should provide a clear and concise description of the situation.
Jericho	You are now a situation summarizer. Please summarize the information related to environmental tasks and interactive historical observations. Note that you should first carefully consider your task. Request: Extract only the most essential facts and conclusions, using highly concise language to avoid missing any key information. You should provide a clear and concise description of the situation. Keep the abstract within 3 sentences.

Figure 12: Summarization pormpts.

ALFWorld	You are an intelligent guide in an interactive household environment. Your task is to assist Agent in accomplishing household tasks within the environment. Please analyze the current situation and use your reasoning ability to provide solutions or guidance. The available actions are: go to {recep} take {obj} from {recep} move {obj} to {recep} open {recep} close {recep} use {obj} clean {obj} with {recep} heat {obj} with {recep} cool {obj} with {recep} where {obj} and {recep} correspond to objects and receptacles. If current is "Nothing happened", that means the previous action is invalid or incorrect and you should try more options. To ensure universality, please do not specify the specific item. Remember, you are currently providing high-level and universal guidance that can assist the agent in executing available actions. Please do not include specific targets or locations. Before taking any action, please check if you are in a suitable location to perform the action related to the task. Keep your response to two or three sentences each turn. **Output Format:** Your output must strictly follow this format: Thought: your thoughts.

Figure 13: Prompts for agent on ALFWorld.

ScienceWorld	You are a helpful assistant to do some scientific experiment in an environment. You should explore the environment and find the items you need to complete the experiment. Please analyze the current situation and use your reasoning ability to provide solutions or guidance. The available actions are: open OBJ: open a container close OBJ: close a container activate OBJ: activate a device deactivate OBJ: deactivate a device connect OBJ to OBJ: connect electrical components disconnect OBJ: disconnect electrical components dunk OBJ in OBJ eat OBJ flush OBJ use OBJ on OBJ look around: describe the current room look at OBJ read OBJ: read a note or book move OBJ to OBJ: move an object to a container pick up OBJ: move an object to the inventory pour OBJ in OBJ put down OBJ mix OBJ: chemically mix a container go LOC: teleport to a specific room focus on OBJ: signal intent on a task object wait: task no action for 10 steps wait1: task no action for a step To ensure universality, please do not specify the specific item. Remember, you are currently providing high-level and universal guidance that can assist the agent in executing available actions. The specific color of the task object does not appear in think of content. Keep your response to two or three sentences each turn. **Output Format:** Your output must strictly follow this format: Thought: your thoughts.

Figure 14: Prompts for agent on ScienceWorld.

Jericho	You are a game master in fictional text games. You are in a fictional game environment and you need to accomplish goals by performing actions. Please analyze the current situation and use your reasoning ability to provide solutions or guidance. Here are the actions you can do: Examine <place/obj>: check the details of something. Take <obj>: pickup obj. Put down <obj>: leave a obj at your current place. Drop <obj> : drop obj. Check valid actions: Check actions you can use. South: go south. North: go north. East: go east. West: go west. Up: go up. Down: go down. Other available actions could be determined through check valid actions. The execution object of the action must be in available_objects list, otherwise it is invalid. Remember, you are currently providing high-level and universal guidance. Keep your response to two or three sentences each turn. **Output Format:** Your output must strictly follow this format: Thought: your thoughts.

Figure 15: Prompts for agent on Jericho.

You are an expert evaluator specialized in logical verification of decision-making processes. Your task is to assess the logical consistency and validity of an agent's thought given a specific state in a task. Your assessment must determine whether the thought is a valid and logically sound derivation from the given state, not whether it is the smartest or most optimal decision.

Input:
State: [Insert the full state description here. This should include the agent's current observations, relevant inventory, current goal, and any other critical context.]
Thought: [Insert the candidate thought string to be evaluated here.]

Evaluation Guidelines:
1. Reasoning Rationality: Whether the reasoning steps in the thought naturally follow from the state and conform to common sense or domain-specific logic. Focus on logical structure and contextual fit.
2. Non-Contradiction: The conclusions or intermediate steps of the thought cannot conflict with any information in the state.

Important Instructions:
Focus on logical structure and contextual fit, not on the optimality or brilliance of the thought.

You must output ONLY a single line in the exact following format.
Don’t include any other text, commentary, or formatting. You must output your evaluation strictly in the following format.
Output Format:
is_valid: <True|False>

Figure 16: Prompt for LLM agent logicial check.