Evolving Generalist Virtual Agents with Generative and Associative Memory

Zhenkui Zhang[1,*], Wendong Bu[1,*], Kaihang Pan[1,†], Bingchen Miao[1], Wenqiao Zhang[1,✉], Guoming Wang[1,✉], Wei Ji[2], Rui Tang[3], Juncheng Li[1,✉], Siliang Tang[1]

[1]Zhejiang University,
[2]Nanjing University,
[3]Manycore Tech Inc.
{zhenkui, wendongbu, kaihangpan, miaobingchen23, junchengli}@zju.edu.cn

Abstract

Generalist Virtual Agents (GVAs) powered by Multimodal Large Language Models (MLLMs) exhibit impressive capabilities. However, their long-term learning is hampered by a core limitation: a failure to evolve beyond existing trajectories. This stems from memory systems that treat experiences as isolated fragments and rely on brittle semantic retrieval, preventing the synthesis of novel solutions from disparate knowledge. To address this, we introduce CA3Mem, a framework inspired by the human hippocampus that organizes experiences into a structured memory graph. Leveraging this graph, CA3Mem features two key innovations: 1) a generative memory recombination mechanism that synthesizes novel solutions to drive agent evolution, and 2) an associative retrieval algorithm that employs spreading activation to recall a comprehensive and contextually-aware set of experiences. Experiments on OSWorld and WebArena demonstrate that CA3Mem significantly enhances agent capabilities, leading to marked improvements in long-horizon planning, compositional generalization for novel tasks, and continuous adaptation from experience.

1 Introduction

Recent advances in multimodal large language models (MLLMs) have substantially expanded the capabilities of Generalist Virtual Agents (GVAs) (Gao et al. 2024; Zheng et al. 2024; Li et al. 2023b; Hong et al. 2024), enabling them to interpret user instructions, perceive graphical user interfaces (GUIs), and reason over digital environments with strong visual and contextual understanding (Hu et al. 2024; Li et al. 2023a; Zhang et al. 2024a; Li et al. 2025; Wang et al. 2024a). Leveraging these MLLM-enabled capabilities, GVAs can generate executable actions and interact with device interfaces, thereby assisting users in completing diverse tasks (He et al. 2024; Li et al. 2024). Despite their perceptual and reasoning strengths, MLLM-based GVAs rely on limited context windows to capture and retain interaction history, confining them to single-task scopes and hindering persistent knowledge reuse. This limitation becomes particularly pronounced in the face of a vast and continuously evolving landscape of applications and websites, requiring agents to continually acquire up-to-date domain knowledge from open-world experience. To address these challenges, recent research has focused on endowing agents with long-term memory mechanisms that enable continuous knowledge accumulation and autonomous adaptation over time (Jiang et al. 2024; Azam et al. 2024). These mechanisms typically accumulate reusable knowledge by preserving task-specific interaction histories (Zheng et al. 2023; Wang et al. 2024b) or abstracting experiential patterns, which are often encoded in isolation across tasks (Chen et al. 2024; Wu et al. 2024a; Tan et al. 2024; Zhao et al. 2023; Fu et al. 2024). At inference time, relevant information is retrieved via semantic similarity and integrated into the MLLM's context to support downstream decision-making.

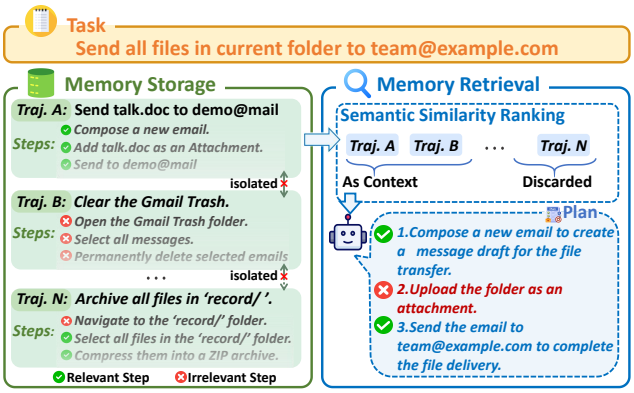

Figure 1: A case illustrating the limitations of existing memory mechanisms. Semantic similarity retrieves a partially relevant trajectory (Traj. A) but misses the crucial archiving skill in Traj. N. Furthermore, because memories are stored as isolated trajectories, this prevents the agent from synthesizing knowledge from both, leading to an incorrect plan that attempts to attach a folder directly.

However, the long-term memory mechanisms of existing GVAs exhibit significant limitations in both memory representation and utilization: **1) Static memory representation hinders evolution beyond existing trajectories:** Existing memory systems typically treat past experiences as isolated trajectories, relying on static extraction or abstraction of

*Equal Contribution.
†Project Leader.
✉Corresponding Author.
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

knowledge. While this allows for the reuse of previously observed solutions, their evolutionary potential is fundamentally limited. By failing to model the latent relationships between experiential fragments from different contexts, these systems lack a mechanism to **generatively synthesize novel task solutions**. Consequently, their problem-solving capabilities are largely bound by the scope of their explicit memories, restricting their ability to adapt to challenges that require inventive compositions of past experiences. **2) Trajectory matching based retrieval methods limit experience reuse:** Current retrieval methods primarily rely on semantic similarity, treating trajectories as strings and performing simple string matching. Given the complexity of GVA tasks, however, it is rare for a new task to have a perfect one-to-one match with a single past trajectory. As illustrated in Figure 1, if the result scope is constrained to only the most semantically similar trajectories, the agent may miss key knowledge; conversely, expanding the scope to ensure sufficient coverage typically introduces substantial noise that confounds the agent's decision-making. Furthermore, conventional methods essentially return a collection of memory fragments that are only semantically similar to the task, which may lack true contextual relevance and thus hinder the agent's reuse of experience.

In contrast, the human hippocampal cornu ammonis area 3 (CA3) (Sammons et al. 2024) offers a compelling blueprint for overcoming these limitations through two distinct mechanisms. First, it enables **generative knowledge evolution**. Through memory replay, the brain recombines fragments of past episodes to construct novel experiences that have never been directly encountered (Kurth-Nelson et al. 2023; Atherton, Dupret, and Mellor 2015; Nakashiba et al. 2009). This ability to synthesize new scenarios from existing memories in a bottom-up fashion inspires our approach to agent evolution. Second, it facilitates robust **associative memory retrieval**. Functioning as an auto-associative network (Rolls 2013; Marr 1971), CA3 can reconstruct full memory representations from partial cues via a process known as pattern completion (Rolls and Kesner 2006; Rolls and Treves 1997). This principle of associative retrieval thus diverges significantly from mechanisms dependent on direct semantic matching. Its core process, wherein activation from an initial cue progressively spreads through contextually linked memories, directly informs our design of a more sophisticated retrieval system.

Inspired by the aforementioned neuroscientific principles, we propose **CA3Mem**, a long-term memory framework designed to support cross-task generalization and continual learning in GVAs. Specifically: **1) We introduce a generative memory framework for agent evolution.** First, we propose CA3-Net, a biologically inspired memory graph that organizes experiences into subtask nodes linked by contextual dependencies, capturing latent associations across diverse task trajectories. Built upon this structure, we introduce a memory recombination mechanism that enables the agent to reason within its memory. By dynamically composing subtask nodes from different task histories in a bottom-up manner, the agent can generate novel, executable trajectories that have never been explicitly encountered. This process of synthesizing new knowledge from fragmented memories drives the agent's memory-driven evolution, allowing it to continually expand its capabilities beyond simple knowledge recall. **2) We propose an associative retrieval mechanism to enhance experience reuse.** To achieve this during online task execution, we employ a spreading activation process (Anderson 1983; Crestani 1997) that leverages both semantic similarity and contextual dependencies within CA3-Net. The process begins by identifying an initial set of seed nodes that are semantically relevant to the current task state. From these entry points, activation iteratively propagates throughout the network via contextual links, expanding the search to structurally connected memory regions. This approach enables the agent to bridge the semantic gap and retrieve crucial, task-relevant experiences, even those that are semantically distant.

Extensive experiments validate that CA3Mem significantly advances agent capabilities, achieving a **state-of-the-art** overall success rate of 23.85% on the comprehensive OSWorld benchmark. This superiority is particularly evident on difficult **Workflow** tasks requiring multi-step, cross-application coordination, where CA3Mem achieves a 10.3% absolute improvement in success rate over the GPT-4o baseline. Furthermore, dedicated experiments confirm the framework's capacity for continuous adaptation, demonstrating its ability to learn from ongoing experience. Moreover, its strong performance on the WebArena benchmark highlights its overall robustness and generalizability across diverse environments, and these results collectively underscore the broad applicability and strong potential of CA3Mem as a long-term memory framework for generalist virtual agents. Our contributions can be summarized as follows:

- We propose a generative memory framework, **CA3Mem**, that enables agent evolution by synthesizing novel, executable task solutions from a structured graph of past experiences via memory recombination.
- We develop an **associative retrieval mechanism** that employs spreading activation over the memory graph. This approach bridges the semantic gap in memory search by retrieving experiences based on contextual relevance, not just semantic similarity.
- We conduct extensive experiments across comprehensive task settings, demonstrating that CA3Mem shows improved performance in both relatively simple scenarios such as Daily and OS, as well as in more **complex, cross-application** environments like Workflow.

2 Related Work

Memory Management Mechanisms for Virtual Agents. To enhance agent adaptability, a growing body of research has focused on long-term memory mechanisms (Zhang et al. 2024c). Prior work has explored storing experiences at various levels of granularity, from complete task trajectories and workflows (Zheng et al. 2023; Wang et al. 2024b; Agashe et al. 2024) to finer-grained procedural skills and hierarchical knowledge modules (Tan et al. 2024; Wu et al. 2024a; Wang et al. 2023). Although these methods are effective at preserving past solutions, they typically treat memories as

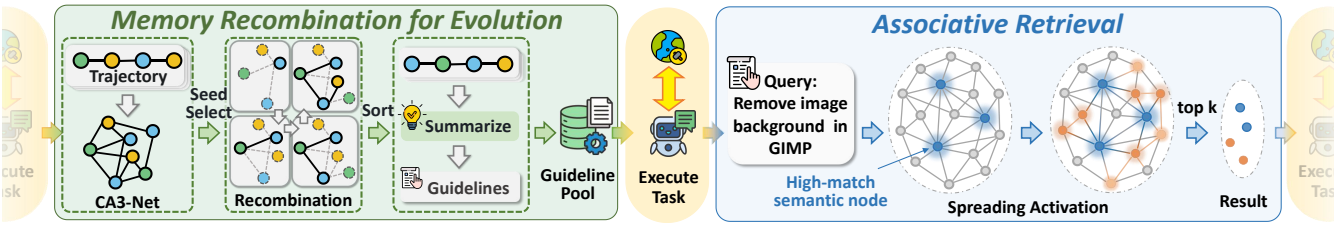

Figure 2: Overview of the CA3Mem framework. **Left:** Task trajectories are integrated into the CA3-Net memory graph, enabling recombination into new executable trajectories. These are abstracted into high-level guidelines and stored for future use. **Right :** During task execution, the current task Context triggers semantic retrieval, followed by spreading activation to identify relevant past experiences for decision-making.

isolated units and rely on retrieval mechanisms based on semantic similarity (Sumers et al. 2023). This fragmented storage and brittle retrieval process hinders agents from reusing and composing knowledge from experiences that are contextually relevant but semantically distant, thereby limiting their effectiveness in open-world environments (Xu et al. 2025a).

Mechanisms for Self-Evolution in Virtual Agents. To support self-evolution, several works introduce mechanisms that enable virtual agents to derive reusable rules or skills from their interaction history (Liang et al. 2024). Prevailing methods distill experiences into various forms of structured knowledge, such as context-aware rules (Fu et al. 2024), procedural manuals (Chen et al. 2024), or natural-language lessons and skills (Zhao et al. 2023). Other prominent strategies involve incorporating self-reflection mechanisms to optimize behavior (Shinn et al. 2023; Wu et al. 2024a; Zhang et al. 2024b) or abstracting generalizable patterns from successful executions (Tan et al. 2024; Wang et al. 2023). While these approaches are pivotal for distilling high-level knowledge from raw experience, their evolutionary potential is fundamentally bound by the scope of previously observed trajectories. They excel at abstraction and refinement but lack a mechanism to **generatively synthesize novel task solutions** by recombining experiential fragments from disparate contexts, a limitation that significantly hinders true autonomous evolution.

3 CA3Mem

CA3Mem is inspired by the CA3 region of the human hippocampus and is designed to equip virtual agents with a structured, association-capable long-term memory system. The framework segments task execution trajectories into fine-grained memory fragments and models contextual dependencies to construct the relational **memory graph CA3-Net** (**Section 3.1**). Leveraging this graph, the **memory recombination** module drives knowledge evolution by synthesizing novel, executable trajectories from the graph and summarizing them into high-level, transferable Guidelines (**Section 3.2**). During online task execution, the agent leverages a dual-retrieval process to inform its decision-making: it retrieves relevant Guidelines via semantic search, while applying the **associative retrieval** process to the CA3-Net to recall contextually relevant subtask nodes (**Section 3.3**). Finally, newly successful trajectories

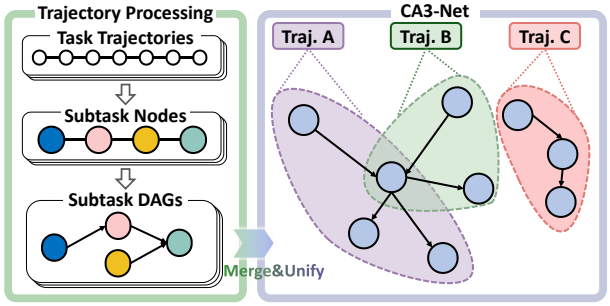

Figure 3: Construction of CA3-Net. Each trajectory is parsed into subtask nodes and structured as a DAG **(left)**, which is then merged into the unified memory graph**(right)**.

from online execution are in turn used to dynamically update and expand the CA3-Net, enabling a continuous cycle of learning and adaptation (Figure 2).

3.1 Construction of CA3-Net

The construction of CA3-Net transforms the agent's raw experiences into a structured graph, as illustrated in Figure 3. The process starts with task trajectories, which serve as the fundamental records of the agent's interactions. During task execution, the agent's experiences are collected as trajectories. A trajectory τ, representing a complete task instance, is formally defined as:

$$\tau = (G, \{(O_1, A_1), (O_2, A_2), \ldots, (O_T, A_T)\}) \quad (1)$$

Where G denotes the task goal, and $\{(O_i, A_i)\}_{i=1}^{T}$ is the sequence of observation-action pairs from the agent's interaction with the environment. Following execution, an autonomous evaluator validates each trajectory, and only successful ones are retained for memory construction. The detailed evaluation procedures are described in Appendix E.

From Trajectories to Subtask Nodes. Each successful task trajectory is then decomposed into a set of *memory units* based on the principle of functional cohesion, where each unit corresponds to a specific subtask. These units are instantiated as *subtask nodes* in the memory network, each encoding structured information such as the subtask's functional description, execution steps, environmental context, and associated applications. To enhance generalizability, subtask nodes omit low-level interface details such as the exact screen coordinates of individual actions. The prompt

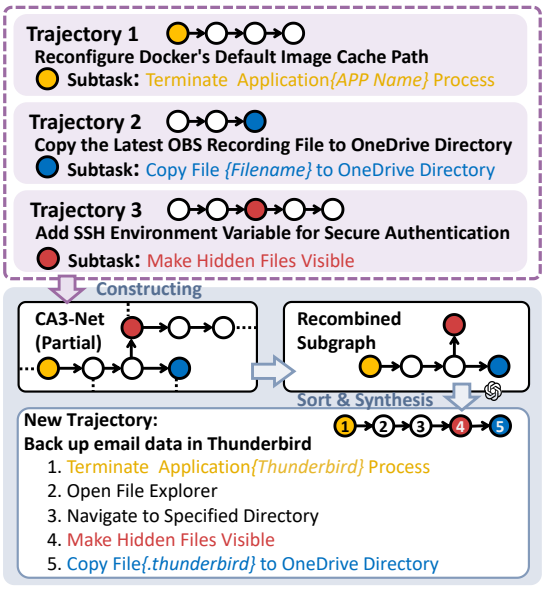

Figure 4: A memory recombination example where the agent synthesizes a novel trajectory by composing subtask knowledge from multiple prior tasks. Detailed descriptions of the source trajectories are provided in Appendix F.

used for subtask node extraction, along with an example of a generated subtask node, is provided in Appendix A.

From Subtask Nodes to Memory Networks. After subtask node extraction, each task trajectory is segmented into a set of functionally independent subtask nodes, enabling fine-grained functional decomposition. Although the sequential order of actions in a trajectory provides an explicit execution sequence, such a linear structure is often insufficient to accurately capture the underlying dependencies among subtasks. To address this, we further model the contextual dependencies between subtask nodes and construct a dependency graph that reflects their interrelations.

Specifically, we infer upstream and downstream dependencies by analyzing the preconditions and postconditions of each subtask node. The system performs dependency analysis within each trajectory's subtask node set. By leveraging contextual features such as subtask functional description and environmental resources, it uncovers latent dependencies that may be overlooked by the raw execution order and establishes directed edges between the corresponding subtask nodes.

The resulting structure is a *Directed Acyclic Graph (DAG)* that captures the subtask dependency structure of a trajectory. Each trajectory is transformed into an individual DAG $g_i = (V_i, E_i)$, where E_i is the set of directed edges representing contextual dependencies, and $V_i = \{v_{i1}, v_{i2}, \ldots, v_{in_i}\}$ is the set of its constituent subtask nodes. Collectively, these individual graphs form a comprehensive set $\mathcal{G} = \{g_1, g_2, \ldots, g_M\}$. Consequently, the global set of all subtask nodes across all trajectories is the union of these node sets: $U = \bigcup_{i=1}^{M} V_i$.

To establish cross-trajectory associations within our memory graph, we construct CA3-Net by merging functionally equivalent subtask nodes from the global set U. The equivalence is determined by a semantic similarity criterion: two nodes are merged if the cosine similarity of their functional description embeddings exceeds a predefined threshold, θ_{merge}. The resulting merged node unifies the attributes and consolidates the incoming and outgoing edges of its constituents. This process provides a robust semantic and structural foundation for the subsequent associative retrieval and memory recombination.

3.2 Memory Recombination for Evolution

As shown in the Figure 4, to overcome the limitations of processing trajectories in isolation, we propose a memory recombination-driven evolutionary strategy. This strategy synthesizes novel task trajectories through a **bottom-up composition** of subtask nodes within the CA3-Net graph. The composition process is governed by two key factors: it is constrained by the topological structure of CA3-Net, which encodes valid subtask dependencies to ensure logical reliability, and it is guided by our proposed *Coverage-based Overlap Score* to promote the discovery of novel knowledge. These synthesized trajectories serve as a basis for abstracting transferable knowledge, enabling the agent to continuously evolve in a memory-driven, self-supervised manner.

Coverage-based Overlap Score (COS). To formally quantify the novelty of a recombined trajectory, we introduce the *Coverage-based Overlap Score* (COS), which serves as the primary guidance metric for our recombination strategy. This metric assesses the trajectory's overlap with existing ones, framed as a set cover problem. Given a recombined trajectory represented by a set of subtask nodes S, and the set of all prior trajectory DAGs $\mathcal{G} = \{g_1, g_2, \ldots, g_M\}$, we first define the minimum coverage count C:

$$C = \arg\min_{\mathcal{G}' \subseteq \mathcal{G}} |\mathcal{G}'| \quad \text{s.t.} \quad \forall s \in S,\ \exists g \in \mathcal{G}' \text{ such that } s \in V_g \tag{2}$$

where V_g is the node set of a graph $g \in \mathcal{G}$. Intuitively, C is the smallest number of prior trajectories required to cover all nodes in S. The COS is then computed as:

$$\mathrm{COS}(S) = \frac{|S| - C}{|S| - 1}, \qquad |S| \geq 2 \tag{3}$$

While the general set cover problem is NP-hard, the overhead of computing COS is negligible in our framework, as synthesized trajectories inherently consist of a small number of subtask nodes ($|S|$). A score of $\mathrm{COS}(S) = 0$ indicates maximum novelty, as each constituent node is drawn from a distinct trajectory. Conversely, a score of 1 signifies no novelty, meaning all nodes originate from a single trajectory.

Memory Recombination-driven Evolutionary Strategy. Our strategy employs a novelty-guided greedy search to construct recombined subgraphs. Using the COS metric as a heuristic, the algorithm incrementally expands a subgraph from a seed node, prioritizing candidates that maximize novelty while maintaining structural integrity. Subgraphs that

meet a predefined novelty threshold are retained for the subsequent knowledge abstraction phase. The full algorithm is detailed in Appendix B.

These subgraphs, which contain contextually related subtask nodes drawn from different prior tasks, are then converted into complete trajectories from which the agent abstracts generalized GUI task guidelines. This process continually enriches the agent's knowledge base, `GuidelinePool`, driving its evolution.

3.3 Associative Retrieval via Spreading Activation

The hippocampal CA3 region forms an autoassociative network that enables the human brain to recall related memories by progressively activating neurons linked through contextual associations. As shown in Figure 2, to emulate this biological mechanism in CA3-Net, we design an *associative retrieval mechanism* based on the Spreading Activation model (Anderson 1983; Crestani 1997), which incrementally propagates activation across memory nodes via contextual links. This enables the agent to retrieve relevant memories beyond those directly matched, expanding the recall scope while preserving contextual relevance.

Initialization Phase. Given a query embedding that encodes the current task context, the model first performs a semantic search across all subtask nodes $v \in U$ in the memory graph. It computes the cosine similarity between the query and each node's embedding $\mathbf{e}_v$. Nodes whose similarity scores exceed a predefined threshold F form the initial activation set, denoted as S_0. Each node in this set is assigned an initial activation value equal to its similarity score. This set serves as both the starting point for activation spreading and the initial result set $\mathcal{R}_0$.

Spreading Activation Phase. Following initialization, the retrieval proceeds with an iterative *spreading activation phase*. At each iteration t, every active node v in the current source set S_t propagates a fraction of its activation signal $w_v^{(t)}$ to its direct neighbors. This signal is scaled by a decay factor $\lambda \in (0, 1]$, ensuring that activation attenuates as it spreads further from the original semantic source nodes, thus emphasizing closer contextual relationships. After each propagation step, the system evaluates the set of all nodes that received activation. Any node u whose cumulative activation $w_u^{(t+1)}$ meets or exceeds the threshold F and is not already present in the current result set $\mathcal{R}_t$ is considered newly activated. These nodes form the next source set S_{t+1} and are used to update the result set via $\mathcal{R}_{t+1} = \mathcal{R}_t \cup S_{t+1}$. This process allows activation to dynamically expand across the graph, uncovering increasingly relevant regions of memory.

The iterative process continues until one of two termination conditions is met: either the source set for the next iteration becomes empty ($\mathcal{R}_{t+1} = \mathcal{R}_t$), indicating that no new nodes have met the activation threshold and the spread has naturally converged; or a maximum number of iterations $T_{\max}$ is reached, preventing an overly broad or potentially infinite search. Once terminated, all nodes in the final result set $\mathcal{R}_t$ are ranked according to their cumulative activation values w_v. The top-k most strongly and consistently activated nodes are selected as the retrieved memory entries. These entries provide both semantic relevance and contextual alignment, supporting downstream decision-making in the agent's planner. A full symbolic specification of this process is provided in Appendix C.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate CA3Mem on two public benchmarks, OSWorld and WebArena, chosen to test a diverse range of agent capabilities. **OSWorld** assesses an agent's proficiency in complex GUI tasks within full-featured desktop operating systems. In contrast, **WebArena** focuses on long-horizon challenges, requiring agents to navigate and interact with dynamic, hierarchical web environments. To ensure comparability with prior work, our evaluation on WebArena is centered on its Reddit domain (Fu et al. 2024; Wang et al. 2024b; Chen et al. 2024).

Baselines. To evaluate its effectiveness and generalizability, we integrate **CA3Mem** with a GPT-4o backbone and benchmark it against a diverse set of baselines. This comparison suite includes not only foundational MLLMs like GPT-4o and Claude-3 but also a spectrum of mainstream agentic architectures, which we categorize into three types: *End-to-End Agent Models* (e.g., CogAgent (Hong et al. 2024) and AGUVIS (Xu et al. 2025b)), modular *Planner-Grounder Agents* (Yang et al. 2024; Cheng et al. 2024; Wu et al. 2024b), and *Agent Frameworks* designed for long-term memory or self-evolution (e.g., ExpeL (Zhao et al. 2023), Agent Workflow Memory (AWM) (Wang et al. 2024b), AutoGuide (Fu et al. 2024), Cradle (Tan et al. 2024), and AgentS (Agashe et al. 2024; Wang et al. 2025)). For a fair comparison, all experiments adhere to the original evaluation protocols and a fixed 15-step execution budget, with all agent frameworks, including our proposed CA3Mem, utilizing GPT-4o as the backbone MLLM.

4.2 Main Results

Online Performance in GUI Environments. Our empirical results demonstrate that CA3Mem significantly enhances the capabilities of GUI agents across diverse task types, achieving state-of-the-art overall success rates of 23.85% on **OSWorld** and 63.2% on WebArena. **1) CA3Mem achieves the best performance on the comprehensive OSWorld benchmark**, as shown in Table 1. It achieves an overall success rate of 23.85%, outperforming all baselines across categories. Compared to the end-to-end agent UI-TARS (22.70%), CA3Mem exhibits superior performance. Methods such as UI-TARS and CogAgent rely on pretraining with large-scale UI interaction datasets; however, the resulting static knowledge impairs their adaptability to dynamic and evolving environments. Furthermore, CA3Mem surpasses AgentS (20.58%), which relies on feedback-driven memory refinement but lacks structured memory management and primarily depends on semantic similarity for retrieval.

2) In long-horizon tasks, particularly in the *Workflow* domain of OSWorld, which requires coordinating actions

Category	Method	OS	Office	Daily	Profess.	Workflow	Overall
MLLM	GPT-4o	8.33	3.58	6.07	4.08	5.58	5.03
	Claude-3	12.50	3.57	5.27	8.16	1.00	4.41
	Gemini-Pro-1.5	12.50	3.58	7.83	8.16	1.52	5.10
	GPT-4V	16.66	6.99	24.50	18.37	4.64	12.17
	Qwen-VL-Max	29.17	3.58	8.36	10.20	2.61	6.87
End-to-End Agent Model	CogAgent	4.17	0.85	2.71	0.62	0.09	1.32
	AGUVIS-72B	-	-	-	-	-	10.26
	UI-TARS-72B-DPO	-	-	-	-	-	22.70
Planner-Grounder Agent	GPT-4o + OS-Atlas-7B	25.00	10.26	23.08	18.37	8.91	14.64
	GPT-4o + SeeClick	16.67	5.98	12.81	10.21	7.92	9.21
	GPT-4o + Aria-UI	25.00	9.58	25.72	20.41	8.55	15.15
Agent Framework	CRADLE	16.67	3.58	6.55	20.41	5.48	7.81
	Operator	-	-	-	-	-	19.70
	Agent S	45.83	13.00	27.06	36.73	10.53	20.58
	CA3Mem	**50.00**	**14.53**	**30.77**	**38.78**	**15.84**	**23.85**

Table 1: Task Success Rates (%) on OSWorld Subdomains under a 15-Step Evaluation.

Method	Success Rate
ReAct (Yao et al. 2023)	6.0
Expel (Zhao et al. 2023)	21.8
AutoGuide (Fu et al. 2024)	47.1
AWM (Wang et al. 2024b)	50.9
Step (Sodhi et al. 2024)	59.0
CA3Mem	**63.2**

Table 2: Task Success Rates (%) on WebArena(Reddit)

across multiple applications, CA3Mem achieves a success rate of 15.84%, significantly outperforming the next best framework (10.53%) and nearly tripling the performance of its GPT-4o backbone (5.58%). **3) CA3Mem also delivers strong performance in web-based environments**, as demonstrated on the **WebArena** benchmark, which challenges agents to handle hierarchical UI structures, conduct long-horizon navigation, and execute multi-step form interactions. It achieves a success rate of 63.2%, significantly outperforming the no-memory baseline ReAct (6.0%), as well as strong baselines such as AWM (50.9%), AutoGuide (47.1%), and Expel (21.8%). Among these, AWM retrieves previously successful workflows as exemplars for reuse, while AutoGuide distills task guides from historical trajectories by embedding contextual triggers and action suggestions. Although both methods provide valuable prior knowledge, they lack the ability to integrate and recombine information across multiple task trajectories.

Dynamic Knowledge Accumulation and Reuse. To investigate the agent's ability to learn from ongoing experience, we conducted a controlled experiment. We used the entire OSWorld *Daily* domain as a fixed, unseen test set. To simulate an autonomous learning process, we utilized GPT-4o to synthetically generate a series of task instructions based on the applications found in the *Daily* domain. The agent was sequentially exposed to four stages of this learning process, with each stage comprising 50 of these synthetic

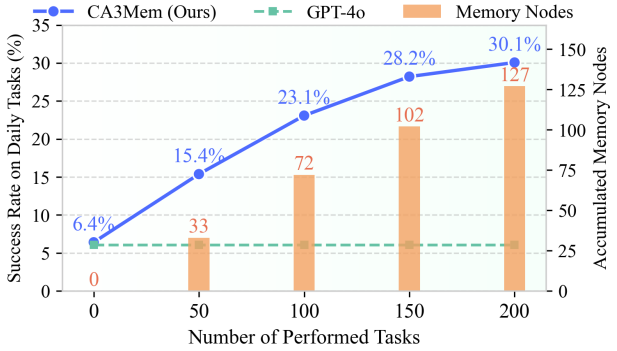

Figure 5: Success rate and memory growth curves with respect to the number of performed tasks on the OSWorld Daily domain.

tasks. After each stage, the agent's memory was frozen, and its performance was evaluated on the real *Daily* domain tasks to assess its current capabilities. The results, presented in Figure 5, show that CA3Mem successfully learns from its prior experiences and effectively transfers this knowledge to subsequent, unseen tasks. As the agent processes more tasks, its success rate on the real *Daily* test set improves dramatically, rising from an initial 6.41% to 30.07% after four stages. Crucially, this substantial performance gain directly correlates with the number of memory nodes accumulated. **These findings indicate that** our framework enables a virtuous cycle of improvement, effectively turning raw experience into enhanced capabilities for novel tasks and thereby demonstrating a strong potential for lifelong learning.

Knowledge Recombination and Evolution. Beyond executing predefined procedures, CA3Mem constructs novel task solutions by recombining subtasks stored in memory. Figure 4 presents a concrete example where a new task, backing up email data in Thunderbird, is generated by integrating subtask segments from three distinct trajectories: terminating an application process (from a Docker configuration task), revealing hidden files (from an SSH setup

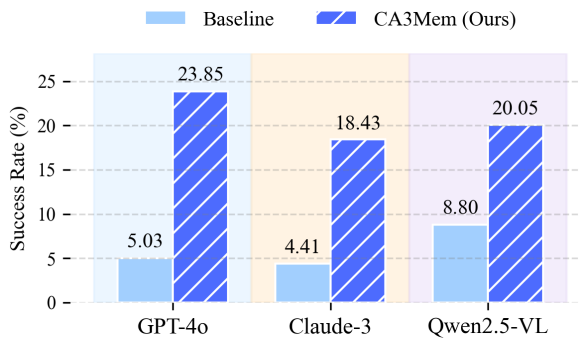

Figure 6: Task success rates (%) on OSWorld for backbone models, before and after applying CA3Mem.

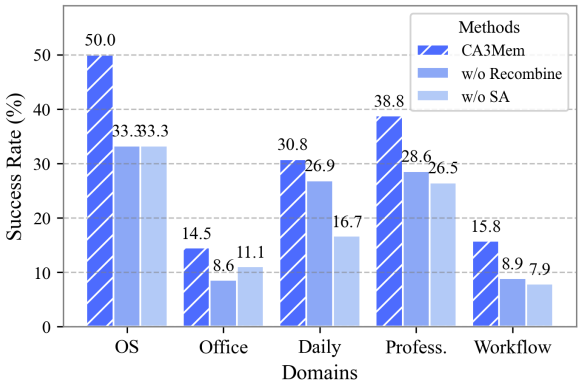

Figure 7: The success rate of CA3Mem ablation experiments on OSWorld.

task), and copying data to a OneDrive directory (from an OBS backup task). Through this recombination process, CA3Mem forms a coherent plan guided by the structural constraints of the CA3-Net, ensuring the recombined actions remain both contextually appropriate and executable. This example highlights CA3Mem's **capability to transcend rote imitation** by leveraging past experiences to synthesize new and functionally valid behaviors.

4.3 In-Depth Analysis

Impact of Memory Recombination on Compositional Generalization. To analyze how memory recombination supports generalization across task variations, we conduct ablation studies by disabling the recombination module. As shown in Figure 7, removing recombination (w/o Recombine) results in a substantial drop in success rate across all OSWorld domains, with the overall performance decreasing from 23.85% to 16.80%. The degradation is especially pronounced in scenarios that require assembling skills from different sources. For example, in the Workflow domain, where tasks span multiple applications and reusable trajectories are rare, the success rate drops from 15.84% to 8.91%. Similarly, in the Office domain, performance falls from 14.53% to 8.55%. These results suggest that CA3Mem's **recombination module is crucial for solving tasks where direct reuse of full trajectories is insufficient.** By recombining subtasks across trajectories, the agent generalizes to novel instructions beyond those seen during memory construction.

Impact of Associative Retrieval on Experience Reuse. We also evaluate the role of the spreading activation (SA) mechanism in retrieving contextually appropriate subtasks. In the ablated variant (w/o SA), CA3Mem uses a flat semantic similarity metric to retrieve subgoals, ignoring the relational structure in CA3-Net. This leads to a sharp drop in overall performance, with the success rate falling to 14.91%. The impact is especially pronounced in domains that require strong contextual matching capabilities. In the Daily domain, performance drops from 30.77% to 16.67%. Many tasks in this domain involve common applications such as Chrome and email clients, where instructions often share high semantic similarity. This dense overlap introduces retrieval noise that disrupts the agent's decision-making process (Zhang et al. 2025). In the Workflow domain, the success rate similarly declines from 15.84% to 7.92%. Workflow tasks often require coordinating knowledge across multiple applications. Similar to the example in Figure 2, these tasks frequently involve knowledge that is semantically distant from the instruction, making accurate retrieval more challenging.

Generalizability Across Models and Environments. CA3Mem demonstrates strong generalizability across both diverse UI environments and underlying model backbones. It delivers consistently high performance in desktop-based OSWorld tasks as well as in the more complex web-based interactions of WebArena. Furthermore, Figure 6 illustrates the **model-agnostic nature of CA3Mem**. The framework yields substantial performance lifts when applied to a variety of models, including GPT-4o, Claude-3, and Qwen2.5-VL (Bai et al. 2025): It elevates the success rate of Claude-3 on OSWorld from 4.41% to 18.43% and Qwen2.5-VL from 8.80% to 20.05%. This characteristic validates that CA3Mem provides a fundamental architectural enhancement, highlighting its broad applicability and potential for widespread adoption.

5 Conclusion

In this paper, we addressed a critical limitation of existing GVAs: their inability to evolve beyond pre-observed experiences due to memory systems that are fragmented and reliant on brittle semantic retrieval. We introduced CA3Mem, a long-term memory framework inspired by the human hippocampus, that enables genuine agent evolution through two synergistic mechanisms: memory recombination for synthesizing novel, executable solutions from disparate experiences, and associative retrieval for accessing comprehensive, context-aware knowledge. Our extensive experiments demonstrate that CA3Mem significantly improves task success rates on the OSWorld and WebArena benchmarks, showing marked advantages in continuous adaptation. By transforming memory from a static repository into a generative and dynamic knowledge source, CA3Mem provides a new paradigm for creating more adaptive, resourceful, and continually evolving virtual agents.

Acknowledgments

This work was supported by the Key R&D Projects in Zhejiang Province (No. 2024C01106, 2025C01030), the NSFC (62272411), and the Zhejiang NSF (LRG25F020001), the Fundamental Research Funds for the Central Universities (226-2025-00017), Ningbo Yongjiang Talent Introduction Programme(2024A-401-G), Zhejiang University Education Foundation Qizhen Scholar Foundation.

References

Agashe, S.; Han, J.; Gan, S.; Yang, J.; Li, A.; and Wang, X. E. 2024. Agent s: An open agentic framework that uses computers like a human. *arXiv preprint arXiv:2410.08164*.

Anderson, J. R. 1983. A spreading activation theory of memory. *Journal of verbal learning and verbal behavior*, 22(3): 261–295.

Atherton, L. A.; Dupret, D.; and Mellor, J. R. 2015. Memory trace replay: the shaping of memory consolidation by neuromodulation. *Trends in neurosciences*, 38(9): 560–570.

Azam, M.; Rafiq, T.; Naz, F. G.; Ghafoor, M.; Nisa, M. U.; and Malik, H. 2024. A novel model of narrative memory for conscious agents. *International Journal of Information Systems and Computer Technologies*, 3(1): 12–22.

Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; et al. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Chen, M.; Li, Y.; Yang, Y.; Yu, S.; Lin, B.; and He, X. 2024. Automanual: Constructing instruction manuals by llm agents via interactive environmental learning. *Advances in Neural Information Processing Systems*, 37: 589–631.

Cheng, K.; Sun, Q.; Chu, Y.; Xu, F.; Li, Y.; Zhang, J.; and Wu, Z. 2024. Seeclick: Harnessing gui grounding for advanced visual gui agents. *arXiv preprint arXiv:2401.10935*.

Crestani, F. 1997. Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11: 453–482.

Fu, Y.; Kim, D.-K.; Kim, J.; Sohn, S.; Logeswaran, L.; Bae, K.; and Lee, H. 2024. Autoguide: Automated generation and selection of context-aware guidelines for large language model agents. *arXiv preprint arXiv:2403.08978*.

Gao, M.; Bu, W.; Miao, B.; Wu, Y.; Li, Y.; Li, J.; Tang, S.; Wu, Q.; Zhuang, Y.; and Wang, M. 2024. Generalist virtual agents: A survey on autonomous agents across digital platforms. *arXiv preprint arXiv:2411.10943*.

He, H.; Yao, W.; Ma, K.; Yu, W.; Dai, Y.; Zhang, H.; Lan, Z.; and Yu, D. 2024. WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 6864–6890.

Hong, W.; Wang, W.; Lv, Q.; Xu, J.; Yu, W.; Ji, J.; Wang, Y.; Wang, Z.; Dong, Y.; Ding, M.; et al. 2024. Cogagent: A visual language model for gui agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14281–14290.

Hu, X.; Xiong, T.; Yi, B.; Wei, Z.; Xiao, R.; Chen, Y.; Ye, J.; Tao, M.; Zhou, X.; Zhao, Z.; et al. 2024. Os agents: A survey on mllm-based agents for general computing devices use.

Jiang, X.; Li, F.; Zhao, H.; Wang, J.; Shao, J.; Xu, S.; Zhang, S.; Chen, W.; Tang, X.; Chen, Y.; et al. 2024. Long term memory: The foundation of ai self-evolution. *arXiv preprint arXiv:2410.15665*.

Kurth-Nelson, Z.; Behrens, T.; Wayne, G.; Miller, K.; Luettgau, L.; Dolan, R.; Liu, Y.; and Schwartenbeck, P. 2023. Replay and compositional computation. *Neuron*, 111(4): 454–469.

Li, J.; Gao, M.; Tang, S.; Wei, L.; Xiao, J.; Wu, F.; Hong, R.; Wang, M.; and Tian, Q. 2025. Structure-Induced Gradient Regulation for Generalizable Vision-Language Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Li, J.; Pan, K.; Ge, Z.; Gao, M.; Ji, W.; Zhang, W.; Chua, T.-S.; Tang, S.; Zhang, H.; and Zhuang, Y. 2023a. Fine-tuning multimodal llms to follow zero-shot demonstrative instructions. In *The Twelfth International Conference on Learning Representations*.

Li, J.; Tang, S.; Zhu, L.; Zhang, W.; Yang, Y.; Chua, T.-S.; and Wu, F. 2023b. Variational Cross-Graph Reasoning and Adaptive Structured Semantics Learning for Compositional Temporal Grounding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Li, Y.; Zhang, C.; Yang, W.; Fu, B.; Cheng, P.; Chen, X.; Chen, L.; and Wei, Y. 2024. Appagent v2: Advanced agent for flexible mobile interactions. *arXiv preprint arXiv:2408.11824*.

Liang, X.; He, Y.; Xia, Y.; Song, X.; Wang, J.; Tao, M.; Sun, L.; Yuan, X.; Su, J.; Li, K.; et al. 2024. Self-evolving Agents with reflective and memory-augmented abilities. *arXiv preprint arXiv:2409.00872*.

Marr, D. 1971. Simple Memory: A Theory for Archicortex. *Philosophical Transactions of the Royal Society B*, 262(841): 23–81.

Nakashiba, T.; Buhl, D. L.; McHugh, T. J.; and Tonegawa, S. 2009. Hippocampal CA3 output is crucial for ripple-associated reactivation and consolidation of memory. *Neuron*, 62(6): 781–787.

Rolls, E.; and Treves, A. 1997. *Neural networks and brain function*. Oxford university press.

Rolls, E. T. 2013. A quantitative theory of the functions of the hippocampal CA3 network in memory. *Frontiers in cellular neuroscience*, 7: 98.

Rolls, E. T.; and Kesner, R. P. 2006. A computational theory of hippocampal function, and empirical tests of the theory. *Progress in neurobiology*, 79(1): 1–48.

Sammons, R. P.; Vezir, M.; Moreno-Velasquez, L.; Cano, G.; Orlando, M.; Sievers, M.; Grasso, E.; Metodieva, V. D.; Kempter, R.; Schmidt, H.; and Schmitz, D. 2024. Structure and function of the hippocampal CA3 module. *Proceedings of the National Academy of Sciences*, 121(6): e2312281120.

Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; and Yao, S. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36: 8634–8652.

Sodhi, P.; Branavan, S. R. K.; Artzi, Y.; and McDonald, R. 2024. SteP: Stacked LLM Policies for Web Actions. arXiv:2310.03720.

Sumers, T.; Yao, S.; Narasimhan, K.; and Griffiths, T. 2023. Cognitive architectures for language agents. *Transactions on Machine Learning Research*.

Tan, W.; Zhang, W.; Xu, X.; Xia, H.; Ding, Z.; Li, B.; Zhou, B.; Yue, J.; Jiang, J.; Li, Y.; et al. 2024. Cradle: Empowering foundation agents towards general computer control. *arXiv preprint arXiv:2403.03186*.

Wang, G.; Xie, Y.; Jiang, Y.; Mandlekar, A.; Xiao, C.; Zhu, Y.; Fan, L.; and Anandkumar, A. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Wang, J.; Xu, H.; Jia, H.; Zhang, X.; Yan, M.; Shen, W.; Zhang, J.; Huang, F.; and Sang, J. 2025. Mobile-Agent-v2: Mobile Device Operation Assistant with Effective Navigation via Multi-Agent Collaboration. *Advances in Neural Information Processing Systems*, 37: 2686–2710.

Wang, S.; Liu, W.; Chen, J.; Zhou, Y.; Gan, W.; Zeng, X.; Che, Y.; Yu, S.; Hao, X.; Shao, K.; et al. 2024a. Gui agents with foundation models: A comprehensive survey. *arXiv preprint arXiv:2411.04890*.

Wang, Z. Z.; Mao, J.; Fried, D.; and Neubig, G. 2024b. Agent workflow memory. *arXiv preprint arXiv:2409.07429*.

Wu, Z.; Han, C.; Ding, Z.; Weng, Z.; Liu, Z.; Yao, S.; Yu, T.; and Kong, L. 2024a. Os-copilot: Towards generalist computer agents with self-improvement. *arXiv preprint arXiv:2402.07456*.

Wu, Z.; Wu, Z.; Xu, F.; Wang, Y.; Sun, Q.; Jia, C.; Cheng, K.; Ding, Z.; Chen, L.; Liang, P. P.; et al. 2024b. Os-atlas: A foundation action model for generalist gui agents. *arXiv preprint arXiv:2410.23218*.

Xu, W.; Mei, K.; Gao, H.; Tan, J.; Liang, Z.; and Zhang, Y. 2025a. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*.

Xu, Y.; Wang, Z.; Wang, J.; Lu, D.; Xie, T.; Saha, A.; Sahoo, D.; Yu, T.; and Xiong, C. 2025b. Aguvis: Unified Pure Vision Agents for Autonomous GUI Interaction. arXiv:2412.04454.

Yang, Y.; Wang, Y.; Li, D.; Luo, Z.; Chen, B.; Huang, C.; and Li, J. 2024. Aria-UI: Visual Grounding for GUI Instructions. *arXiv preprint arXiv:2412.16256*.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. arXiv:2210.03629.

Zhang, C.; He, S.; Qian, J.; Li, B.; Li, L.; Qin, S.; Kang, Y.; Ma, M.; Liu, G.; Lin, Q.; et al. 2024a. Large language model-brained gui agents: A survey. *arXiv preprint arXiv:2411.18279*.

Zhang, W.; Tang, K.; Wu, H.; Wang, M.; Shen, Y.; Hou, G.; Tan, Z.; Li, P.; Zhuang, Y.; and Lu, W. 2024b. Agent-pro: Learning to evolve via policy-level reflection and optimization. *arXiv preprint arXiv:2402.17574*.

Zhang, Z.; Dai, Q.; Bo, X.; Ma, C.; Li, R.; Chen, X.; Zhu, J.; Dong, Z.; and Wen, J.-R. 2024c. A survey on the memory mechanism of large language model based agents. *ACM Transactions on Information Systems*.

Zhang, Z.; Hu, X.; Zhang, H.; Zhang, J.; and Wan, X. 2025. ICR Probe: Tracking Hidden State Dynamics for Reliable Hallucination Detection in LLMs. *ArXiv*, abs/2507.16488.

Zhao, A.; Huang, D.; Xu, Q.; Lin, M.; Liu, Y.-J.; and Huang, G. 2023. ExpeL: LLM Agents Are Experiential Learners. *arXiv preprint arXiv:2308.10144*.

Zheng, B.; Gou, B.; Kil, J.; Sun, H.; and Su, Y. 2024. GPT-4V (ision) is a generalist web agent, if grounded. In *Proceedings of the 41st International Conference on Machine Learning*, 61349–61385.

Zheng, L.; Wang, R.; Wang, X.; and An, B. 2023. Synapse: Trajectory-as-exemplar prompting with memory for computer control. *arXiv preprint arXiv:2306.07863*.