AI agent memory, learning and evolution mechanism: a comprehensive review in 2026

Authors: Zhenyu Zhang, Jingling Weng, Xiaozheng Lai, Xiaozheng Lai

Date: 2026-03-04T15:32:35+00:00

Abstract

In February 2026, AI Agent technology has entered the stage of complex systems with autonomous, persistent, and adaptive characteristics. 2025 will become the "Year of AI Agents" , promoting the transformation of large language models into active task executors. Memory, learning, and evolution are the three core pillars supporting its intelligence. This article is based on academic and industry achievements from 2024 to early 2026, and systematically reviews three major dimensions of technological progress: the evolution of memory domain parsing vector databases to neural symbol hybrid architectures, and performance trade-offs of representative systems such as MemGPT; Exploring frameworks such as Reflexion in the field of learning to implement adaptive testing mechanisms through environmental feedback; The field of evolution reveals the operational principles of genetic algorithm capability transfer protocols, MemEvolve, and other technologies in multi-agent systems. At the same time, this article compared benchmark indicators such as LoCoMo to evaluate the current implementation status of top enterprises, and found that the production level AI Agent system integrating the three pillars is still in the early stages of exploration, and urgently needs to address challenges such as retrieval latency, catastrophic forgetting, and secure alignment.

Full Text

Preamble

AI agent memory, learning and evolution mechanism: a comprehensive review in 2026 Zhenyu Zhang†, Jingling Weng†, Xiaozheng Lai∗ School of Computer Science and Engineering, South China University of Technology, Guangzhou 510641, China

Abstract

In February 2026, AI Agent technology has entered the stage of complex systems with autonomous, persistent, and adaptive characteristics. 2025 will become the "Year of AI Agents", promoting the transformation of large language models into active task executors [51]. Memory, learning, and evolution are the three core pillars supporting its intelligence. This article is based on academic and industry achievements from 2024 to early 2026, and systematically reviews three major dimensions of technological progress: the evolution of memory domain parsing vector databases to neural symbol hybrid architectures, and performance trade-offs of representative systems such as MemGPT [4, 55]; Exploring frameworks such as Reflexion in the field of learning to implement adaptive testing mechanisms through environmental feedback [17, 43]; The field of evolution reveals the operational principles of genetic algorithm capability transfer protocols, MemEvolve, and other technologies in multi-agent systems [11, 26]. At the same time, this article compared benchmark indicators such as LoCoMo to evaluate the current implementation status of top enterprises [33, 57], and found that the production level AI Agent system integrating the three pillars is still in the early stages of exploration, and urgently needs to address challenges such as retrieval latency, catastrophic forgetting, and secure alignment [46, 56].

Keywords: AI agent; Long-term memory; Self-evolving; Meta-learning; Multi-agent systems †Equal contribution (Co-first authors). E-mail: cs_{zhangzhenyu}@mail.scut.edu.cn (Z. Zhang); 202430452950@mail.scut.edu.cn (J.

Weng) ∗Corresponding Author. E-mail: laixz@scut.edu.cn (Xiaozheng Lai) Contents

2.1 Paradigm Shift of Memory Architecture: From Vector Database to Cognitive

Mapping . .

7 Future Prospects and Conclusions

Implementation Status of Production Environment .

1.1 Research Background and 2025-2026 Technical Context

With the maturity of transformer architecture and the improvement of computing power, the field of artificial intelligence has experienced a profound paradigm shift from 2025 to 2026. The traditional generative AI mainly focuses on the quality of content generation, while the new generation of AI agents emphasize the ability of perception, planning, action and reflection in a dynamic environment [1]. This transformation requires that the agent system not only has a powerful reasoning core, but also has a "cognitive life cycle" similar to organisms:

it can remember past experience (memory), learn from mistakes (learning), and pass the optimal strategy (Evolution) in the group.

At the beginning of 2025, Microsoft announced that it had entered the " AI agent era" and embedded the agent function into its entire technology stack [57]. Sam Altman, CEO of openai, predicts that the first AI agent will join the labor force in 2025 [58]. However, in order to realize the real independent agent, simply relying on the expansion of model parameters has encountered a bottleneck, and the focus of research has turned to architecture innovation. Especially in long-horizon tasks, the restriction of context window makes the external memory system a rigid requirement [6]. At the same time, the static model weight cannot adapt to the rapidly changing environment, so feedback based self reflection and population-based evolutionary algorithm have become the key paths to improve the adaptability of agents [15, 26].

1.2 Definition of Core Research Dimensions

In order to systematically understand this complex field, this report defines the research scope as the following three interrelated dimen- sions: • Memory: refers to how the agent realizes the persistence of state, the cross round management of context, and the evolution of the architecture of short-term and long-term memory. This includes the application of vector database, the distinction and integration of episodic memory and semantic memory, and the advanced variant of retrieval enhanced generation (RAG) [1, 7]. • Learning: refers to how a single agent can achieve self iteration of its ability through environmental feedback, self reflection or experience accumulation. This covers test time learning, meta learning and strategy optimization based on language feedback [14,21]. • Evolution: refers to how swarm intelligence emerges in a multi-agent collaborative ecosystem; How are the capabilities and Strategies of agents encapsulated, shared, inherited and naturally selected in the network. This involves evolutionary algorithms (EAS), capability transfer protocols and the unified evolutionary architecture [32, 36].

1.3 Report Structure and Objectives

Figure 1 [Figure 1: see original paper] systematically presents the complete cognitive lifecycle of AI agents in 2026, visually integrating the technological systems and interrelationships of the three core pillars of memory, learning, and evolution. The memory mechanism section shows the evolution path from traditional vector databases to hierarchical, graph structured, and biologically inspired architectures; The learning and self reflection mechanism focuses on core processes such as closed-loop feedback and introspective learning, reflecting the iterative logic of agents from output to feedback and then to optimization; Evolution and genetic mechanisms depict the application of evolutionary algorithms, ability transfer, and the emergence of swarm intelligence in multi-agent ecosystems. The three pillars are ultimately integrated into a unified evolutionary architecture such as MemEvolve and ReMem. At the same time, the chart outlines the current industry implementation status, mainstream benchmark testing frameworks, and core challenges such as security and efficiency during deployment, providing an overall technical framework for the hierarchical analysis of subsequent chapters in this article.

This report aims to provide researchers and practitioners with a detailed technical map. We will first analyze the latest design of memory architecture and compare the tradeoffs between efficiency and accuracy of different systems. Then, we discuss how the learning mechanism can improve the robustness of agents through closed-loop feedback. Then, the evolutionary dynamics and its technical implementation in multi-agent system are analyzed. Finally, combined with benchmark data and industrial deployment cases, the maturity and future direction of current technology are evaluated. All discussions are based on public literature and search results from 2024 to early 2026 to ensure the timeliness and accuracy of information.

2 Deep Evolution of AI Agent Memory Mechanism

Memory is one of the core characteristics of AI agent different from traditional chat robot. It endows agents with continuity in time dimension, enabling them to understand long-term dependencies and accumulate domain knowledge. From 2024 to 2025, the memory archi- tecture has undergone rapid iterations from simple vector storage to complex neural symbol hybrid systems.

2.1 Paradigm Shift of Memory Architecture: From Vector Database to Cognitive Mapping

Early AI agent memory systems mainly relied on vector databases for semantic similarity search [5]. Although this architecture is simple to implement, it performs poorly in dealing with complex reasoning and multi hop query, and it is difficult to maintain the logical association between information. The research trend in 2025 shows that the memory system is developing in the direction of stratification, graph structure and biological heuristics.

Hierarchical Memory Systems (HMS) is the mainstream solution to the context window restriction. MemGPT is a pioneer in this field. It adopts the design inspired by the operating system and divides memory into " main context" , " external storage" and " archive" through virtual context management [6, 12]. This mechanism allows the agent to manage tokens like hard disk and memory, so as to handle unlimited dialog history in limited working memory. However, MemGPT has limitations in retrieval efficiency, especially in tasks requiring precise time reasoning [53].

Graph-based Memory has made significant progress in 2025. MAGMA(cid:0)Multi-Graph Agent Memory Architecture(cid:0) The multi graph structure is introduced to organize the agent memory, which can more effectively capture the complex relationship between entities [4]. Zep uses the time knowledge map architecture and takes the time stamp as an important attribute of the graph node, thus enhancing the agent' s ability to understand the event sequence [8]. The advantage of graph structure is that it supports multi hop reasoning, but the computational overhead in writing and updating is large, and it needs to balance the depth of graph traversal and retrieval delay.

Bio-Inspired Architectures attempted to simulate the function of the human hippocampus. The hippocampus module released in 2024-2025 uses binary signature and dynamic wave matrix for fast search, aiming to solve the high cost problem of existing memory systems in terms of retrieval delay and storage scalability [1]. This design shows significant performance advantages in the locomo benchmark, especially in reducing the query token overhead [1].

Figure 1: A comprehensive overview of the AI agent cognitive life cycle in 2026. The diagram illustrates the three core pillars of agent intel- ligence: Memory Mechanisms (transitioning from vector databases to bio-inspired and hierarchical architectures), Learning & Self-Reflection Mechanisms (focusing on closed-loop feedback and introspective adaptation), and Evolution & Genetic Mechanisms (detailing swarm in- telligence and capability transfer within multi-agent ecosystems). The roots represent the synthesis of these pillars into Unified Evolution Architectures (e.g., MemEvolve, ReMem), while the peripheral panels summarize the current industrial landscape, benchmarking frameworks, and critical deployment challenges.

2.2 The Integration of Episodic Memory and Semantic Memory

In cognitive psychology, memory is divided into episodic memory (memory about specific events) and semantic memory (memory about facts and concepts). AI agent systems are gradually reproducing this distinction to improve the accuracy of reasoning.

The Implementation of Episodic Memory mainly focuses on the storage and retrieval of historical interaction trajectories. The AMEM (Agent Memory) system is inspired by the Zettelkasten method and simulates episodic memory by dynamically linking and organizing memory notes [9]. This system not only stores the original dialogue, but also stores the agent' s thinking process and action results at that time. The EMemBench benchmark specifically evaluates the interactive ability of episodic memory, emphasizing the difficulty for agents to maintain consistency in long conversations [10].

The Construction of Semantic Memory focuses on extracting general knowledge from a large number of scenarios. SHIMI (Semantic Hierarchical Memory Index) proposes a semantic hierarchical memory index for scalable proxy inference [2]. It reduces the redundancy of the memory bank by abstracting specific interactive experiences into universal rules or facts. The Mem0 architecture focuses on continuous retrieval and retrieval of memory, automatically distinguishing which information needs to be retained as semantic knowledge for a long time and which is only used as temporary context [10, 13].

The Challenge of Integration lies in how to dynamically switch between the two. The A-MEM system attempts to organize past inter- actions through a dynamic indexing and linked note network, with linking and refining functions, attempting to construct semantic associations while maintaining situational details [4, 9]. However, existing fusion mechanisms still face the risk of " catastrophic forgetting" , where the writing of new knowledge may lead to the coverage of old knowledge or a decrease in retrieval weights [16].

2.3 Analysis of Representative Memory System Techniques

In order to gain a deeper understanding of the technical details, we analyzed several of the most representative memory systems from 2025 to 2026.

MemGPT: As a representative of hierarchical memory, the core of MemGPT lies in " context swapping" . It treats LLM' s context window as limited RAM and external vector storage as a hard disk. When the context is full, the system will automatically archive unimportant information and retrieve relevant information to load back into the context [10]. This design allows the limited context model to behave as if it has a larger working memory. In the MemBench evaluation, MemGPT performs outstandingly in terms of " participation accuracy" , but has low efficiency and high token consumption [53].

Hippocampus: This system emphasizes efficiency and scalability. It uses binary signatures to compress memory content and utilizes dynamic wave matrices for fast similarity matching [1]. On the LongMemEval-s benchmark, Hippocampus demonstrated extremely low to- ken consumption per query, demonstrating a significant efficiency advantage compared to resource intensive systems such as MemGPT [1].

However, in complex reasoning tasks that require deep semantic understanding, binary compression may result in a loss of some information accuracy.

MAGMA: Architecture based on multi graph structure. MAGMA has been evaluated on the LoCoMo benchmark, demonstrating its effectiveness in multi session, single session users, and knowledge update tasks [4]. It processes different types of information (such as user preferences, task status, world knowledge) separately by maintaining multiple subgraphs, and performs graph fusion during retrieval. This design improves the structuring of reasoning, but increases the engineering complexity of the system.

MemEvolve: This is a relatively new meta evolution framework that not only manages memory content, but also evolves the memory architecture itself [3, 11]. It proposes a unified modular design space (encoding, storage, retrieval, management) that allows agents to adjust their memory strategies based on task feedback. For example, if the retrieval hit rate is low, the system may automatically adjust the parameters or retrieval threshold of the embedded model [11].

2.4 Benchmarking and Performance Evaluation of Memory Systems

With the diversification of the memory system, evaluation criteria have become crucial. Multiple specialized benchmark test suites have emerged in 2025 to quantify and compare the performance of different architectures.

LoCoMo (Long Context Understanding and Reasoning): This is currently one of the most widely used long-term memory bench- marks [33]. It covers task categories such as single hop/multi hop retrieval, temporal reasoning, common sense reasoning, and adversarial reasoning [64]. On LoCoMo, Hippocampus achieved an F1 value of 38.3 in temporal inference tasks, significantly higher than MemGPT' s 26.5, indicating its advantage in handling temporal dependencies [1]. MAGMA has also demonstrated superiority on this benchmark, particularly in tasks that require structured knowledge [4].

LongMemEval: This benchmark focuses on evaluating the accuracy and efficiency of long-term memory [65]. The evaluation metrics include average number of processed tokens, open domain performance, and overall score [33]. On LongMemEval-s, there are significant differences in token efficiency among different systems. MemeGPT is recorded as one of the systems with the highest resource consumption, while Memory Bank shows higher efficiency [1]. This reveals the classic trade off between accuracy and efficiency in the memory system.

MemBench: Built a more comprehensive dataset to evaluate the effectiveness, efficiency, and capacity of LLM agent memory capa- bilities [46]. It proposes interactive scenarios such as participation and observation, and points out that the current challenges facing evaluation include insufficient diversity and lack of comprehensive indicators [46]. MemBench' s 3D scatter plot visually demonstrates the performance trade-offs of state-of-the-art proxy memory systems in terms of F1 score, average token consumption, and average total time, indicating that achieving a perfect balance of high accuracy, low resource consumption, and low execution time remains a challenge [1].

Other indicators: In addition to task success rate, latency and token cost are also key concerns in the production environment. Hip- pocampus' evaluation on LoCoMo and LongMemEval benchmarks showed significant performance advantages, such as reduced retrieval latency and query token overhead [1]. The Evo Memory benchmark introduces metrics such as Step Efficiency and Sequence Robustness to evaluate the adaptability of agents in dynamic environments [43].

2.5 Technical Challenges and Limitations of Memory Mechanisms

Figure 2 [Figure 2: see original paper]: Deep evolution and architectural innovations of AI agent memory mechanisms. This figure delineates the paradigm shift from traditional vector-based semantic search to modern neural-symbolic hybrid architectures designed for long-term dependency management. It categorizes key memory innovations into four domains: Hierarchical Memory Systems (e.g., MemGPT) which utilize OS-inspired virtual con- text management; Graph-Based Memory (e.g., MAGMA, Zep) for capturing complex entity relationships and supporting multi-hop reasoning; Bio-Inspired Architectures (e.g., Hippocampus) that employ binary signatures and dynamic wave matrices for high-speed retrieval; and Meta- Evolutionary Frameworks (e.g., MemEvolve) which optimize memory strategies based on task feedback. The bottom sections further illustrate the functional integration of episodic and semantic memory types—addressing the transition from specific interactive trajectories to generalized knowledge—while highlighting persistent technical challenges such as the latency-scalability contradiction and catastrophic forgetting.

Figure 2 provides a panoramic visualization analysis of the deep evolution and architectural innovation of AI Agent memory mecha- nisms from 2024 to 2026, clearly defining the core goals of paradigm shift from traditional vector databases to neural symbol hybrid architec- tures: addressing pain points such as long-range dependency management, complex reasoning, and multi hop queries. The chart divides the current mainstream memory architecture innovations into four categories, respectively illustrating the operating system based virtual context management of the hierarchical memory system (MemGPT), the entity relationship capture capability of the graph based memory architec- ture (MAGMA, Zep), the binary signature and dynamic wave matrix high-speed retrieval mechanism of the biological heuristic architecture (Hippocampus), and the modular design space and policy adaptive logic of the meta evolution framework (MemEvolve). The performance ad- vantages and limitations of each architecture are compared. At the same time, the lower part of the chart shows the integration path of episodic memory and semantic memory, using systems such as AMEM and SHIMI as examples to illustrate the functional characteristics of the two types of memory. Ultimately, it focuses on the core technical challenges faced by the current memory mechanism, such as delay scalability contradiction, adaptive loss, illusion and consistency issues, privacy and security, which accurately corresponds to the analysis content of this section and intuitively presents the current technical status and unresolved problems of memory mechanism development.

Despite significant progress, memory mechanisms still face many challenges. Firstly, there is a contradiction between retrieval latency and storage scalability. As the memory bank grows, the time complexity of vector search increases, affecting the real-time interaction experience [1]. Secondly, there is a lack of adaptability. Most memory systems are static and cannot dynamically adjust retrieval strategies based on task types [11]. The third issue is hallucinations and consistency. The retrieved erroneous information may cause agents to experience hallucinations, and information conflicts in long-term memory are difficult to resolve automatically [47]. Finally, privacy and security are also important considerations, and persistent storage of user data requires strict access control and encryption mechanisms [38].

3 AI Agent Learning and Self Reflection Mechanism

If memory is the agent' s ' past' , then learning is the agent' s bridge to the ' future' . In 2025-2026, the learning mechanism of AI agents will no longer be limited to weight updates in the pre training stage, but will shift towards self reflection during testing, utilization of environmental feedback, and experience replay.

3.1 The Core Mechanism of Self-Reflection

Self reflection refers to the ability of an agent to review and analyze their actions and decision-making processes after completing a task, in order to generate improvement strategies. This is the key to enhancing the planning capability and robustness of agents.

The Reflexion Framework is a milestone work in this field. It reflects on itself through language feedback, storing experiences from task failures or successful trajectories in a memory buffer to guide subsequent attempts [17, 19]. Reflexion maintains an 'Event Buffer'that stores verbal reflections. When agents encounter similar situations, they will retrieve these reflections to avoid repeating the same mistakes.

This method does not require fine-tuning of model weights and can significantly improve performance during testing [18].

The ReMem Architecture further integrates reflection and memory evolution. ReMem is described as a self evolving proxy baseline that dynamically manages memory through an iterative process [50]. Its core is the " Think Act Refine" cycle [43]. • Thinking Module: Responsible for internal reasoning and task decomposition. • Action Module: Perform specific operations or generate final responses. • Optimize Memory Module: After each action, explore, retrieve, trim, and organize memory content to improve subsequent task performance [43]. ReMem emphasizes episodic memory and cross event reasoning, aiming to address the shortcomings of language agents in recalling and inferring interactive histories [43]. ReMem consistently outperforms the History baseline on the GPQA, TOOLBENCH, and MMLU-PRO-ING datasets, indicating improvements in convergence speed and final accuracy [43].

Introspective Learning allows agents to analyze their own performance during testing time and generate verbal criticism or insights [16]. This mechanism enables agents to quickly recover from a single error and learn from experience, thereby transforming into stronger and more efficient performance [21]. Some methods also optimize their self reflection ability by re collecting model guided revision trajectories and conducting supervised fine-tuning (SFT) [21].

3.2 Environmental Feedback and Reinforcement Learning

Environmental feedback is a key signal source for AI agent learning and improvement [15]. Unlike traditional supervised learning, agent learning relies more on sparse reward signals or diagnostic signals.

Feedback Type: The environment or evaluator provides reward signals or diagnostic signals to help the agent identify areas for improve- ment and adjust behavior [18, 22]. These feedbacks can be explicit (such as task success/failure markers) or implicit (such as user satisfaction, code execution error logs). Agents interact with external sources (such as knowledge bases or environments) to obtain additional information, forming a closed loop [24].

Iterative Learning Process: AI agents rapidly improve their decision-making ability through an iterative process of trial and error, self reflection, and continuous memory [22]. The cycle of generating output ->environmental feedback ->self reflection ->generating new responses constitutes the basic unit of agent learning [20]. For example, in a code generation task, the agent executes the code, captures error information, reflects on the cause of the error, corrects the code until it passes the test case.

Experience Replay: Although the search results do not directly mention " experience replay" as an independent term, multiple pages mention concepts such as " experience,"" trajectory,"" memory," and " iterative feedback," all of which are related to the core idea of experience replay [25]. Agents improve by storing and reviewing past trajectories, reflecting, and remembering [19,23]. This mechanism draws inspiration from the experience replay pool in deep reinforcement learning, but in LLM agents, it manifests as historical log retrieval in textual form.

3.3 Meta learning and Gradient Updating Strategy

Figure 3 [Figure 3: see original paper] constructs a comprehensive framework for the learning and self reflection mechanism of AI agents in 2026, which fully visualizes the iterative process of adaptive and continuous self-improvement of agents during testing, forming a precise correspondence with the technical content in sections 3.1-3.3 of this chapter. The upper left area of the chart analyzes the core mechanism of self reflection, focusing on the " event buffer" verbal reflection storage and reuse logic of the Reflexion framework, as well as the optimization path of situational memory and cross event reasoning through the " think act optimize" closed-loop of the ReMem architecture; The upper right and middle regions explain the environmental feedback and reinforcement learning system, clarify the types of explicit/implicit feedback signals, the iterative learning process of agents from output to feedback and then to reflective optimization, and the experience replay mechanism based on text-based historical log retrieval; The bottom area focuses on meta learning and gradient update strategies, demonstrating the core logic of the UMEM framework, which considers external memory banks as evolutionary non differentiable parameters and achieves experience extraction through forward propagation and backward optimization, as well as the meta update rules of the MemEvolve framework for co evolution of experience knowledge and memory architecture. Overall, this chart demonstrates how learning and self reflection mechanisms transform agents from static response generators to adaptive entities with continuous self-improvement and positive knowledge transfer capabilities, providing a comprehensive technical perspective for the analysis of meta learning and gradient update strategies.

In order to achieve continuous adaptation, some studies have begun to explore the application of meta learning to agent systems. Meta- Learning aims to achieve continuous self-improvement by driving agents to enhance themselves through interaction [72].

The UMEM Framework proposes a unified memory extraction and management framework that considers adaptive agents as a pa- rameter system, where the external memory bank is an evolving non differentiable parameter [34]. It is conceptually similar to neural network optimization, including forward propagation (retrieving memory) and backward optimization (updating memory content) to extract and consol- idate experience [34]. This perspective transforms memory management into an optimization problem, allowing the use of gradient information (if differentiable) or evolutionary strategies to update the memory bank.

Meta-Update Rule: Some work has explored meta learning loop optimization of proxy systems themselves [73]. MemEvolve, as a meta evolutionary framework, aims to jointly evolve agents' experiential knowledge and memory architecture, allowing agents to not only accumulate experience but also gradually optimize their learning methods [54]. This means that agents not only learn ' what to do' , but also ' how to remember' and ' how to learn' .

Adaptive Mechanism: Adaptive agents need to continuously adapt to new tasks while retaining previous knowledge, which involves continuous learning and catastrophic forgetting problems [16]. Live-Evo is an online adaptive memory system that learns from continuous Figure 3: Comprehensive framework of AI agent learning and self-reflection mechanisms. This figure illustrates the iterative processes through which agents achieve self-improvement and test-time adaptation. The Core Mechanisms of Self-Reflection (top left) detail the Reflexion Framework, which utilizes an " Event Buffer" for verbal reflections, and the ReMem Architecture, which implements a " Think-Act-Refine" cycle to optimize episodic memory and cross-event reasoning. The Environmental Feedback & Reinforcement Learning section (top right and middle) explores how agents interpret explicit and implicit signals to generate revised trajectories and perform introspective error recovery. The bottom panel highlights Meta-Learning & Gradient-Based Strategies, featuring the UMEM framework for treating memory as a non-differentiable parameter system and the MemEvolve framework for the co-evolution of experiential knowledge and memory architecture. Collectively, these mechanisms enable agents to transition from static response generators to adaptive entities capable of continuous self-improvement and positive knowledge transfer. feedback and manages memory through experience weight updates and forgetting [50]. MemRL achieves adaptation by optimizing memory efficiency to minimize catastrophic forgetting and maximize positive transfer [74]. Evo-MAML mentions " meta learning and evolutionary gradients" , implying dynamic or evolutionary gradient update strategies [75].

3.4 Evaluation Indicators for Learning Mmechanisms

Evaluating the learning ability of agents requires specific indicators. The Evo-Memory benchmark evaluates learning, adaptation, and knowledge reuse capabilities, and mentions metrics such as step efficiency and sequence robustness [43]. Mem2ActBench provides metrics on conversation, turn, token, QA pairs, inference, and memory evolution [45]. In addition, task success, such as GAIA benchmark, and tool use ability are also important dimensions for measuring learning effectiveness [45, 47].

4 Evolution and Genetic Mechanisms in Multi Agent Ecology

When a single Agent extends to Multi Agent Systems (MAS), the emergence of intelligence undergoes a qualitative change. The evolutionary mechanism introduces the concepts of natural selection, genetics, and variation in biology, enabling agent populations to self optimize without central control.

4.1 The Application of Evolutionary Algorithm in Agent System

Evolutionary algorithms (EAs) are a class of stochastic search algorithms inspired by biological evolution, which generate new solutions and retain improved ones through basic mechanisms such as selection, mutation, and recombination [26]. In 2025-2026, EAs will be widely used in multi-agent control problems and automatic agent generation.

EvoAgent: This study explores the use of evolutionary algorithms for automatic multi-agent generation [32]. By defining the genes of the agent (such as prompt words, tool configurations, and memory strategies), the system can automatically mutate new instances of the agent and conduct fitness assessments in the task environment. The best performing agent will be retained and serve as the parent of the next generation.

Natural Selection and Reproduction: In the ecosystem architecture based on genetic algorithms, the best performing agents will be reproduced through crossover and mutation operations [27]. Evolutionary selection involves inheritance and mutation, and successful agents will pass on their behavior to their offspring [30]. This mechanism enables excellent strategies, such as efficient prompt word templates or memory retrieval logic, to spread within the group.

Co-evolutionary models: used to solve problems in multi-agent systems [29]. In coevolution, different types of agents (such as preda- tors and prey, or task decomposers and executors) exert selection pressure on each other, promoting the joint improvement of their abilities.

Evolutionary computation is applied in cooperative multi-agent environments to reduce the need for prior knowledge [28].

4.2 Capability Transfer and Protocol Standardization

In the multi-agent ecosystem, capability transfer is the key to achieving resource optimization and task collaboration. This involves how agents exchange code, state, or policies between each other.

Technical Implementation: The framework provides primitives such as " requestToTransfer" and " begin Transfer" for intelligent agents to move between different environments [36]. The migration of intelligent agents involves code and state information, and secure migration is challenging, but communication based migration methods are effective [38]. Mobile intelligent agents can be seen as messages containing task descriptions and executing code [37].

Protocol and specifications: • FIPA and JADE: Traditional multi-agent systems rely on the FIPA (Foundation for Intelligent Physical Agents) specification and JADE (Java Agent Development Framework) [66,68]. FIPA defines Proxy Communication Language (ACL) and Interaction Protocols (IPs) [67]. JADE provides APIs for sending and receiving messages, and supports the standard interaction protocol in the FIPA specification [69]. • New framework: Although FIPA is the foundation, the new framework for 2025 focuses more on LLM driven capability exchange.

The axiom of ' exchange_{capability}' was mentioned in [42], which defines the limitations on the exchange capabilities between intel- ligent agents. A modeling language for intelligent agents that implements protocols through capabilities has been proposed in [40]. A protocol language has been proposed that can clearly specify multi-agent protocols and automatically generate code through protocol specifications in [41]. • API and Data Format: The FIPA specification defines that message formats are specified by ACL language [70]. Although HTTP MTP and XML encoding may not fully comply with web service standards in some aspects, FIPA itself does not mandate the specifi- cation of transmission methods [71]. New research tends to use lighter weight JSON-RPC or gRPC combined with semantic protocols to implement capability invocation [76].

Capability-Based Reconfiguration: The multi-agent framework includes Capability Matching Agents (CMA) and Product Matching Agents (PMA), implying capability based reconfiguration [39]. This means that the system can dynamically combine agents with different abilities according to task requirements, forming temporary collaborative teams.

4.3 Unified Evolution Architecture: MemEvolve and UMEM

At the end of 2025 to the beginning of 2026, the research community began to attempt to integrate memory, learning, and evolution into a unified architecture.

MemEvolve: This is a meta evolution framework aimed at achieving meta evolution of self evolving memory systems through a unified modular design space (encode, store, retrieve, manage) [11]. It proposes EvolveLab, a unified self evolving memory code repository that extracts twelve representative memory systems into modular design spaces [54]. The core idea of MemEvolve is that an improved outer loop memory architecture can enhance the learning efficiency of agents, while more powerful agents can provide more accurate fitness signals to drive the next round of architecture evolution [11]. It has achieved significant performance improvement and strong cross task generalization ability on multiple benchmarks [54].

UMEM: conceptually similar to neural network optimization, it includes forward propagation and backward optimization to extract and consolidate experience [34]. It considers external memory banks as evolutionable parameters [34].

ReMem and Evolution: ReMem establishes a new standard for adaptive and self improving LLM agents by combining reflection and memory evolution [43]. The evolutionary memory mechanism allows agents to autonomously optimize strategies through feedback [49].

Data Flow and Interaction: Although lacking standardized detailed data flow diagrams, ReMem' s " Think Act Optimize" loop provides high-level process descriptions [43]. MemEvolve evolves the implementation of programmatic modules through feedback loops [11]. These architectures attempt to break down the boundaries of memory, learning, and evolution, making them different facets of the same optimization process.

4.4 The Emergence of Swarm Intelligence

Figure 4 [Figure 4: see original paper] presents a four quadrant analysis of the evolutionary dynamics and genetic mechanisms in the multi-agent ecosystem in 2026, fully demonstrating the emergence and dissemination patterns of swarm intelligence, which is in line with the technical content in sections 4.1- 4.4 of this chapter. Quadrant A displays the evolutionary cycle of multi-agent systems, clearly depicting the selection, crossover, and mutation operations based on " agent genes" (prompt words, tool configurations, memory strategies, etc.), as well as the fitness evaluation and reproduction process from the t-th generation population to optimized offspring, corresponding to the core application of evolutionary algorithms in intelligent agent systems; Quadrant B provides a detailed explanation of capability migration and protocol standardization, presenting the technological evolution from traditional FIPA/JADE frameworks to lightweight LLM driven capability exchange and secure migration primitives, including the application of new APIs and semantic protocols such as JSON-RPC/gRPC; Quadrant C focuses on a unified evolutionary architecture, using MemEvolve as an example to illustrate its dual loop collaborative mechanism of " architecture meta evolution outer loop+intelligent agent learning inner loop" , as well as the core supporting role of EvolveLab modular design space; Quadrant D integrates the driving factors of swarm intelligence and outlines how biological heuristic mechanisms (sexual selection, coevolution) and macro population operations (agent fusion, genetic material transfer) can drive multi-agent systems to solve complex problems that cannot be handled by a single agent, providing an intuitive technical framework for analyzing the emergence of swarm intelligence.

How did Swarm Intelligence emerge in the multi-agent collaborative ecosystem? The introduction of co evolutionary models and genetic algorithms enables agent populations to solve complex problems that a single agent cannot handle [29]. Evolutionary computing technology introduces macro operations such as agent merging, genetic material transfer, and population differentiation [28]. The Sexual Selection Mechanism is described in co evolutionary multi-agent systems and is used to determine which agent strategies can be combined [31].

This biologically inspired approach reduces reliance on manually designed collaboration protocols, allowing collaboration patterns to naturally form during evolution.

5 Benchmarking and Quantitative Performance Analysis

Figure 5 [Figure 5: see original paper] provides a systematic analysis of the benchmark testing pattern and quantitative performance evaluation of AI Agent memory systems in 2026, providing a panoramic visualization framework for the performance analysis in this chapter. The four quadrants correspond to the benchmark testing suite, core performance indicators, precision efficiency trade-offs, and evaluation challenges. Quadrant A has sorted out the current mainstream benchmark testing suite system, covering sub directions such as Long Context Reasoning (LoCoMo), Long Term Memory Accuracy and Efficiency Assessment (LongMemEval), Memory Ability Comprehensive Assessment (MemBench), Self Evolution- ary Memory Assessment (Evo Memory), Task Oriented Memory Utilization (Mem2ActBench), and Trusted Testing Evolution (TAME), and clarified the core evaluation objectives of each suite; Quadrant B compared the key performance indicators of the memory system (F1 score, latency, token efficiency, retrieval latency, etc.) and quantified the performance differences between different architectures - Hippocampus had significantly higher F1 scores than MemGPT in temporal inference tasks, MAGMA performed well in multi session knowledge update tasks, and Hippocampus had lower token consumption and retrieval latency; Quadrant C visualizes the core challenge of the " precision efficiency trade-off" in AI Agent memory systems, intuitively demonstrating the inherent contradiction of high fidelity reasoning often accompanied by high computational overhead and high resource efficiency, which may sacrifice the accuracy of complex semantic understanding; Quadrant D points out the key shortcomings of the current evaluation methodology, including insufficient diversity of testing tasks, lack of dynamic Figure 4: Evolutionary dynamics and genetic mechanisms in multi-agent ecosystems. This figure provides a four-quadrant analysis of how intelligence emerges and propagates within agent populations. Panel A illustrates the Multi-Agent Evolutionary Cycle, where populations at

undergo fitness assessment and reproduction via crossover and mutation of " agent genes" (e.g., prompts and tool configura- tions) to produce optimized successors. Panel B details Capability Transfer and Protocol Standardization, showing the shift from traditional FIPA/JADE frameworks to lightweight, LLM-driven capability exchange and secure migration primitives. Panel C presents Unified Evolution Architectures, specifically the MemEvolve framework, which couples an outer loop for architectural meta-evolution with an inner loop for agent learning, utilizing the modular design space of EvolveLab. Finally, Panel D synthesizes the Drivers of Swarm Intelligence, mapping biologically-inspired mechanisms and macro-scale population operations that enable non-centralized, complex problem-solving capabilities.

Figure 5: Benchmarking landscape and quantitative performance evaluation of AI agent memory systems. This figure provides a systematic analysis of the evaluative frameworks and metrics used to measure agentic performance as of 2026. Panel A categorizes the primary benchmark suites, ranging from long-context reasoning (LoCoMo) to task-oriented memory utilization (Mem2ActBench) and trustworthy testing-time evo- lution (TAME). Panel B compares Key Performance Indicators (KPIs), demonstrating that while graph-based systems like MAGMA excel in multi-session knowledge updates, bio-inspired architectures like Hippocampus offer superior token efficiency and reduced retrieval latency compared to hierarchical baselines like MemGPT. Panel C visualizes the " Accuracy-Efficiency Trade-off Challenge," illustrating the inher- ent tension between high-fidelity reasoning and resource-constrained execution. Finally, Panel D identifies critical gaps in current evaluation methodologies, emphasizing the urgent need for more interactive, diverse, and standardized indicators to support the transition toward au- tonomous AI ecosystems. interactive scenario evaluation, incomplete indicator system, and difficulty in horizontal comparison, providing direction for optimizing the benchmark testing system in the future.

In order to objectively evaluate the effectiveness of the above mechanisms, the research community has established a series of benchmark tests. However, there is still a lack of a completely unified super benchmark that covers all dimensions.

5.1 Main Benchmark Suite

• LoCoMo: focuses on long context understanding and reasoning, covering tasks such as single hop/multi hop, temporal, etc [33]. • LongMemEval: Evaluating the accuracy and efficiency of long-term memory [65]. • MemBench: Evaluate the effectiveness, efficiency, and capacity of memory ability [46]. • Evo-Memory: Focused on benchmarking LLM agents' self evolving memory, evaluating their learning, adaptation, and knowledge reuse abilities [43]. • Mem2ActBench: Evaluating task oriented autonomous agent long-term memory utilization [45]. • TAME: Proposing a trustworthy proxy memory testing time evolution method [35].

5.2 Comparison of Key Performance Indicators

Although there is a lack of a complete and unified table that includes all systems, we can extract key data points from scattered literature for comparative analysis.

Accuracy/F1: • In LoCoMo' s temporal inference task, Hippocampus has an F1 value of 38.3, while MemGPT has a value of 26.5 [1]. This indicates that the biological heuristic architecture is superior to traditional hierarchical architectures in processing time series data. • The performance comparison of MAGMA on the LongMemEval dataset shows that it outperforms Full context and Nemori baselines in multi session and knowledge update tasks [4]. • ReMem consistently outperforms the History baseline on datasets such as GPQA and TOOLBENCH [43].

Latency&Token Efficiency: • Token Consumption: On the LongMemEval-s benchmark, MemeGPT is one of the systems with the highest resource consumption, while MemoryBank is the most efficient [1]. Hippocampus maintains accuracy on LoCoMo and LongMemEval benchmarks while reducing end-to-end retrieval latency and per query token usage [1]. • Retrieval Latency: Hippocampus aims to address the high cost issues of retrieval latency and storage expansion in existing memory systems [1]. MemGPT has relatively low efficiency due to frequent context exchange [53].

Capacity&Scalability: • MemBench evaluates multiple aspects of memory capacity [46]. • Hippocampus is described as an efficient and scalable proxy memory module [1]. • SHIMI is used for scalable proxy inference [2].

Trade-offs Analysis: MemBench' s 3D scatter plot visually illustrates the performance trade-offs of the most advanced proxy memory systems in terms of F1 score, average token consumption, and average total time [1]. This indicates that achieving a perfect balance of high accuracy, low resource consumption, and low execution time remains a challenge. Usually, high accuracy (such as MAGMA' s graph structure) comes with high computational overhead, while high efficiency (such as Hippocampus'binary signature) may compromise slightly on complex semantic understanding.

5.3 Assessment Challenge

The main challenges facing the current evaluation include insufficient diversity, lack of interactive scenarios, and lack of comprehensive indicators [46]. Most benchmarks focus on static question answering and lack evaluation of dynamic environment interaction, multi round collaboration, and long-term evolutionary capabilities [48]. In addition, the coexistence of methods that indirectly evaluate the effectiveness of proxy memory through downstream tasks or directly evaluate it through memory target tasks makes it difficult to directly compare the results horizontally [47].

6 Current Status and Challenges of Industrial Deployment

Although significant progress has been made in academia from 2025 to 2026, the production deployment in industry is still in a cautious exploration stage.

6.1 Dynamics of Major Technology Companies

• Microsoft announces 2025 as the ' AI agent era' and embeds it throughout its entire stack [57]. Microsoft will use Azure AI Agent Ser- vice to build multi-agent solutions by 2025 [63]. This indicates that Microsoft is committed to making proxy capabilities infrastructure available for enterprise developers to call upon. • OpenAI: CEO Sam Altman predicts that by 2025 we may see ' the first AI agent joining the workforce and substantially changing company output' [58]. OpenAI released o3 and o4-mini systems in 2025 and launched the " Operator" project [59, 77]. However, OpenAI also considers " self replication without human intervention"as a high-risk in cutting-edge AI risk models [56]. • Google DeepMind estimates that AI with abilities at least equivalent to humans in most cognitive tasks may emerge in the "next few years"(2030), and mentions the release of Gemini 2.5 in 2025 [58,59]. Google Agent Development Kit (ADK) and Project Astra will announce their layout on the agent development toolchain in 2025 [60, 61]. • Anthropic: CEO Dario Amodei believes that in 2026-2027, there may be AI that is smarter than almost all humans in almost ev- erything [58]. Anthropic invests heavily in AGI research and focuses on the risks of autonomous replication and adaptive models (ARA) [56, 62].

6.2 Implementation Status of Production Environment

The search results show that there is no explicit mention of any specific company or organization deploying self evolving AI agent systems with memory and learning capabilities in public documents during the period of 2025-2026 [80]. Although the widespread application and trends of AI agents have been mentioned, specific deployment cases often remain at the conceptual, framework, or internal testing stage. • Enterprise Application Trend: The use of AI agents has become a mainstream practice in enterprise environments by 2025 [57]. 99% of enterprise developers are exploring AI agents [52]. Verizon will adopt Google

Gemini AI by 2025 [78]. • Obstacles to Landing: The AI2Agent case study shows that although it significantly shortens deployment time, improves success rates, and reduces errors, this is more of an automated deployment rather than a self evolving agent [79]. The true ' self-evolving' system (i.e. a system that can autonomously modify its own code or weights) has not yet been widely deployed due to security risks such as uncontrollable mutations [56]. • Infrastructure: AI agents are moving from research to production infrastructure [1]. Many fields require agents to have active memory management capabilities [44].

6.3 Safety and Ethical Risks

With the enhancement of Agent capabilities, security risks are becoming increasingly prominent. • Self-Replication Risk: OpenAI, Google DeepMind, and Anthropic consider " self replication without human intervention" as high- risk [56]. The society is still far from AI systems with self replicating capabilities. • Unpredictable: Evolutionary mechanisms may lead to agents developing behavior strategies that humans cannot understand. • Data Privacy: Persistent memory systems store a large amount of user interaction data, and ensuring that this data is not abused or leaked is a prerequisite for production deployment [38].

7.1 Technology Fusion Trend: The Rise of Unified Architecture

Research from 2025 to 2026 clearly indicates that memory, learning, and evolution are no longer isolated research directions, but are merging into Unified Agent Architects. The proposal of frameworks such as MemEvolve, UMEM, and ReMem marks our attempt to build systems that can simultaneously manage the memory lifecycle, optimize learning strategies, and adapt to environmental changes [11, 34]. The future agents will no longer be static model callers, but dynamic, self-organizing cognitive entities.

Modular design space will become the standard. As shown in EvolveLab, the memory system is disassembled into encoding, storage, retrieval, and management modules, allowing for dynamic assembly according to task requirements [54]. This flexibility is key to addressing diverse application scenarios.

7.2 From Automation to Autonomy

The current industrial deployment is mostly focused on automation tasks such as customer service and code generation, while the future goal is autonomy. This means that agents can set their own goals, plan long-range paths, and self correct when encountering obstacles. This requires memory mechanisms to support longer-term dependencies (months or even years), learning mechanisms to support more complex causal reasoning, and evolutionary mechanisms to support safer strategy exploration.

7.3 Challenges and Unsolved Mysteries

Despite the broad prospects, there are still key challenges to be addressed: 1. Lack of Standardization: Lack of unified capability transfer protocols and memory data format standards, resulting in difficulty in interoperability between different Agent systems [71, 76]. 2. The Evaluation System is Incomplete: existing benchmarks mostly focus on static tasks, lacking quantitative evaluation of dynamic evolutionary capabilities and group collaboration effects [46]. 3. Security Alignment: The biggest obstacle to production deployment is how to ensure that the behavior of agents always conforms to human values and security norms while allowing them to evolve on their own [56]. 4. Resource Efficiency: The complex memory and evolution mechanisms bring high computational and token costs, and how to deploy efficient agents on edge devices remains a challenge [1].

7.4 Conclusion

In summary, 2025-2026 is a crucial turning point for AI Agent technology to transition from being a " toy" to a " tool" . Innovations in the architecture of memory mechanisms, such as Hippocampus and MAGMA, have solved the problem of long-range dependencies; The reflective closed loop of learning mechanisms, such as Reflexion and ReMem, enhances the robustness of single point tasks; The introduction of evolutionary mechanisms such as MemEvolve and EvoAgent has paved the way for swarm intelligence and sustained adaptation. Although fully autonomous and self evolving production level systems have not yet been widely adopted, the underlying components of the technology stack are becoming increasingly mature.

For researchers, future opportunities lie in building more efficient unified architectures, developing standardized interoperability pro- tocols, and developing benchmark tests that are more closely aligned with real-world dynamics. For practitioners, the focus should be on how to securely integrate existing memory and reflection modules into business workflows, while strictly controlling risks while improving effi- ciency. The evolutionary journey of AI agents has just begun, and the interweaving of memory, learning, and evolution will eventually shape the prototype of the next generation of general artificial intelligence.

References

[1] Yi Li, Lianjie Cao, Faraz Ahmed, Puneet Sharma, and Bingzhe Li. Hippocampus: An efficient and scalable memory module for agentic AI, 2026. [2] Tooraj Helmi. Decentralizing AI Memory: SHIMI, a Semantic Hierarchical Memory Index for Scalable Agent Reasoning, [3] Gloria Felicia, Nolan Bryant, Handi Putra, Ayaan Gazali, Eliel Lobo, Esteban Rojas. From Perception to Action: Spatial AI Agents and World Models, 2026. [4] Dongming Jiang, Yi Li, Guanpeng Li, Bingzhe Li. MAGMA: A Multi-Graph based Agentic Memory Architecture for AI Agents,2026.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, F. Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, M. Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks, 2020. [6] Gaurav Srivastava, Aafiya Hussain, Chi Wang, Yingyan Celine Lin, Xuan Wang. EffGen: Enabling Small Language Models as Capable Autonomous Agents, 2026. [7] Yifeng He, Ethan Wang, Yuyang Rong, Zifei Cheng, and Hao Chen. Security of AI agents. In 2025 IEEE/ACM Interna- tional Workshop on Respon, 2025.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory, 2025. [9] Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents, 2025. [10] Xinze Li, Ziyue Zhu, Siyuan Liu, Yubo Ma, Yuhang Zang, Yixin Cao, Aixin Sun. EMemBench: Interactive Benchmark- ing of Episodic Memory for VLM Agents, 2026. [11] Guibin Zhang, Haotian Ren, Chong Zhan, Zhenhong Zhou, Junhao Wang, He Zhu, Wangchunshu Zhou, and Shuicheng Yan. MemEvolve: Meta-evolution of agent memory systems, 2025. [12] Charles Packer, Vivian Fang, Shishir G. Patil, Kevin Lin, Sarah Wooders, and Joseph Gonzalez. MemGPT: Towards LLMs as operating systems, 2023. [13] Jinyuan Fang, Yanwen Peng, Xi Zhang, Yingxu Wang, Xinhao Yi, Guibin Zhang, Yi Xu, Bin Wu, Siwei Liu, Zihao Li, Zhaochun Ren, Nikos Aletras, Xi Wang, Han Zhou, Zaiqiao Meng. A Comprehensive Survey of Self-Evolving AI Agents:

A New Paradigm Bridging Foundation Models and Lifelong Agentic Systems,2025. [14] Pengyu Zhao, Zijian Jin, and Ning Cheng. An in-depth survey of large language model-based artificial intelligence agents, [15] Shangheng Du, Jiabao Zhao, Jinxin Shi, Zhentao Xie, Xin Jiang, Yanhong Bai, Liang He. A Survey on the Optimization of Large Language Model-based Agents, 2025. [16] Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Yiran Wu, Hongru Wang, Han Xiao, Yuhang Zhou, Shaokun Zhang, Jiayi Zhang, Jinyu Xiang, Yixiong Fang, Qiwen Zhao, Dongrui Liu, Qihan Ren, Cheng Qian, Zhenhailong Wang, Minda Hu, Huazheng Wang, Qingyun Wu, Heng Ji, Mengdi Wang. A Survey of Self-Evolving Agents: On Path to Artificial Super Intelligence, 2025. [17] Noah Shinn, Federico Cassano, Beck Labash, A. Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In Neural Information Processing Systems, NeurIPS ' 23, 2023. [18] Ju Wu, Calvin K.L. Or. Position Paper: Towards Open Complex Human-AI Agents Collaboration Systems for Problem Solving and Knowledge Management, 2025. [19] Noah Shinn, Beck Labash, and A. Gopinath. Reflexion: an autonomous agent with dynamic memory and self-reflection, [20] Sahil Girhepuje, Siva Sankar Sajeev, Purvam Jain, Arya Sikder, Adithya Rama Varma, R. George, Akshay Govind Srini- vasan, Mahendra Kurup, Ashmit Sinha, Sudip Mondal. RE-GAINS & EnCHANT: Intelligent Tool Manipulation Systems For Enhanced Query Responses, 2024. [21] Siyu Yuan, Zehui Chen, Zhiheng Xi, Junjie Ye, Zhengyin Du, and Jiecao Chen. Agent-R: Training language model agents to reflect via iterative self-training, 2025. [22] Sara Candussio. A Dialectic Pipeline for Improving LLM Robustness, 2026. [23] Dasheng Bi, Yubin Hu, Mohammed N. Nasir. Real-Time Procedural Learning From Experience for AI Agents, 2025. [24] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models, 2022. [25] Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Y. Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. In Proceedings of the AAAI Conference on Artificial Intelligence, 2023. [26] Aida Rahmattalabi, Jen Jen Chung, M. Colby, and Kagan Tumer. D++: Structural credit assignment in tightly coupled multiagent domains. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2016. [27] Miquel Montaner Rigall. Collaborative recommender agents based on case-based reasoning and trust, 2003. [28] L. Bull, T. Fogarty. EVOLUTIONARY COMPUTING IN COOPERATIVE MULTI-AGENT ENVIRONMENTS, 1996. [29] Zhang Xiang-feng. Adaptive Co-evolution Model for Multi-agent System, 2010. [30] Honglin Bao, Qiqige Wuyun, and W. Banzhaf. Evolution of cooperation through genetic collective learning and imitation in multiagent societies. In 2018 IEEE Symposium on Artificial Life. [31] W. Kosinski. Advances in Evolutionary Algorithms, 2008. [32] Siyu Yuan, Kaitao Song, Jiangjie Chen, Xu Tan, Dongsheng Li, and Deqing Yang. EvoAgent: Towards automatic multi- agent generation via evolutionary algorithms. In North American Chapter of the Association for Computational Linguis- tics, 2024. [33] Xiaochen Zhao, Kaikai Wang, Xiaowen Zhang, Chen Yao, and Aili Wang. HyMem: Hybrid memory architecture with dynamic retrieval scheduling, 2026. [34] Yongshi Ye, Hui Jiang, Feihu Jiang, Tian Lan, Yichao Du, Biao Fu, Xiaodong Shi, Qianghuai Jia, Longyue Wang, Weihua Luo. UMEM: Unified Memory Extraction and Management Framework for Generalizable Memory, 2026. [35] Yu Cheng, Jiuan Zhou, Yongkang Hu, Yihang Chen, Huichi Zhou, Mingang Chen, Zhizhong Zhang, Kun Shao, Yuan Xie, Zhaoxia Yin. TAME: A Trustworthy Test-Time Evolution of Agent Memory with Systematic Benchmarking, 2026. [36] Henri Avancini, A. Amandi, and F. D. C. Exactas. A Java Framework for Multi-agent Systems, 2000. [37] J. T. UniversiteitAmsterdamtreur. Implementing Multi-Agent Systems: Languages, Frameworks, and Standards [Ex- tended Abstract], 1998. [38] N. Khalid, G. Tahir, and P. Bloodsworth. Persistent and scalable JADE: A cloud-based in-memory multi-agent framework, [39] Nikolas Antzoulatos, Elkin Castro, Lavindra de Silva, A. Rocha, S. Ratchev, and J. Barata. A multi-agent framework for capability-based reconfiguration of industrial assembly systems. International Journal of Production Research, 2017. [40] Nikolaos I. Spanoudakis and Pavlos Moraitis. An Agent Modeling Language Implementing Protocols through Capabili- ties. In 2008 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2008. [41] Stefan Philipps, Jürgen Lind. Ein System zur Definition und Ausführung von Protokollen für Multi-Agentsysteme, 1999. [42] R. M. Barbosa. Especificação formal de organizações de sistemas multiagentes, 2011. [43] Tianxin Wei, Noveen Sachdeva, Benjamin Coleman, Zhankui He, Yuanchen Bei, Xuying Ning, Mengting Ai, Yunzhe Li, Jingrui He, Ed H. Chi, Chi Wang, Shuo Chen, Fernando Pereira, Wang-Cheng Kang, and Derek Zhiyuan Cheng.

Evo-Memory: Benchmarking LLM agent test-time learning with self-evolving memory, 2025. [44] 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,Zhenrong Cheng,Xuanbo Fan,Jiaxin Guo,Xinlei Yu,Zhenhong Zhou,Zewen Hu,Jiahao Huo,Junhao Wang,Yuwei Niu,Yu Wang,Zhenfei Yin,Xiaobin Hu,Yue Liao,Qiankun Li,Kun Wang,Wangchunshu Zhou,Yixin Liu,Dawei Cheng,Qi Zhang,Tao Gui,Shirui Pan,Yan Zhang,Philip Torr,Zhicheng Dou,Ji-Rong Wen,Xuanjing Huang,Yu-Gang Jiang,Shuicheng Yan. Memory in the Age of AI Agents, 2025. [45] Yiting Shen, Kun Li, Wei Zhou, Songlin Hu. Mem2ActBench: A Benchmark for Evaluating Long-Term Memory Uti- lization in Task-Oriented Autonomous Agents, 2026. [46] Haoran Tan, Zeyu Zhang, Chen Ma, Xu Chen, Quanyu Dai, and Zhenhua Dong. MemBench: Towards more comprehen- sive evaluation on the memory of LLM-based agents, 2025. [47] Xiaofang Yang, Lijun Li, Heng Zhou, Tong Zhu, Xiaoye Qu, Yuchen Fan, Qianshan Wei, Rui Ye, Li Kang, Yiran Qin, Zhiqiang Kou, Daizong Liu, Qi Li, Ning Ding, Siheng Chen, Jing Shao. Toward Efficient Agents: Memory, Tool learning, and Planning, 2026. [48] Jiachen Zhu, Menghui Zhu, Renting Rui, Rong Shan, Congmin Zheng, Bo Chen, Yunjia Xi, Jianghao Lin, Weiwen Liu, Ruiming Tang, Yong Yu, Weinan Zhang. Evolutionary Perspectives on the Evaluation of LLM-Based AI Agents: A Comprehensive Survey, 2025. [49] Songze Li, Ruishi He, Xiaojun Jia, Jun Wang, Zhihui Fu. Knowledge-Driven Multi-Turn Jailbreaking on Large Language Models, 2026. [50] Yaolun Zhang, Yiran Wu, Yijiong Yu, Qingyun Wu, Huazheng Wang. Live-Evo: Online Evolution of Agentic Memory from Continuous Feedback, 2026. [51] Mengyuan Chen, Chengjun Dai, Xinyang Dong, Chengzhe Feng, Kewei Fu, Jianshe Li, Zhihan Peng, Yongqi Tong, Junshao Zhang, Hong Zhu. Dingtalk DeepResearch: A Unified Multi Agent Framework for Adaptive Intelligence in Enterprise Environments, 2025. [52] Woong Shin,Renan Souza,Daniel Rosendo,Fr' ed' eric Suter,Feiyi Wang,Prasanna Balaprakash,Rafael Ferreira da Silva.

The (R)evolution of Scientific Workflows in the Agentic AI Era: Towards Autonomous Science, 2025. [53] Haoran Tan, Zeyu Zhang, Chen Ma, Xu Chen, Quanyu Dai, and Zhenhua Dong. MemBench: Towards More Compre- hensive Evaluation on the Memory of LLM-based Agents, 2025. [54] Guibin Zhang, Haotian Ren, Chong Zhan, Zhen Zhou, Junhao Wang, He Zhu, Wangchunshu Zhou, Shuicheng Yan.

MemEvolve: Meta-Evolution of Agent Memory Systems, 2025. [55] Zixia Jia, Jiaqi Li, Yipeng Kang, Yuxuan Wang, Tong Wu, Quansen Wang, Xiaobo Wang, Shuyi Zhang, Junzhe Shen, Qing Li, Siyuan Qi, Yitao Liang, Di He, Zilong Zheng, Song-Chun Zhu. The AI Hippocampus: How Far are We From Human Memory? 2026. [56] Xudong Pan, Jiarun Dai, Yihe Fan, Minyuan Luo, Changyi

Li, and Min Yang. Large language model-powered AI systems achieve self-replication with no human intervention, 2025. [57] Kaiyu Zhou, Yongsen Zheng, Yicheng He, Meng Xue, Xueluan Gong, Yuji Wang, Kwok-Yan Lam. Beyond Max Tokens:

Stealthy Resource Amplification via Tool Calling Chains in LLM Agents, 2026. [58] Charlotte Stix, Matteo Pistillo, Girish Sastry, Marius Hobbhahn, Alejandro Ortega, Mikita Balesni, Annika Hallensleben, Nix Goldowsky-Dill, Lee Sharkey. AI Behind Closed Doors: a Primer on The Governance of Internal Deployment, 2025. [59] Jonathan Kutasov, Yuqi Sun, Paul Colognese, Teun van der Weij, Linda Petrini, Chen Bo Calvin Zhang, John Hughes, Xi- ang Deng, Henry Sleight, Tyler Tracy, Buck Shlegeris, Joe Benton. SHADE-Arena: Evaluating Sabotage and Monitoring in LLM Agents, 2025. [60] Mathieu Andreux, Märt Bakler, Yanael Barbier, Hamza Benchekroun, Emilien Biré, Antoine Bonnet, Riaz Bordie, Nathan Bout, Matthias Brunel, Aleix Cambray, Pierre-Louis Cedoz, Antoine Chassang, Gautier Cloix, Ethan Connelly, Alexandra Constantinou, Ramzi De Coster, Hubert de la Jonquiere, Aurélien Delfosse, Maxime Delpit, Alexis Deprez, Augustin Derupti, Mathieu Diaz, Shannon D' Souza, Julie Dujardin, Abai Edmund, Michael Eickenberg, Armand Fatalot, Wissem Felissi, Isaac Herring, Xavier Koegler, Erwan Le Jumeau de Kergaradec, Aurélien Lac, Maxime Langevin, Corentin Lauverjat, Antonio Loison, Avshalom Manevich, Axel Moyal, Axel Nguyen Kerbel, Marinela Parovic, Julien Revelle, Guillaume Richard, Mats Richter, Ronan Riochet, María Santos, Romain Savidan, Laurent Sifre, Maxime Theillard, Marc Thibault, Ivan Valentini, Tony Wu, Laura Yie, Kai Yuan, Jevgenij Zubovskij. Surfer 2: The Next Generation of Cross-Platform Computer Use Agents, 2025. [61] Xinyi Hou, Yanjie Zhao, Haoyu Wang. LLM Applications: Current Paradigms and the Next Frontier, 2025. [62] Yeliz Figen Doker. The Wicked Nature of AGI. Law, Technology and Humans, 2025. [63] Margaret Mitchell, Avijit Ghosh, Alexandra Sasha Luccioni, Giada Pistilli. Fully Autonomous AI Agents Should Not be Developed, 2025. [64] Yifei Li, Weidong Guo, Lingling Zhang, Rongman Xu, Muye Huang, Hui Liu;Lijiao Xu, Yu Xu, Jun Liu. Locomo-Plus:

Beyond-Factual Cognitive Memory Evaluation Framework for LLM Agents, 2026. [65] Yaxiong Wu, Yongyue Zhang, Sheng Liang, Yong Liu. SGMem: Sentence Graph Memory for Long-Term Conversational Agents, 2025. [66] F. Bellifemine, A. Poggi, and G. Rimassa. Developing multi-agent systems with JADE, 2007. [67] Eleonora Iotti. An agent-oriented programming language for JADE multi-agent systems, 2018. [68] F. Bellifemine, A. Poggi, G. Rimassa. Jade - a fipa-compliant agent framework, 1999. [69] G. Kardas, Emine Bircan, and Moharram Challenger. Supporting the platform extensibility for the model-driven devel- opment of agent systems by the interoperability between domain-specific modeling languages of multi-agent systems.

Comput. Sci. Inf. Syst., 2017. [70] Giovanni Caire. JADE: Java agents development framework, 2012. [71] Esteban León-Soto. Agent communication using web services, a new FIPA message transport service for Jade, 2007. [72]

Huimin Peng. A Comprehensive Overview and Survey of Recent Advances in Meta-Learning, 2020. [73] Tianxin Wei, Ting-Wei Li, Zhining Liu, Xuying Ning, Ze Yang, Jiaru Zou, Zhichen Zeng, Ruizhong Qiu, Xiao Lin, Dongqi Fu, Zihao Li, Mengting Ai, Duo Zhou, Wenxuan Bao, Yunzhe Li, Gaotang Li, Cheng Qian, Yu Wang, Xiangru Tang, Yin Xiao, Liri Fang, Hui Liu, Xianfeng Tang, Yuji Zhang, Chi Wang, Jiaxuan You, Heng Ji, Hanghang Tong, Jingrui He. Agentic Reasoning for Large Language Models, 2026. [74] Shengtao Zhang, Jiaqian Wang, Ruiwen Zhou, Junwei Liao, Yuchen Feng, Weinan Zhang, Ying Wen, Zhiyu Li, Feiyu Xiong, Yutao Qi, Bo Tang, Muning Wen. MemRL: Self-Evolving Agents via Runtime Reinforcement Learning on Episodic Memory, 2026. [75] Jiaxing Chen, Weilin Yuan, Shaofei Chen, Zhenzhen Hu, Peng Li. Evo-MAML: Meta-Learning with Evolving Gradient, [76] Vaibhav Tupe, Shrinath Thube. AI Agentic workflows and Enterprise APIs: Adapting API architectures for the age of AI agents, 2025. [77] Laurie Hughes, Yogesh K. Dwivedi, Tegwen Malik, Mazen Shawosh, M. Albashrawi, Il Jeon, Vincent Dutot, Mandanna Appanderanda, Tom Crick, Rahul De' , Mark Fenwick, Senali Madugoda Gunaratnege, Paulius Jurcys, A. Kar, N. Kshetri, Keyao Li, Sashah Mutasa, Spyridon Samothrakis, Michael Wade, and Paul Walton. AI Agents and Agentic Systems: A Multi-Expert Analysis. Journal of Computer Information Systems, 2025. [78] Segev Shlomov,Alon Oved,Sami Marreed,Ido Levy,Offer Akrabi,Avi Yaeli,Lukasz Strak,Elizabeth Koumpan,Yinon Goldshtein,Eilam Shapira,Nir Mashkif,Asaf Adi. From Benchmarks to Business Impact: Deploying IBM Generalist Agent in Enterprise Production, 2025. [79] Jiaxiang Chen,Jingwei Shi,Lei Gan,Jiale Zhang,Qingyu Zhang,Dongqian Zhang,Xin Pang,Zhucong Li,Yinghui Xu.

AI2Agent: An End-to-End Framework for Deploying AI Projects as Autonomous Agents, 2025. [80] Yuqing Zhou,Zhuoer Wang,Jie Yuan,Hong Wang,Samson Koelle,Ziwei Zhu,Wei Niu. WISE-Flow: Workflow-Induced Structured Experience for Self-Evolving Conversational Service Agents, 2026.

Note: Figure translations are in progress. See original paper for figures.