SelfEvolveAgent: Continual Learning in AI Agents using Memory

Hiva Mohammadzadeh (Zaad)
Stanford University
hiva@stanford.edu

Shayan Talaei (TBD)**
Stanford University

1 Introduction

The rapid advancement of large language model agents has created opportunities for building autonomous systems capable of complex reasoning and decision-making, from web browsing to software engineering. These agents demonstrate remarkable capabilities in reasoning through complex problems, and executing multi-step tasks. However, existing agents largely fail to learn from their experiences, forcing them to repeat past errors and discard valuable insights. This constraint manifests as a critical bottleneck in the development of truly autonomous systems. Agents fail to transfer successful strategies across similar tasks, and cannot develop increasingly sophisticated reasoning capabilities over time. This inability to learn from experience not only limits performance but also prevents the emergence of the kind of cumulative intelligence that characterizes effective human learning and problem-solving. The recent introduction of ReasoningBank represents a breakthrough in addressing these limitations through a novel memory framework that distills generalizable reasoning strategies from an agent's self-judged successful and failed experiences. By integrating memory with test-time scaling, ReasoningBank demonstrates substantial improvements across challenging benchmarks. However, the gap between theoretical frameworks and practical, scalable implementations remains substantial, limiting broader adoption and further research development.

2 Motivation

This project addresses critical gaps in current landscape of agent development and continual learning research. While ReasoningBank demonstrates the potential of memory driven agents, there are some limitations that motivate our investigation into enhancing the performance of such technique.

One major limitations is the lack of comprehensive, open-source implementations which limits broader research adoption. Existing implementations are often proprietary, incomplete, or lack the documentation necessary for reproducibility and extension by the research community. Another limitation is that current memory systems for LLM agents often rely on simple storage mechanisms (vector or graph databases) that do not scale effectively with prolonged use. As agents encounter thousands of experiences over extended deployments, naive memory architectures become inefficient, leading to degraded retrieval performance and increased storage costs. Finally, existing agent memory systems typically operate within single domains, failing to leverage cross-domain transfer of reasoning strategies. An agent's successful web browsing strategies could potentially inform software engineering tasks, but current systems lack mechanisms for such knowledge transfer.

The convergence of these limitations creates a significant opportunity. By developing a comprehensive repository that implements and extends ReasoningBank's innovations while addressing scalability and transfer challenges, we can accelerate research in self evolving agents and enable practical deployment of continually learning autonomous systems.

3 Related Literature

Memory augmentation for LLMs has gained significant attention, with approaches ranging from external knowledge retrieval to episodic memory systems. ReasoningBank introduces a novel memory framework that distills generalizable reasoning strategies from an agent's self-judged successful and failed experiences, proposing Memory-Aware Test-Time Scaling (MaTTS) that integrates memory with test-time compute scaling, achieving substantial improvements across WebArena, Mind2Web, and SWE-Bench-Verified benchmarks.

A Survey of Continual Reinforcement Learning provides a comprehensive examination of Continual Reinforcement Learning (CRL), proposing a new taxonomy that categorizes methods into policy-focused, experience-focused, dynamic-focused, and reward-focused approaches. The survey highlights the balance among plasticity, stability, and scalability in CRL systems and identifies key challenges including catastrophic forgetting and knowledge transfer across sequential tasks.

StreamBench introduces the first benchmark designed to evaluate continuous improvement of LLM agents over input-feedback sequences. The framework simulates online learning environments where agents receive continuous feedback streams and iteratively enhance performance through updating prompt templates, retrievers, memory, or model parameters. StreamBench addresses the gap in existing benchmarks that evaluate only innate capabilities rather than learning over time.

Lifelong Learning of LLM Agents Roadmap systematically categorizes lifelong learning LLM agents into three core modules: perception (multimodal input integration), memory (storing and retrieving evolving knowledge), and action (grounded environment interactions). The survey highlights techniques including memory replay, retrieval-augmented generation, and self-reflection to prevent catastrophic forgetting while enabling continuous adaptation in dynamic environments.

4 Contributions

Our key contributions include:

1. First comprehensive, open-source implementation of ReasoningBank's core framework, including memory schema design, and Memory-Aware Test-Time Scaling.
2. Evaluation pipelines for WebArena, Mind2Web, and SWE-Bench-Verified, and other datasets with performance analysis against reported baselines.
3. Algorithms for compressing and abstracting memory content while preserving reasoning utility, enabling long-term deployment.
4. Mechanisms for transferring reasoning strategies learned in one domain to related domains, expanding the generalizability and practical applicability of memory-driven agent learning.

5 Ideal Results and Evaluation

We aim to replicate the RasoningBank's performance as our State of the art baseline. We want to achieve 14.5% improvement on WebArena, 23.0% improvement on Mind2Web, and 8.8% improvement on SWE-Bench-Verified compared to baseline ReAct agents. We plan to expand this to other Software engineering and reasoning tasks, as well as additional Large language Models.

Ideally, we also achieve additional improvement through efficient memory storage and demonstrate improvement on cross-domain memory transfer tasks compared to domain-isolated learning.

6 Project Plan

Week 1	Project setup, environment configuration, No memory baseline implementation.
Week 2	ReasoningBank implementation and replication of results in the paper.
Week 3	**Midterm presentation & deliverables:** Functional basic agent without memory enhancement ("No Memory" baseline); Implementation of the ReasoningBank WebArena results with a working memory storage system; Replication of accuracies shown in the ReasoningBank paper.
Weeks 4–6	Explore improvements to the memory storage.
Weeks 7–9	Exploring improvements in cross-domain memory transfer.
Week 11	Final evaluation and optimization.
Week 12	**Final report & presentation & deliverables:** Potential improvements to the memory storage and retrieval systems; Experiments to show if basic cross-domain memory transfer functionality helps accuracy.
Compute:	API credits for Google Gemini and Anthropic Claude and OpenAI GPT-5 models; AWS credits for hosting WebArena.

7 References

Shunyu Yao, Google. 2025. ReAct: Synergizing Reasoning and Acting in Language Models. https://arxiv.org/abs/2210.03629

Google. (2025). ReasoningBank: Scaling Agent Self-Evolving with Reasoning Memory. https://arxiv.org/abs/2509.25140v1

Chaofan Pan, IEEE. 2025. A Survey of Continual Reinforcement Learning. https://arxiv.org/abs/2506.21872

Cheng-Kuang Wu, Appier AI Research. 2024. StreamBench: Towards Benchmarking Continuous Improvement of Language Agents. https://arxiv.org/abs/2406.08747v1

Junhao Zheng, IEEE. 2025. Lifelong Learning of Large Language Model based Agents: A Roadmap. https://arxiv.org/pdf/2501.07278v1