From Static LLMs to Lifelong Intelligence

A Roadmap for Self-Evolving Agent via Evo-Memory

凌川

Shanghai Jiao Tong University

2026 年 4 月 7 日

Overview

Why Evolve? From Static to Autonomous Agents

- **Break the Static Bottleneck**
  LLMs are frozen post-training; evolution enables adaptation to novel tasks and dynamic contexts.
- **Shift the Locus of Autonomy**
  From human-curated updates to self-directed improvement from real-time experience and interaction.
- **Enable Continuous Growth**
  Autonomous evolution of **Models, Memory, and Tools** for long-term robustness and specialized expertise.
- **The Bridge to *ASI***
  A fundamental requirement for reaching Artificial Super Intelligence through recursive self-improvement.

A Map of Self-Evolving Agent

Static LLMs $\rightarrow$ Continuous Self-Improvement

1. WHAT to Evolve

- **Model** (Weights)
  [SCoRe, SCA]
- **Context** (Memory)
  [Mem0, **Evo-Memory**]
- **Tools** (Skills/Code)
  [Voyager, Alita]

2. WHEN to Evolve

- **Intra-task** (Step-level)
  [Reflexion, AdaPlanner]
- **Inter-task** (Lifelong)
  [LifelongBench, **Evo-Memory**]

3. HOW to Evolve

- **Reward-based** (Env)
  [TextGrad, **Evo-Memory**]
- **Imitation** (Self-train)
  [Self-Instruct, STaR]
- **Population-based**
  [AFlow, DGM]

*Focus: **Evo-Memory** bridges **Context**, **Inter-task**, and **Reward**.*

Motivation: Filling the Critical Gaps

WHAT **Facts/Skills → Strategies**

- Gap: Existing benchmarks (*StreamBench*, *Lifelong-Bench*) focus on retention, not **trajectory reuse**.

WHEN **Static Consistency → Deployment Evolution**

- Gap: Current dialogue studies test **staying the same**, not **becoming better** over task streams.

HOW **Passive Buffers → Structured Updates**

- Gap: Lack of modeling for **memory integration** and **refinement** mechanisms.

Existing agents **remember**, but they do not **evolve**.

Positioning Evo-Memory on the Map

1. WHAT: Context (Memory)

- **Object:** The agent's memory pool (keeping model weights frozen).
- **Key Shift:** From static fact memorization to dynamic **Experience Reuse** (reasoning strategies).

2. WHEN: Inter-test-time

- Continuous sequential task streams (**Inter-test-time**).
- Updating memory *between* tasks based on intra-task feedback.

3. HOW: Synthesis & Refinement

- **Mechanism:** Using text-based feedback to prune, synthesize, and rewrite past trajectories.
- **Methodology: ExpRAG** (Retrieval baseline) and **ReMem** (Action-Think-Refine pipeline).

WHAT to Evolve: Fact vs. Experience

Traditional RAG (Static)

Goal: Conversational Recall

- **Stored:** Facts and dialogue history.
- **Example:** "Solutions to $2x^2 + 3x - 1 = 0$."
- **Limitation:** Remembers *what was said*, but not *how to solve it*.

Evo-Memory (Dynamic)

Goal: Experience Reuse

- **Stored:** Procedural knowledge and strategies.
- **Example:** "Using the quadratic formula."
- **Advantage:** Abstracts reasoning paths to solve *future, unseen* problems.

The Essence of "Memory"

Evolution requires accumulating **HOW** to reason, not just **WHAT** was observed.

WHEN to Evolve: A New Testing Expectation

Core Presupposition

Evo-Memory explicitly evaluates an agent's capability for **Inter-test-time Self-Evolution** across a stream of tasks.

Traditional (Static)

- Isolated Evaluation
- **Timing:** Independent task sampling.
- **Context:** Reset after each episode.
- **Result:** Zero cross-task learning.

Evo-Memory (Dynamic)

- Continuous Evolution
- **Timing:** Constructed task streams.
- **Context:** Memory carries over $t \to t+1$.
- **Result:** Generalizes strategies over time.

WHEN to Evolve: Instantiating the Benchmark

Meeting the Expectation

To **satisfy the expectation of Test-time Evolution**, the paper formalizes a unified pipeline and provides two representative implementations.

Implementation 1: ExpRAG

- **Type:** Retrieval-based Memory.
- **Timing:** Task-level (*Inter-test-time*).
- **Evolution:** Appends completed task trajectories into a global pool.
- **Focus:** Passive experience retrieval.

Implementation 2: ReMem

- **Type:** Agentic Self-Evolving Memory.
- **Timing:** Task-level (*Inter-test-time*)+Step-level (*Intra-test-time*).
- **Evolution:** Continuously distills feedback during execution.
- **Focus:** Active experience refinement.

HOW to Evolve: Reward-based Evolution

Positioning in the Map: Reward-based (Text signals)

Instead of Imitation or Population-based methods, the paper exclusively utilizes **environmental feedback (rewards/text)** to drive the evolution of Context.

The Formal Problem Setting

- **Task Stream:** Sequence of queries $X = \{x_1, \ldots, x_T\}$.
- **Feedback:** Action $\hat{y}_t$ yields environment feedback f_t.
- **Core Goal:** Translate f_t into experience, evolving $M_t \to M_{t+1}$.

The Unified Pipeline

- **Search:** Retrieves history $R_t = \mathcal{R}(M_t, x_t)$ via input x_t.
- **Synthesis:** Forms context $\tilde{C}_t = \mathcal{C}(x_t, R_t)$ to output $\hat{y}_t$.
- **Evolve:** Builds entry $m_t = h(x_t, \hat{y}_t, f_t)$ to update $M_{t+1} = \mathcal{U}(M_t, m_t)$.

HOW to Evolve: Paradigms for Future Exploration

Operationalizing the Pipeline

The paper instantiates the evolution pipeline with two **directional paradigms**, showcasing the leap from passive retrieval to active agentic reasoning.

Paradigm 1: Passive Retrieval

- **Mechanism:** Standard In-Context Learning augmented by Top-k retrieval.
- **Update Rule:** $M_{t+1} = M_t \cup \{\text{trajectory}_t\}$.
- **Nature:** Passive accumulation. It blindly appends raw experiences, making it highly vulnerable to noise poisoning.

Paradigm 2: Active Agentic Reasoning

- **Mechanism:** Formulates memory evolution as an active Markov Decision Process (MDP).
- **Action Space:** $a_t \in \{\text{Think}, \text{Act}, \textbf{Refine}\}$.
- **Nature:** Active orchestration. The agent autonomously distills feedback and prunes noise *during* execution.

Empirical Insights: Evaluation Overview

From HOW to Empirical Validation

Based on the proposed paradigms for memory evolution, we now test whether they lead to gains in **effectiveness**, **robustness**, and **learning dynamics**.

1. Overall Effectiveness & Efficiency

RQ1 + RQ2: Do evolving-memory agents achieve stronger performance and reduce repeated trial-and-error through experience reuse?

2. Robustness under Realistic Streams

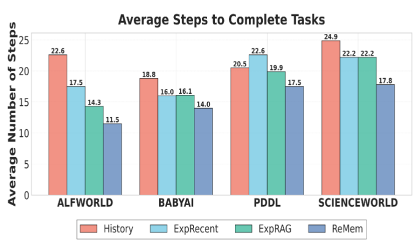
RQ3 + RQ4: Are these memory evolution mechanisms robust to task ordering and noisy failed trajectories?

3. Learning Dynamics over Time

RQ5: Do the gains of self-evolving memory accumulate as the task stream progresses?

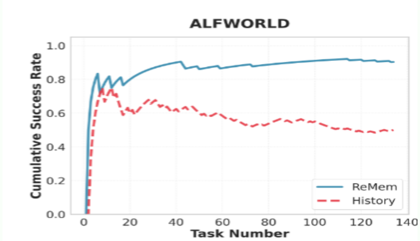
Empirical Insights: Evidence-based Answers

RQ1 & RQ2: Effectiveness & Efficiency



- Success: **0.24** → **0.78** (Claude 3.7)
- Steps: **22.6** → **11.5** (AlfWorld)

RQ5: Learning Dynamics over Time



- Continuous improvement along task streams.
- Stable retention of procedural knowledge.

RQ3 & RQ4: Robustness to Order & Feedback Noise

- **Order Sensitivity (RQ3):** Stable performance across easy-to-hard / hard-to-easy task sequences.
- **Noise Handling (RQ4):** *Active Refinement* is crucial to prevent Memory Poisoning from failed trajectories.

Conclusion: Toward Lifelong Intelligence

From Static LLMs to Lifelong Intelligence

Evo-Memory shows a concrete pathway toward self-evolving agents: evolving **memory as experience**, **across task streams**, through **feedback-driven refinement**.

Three Takeaways

- **WHAT:** Memory should support **experience reuse**, shifting from static fact recall to reusable procedural knowledge.
- **WHEN:** Evolution should happen **during deployment**, with gains accumulating across sequential task streams.
- **HOW:** Stable self-improvement requires **active refinement**, rather than passively accumulating raw trajectories.

Limitations of Evo-Memory

Open Questions for Self-Evolving Agents

Evo-Memory provides a strong starting point, but several challenges remain before memory evolution can support truly lifelong agents.

Three Directions for Improvement

- **WHAT & HOW (The RAG Bottleneck):** Evo-Memory relies on textual RAG to bridge the memory pool and the limited context window. However, **semantic similarity** $\neq$ **logical relevance**. Keyword-based retrieval introduces "Semantic Noise" that degrades reasoning.
- **WHEN & HOW (The Granularity Bottleneck):** Evolution is currently triggered at the **Trajectory-level**. This holistic approach lacks the capability for **fine-grained (atomic) experience reuse**. It is difficult for the agent to extract and apply a specific, highly reusable sub-skill from a long execution process to new scenarios.
- **DOMAIN (The Semantic Trap):** Reuse is brittle across heterogeneous environments (e.g., Kitchen $\rightarrow$ Lab) because the memory is tied to domain-specific vocabulary rather than abstract causal logic.

Future Directions: Beyond Evo-Memory

Open Questions for Self-Evolving Agents

From the limitations of Evo-Memory, there are perhaps several directions we could explore for more robust lifelong intelligence.

Three Conjectures to Explore in the Future

- **WHAT & HOW: Progressive Unfolding via Skills.** Structure memory as callable skill modules to bypass context limits.
- **WHEN & HOW: Atomic Experience Decoupling.** Shift from holistic trajectory memorization to extracting and recombining **step-level atomic strategies**.
- **DOMAIN: Logic-based Causal Invariance.** Transfer the underlying **task logic and topology** rather than brittle textual keywords.

Selected References

- **Wei, T., et al. (2025).** Evo-Memory: Benchmarking LLM Agent Test-time Learning with Self-Evolving Memory. *arXiv preprint arXiv:2511.20857*.
- **Liu, S., et al. (2025).** A Survey of Self-Evolving Agents: On Path to Artificial Super Intelligence. *arXiv preprint arXiv:2507.21046*.
- **Zhang, G., et al. (2025).** A Comprehensive Survey of Self-Evolving AI Agents: A New Paradigm Bridging Foundation Models and Lifelong Agentic Systems. *arXiv preprint arXiv:2508.07407*.

Note: Only core references are listed.

Thank you all!