

# MASkills: Continual Skills Optimization for Multi-Agent LLM Systems

Huaiyuan Yao<sup>1</sup> Xiaouu Liu<sup>1</sup> Charles Fleming<sup>2</sup> Tianlong Chen<sup>3</sup> Hua Wei<sup>1</sup>

<sup>1</sup> Arizona State University {huaiyuan, xiaoouli, hua.wei}@asu.edu

<sup>2</sup> Cisco Research chfleming@cisco.com

<sup>3</sup> University of North Carolina at Chapel Hill tianlong@cs.unc.edu

## Abstract

LLM-based multi-agent systems have shown strong performance on complex tasks, yet continual improvement from interaction experience remains challenging. Existing self-reflection methods build experience memories, but memories are mostly hard to invoke, refine, or scale, while agent skills offer a more actionable unit: structured procedural knowledge that specifies when to act, how to act, and which resources or tools to use. We introduce **MASkills**, a continual learning framework that optimizes multi-agent LLM systems through agent skills. MASkills presents a new agent-optimization pipeline that integrates skill-conditioned credit assignment, hierarchical credit aggregation, and momentum-smoothed optimization, enabling agent skill libraries to evolve through refinement, induction, consolidation, and pruning. Experiments on HotpotQA, LoCoMo, and GAIA demonstrate the effectiveness of MASkills across multiple agentic tasks. Our code is available at <https://github.com/DaRL-GenAI/MASkills>

## 1 Introduction

LLM-based multi-agent systems have emerged as a practical paradigm for solving complex tasks through coordination, role specialization, and long-horizon interaction (Liu et al., 2026; Yao et al., 2025a; Zhao et al., 2025). However, how to continually improve a multi-agent LLM system over time remains challenging. Existing methods typically use self-reflection to construct an experience memory (Tsui, 2025; Yao et al., 2025b). Such memories can preserve useful experience, but they remain a weak basis for continual improvement: they record what happened in past trajectories, but not which action policy should be reused; they lack reliable invocation conditions; and as the memory grows, useful lessons become mixed with noisy, redundant, or stale free-form records (Zhang et al., 2024).

A more useful unit for continual improvement is a *skill*. According to Anthropic (Anthropic, 2025), a skill is a structured package of procedural knowledge that tells an agent how to perform a class of tasks, including when to invoke it, how to act, which resources or tools to use. Skills make experience reusable: agents can discover relevant skills from lightweight descriptions, load detailed instructions and resources on demand, and refine them as new experience accumulates. This progressive-disclosure design makes skills more scalable and actionable than large unstructured memory stores. Recent methods such as MEMSKILL (Zhang et al., 2026), EVOSKILL (Alzubi et al., 2026), and POLYSKILL (Yu et al., 2026) develop such reusable procedural abstractions, but remain predominantly single-agent and do not optimize skills under the coordination dynamics that determine team-level utility in multi-agent systems.

Extending skill evolution to multi-agent settings requires agents to refine their skills through interaction, so that skill updates reflect not only individual outcomes but also their contributions to team-level coordination. This goal raises three concrete challenges: **(1) Skill-level credit assignment.** Team rewards should be attributed not only to the correct agent, but also to the correct skill. In multi-agent settings, a skill may only be invoked at a few timesteps, and its effect is entangled with the behaviors and skills of other agents. As a result, team-level or even agent-level feedback is insufficient to determine which specific skills should be improved. **(2) Noisy and heterogeneous interaction.** Language-based feedback extracted from trajectories is often noisy, inconsistent, and unstable across learning cycles. Moreover, multi-agent systems may operate under very different coordination structures, such as centralized planning or decentralized peer communication, which induce different patterns of interaction dependency and credit propagation. **(3) Discrete and high-risk**

**skill updates.** Unlike parameter vectors, skills are discrete language artifacts such as SKILL.md files, scripts, and references. Skill evolution, therefore, requires open-ended structural modifications rather than smooth numerical updates. Incorrect edits may directly alter the agent’s effective action space, making continual skill optimization inherently high-risk and difficult to stabilize.

To address these challenges, we introduce **MASkills**, a language-based continual learning framework that performs policy improvement directly in agents’ skill spaces rather than parameter spaces. Under decentralized execution, each agent invokes reusable skill artifacts as part of its latent policy reasoning, while the system continually evolves these skills through language-space optimization. MASkills adopts a unified language-space policy-gradient-style framework, following the structural interpretation of TextGrad (Yuksekgonul et al., 2025) and LangMARL (Yao et al., 2026a), with three components corresponding to the challenges above: (1) skill-conditioned credit assignment, where a critic attributes trajectory feedback to specific skill invocations through counterfactual comparison; (2) hierarchical credit aggregation and momentum-smoothed optimization, which stabilize noisy feedback across trajectories, agents, skills, and interaction topologies; and (3) credit-driven skill-space optimization, where stable credit signals drive skill refinement, induction, consolidation, and pruning, together with held-out validation and rollback mechanisms.

Our contributions are summarized as follows: (1) We introduce a new paradigm for continual multi-agent LLM learning that casts policy improvement as optimization over reusable skill spaces distilled from interaction traces. (2) We propose **MASkills**, a continual learning framework for multi-agent skill optimization that combines skill-conditioned credit assignment, aggregation, and optimization to evolve agent skill libraries through refinement, induction, consolidation, and pruning, together with held-out validation and rollback for stable updates. (3) We evaluate MASkills on HotpotQA, LoCoMo, and GAIA, demonstrating strong performance across memory and agentic tasks.

## 2 Related Work

**Single-agent skill discovery.** VOYAGER (Wang et al., 2023) pioneered an ever-growing skill library for an embodied LLM agent. Recent work has

extended this paradigm in three complementary directions. MEMSKILL (Zhang et al., 2026) treats memory operations themselves as a learnable skill bank: a controller selects a Top- $K$  subset of skills per text span, an LLM executor applies them, and a designer periodically revises the bank from clustered hard cases. EVOSKILL (Alzubi et al., 2026) casts skill discovery as iterative textual feedback descent: an Executor runs tasks, a Proposer diagnoses failures, and a Skill-Builder materializes proposals into structured SKILL.md folders, retained on a Pareto frontier only if they improve held-out validation. POLYSKILL (Yu et al., 2026) introduces polymorphic abstraction, separating a skill’s abstract goal from its concrete site-specific implementation, enabling cross-website transfer. However, these methods assume single-agent feedback and cannot directly attribute team-level outcomes to interacting agents and skills.

**Multi-agent LLM systems.** LLM multi-agent frameworks such as AutoGen (Wu et al.), MetaGPT (Hong et al., 2024), agentic neural networks (Ma et al., 2026), and symbolic learning (Zhou et al., 2024) have demonstrated that division of labor among LLMs improves coverage on complex tasks. Most of these systems rely on hand-engineered roles and prompts (Yao et al., 2026b; Chen et al., 2026; Hu et al., 2026); the few that self-improve do so by either rewriting prompts globally such as TextGrad (Yuksekgonul et al., 2025), DSPy (Khatab et al., 2023) or sharing a memory like Reflexion (Shinn et al., 2023). LangMARL (Yao et al., 2026a) introduced explicit credit assignment into language-space optimization.

## 3 Problem Formulation

We formalize MASkills on top of a cooperative multi-agent setting, a skill artifact, and a language-based notion of credit.

**Cooperative multi-agent setting.** We model a team of  $N$  LLM agents as a Dec-POMDP  $\mathcal{M} = \langle S, \{A_i\}_{i=1}^N, \{O_i\}_{i=1}^N, P, R, \gamma \rangle$ . The agents share a team reward  $r_t = R(s_t, a_t^1, \dots, a_t^N)$ , and each agent  $i$  executes a decentralized policy  $\pi_i(a_t^i | o_t^i)$  that maps its local observation to an action. The team objective is the expected discounted return  $J(\pi) = \mathbb{E}_\pi[\sum_t \gamma^t r_t]$ . MASkills inherits this Dec-POMDP and re-parameterizes each  $\pi_i$  through a skill library.

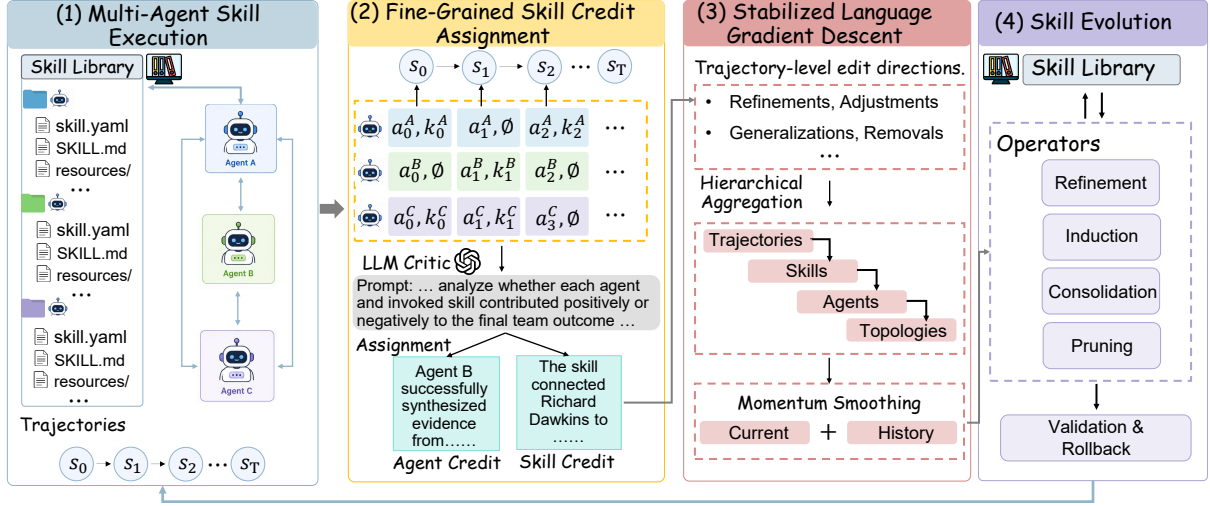


Figure 1: Overview of the MASkills pipeline as a four-step language-space analog of policy optimization. Agents execute skills to generate trajectories, receive skill-conditioned linguistic credit, aggregate temporally stable advantages, and update the skill library through refinement, induction, consolidation, and pruning.

**Skill artifacts.** Following (Alzubi et al., 2026), we model each skill artifact as a structured, file-system-based representation. For agent  $i$ , the  $j$ -th skill artifact is defined as

$$k_i^{(j)} = \left( y_i^{(j)}, m_i^{(j)}, \mathcal{R}_i^{(j)} \right), \quad (1)$$

where  $y_i^{(j)}$  denotes the metadata file, such as `skill.yaml`;  $m_i^{(j)}$  denotes the procedural instruction file, such as `SKILL.md`; and  $\mathcal{R}_i^{(j)}$  denotes a set of auxiliary resources, such as scripts, references, assets, and configuration files. In implementations such as Claude/Codex-style skills, this representation often corresponds to a directory containing a procedural instruction file and optional resource folders, while some variants separate metadata into a dedicated `skill.yaml` file. This structure supports concise procedural abstraction, actionable execution guidance, hierarchical disclosure, and reusable skill composition.

Each agent  $i$  maintains an evolving *skill library*

$$\mathcal{K}_i = \{k_i^{(1)}, k_i^{(2)}, \dots, k_i^{(M_i)}\}, \quad (2)$$

where  $M_i$  denotes the current number of skills available to agent  $i$ . We treat  $\mathcal{K}_i$  as the learnable component of agent  $i$ 's policy. MASkills therefore optimizes reusable structured skill artifacts that parameterize decentralized agent behavior.

**Skill-space optimization.** Whereas classical MARL optimizes network parameters  $\theta$ , MASkills directly optimizes the agents' skill libraries:

$$\max_{\{\mathcal{K}_i\}_{i=1}^N} \mathbb{E}_{\tau \sim \pi(\cdot|\{\mathcal{K}_i\})} [R(\tau)]. \quad (3)$$

Here, the optimization variables  $\{\mathcal{K}_i\}$  are discrete language-structured skill artifacts.

We approach Eq. 3 through a *language-space policy-gradient-style* framework, following the structural interpretation introduced by TextGrad (Yuksekgonul et al., 2025). The analogy is conceptual rather than differentiable: decentralized rollouts provide trajectory sampling, language critics provide skill-level advantage-like feedback, hierarchical aggregation and momentum smoothing approximate Monte-Carlo expectation and variance reduction, and credit-driven skill editing operators act as pseudo-gradient updates over each  $\mathcal{K}_i$ . Throughout the paper, terms such as *gradient*, *advantage*, and *momentum* should therefore be understood as language-space analogies rather than literal derivatives.

## 4 MASkills Framework

MASkills operationalizes the skill-space objective as a closed-loop process structured as a language-space analogue of a policy gradient: decentralized rollouts produce skill traces; a language critic emits per-agent, per-skill counterfactual credits; a hierarchical aggregator over a batch plus momentum smoothing across cycles produces a temporally stable skill advantage; and this advantage drives four credit-driven operators on each skill space under a held-out validation, as shown in Fig. 1.

### 4.1 Multi-Agent Skill Execution

Each agent is an LLM equipped with a skill space  $\mathcal{K}_i$ . At rollout time, lightweight metadata from

each `skill.yaml` (e.g., skill name and short description) is injected into the agent prompt, exposing the currently available skills to the model. The agent then autonomously decides whether to invoke a skill, which skill to invoke, and how to compose multiple skills during reasoning.

Skills are exposed as callable tools rather than fully expanded prompts. When a skill is selected, its corresponding `SKILL.md` and auxiliary resources  $\mathcal{R}$  are dynamically loaded into the context. This keeps the active context compact while enabling specialized procedural behavior.

Formally, the agent policy is conditioned on its local observation  $o_t^i$ , interaction history  $h_t^i$ , and skill space  $\mathcal{K}_i$ :

$$a_t^i \sim \pi_\theta(\cdot \mid o_t^i, \mathcal{K}_i, h_t^i). \quad (4)$$

Let  $c_t^i \in \mathcal{K}_i \cup \{\emptyset\}$  denote the skill invoked by agent  $i$  at timestep  $t$ , where  $c_t^i = \emptyset$  means the agent acted directly without loading a skill. Since skill invocations are explicit tool calls, the resulting skill trace  $\xi_i = \{(t, c_t^i)\}_{t=0}^{T-1}$  is directly observable from execution logs.

The resulting trajectory becomes

$$\tau = \langle s_0, \{a_0^i, \xi_0^i\}_{i=1}^N, \dots, s_T \rangle, \quad (5)$$

augmenting the standard multi-agent trajectory with explicit skill-invocation records that later support skill-level credit assignment.

## 4.2 Fine-Grained Skill Credit Assignment

Optimizing LLM agents at the skill level requires identifying which invoked skills actually contributed to team success or failure. Agent-level feedback alone is insufficient: a single trajectory may involve multiple interacting skills across multiple agents, and only a subset of them may be responsible for the final outcome.

MASkills therefore assigns credit at the granularity of individual skill invocations. Given a trajectory  $\tau$  with skill traces, a centralized language critic evaluates how each invoked skill affected the team outcome relative to the counterfactual where that skill had not been used.

For each skill  $k \in \mathcal{K}_i(\tau)$  actually invoked by agent  $i$  along trajectory  $\tau$ , the critic produces a structured natural-language credit:

$$C_i^{\text{text}}(\tau, k) = \text{LLM}_{\text{critic}}(\tau, i, k, \xi_i), \quad (6)$$

which explains whether the skill helped coordination, was redundant, caused a failure, or should be generalized or specialized.

In addition to per-skill feedback, the critic also emits an agent-level residual credit:

$$C_i^{\text{text}}(\tau) = \text{LLM}_{\text{critic}}(\tau, i), \quad (7)$$

which captures effects not attributable to any single skill. This residual is particularly important for skill induction: if a failure cannot be explained by existing skills, the agent-level critique signals that a new skill may be needed.

## 4.3 Stabilized Language Gradient Descent

Trajectory-level language critiques are inherently noisy and often contradictory across rollouts. To stabilize optimization, MASkills converts raw critiques into stable language-space gradients through gradient extraction, hierarchical aggregation, and momentum-based updates.

**Trajectory-level edit directions.** Given a trajectory  $\tau$  and an invoked skill  $k$ , the language critic first converts the textual credit signal into a trajectory-level edit direction:

$$g_i(\tau, k) = \text{Grad}\left(C_i^{\text{text}}(\tau, k)\right), \quad (8)$$

where  $\text{Grad}(\cdot)$  extracts a structured update direction from the credit signal, including suggested refinements, behavioral adjustments, generalizations, or removals in skill space. Unlike numerical gradients,  $g_i(\tau, k)$  is a structured natural-language edit direction over skills.

**Hierarchical aggregation.** Given a rollout batch  $\mathcal{B} = \{\tau_1, \dots, \tau_B\}$ , trajectory-level gradients are recursively aggregated across trajectories, skills, agents, and interaction topologies. At the skill level, the gradients associated with skill  $k$  are merged into a single language-space gradient estimate:

$$G_i^{(m)}(k) = \text{Agg}\left(\{g_i(\tau_b, k)\}_{b=1}^B\right), \quad (9)$$

where  $\text{Agg}(\cdot)$  is an LLM-based operator that merges recurring behavioral patterns, resolves conflicting update directions, removes redundancy, and summarizes the coordination utility of the skill. Higher-level aggregation recursively combines these gradients across agents into a stable language gradient estimate.

**Momentum-stabilized skill editing.** Even after aggregation, edit directions may still fluctuate across optimization cycles. Instead of treating each

cycle independently, MASkills carries forward persistent historical edit directions together with the current aggregated feedback.

The skill-space update therefore follows a language-space analogue of momentum gradient descent:

$$\mathcal{K}_i^{(m+1)} = \text{Edit}(\mathcal{K}_i^{(m)}, G_i^{(m)}, G_i^{(m-1)}), \quad (10)$$

where  $\text{Edit}(\cdot)$  denotes a language-space editing procedure that modifies the skill library using both the current aggregated feedback and persistent historical edit directions from previous optimization cycles. Conceptually, the current edit direction plays the role of the current gradient update, while the historical edit direction acts as a momentum term that preserves persistent behavioral improvements over time. This stabilization suppresses transient or contradictory critiques and encourages consistent long-horizon skill evolution. In practice, the abstract editing operator  $\text{Edit}(\cdot)$  is instantiated through four concrete skill-space transformations: refinement, induction, consolidation, and pruning.

#### 4.4 Skill Evolution Operators

MASkills improves agent behavior by directly modifying the skill space  $\mathcal{K}_i$ . Since skills are structured language artifacts rather than differentiable parameters, policy improvement is realized through a set of credit-driven editing operators acting on different regions of the skill space.

**Refinement.** Useful but imperfect skills are refined through localized diff-style edits guided by the aggregated edit direction:

$$k \leftarrow k \oplus \Delta k, \quad \Delta k = \text{LLM}_{\text{refine}}(k, G_i^{(m)}(k)), \quad (11)$$

where  $\Delta k$  is a structured edit patch rather than a full skill rewrite. The refinement operator modifies only the skill regions implicated by the aggregated feedback while preserving unrelated procedural structure and previously validated behavior.

**Induction.** When persistent failures or unresolved coordination patterns cannot be adequately addressed by the current skill library, MASkills expands the skill space by inducing new skills from difficult trajectories.

Let  $\mathcal{H}_i$  denote the set of hard trajectories associated with persistent unresolved critiques, missing behavioral capabilities, or repeated coordination

failures for agent  $i$ . The system proposes a new skill

$$k_{\text{new}} \sim \text{LLM}_{\text{propose}}(\mathcal{H}_i, G_i^{(m)}), \quad (12)$$

where the proposal is conditioned on both the hard-case trajectories and the aggregated edit directions accumulated during optimization. Unlike refinement, which applies localized edits to existing skills, induction introduces entirely new procedural abstractions into the skill space.

**Consolidation.** To prevent uncontrolled growth and fragmentation of the skill space, MASkills periodically consolidates related skills into higher-level procedural abstractions.

Given a set of functionally overlapping skills  $\{k_a, k_b, \dots\}$ , it constructs a consolidated skill

$$k_{\text{macro}} = \text{Merge}(k_a, k_b, \dots), \quad (13)$$

where  $\text{Merge}(\cdot)$  synthesizes shared behavioral structure, coordination logic, and reusable procedural patterns into a unified skill. Consolidation reduces redundancy in the skill library and encourages reusable higher-level abstractions.

**Pruning.** Not all skills remain useful as the skill space evolves. Some skills become obsolete after consolidation, others exhibit persistently poor coordination utility, and some are rarely invoked across trajectories. To control skill-space growth and remove low-value behaviors, MASkills periodically prunes skills with consistently weak utility signals.

Given the momentum-stabilized edit directions accumulated across optimization cycles, low-utility skills are removed from the skill library:

$$\mathcal{K}_i \leftarrow \mathcal{K}_i \setminus \{k : \text{LowUtility}(G_i^{(m)}(k))\}, \quad (14)$$

where  $\text{LowUtility}(\cdot)$  identifies skills whose behavioral contribution remains persistently negative, redundant, unstable, or negligible across trajectories.

**Validation and rollback.** Because skill updates directly modify the agent’s skill space, incorrect edits may introduce behavioral regressions. To ensure stable skill evolution, all candidate edits are evaluated on a held-out validation set before being committed to the skill library.

Let  $\mathcal{D}_{\text{val}}$  denote a held-out set of validation tasks and rollouts that are not used during the current optimization cycle. Given a candidate updated skill space  $\mathcal{K}'_i$ , MASkills estimates its validation performance while keeping all other agents’ skills fixed:

$$\hat{J}_{\text{val}}(\mathcal{K}'_i | \mathcal{K}_{-i}) = \mathbb{E}_{\tau \sim \mathcal{D}_{\text{val}}} [R(\tau)]. \quad (15)$$

Table 1: Main results across HotpotQA, LoCoMo, and GAIA.

(a) Multi-Hop Reasoning: HotpotQA

| Method | IO   | CoT  | CoT-SC | MedPrompt | MultiPersona | Self-Refine | ADAS | MASkills    |
|--------|------|------|--------|-----------|--------------|-------------|------|-------------|
| F1     | 68.1 | 67.9 | 68.9   | 68.3      | 69.2         | 60.8        | 64.5 | <b>76.3</b> |

(b) Long-horizon Memory: LoCoMo

| Method          | SH-F1        | SH-BLEU      | MH-F1        | MH-BLEU      |
|-----------------|--------------|--------------|--------------|--------------|
| MemoryBank      | 5.00         | 4.77         | 5.56         | 5.94         |
| ReadAgent       | 9.15         | 6.48         | 5.31         | 5.12         |
| LoCoMo          | 25.02        | 19.75        | 12.04        | 11.16        |
| MemGPT          | 26.65        | 17.72        | 9.15         | 7.44         |
| <b>MASkills</b> | <b>27.61</b> | <b>21.30</b> | <b>17.22</b> | <b>12.87</b> |

(c) General AI Assistants Benchmark

| Framework       | L1          | L2          | L3         | Avg.        |
|-----------------|-------------|-------------|------------|-------------|
| Base            | 12.8        | 3.8         | 0.0        | 6.8         |
| Search-o1       | 23.1        | 17.3        | 0.0        | 17.5        |
| Vanilla ReAct   | 28.2        | 15.3        | 0.0        | 18.4        |
| R1-Searcher     | 28.2        | 19.2        | <b>8.3</b> | 20.4        |
| <b>MASkills</b> | <b>35.3</b> | <b>22.6</b> | 0.0        | <b>23.3</b> |

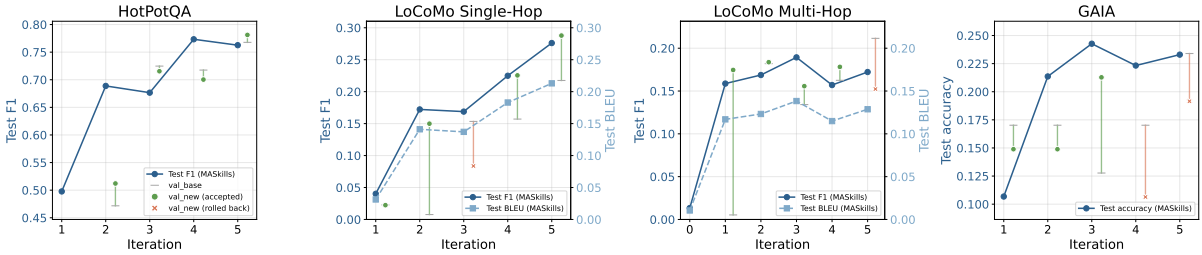


Figure 2: Training curves of MASkills across all benchmarks together with validation-based skill update decisions.

A candidate update is accepted only if

$$\hat{J}_{\text{val}}(\mathcal{K}'_i | \mathcal{K}_{-i}) \geq \hat{J}_{\text{val}}(\mathcal{K}_i | \mathcal{K}_{-i}) - \delta, \quad (16)$$

where  $\delta \geq 0$  absorbs rollout variance and evaluation noise. Otherwise, the proposed edit is discarded and the previous skill space is restored. This validation-and-rollback mechanism acts as a trust-region-style constraint in skill space, preventing unstable edits caused by noisy critiques, or overfitting to recent trajectories.

## 5 Experiments

We organize the empirical evaluation around three research questions (RQ1–RQ3).

- **RQ1 (Task Performance):** Does continual skill optimization improve the performance of multi-agent LLM systems across diverse tasks, and which components contribute most to the improvements?
- **RQ2 (Skill Quality and Transferability):** Do continually optimized skills become more reusable, transferable, and coordination-effective over time?
- **RQ3 (Robustness and Generalization):** Is MASkills robust across different multi-agent coordination topologies and LLM backbones?

### 5.1 Experimental Setup

We instantiate all tasks as cooperative Dec-POMDP environments in which a team of specialized LLM agents interacts under decentralized execution and a shared team-level objective. Each agent maintains a local observation space and executes a role-specialized policy conditioned on its observation history and skill library. To enable coordinated problem solving, agents are assigned complementary functional responsibilities, such as information retrieval, verification, planning, memory tracking, tool use, and decision making. Agents communicate through decentralized peer-to-peer interaction.

For continual skill optimization, we use GPT-5.1 as the optimizer backbone responsible for skill-conditioned credit assignment, trajectory aggregation, skill refinement, induction, consolidation, and pruning. Following prior work (Chhikara et al., 2025; Zhang et al., 2025; Wu et al., 2025), we use GPT-4o-mini (HotpotQA and LoCoMo) and Qwen2.5-7B (GAIA) as the primary actor backbones for agent execution.

### 5.2 Benchmarks

**Tasks.** We instantiate experiments on three benchmarks: (1) **Multi-hop Reasoning:** Hot-

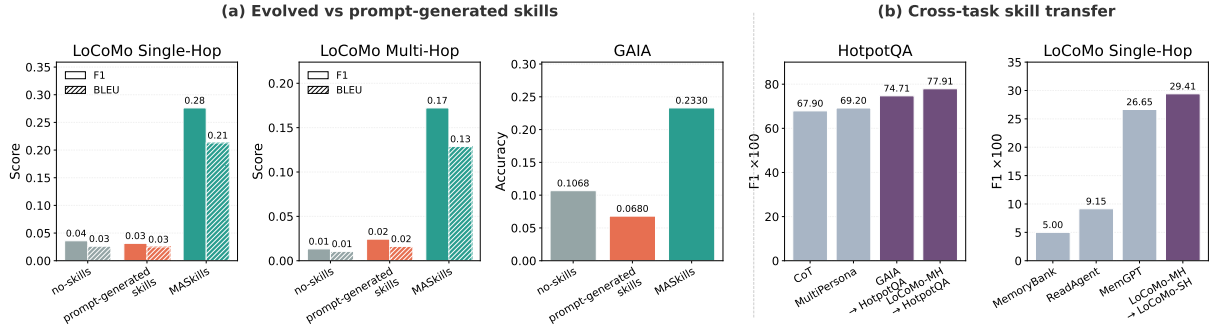


Figure 3: Skill quality and transferability analysis of MASkills. (a) Comparison between evolved skills, prompt-generated skills, and no-skill baselines on LoCoMo and GAIA. (b) Cross-task skill transfer results show that skills learned in one environment generalize effectively to unseen target tasks, indicating that MASkills acquires reusable procedural abstractions rather than task-specific prompting heuristics.

| Topology           | HotpotQA F1  | LoCoMo-SH F1 | LoCoMo-SH BLEU | LoCoMo-MH F1 | LoCoMo-MH BLEU | GAIA-L1      | GAIA-L2      | GAIA-Avg     |
|--------------------|--------------|--------------|----------------|--------------|----------------|--------------|--------------|--------------|
| Centralized        | 72.46        | <b>27.68</b> | 20.76          | <b>22.27</b> | <b>17.70</b>   | 17.65        | 11.54        | 11.76        |
| Hierarchical       | 71.86        | 27.55        | <b>20.85</b>   | 22.03        | 17.62          | 20.59        | 11.54        | 12.75        |
| Decentralized Peer | <b>76.30</b> | 27.61        | 21.30          | 17.22        | 12.87          | <b>35.30</b> | <b>22.60</b> | <b>23.30</b> |

Table 2: Performance under different coordination topologies.

potQA (Yang et al., 2018), a multi-document question answering benchmark requiring compositional reasoning across supporting evidence. (2) **Long-horizon dialogue memory**: LoCoMo (Maharana et al., 2024), a very-long-term multi-session conversation benchmark designed to evaluate persistent conversational memory and temporal reasoning. (3) **General AI Assistants Benchmark (GAIA)** (Mialon et al., 2024) is a benchmark designed to evaluate general AI agents on real-world tasks, with a focus on reasoning, multimodal understanding, web browsing, and tool-use capabilities.

**Metrics.** We report both benchmark-specific task metrics and skill-evolution diagnostics. For *HotpotQA*, we follow the standard evaluation protocol and report answer-level F1. For *LoCoMo*, we report both F1 and BLEU across the four question categories defined by the benchmark (multi-hop, temporal, open-domain, and single-hop memory queries). For *GAIA*, we report averaged success rate across various tasks.

### 5.3 Main Results (RQ1)

Table 1 summarizes the main results on HotpotQA, LoCoMo, and GAIA. Overall, MASkills achieves strong and consistent performance across a diverse set of reasoning, long-context memory, and agentic problem-solving benchmarks, demonstrating the effectiveness of skill optimization in multi-agent systems. On HotpotQA, MASkills outperforms existing prompting-based and multi-agent reason-

ing baselines, indicating stronger multi-hop reasoning and evidence integration capabilities. On LoCoMo, MASkills consistently improves long-context memory retrieval and response quality across both single-hop and multi-hop settings, suggesting better temporal reasoning and memory coordination over extended interactions. On GAIA, MASkills achieves the strongest overall performance among compared agent frameworks, demonstrating improved planning, decomposition, and tool-use abilities in open-ended environments.

### 5.4 Skill Quality and Transferability (RQ2)

We next evaluate whether continually optimized skills become more reusable, transferable, and coordination-effective over time. To this end, we analyze both the intrinsic quality of evolved skills and their ability to generalize across benchmarks.

Figure 3(a) compares MASkills against no-skill baselines and prompt-generated skills across LoCoMo and GAIA. We observe that prompt-generated skills provide only marginal improvements over directly prompting the agents, while continually optimized skills learned by MASkills lead to substantial gains across all evaluated settings. In particular, MASkills significantly improves both F1 and BLEU on LoCoMo single-hop and multi-hop memory tasks, while also achieving the strongest performance on GAIA. These results suggest that iterative skill refinement and trajectory-driven optimization produce substan-

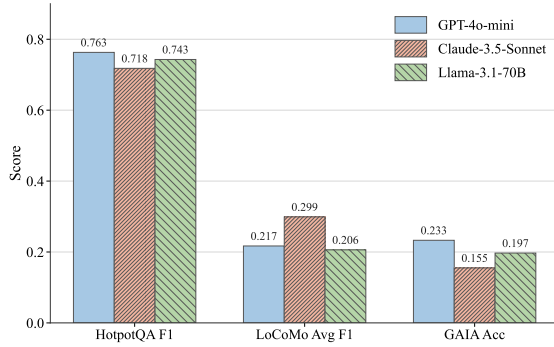


Figure 4: Backbone comparison across different underlying language models.

tially higher-quality procedural abstractions than one-shot prompt-generated skills.

Figure 3(b) further evaluates cross-task skill transfer. We transfer skill libraries learned from one environment into previously unseen target tasks without additional optimization. Transferred skills consistently improve downstream performance across all evaluated source-target pairs. For example, skills optimized on GAIA improve HotpotQA reasoning performance beyond strong prompting-based baselines such as CoT and MultiPersona, while LoCoMo-derived skills improve downstream long-horizon memory reasoning. These findings suggest that MASkills captures reusable behavioral patterns rather than memorizing benchmark-specific heuristics.

### 5.5 Robustness and Generalization (RQ3)

We next evaluate whether MASkills generalizes across different coordination structures and underlying language model backbones.

**Topology Robustness.** We instantiate MASkills under centralized, decentralized peer, and hierarchical structures while keeping the optimization pipeline unchanged. Table 2 shows that MASkills maintains competitive performance across different coordination topologies, suggesting that the proposed skill optimization framework is not tightly coupled with a specific orchestration structure.

Interestingly, the optimal topology is highly task-dependent. On HotpotQA and GAIA, decentralized peer coordination achieves the strongest performance. These tasks benefit from diverse exploration, independent tool or retrieval trajectories. In contrast, centralized coordination performs best on LoCoMo, particularly on multi-hop memory retrieval. Unlike retrieval-centric tasks, LoCoMo requires maintaining globally consistent long-context

| Method                      | LoCoMo-MH   | GAIA        |
|-----------------------------|-------------|-------------|
| MASkills (Full)             | <b>17.2</b> | <b>23.3</b> |
| w/o Skill Credit assignment | 14.2        | 17.1        |
| w/o Momentum Smoothing      | 16.4        | 21.9        |
| w/o Validation Rollback     | 6.6         | 13.5        |
| w/o Consolidation / Pruning | 13.9        | 13.0        |

Table 3: Ablation Study

memory representations, resolving temporal dependencies, and integrating information across multiple conversational sessions. Centralized coordination provides a shared global state that reduces memory fragmentation and inconsistency across agents, leading to more coherent long-horizon reasoning and generation quality. Hierarchical coordination generally yields intermediate performance, balancing structured information aggregation with partial decentralization.

**Backbone Generalization.** We further evaluate MASkills across multiple LLM backbones, including both proprietary and open-source models. Figure 4 shows that MASkills maintains competitive performance across different model families on HotpotQA, LoCoMo, and GAIA. While the relative strengths of the backbones vary across tasks, the overall trends remain consistent: stronger reasoning-oriented models perform better on long-context and agentic tasks, whereas open-weight models still achieve solid performance under the MASkills framework.

### 5.6 Ablation Study

To understand the contribution of different components in MASkills, we conduct ablation studies over skill-conditioned credit assignment, momentum smoothing, validation rollback, and consolidation/pruning. Table 3 shows that removing any component leads to performance degradation, indicating that all modules contribute to stable continual skill optimization. Among them, validation rollback has the largest impact, suggesting that reverting unstable updates is critical for preventing performance collapse during iterative optimization. Skill-conditioned credit assignment also plays an important role by improving the attribution of successful behaviors to specific skills. In addition, momentum smoothing and consolidation/pruning provide consistent gains by stabilizing optimization and reducing redundant or low-quality skills.

## 6 Conclusion

We presented MASkills, a continual learning framework that optimizes multi-agent LLM systems in skill space. Through skill-conditioned credit assignment and evolution, MASkills enables agents to continually refine and reuse skills from interaction experience. Experiments on various benchmarks show that MASkills consistently improves memory and agentic tasks, highlighting the potential of continual skill optimization for scalable and adaptive multi-agent systems.

## Acknowledgement

The work was partially supported by Cisco Faculty Award, Amazon Research Award, and NSF awards #2442477, #2550203, #2616632, #2623317, #2536297, and #2613637. The views and conclusions in this paper should not be interpreted as representing any funding agencies.

## Limitations

Although MASkills demonstrates strong performance across multiple benchmarks, several limitations remain. First, our current experiments mainly focus on cooperative settings with relatively fixed agent roles and communication topologies, leaving more dynamic, adversarial, and open-world environments underexplored. Future work could extend MASkills to adaptive organizational structures, competitive multi-agent games, and large-scale decentralized coordination scenarios. Second, as the skill library continually expands during optimization, scalability challenges may emerge in skill retrieval, consolidation, and coordination efficiency. Future research may investigate hierarchical skill organization, retrieval compression, and lifelong learning mechanisms to support efficient long-term continual skill evolution at scale.

## Ethical Considerations

MASkills aims to improve multi-agent LLM systems by continually refining reusable skill artifacts from interaction experience. While this can enhance agent performance and adaptability, it also introduces several ethical considerations. First, automatically evolved skills may amplify undesirable behaviors inherited from the underlying LLMs, including factual errors, social biases, unsafe tool-use patterns, or overconfident reasoning. To mitigate this risk, MASkills incorporates held-out valida-

tion, rollback, consolidation, and pruning mechanisms before committing skill updates; however, these mechanisms should not be viewed as a substitute for human oversight in high-stakes applications. Second, because the framework learns from interaction trajectories, deployments involving user data should follow strict privacy practices, including data minimization, anonymization, access control, and compliance with the licenses and usage terms of the underlying datasets and tools. Third, improved multi-agent coordination and tool-use ability could be misused for harmful automation if applied without appropriate safeguards. We therefore recommend restricting MASkills to benign use cases, monitoring evolved skills for unsafe or deceptive behavior, and maintaining transparent audit logs of skill modifications. Our experiments are conducted on established research benchmarks, and we do not intend the framework to be used for decision-making in sensitive domains without additional safety evaluation, domain-specific validation, and human accountability.

## AI usage

During the preparation of this work, the authors employed LLMs to assist with code implementation and to refine the manuscript’s grammatical clarity and stylistic flow. All final content, technical contributions, and experimental analyses were rigorously reviewed and verified by the authors.

## References

- Salaheddin Alzubi, Noah Provenzano, Jaydon Bingham, Weiyuan Chen, and Tu Vu. 2026. EvoSkill: Automated skill discovery for multi-agent systems. *arXiv preprint arXiv:2603.02766*.
- Anthropic. 2025. The complete guide to building skills for claude.
- Tiejun Chen, Huaiyuan Yao, Jia Chen, Evangelos E Papalexakis, and Hua Wei. 2026. Every response counts: Quantifying uncertainty of llm-based multi-agent systems through tensor decomposition. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16204–16218.
- Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*.
- Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang,

- Steven Yau, Zijuan Lin, Liyang Zhou, and 1 others. 2024. Metagtpt: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, volume 2024, pages 23247–23275.
- Xu Hu, Yifan Zhang, Songtao Wei, Chen Zhao, Qiannan Li, Bingzhe Li, and Feng Chen. 2026. Small updates, big doubts: Does parameter-efficient fine-tuning enhance hallucination detection? *arXiv preprint arXiv:2602.11166*.
- Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2023. [Dspy: Compiling declarative language model calls into self-improving pipelines](#). *Preprint*, arXiv:2310.03714.
- Xiaou Liu, Longchao Da, Hanyang Chen, Yuan Ling, and Hua Wei. 2026. Androidreality: How far are mobile agents from the real world? *arXiv preprint arXiv:2608.07775*.
- Xiaowen Ma, Yunpu Ma, Chenyang Lin, Sikuan Yan, Jinhe Bi, Zixuan Cao, Yijun Tian, Volker Tresp, and Hinrich Schuetze. 2026. [Self-evolving multi-agent systems via textual backpropagation](#). *Preprint*, arXiv:2506.09046.
- Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870.
- Grégoire Mialon, Clémentine Fourier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2024. Gaia: a benchmark for general ai assistants. In *International Conference on Learning Representations*, volume 2024, pages 9025–9049.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652.
- Ken Tsui. 2025. [Self-correction bench: Revealing the self-correction blind spot in LLMs](#). In *NeurIPS 2025 Workshop on Evaluating the Evolving LLM Lifecycle: Benchmarks, Emergent Abilities, and Scaling*.
- Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.
- Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhenglin Wang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Robert Tang, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. 2025. [Webdancer: Towards autonomous information seeking agency](#). In *Advances in Neural Information Processing Systems*, volume 38, pages 120957–120985. Curran Associates, Inc.
- Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. Autogen: Enabling next-gen llm applications via multi-agent conversation. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [HotpotQA: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.
- Huaiyuan Yao, Longchao Da, Xiaou Liu, Charles Fleming, Tianlong Chen, and Hua Wei. 2026a. [Langmarl: Natural language multi-agent reinforcement learning](#). *Preprint*, arXiv:2604.00722.
- Huaiyuan Yao, Longchao Da, Vishnu Nandam, Justin Turnau, Zhiwei Liu, Linsey Pang, and Hua Wei. 2025a. Comal: Collaborative multi-agent large language models for mixed-autonomy traffic. In *Proceedings of the 2025 SIAM International Conference on Data Mining (SDM)*, pages 409–418. SIAM.
- Huaiyuan Yao, Pengfei Li, Bu Jin, Yupeng Zheng, An Liu, Lisen Mu, Qing Su, Qian Zhang, Yilun Chen, and Peng Li. 2025b. Lilodriver: A lifelong learning framework for closed-loop motion planning in long-tail autonomous driving scenarios. *arXiv preprint arXiv:2505.17209*.
- Huaiyuan Yao, Wanpeng Xu, Justin Turnau, Nadia Kellam, and Hua Wei. 2026b. Instructional agents: Reducing teaching faculty workload through multi-agent instructional design. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4087–4109.
- Simon Yu, Gang Li, Weiyan Shi, and Peng Qi. 2026. Polyskill: Learning generalizable skills through polymorphic abstraction for continual learning. In *International Conference on Learning Representations*, volume 2026, pages 140298–140326.
- Mert Yuksekogonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. 2025. Optimizing generative ai by backpropagating language model feedback. *Nature*, 639:609–616.
- Haozhen Zhang, Quanyu Long, Jianzhu Bao, Tao Feng, Weizhi Zhang, Haodong Yue, and Wenya Wang. 2026. [Memskill: Learning and evolving memory skills for self-evolving agents](#). *Preprint*, arXiv:2602.02474.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, and 1 others. 2025. Aflow: Automating agentic workflow generation. In *International Conference on Learning Representations*, volume 2025, pages 34040–34077.

Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2024. [A survey on the memory mechanism of large language model based agents](#). *Preprint*, arXiv:2404.13501.

Lina Zhao, Zihao Bian, Qingyue Chen, Yafang Li, Zhiyi Luo, Jiaying Bai, Guangbo Li, Min He, Kezhi Li, Huaiyuan Yao, and 1 others. 2025. Fuas-agents: Autonomous multi-modal llm agents for treatment planning in focused ultrasound ablation surgery. *arXiv preprint arXiv:2505.21418*.

Wangchunshu Zhou, Yixin Ou, Shengwei Ding, Long Li, Jialong Wu, Tiannan Wang, Jiamin Chen, Shuai Wang, Xiaohua Xu, Ningyu Zhang, Huijun Chen, and Yuchen Eleanor Jiang. 2024. [Symbolic learning enables self-evolving agents](#). *Preprint*, arXiv:2406.18532.