CoEvo-MAS: Co-Evolving Skills and Agent Organizations in Multi-Agent Systems

Anonymous authors
Paper under double-blind review

Abstract

Large language model (LLM)-based multi-agent systems (MASs) often rely on fixed organizations designed before deployment. As tasks change, these organizations may fail to cover new capability requirements. Skill learning methods can extract reusable knowledge from agent experience, but they typically leave the underlying organization unchanged. Automated organization design methods can adjust agent roles or compositions, yet they often rely on short-term signals such as the current task, isolated failures or raw interaction trajectories. In this paper, we propose **CoEvo-MAS**, a framework that co-evolves skills and multi-agent organizations in a closed loop. From task execution, CoEvo-MAS distills reusable skills and tracks their utility, usage frequency and compatibility with agents. The resulting empirical skill distribution is summarized as a skill profile that reflects recurring capability needs. At regular intervals, a meta agent uses this profile to identify organization-level issues and propose changes to skill assignments, agent roles or agent composition. Each candidate organization is validated on held-out tasks and deployed only if it improves over the current one. CoEvo-MAS is evaluated on ALFWorld and τ^2-bench, where it outperforms the compared methods. Ablation studies further confirm the contribution of skill evolution, skill-profile-guided organization adaptation, and validation-gated updates.

1 Introduction

Large language models (LLMs) have become a common foundation for building agents that can reason, use tools, and interact with external environments **?**. However, many real-world tasks require long-term planning, the use of different tools, and knowledge from multiple domains. It is usually difficult for a single agent to handle all these requirements reliably **?**. LLM-based multi-agent systems (MAS) address this problem by assigning different roles to agents and allowing them to solve tasks through division of labor and collaboration **???**. The performance of an MAS, however, depends not only on the capability of each agent, but also on whether its organization matches the capability needs of the tasks. As Figure 1shows, most existing MAS use organizations that are manually designed before deployment or generated for a given task **??**. Once deployed, these organizations often remain fixed and cannot continuously adapt to the capability needs that emerge from later tasks.

One line of work improves agents by extracting reusable skills from their interaction experience **???**. These methods summarize useful behaviors, plans or workflows from past trajectories and store them in a memory or skill library. The stored skills can then be retrieved and reused in future tasks. Some methods further update, merge or remove skills based on execution feedback **??**. While these methods improve what agents know and how they act, they usually keep the organization that uses the skills unchanged. Consequently the system may learn a useful skill but lack an agent with a suitable role, tool set or duty to use it. In other cases, one agent may need to handle many unrelated skills, while several agents may have highly similar duties. Therefore, learning more skills does not ensure that the MAS organization can make effective use of them.

Another line of work develops dynamic MAS that select, generate or modify agents according to the current task or execution feedback **???**. These methods improve the flexibility of MAS, but their organization changes are often based on the current task, a single failure or raw interaction trajectories **??**. Such evidence may not provide a stable basis for long-term organization changes. Raw

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

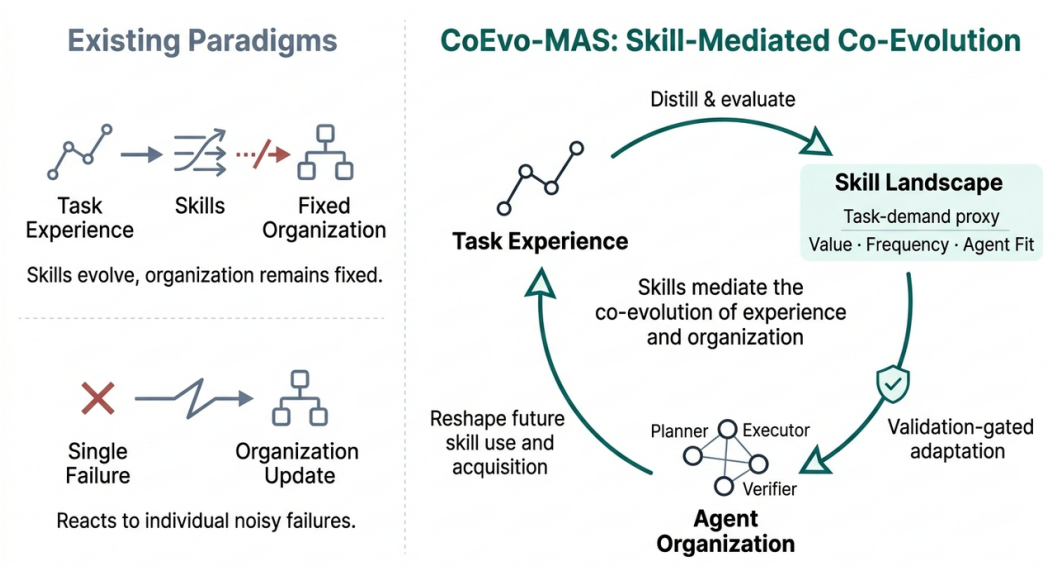


Figure 1: Overview of CoEvo-MAS. The task execution loop generates interaction experience, while the periodic evolution loop updates the SkillBank and adapts the multi-agent organization based on the accumulated skill profile. Candidate organization updates are deployed only after validation.

trajectories contain many messages, intermediate decisions, tool calls and environment responses that may not be directly related to the final result. A failure on one task may also be caused by an execution error rather than a missing capability in the organization. Moreover, directly applying organization changes proposed by an LLM may reduce performance, create repeated duties, or lead to unnecessary growth in the number of agents **??**. It is therefore important to identify capability needs that appear repeatedly across tasks and to verify an organization change before permanently applying it.

Our key observation is that the distribution of accumulated skills provides a compact summary of the capability needs observed during task execution. We characterize each skill by its empirical utility, usage frequency and compatibility with the agents in the current organization. The distribution of accumulated skills forms a structured *skill profile*. Skill utility estimates the effectiveness of a skill based on its actual usage outcomes, usage frequency reflects how often the corresponding capability is needed, and agent compatibility indicates whether the current organization has a suitable agent to use it. The skill profile is not a direct estimate of the underlying task distribution. Instead, it serves as an experience-based proxy for capability needs that repeatedly appear across tasks and provides structured evidence for identifying organization problems, such as missing capabilities, overloaded roles and repeated duties.

Based on this observation, we propose **CoEvo-MAS**, a framework that co-evolves skills and multi-agent organizations at two time scales. In the task execution loop, the meta agent selects a subset of agents from the current agent pool for each task, while the selected agents retrieve and use skills as needed during their interaction with the environment. In the evolution loop, CoEvo-MAS distills new skills from the collected trajectories, updates existing skills based on their actual usage outcomes, and merges or removes skills that are redundant or consistently ineffective. At regular intervals, the system aggregates the empirical distribution of skills in the SkillBank into the current skill profile, and the meta agent uses this profile to determine whether the organization should be adapted. When an update is needed, the meta agent identifies organization-level issues, such as missing capabilities, overloaded roles and repeated duties, and proposes a candidate change through skill reassignment, role update, agent addition or agent removal. Each candidate organization is compared with the current organization on a held-out validation set and is permanently deployed only when it achieves a sufficient improvement; otherwise, the change is rejected. The accepted organization further affects which agents handle later tasks and how skills are used and generated,

forming a closed feedback loop between skill evolution and organization evolution while reducing the risk of harmful or unnecessary updates.

Our main contributions are as follows:

- We propose CoEvo-MAS, a framework that connects skill evolution and multi-agent organization evolution in a closed loop. Experience produces and updates skills, accumulated skill evidence guides organization changes, and the changed organization affects future skill use and production.
- We represent the empirical distribution of accumulated skills through a skill profile that captures their utility, usage frequency and agent compatibility. This profile serves as a compact proxy for recurring capability needs and provides structured evidence for adapting agent roles, skill assignments and agent composition.
- We evaluate CoEvo-MAS across multiple benchmarks. The results show that CoEvo-MAS improves task performance and organizational adaptability over existing baselines, while the validation-gated update reduces harmful organization changes. Ablation studies further verify the contributions of skill evolution, skill-profile-guided organization adaptation and validation gating.

2 Related Work

2.1 LLM-Based Multi-Agent Organizations

LLM-based multi-agent systems organize multiple agents to solve tasks through role assignment, task decomposition, communication and collaborative decision-making **???**. Existing systems often assign specialized roles, such as planner, executor, critic or domain expert, and coordinate them through predefined workflows or communication structures **???**. Other methods construct an agent team for each task by generating roles or selecting agents from an existing pool **??**. These designs have shown that a suitable organization can improve task coverage and reduce the burden on individual agents.

However, most of these methods focus on designing a fixed organization or forming a temporary team for the current task. They do not continuously update the persistent agent pool and role assignments based on capability needs observed across tasks. Task-level agent selection changes which agents participate in one execution, but it does not address whether the underlying organization should add or remove agents, revise their roles or reassign their skills. CoEvo-MAS complements existing organization and routing methods by using the empirical distribution of accumulated skills as long-term evidence for adapting the persistent MAS organization.

2.2 Skill Learning from Agent Experience

LLM agents can improve over time by storing and reusing wisdom gained from previous interactions **???**. Early methods mainly record complete trajectories, successful examples or natural-language reflections in an external memory bank and retrieve relevant experience for later tasks **???**. More recent work converts raw experience into reusable skills, such as action procedures, plans, tool-use programs or code, and maintains them in a skill library **???**. Some methods further use execution feedback to refine, merge or remove stored skills, improving the quality and efficiency of skill reuse **???**.

These methods mainly improve the execution knowledge of individual agents or a fixed agent team. They rarely consider whether the current organization has suitable agents to use the learned skills, or whether the distribution of accumulated skills reveals a need to change agent roles and composition. In contrast, CoEvo-MAS not only evolves skills from task experience, but also uses their utility, usage frequency, and agent compatibility as evidence for adapting the MAS organization.

2.3 Automated Evolution of LLM-Based Multi-Agent Systems

Recent studies have explored the automated design and improvement of LLM-based multi-agent systems. Some methods generate specialized agents or select a team according to the input task **??**. Oth-

ers optimize agent workflows, communication graphs or coordination strategies through search, execution feedback, or evolutionary procedures **???**. Related approaches further revise agent prompts, roles, and collaboration rules based on failures or interaction trajectories, allowing the system to improve through repeated execution **??**.

Most existing methods optimize a team for the current task or search for a better workflow under a fixed evaluation setting. When adaptation is supported, organization changes are often driven by task descriptions, individual failures, or raw trajectories, which may provide noisy or short-term evidence of capability needs. CoEvo-MAS instead uses the empirical distribution of accumulated skills as long-term evidence of recurring capability needs. It periodically adapts the persistent roles, skill assignments, and agent composition, while using a validation gate to reject changes that do not improve performance.

3 METHOD

3.1 PROBLEM FORMULATION

We consider an LLM-based multi-agent system that processes a sequence of tasks $\mathcal{T} = \{x_t\}_{t=1}^{T}$ drawn from a task distribution $\mathcal{D}$. For each task x_t, the system interacts with the environment and produces a trajectory τ_t, which contains agent observations, actions and inter-agent communication. The environment then provides task feedback y_t, such as a success or failure indicator.

At time step t, the system is characterized by a skill bank $\mathcal{B}_t$ and an organizational structure $\mathcal{O}_t$. The skill bank stores reusable knowledge acquired from previous task experience:

$$\mathcal{B}_t = \{s_i\}_{i=1}^{N_t} \tag{1}$$

where each s_i represents a skill that can be applied to future tasks. The organizational structure is represented as

$$\mathcal{O}_t = (\mathcal{A}_t, \mathcal{R}_t, \Phi_t) \tag{2}$$

where $\mathcal{A}_t$ denotes the set of agents, $\mathcal{R}_t$ denotes their roles, and Φ_t specifies the assignment of skills to agents. Together, $\mathcal{B}_t$ and $\mathcal{O}_t$ determine how the system handles the current task.

Given a task x_t, we denote the resulting task performance by

$$R(x_t; \mathcal{B}_t, \mathcal{O}_t) \tag{3}$$

Our goal is to improve performance over the task sequence by jointly adapting the skill bank and the organizational structure:

$$\max_{\{\mathcal{B}_t, \mathcal{O}_t\}_{t=1}^{T}} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{x_t \sim \mathcal{D}} \left[R(x_t, \mathcal{B}_t, \mathcal{O}_t) \right] \tag{4}$$

The key challenge is that the two components are interdependent: accumulated skills provide evidence about the capabilities required by the task distribution, while the organization determines how these skills are assigned and used. We therefore study their joint evolution from task experience, as introduced in the following sections.

3.2 OVERVIEW OF THE CO-EVOLUTION FRAMEWORK

As illustrated in Figure 2, CoEvo-MAS consists of a task execution loop and a periodic evolution loop. For each task, the meta agent selects a subset of agents from the current agent pool. The selected agents retrieve skills from the SkillBank as needed, interact with the environment, and produce execution trajectories. These trajectories record task outcomes and actual skill usage, providing experience for subsequent evolution. After accumulating experience, CoEvo-MAS distills new skills and updates existing ones based on their usage feedback. The empirical distribution of accumulated skills, characterized by their utility, usage frequency and agent compatibility, is summarized as a skill profile. At regular intervals, the meta agent uses this profile to determine whether the current organization should be adapted, and proposes changes to skill assignments, agent roles or agent composition when necessary. Each candidate organization is evaluated against the current organization on held-out validation tasks before deployment. The updated organization then affects subsequent agent selection, skill usage and experience collection, forming a closed loop between skill evolution and organization evolution.

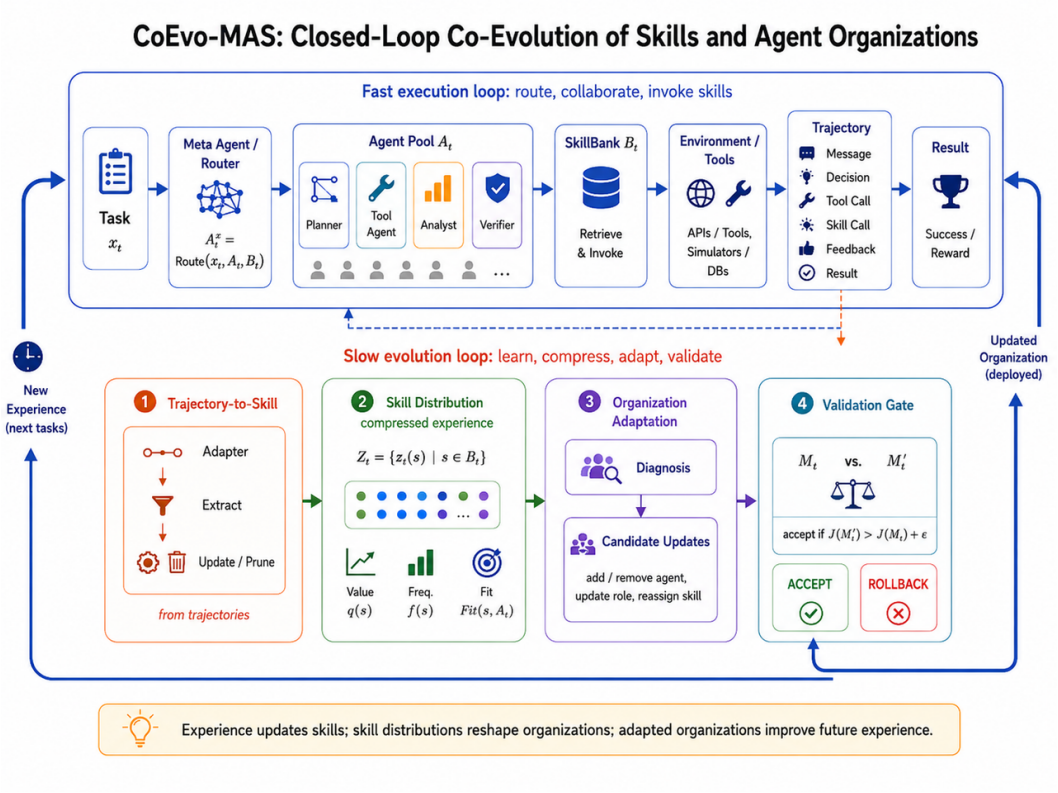


Figure 2: Overview of CoEvo-MAS. The task execution loop generates interaction experience, while the periodic evolution loop updates the SkillBank and adapts the multi-agent organization based on the accumulated skill profile. Candidate organization updates are deployed only after validation.

3.3 Experience-Driven Skill Evolution

3.3.1 Trajectory-Grounded Skill Distillation

CoEvo-MAS periodically extracts reusable skills from a batch of execution trajectories $\{\tau_i\}_{i=1}^{m}$. Each trajectory $\tau_i = \{(o_j, a_j)\}_{j=1}^{L_i}$ contains the observations and actions recorded during task execution.

Grouping related trajectories. We first identify the capability involved in each trajectory from its actions and environment changes, such as manipulating an object, tracking a state change, or completing a subgoal. Trajectories involving similar capabilities are grouped as

$$C_k = \{\tau_i \mid g(\tau_i) = k\}, \tag{5}$$

where $g(\cdot)$ assigns a trajectory to a capability group k.

Extracting reusable procedures. For each group containing at least $w_{\min}$ successful examples, we identify the actions that directly lead to the corresponding state change or subgoal completion. We remove repeated attempts, unnecessary navigation steps and actions that do not contribute to this result. The remaining key steps are then aligned across successful trajectories and generalized into a reusable procedure:

$$\pi_k = \text{Generalize}\left(\{\text{Extract}(\tau_i) \mid \tau_i \in C_k,\ y_i = 1\}\right). \tag{6}$$

The resulting procedure π_k preserves the common order of the key steps while replacing task-specific objects and tools with variable fields, such as `take <object>` or `use <tool> on <target>`.

Skill representation. Each extracted skill is represented as

$$s_k = (p_k, \pi_k, e_k), \tag{7}$$

where p_k describes when the skill can be applied, π_k defines its execution procedure, and e_k describes its expected effect. A new skill is initially marked as *provisional* and added to the SkillBank with utility $u_k(0) = 0.5$.

3.3.2 Feedback-Driven Utility Tracking and Pruning

CoEvo-MAS evaluates each skill based on its actual use in subsequent tasks. A skill is counted as used only when the agent applies it during execution, rather than merely retrieving it from the SkillBank. The system records whether each use produces the expected result according to the environment feedback.

We estimate the utility of skill s_k using its smoothed empirical success rate:

$$u_k(t) = \frac{n^k_{\text{success}}(t) + 1}{n^k_{\text{use}}(t) + 2}, \tag{8}$$

where $n^k_{\text{use}}(t)$ and $n^k_{\text{success}}(t)$ denote the numbers of actual and successful uses up to time t, respectively. This smoothing gives a new skill an initial utility of 0.5 and avoids unreliable estimates based on limited observations.

A provisional skill is marked as *verified* after it receives sufficient usage evidence and its utility exceeds a verification threshold. Skills with consistently low utility are removed, while unused or highly similar skills are removed or merged. Only verified skills are included as evidence for persistent organization updates.

3.4 Skill-Guided Organization Evolution

3.4.1 Skill-Based Organizational Diagnosis and Modification

CoEvo-MAS periodically examines whether the current organization can effectively use the skills accumulated from previous tasks. Rather than estimating the underlying task distribution directly, the system uses verified skills as empirical evidence of recurring capability needs. For each verified skill s_k, we maintain

$$z_t(s_k) = \left[u_t(s_k), f_t(s_k), \phi_t(s_k)\right], \tag{9}$$

where $u_t(s_k)$ is its empirical utility, $f_t(s_k)$ is its usage frequency, and $\phi_t(s_k)$ describes its compatibility with the agents in the current organization. The collection

$$\mathcal{Z}_t = \{z_t(s_k) \mid s_k \in \mathcal{B}^{\text{ver}}_t\} \tag{10}$$

forms the current skill profile, where $\mathcal{B}^{\text{ver}}_t$ denotes the set of verified skills.

At each organization evolution interval, the meta agent uses $\mathcal{Z}_t$ to determine whether the current organization should be modified. It checks for organizational issues such as frequently used and effective skills without a suitable agent, agents assigned to many unrelated skills, overlapping agent roles, and agents that are rarely selected or contribute little. If no clear issue is found, the current organization is retained.

Otherwise, the meta agent proposes a candidate modification:

$$\Delta\mathcal{O}_t \in \{\text{ReassignSkill}, \text{UpdateRole}, \text{AddAgent}, \text{RemoveAgent}\}. \tag{11}$$

The proposed modification specifies both the organization-level issue being addressed and the supporting evidence from the skill profile. Applying it to the current organization produces a candidate organization

$$\widetilde{\mathcal{O}}_t = \mathcal{O}_t \oplus \Delta\mathcal{O}_t. \tag{12}$$

3.4.2 VALIDATION-GATED ORGANIZATION UPDATE AND ROLLBACK

A candidate organization is not directly deployed. We compare $\widetilde{\mathcal{O}}_t$ with the current organization $\mathcal{O}_t$ on the same held-out validation set $\mathcal{V}$. Let

$$J(\mathcal{O}; \mathcal{V}) = \frac{1}{|\mathcal{V}|} \sum_{x \in \mathcal{V}} R(x; \mathcal{O}) \tag{13}$$

denote the average validation performance of organization $\mathcal{O}$, where R is the task-level evaluation score. The organization is updated as

$$\mathcal{O}_{t+1} = \begin{cases} \widetilde{\mathcal{O}}_t, & J(\widetilde{\mathcal{O}}_t; \mathcal{V}) > J(\mathcal{O}_t; \mathcal{V}) + \epsilon, \\ \mathcal{O}_t, & \text{otherwise}, \end{cases} \tag{14}$$

where ϵ is the minimum required improvement. If the candidate does not pass the validation gate, the temporary modification is discarded and the system retains the current organization. This mechanism reduces the risk of deploying harmful changes proposed by the meta agent.

4 EXPERIMENTS

We conduct experiments to answer the following research questions:

- **RQ1: Overall Effectiveness.** Does CoEvo-MAS outperform single-agent, conventional multi-agent and skill-learning methods?
- **RQ2: Effect of Skill Evolution.** How do skill extraction, utility-based updating and skill merging and pruning contribute to task performance?
- **RQ3: Effect of Organization Evolution.** Does skill profile guided MAS organization evolution improve task performance, and how do its main components contribute to the improvement?
- **RQ4: Reliability and Evolution Dynamics.** Does the validation gate reduce harmful organization updates, and how do the learned skills and MAS organization evolve throughout task execution?

4.1 EXPERIMENTAL SETUP

4.2 OVERALL PERFORMANCE (RQ1)

4.3 EFFECT OF SKILL EVOLUTION (RQ2)

4.4 EFFECT OF ORGANIZATION EVOLUTION (RQ3)

4.5 RELIABILITY AND EVOLUTION DYNAMICS (RQ4)

Effect of the Validation Gate

Skill and Organization Evolution over Time

5 CONCLUSION

REFERENCES