CoMAS: Co-Evolving Multi-Agent Systems via Interaction Rewards

ICLR 2026 Poster Presentation

Xiangyuan Xue

March 1, 2026

The Chinese University of Hong Kong

Paper Information

CoMAS: Co-Evolving Multi-Agent Systems via Interaction Rewards

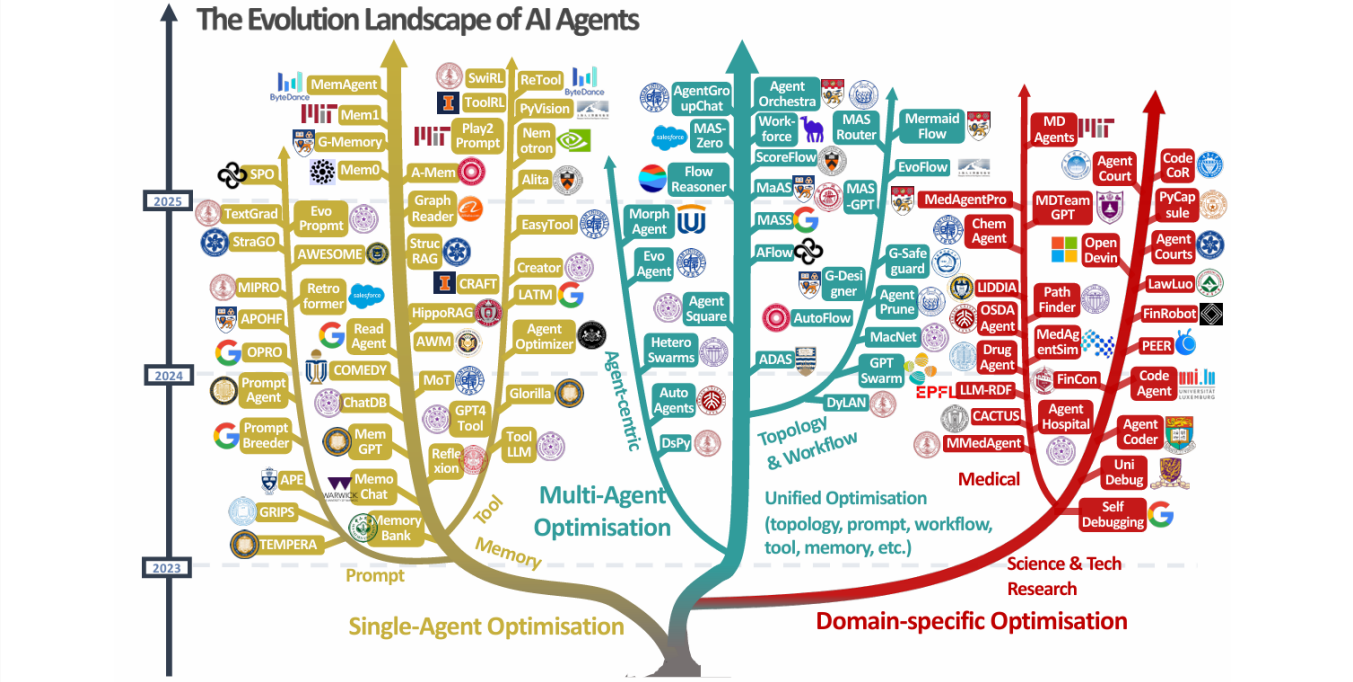
Xiangyuan Xue[1,2] Yifan Zhou[3] Guibin Zhang[4] Zaibin Zhang[5,6] Yijiang Li[7]
Chen Zhang[2] Zhenfei Yin[6]✉ Philip Torr[6] Wanli Ouyang[1,2,8]✉ Lei Bai[2]✉

[1]The Chinese University of Hong Kong [2]Shanghai Artificial Intelligence Laboratory
[3]University of Georgia [4]National University of Singapore
[5]Dalian University of Technology [6]University of Oxford
[7]University of California San Diego [8]Shenzhen Loop Area Institute

- Paper: https://arxiv.org/pdf/2510.08529
- arXiv: https://arxiv.org/abs/2510.08529
- GitHub: https://github.com/xxyQwQ/CoMAS

Research Background

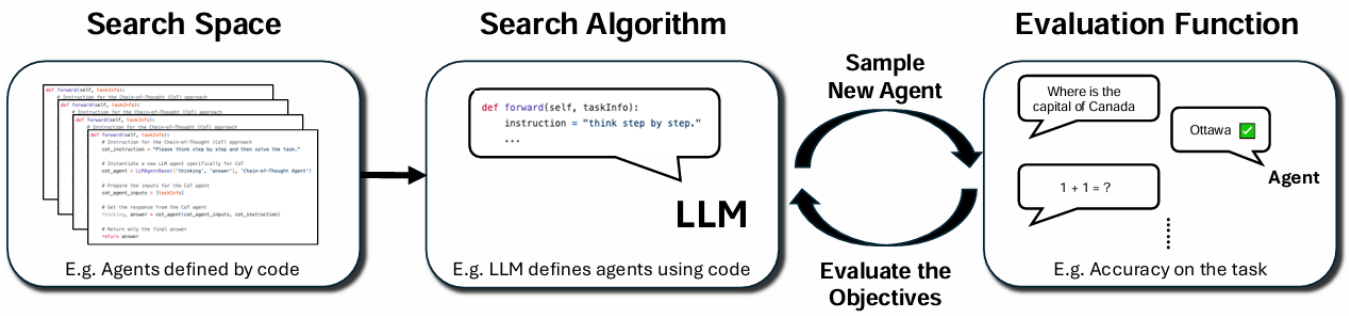
Emergence of Self-Evolving Agents



- **Self-evolution** has become a central research topic, where agents continually evolve rather than remain stagnant.

Fang, Jinyuan, et al. "A comprehensive survey of self-evolving ai agents: A new paradigm bridging foundation models and lifelong agentic systems." arXiv preprint arXiv:2508.07407 (2025).

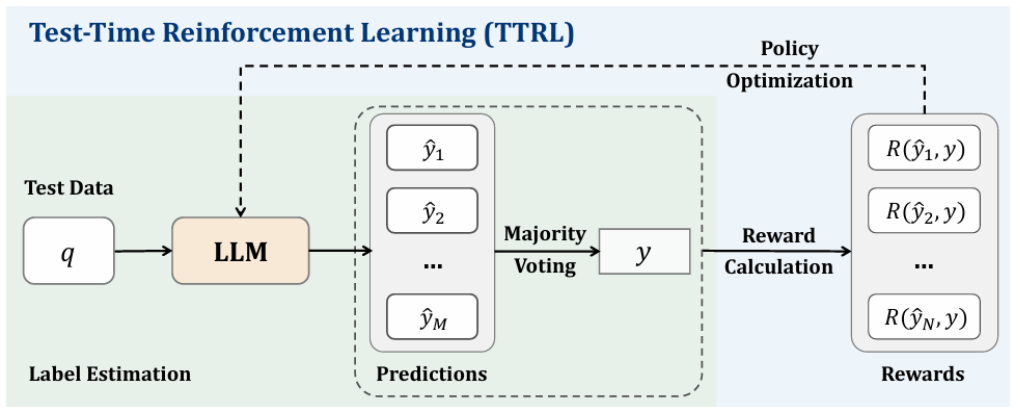
RL-Free Self-Evolution



- Early explorations mainly adopt **RL-free strategies**.
  - Data-Copilot evolves by **expanding external knowledge bases**.
  - RouteLLM evolves by **ensembling multiple agents**.
  - ADAS evolves by **optimizing task workflows**.
  - GPTSwarm evolves by **incorporating symbolic learning**.
- However, these approaches introduce **zero or limited gains** on the capabilities of the foundation models.

Hu, Shengran, Cong Lu, and Jeff Clune. "Automated design of agentic systems." arXiv preprint arXiv:2408.08435 (2024).

RL-Based Self-Evolution



- **RL-based methods** offer a more promising direction.
  - One line of research relies on **external rewards**, such as rule-based verifiers and reward models.
  - Another direction exploits **intrinsic rewards**, such as self-certainty, confidence, semantic entropy, and pseudo-labels.
- Besides, these approaches are restricted to **individual models**.

Zuo, Yuxin, et al. "Ttrl: Test-time reinforcement learning." arXiv preprint arXiv:2504.16084 (2025).

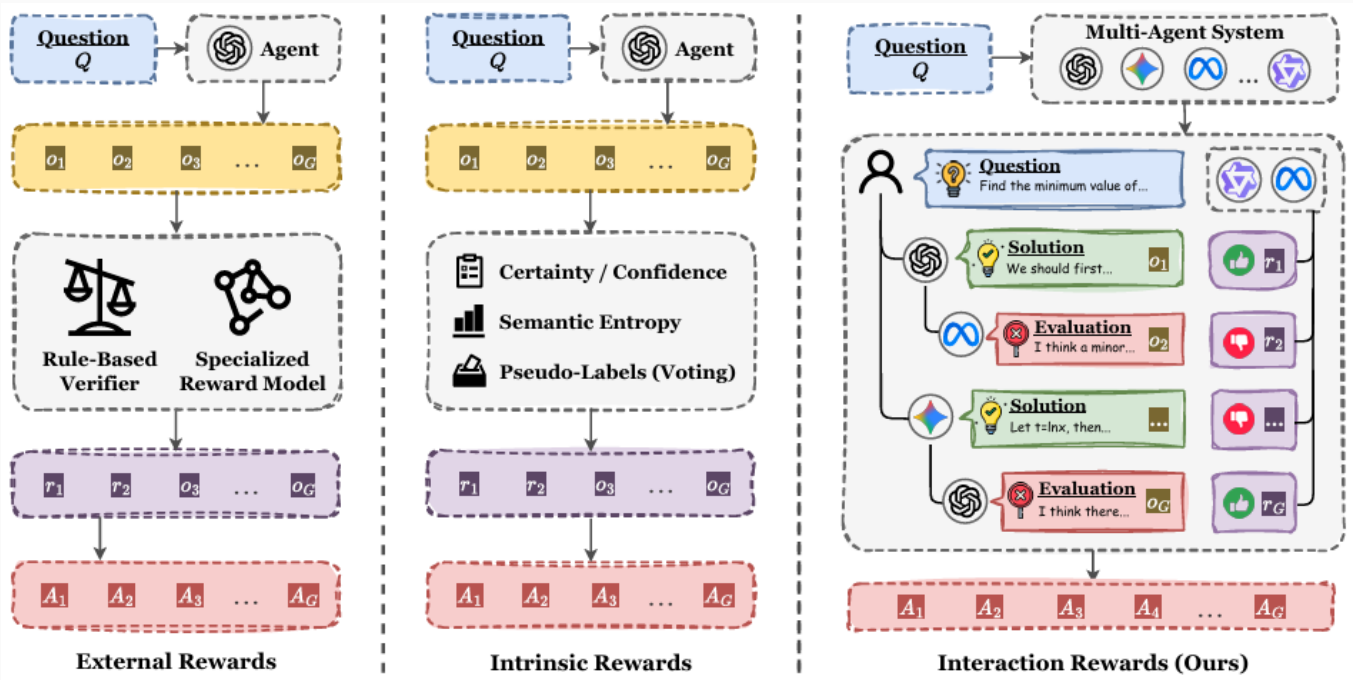
Research Question

- Notably, **human intelligence** provides an alternative paradigm.
  - Human intelligence evolves as a **collective phenomenon** emerging from the diversity and interplay of individuals.
  - In human teams, individuals can learn and improve simply through **mutual discussion and collaboration**.
- This contrast motivates a critical research question.

Research Question

Can agents, akin to human beings, achieve self-evolution by **learning purely from inter-agent interaction** within a multi-agent system, **without relying on external reward signals**?

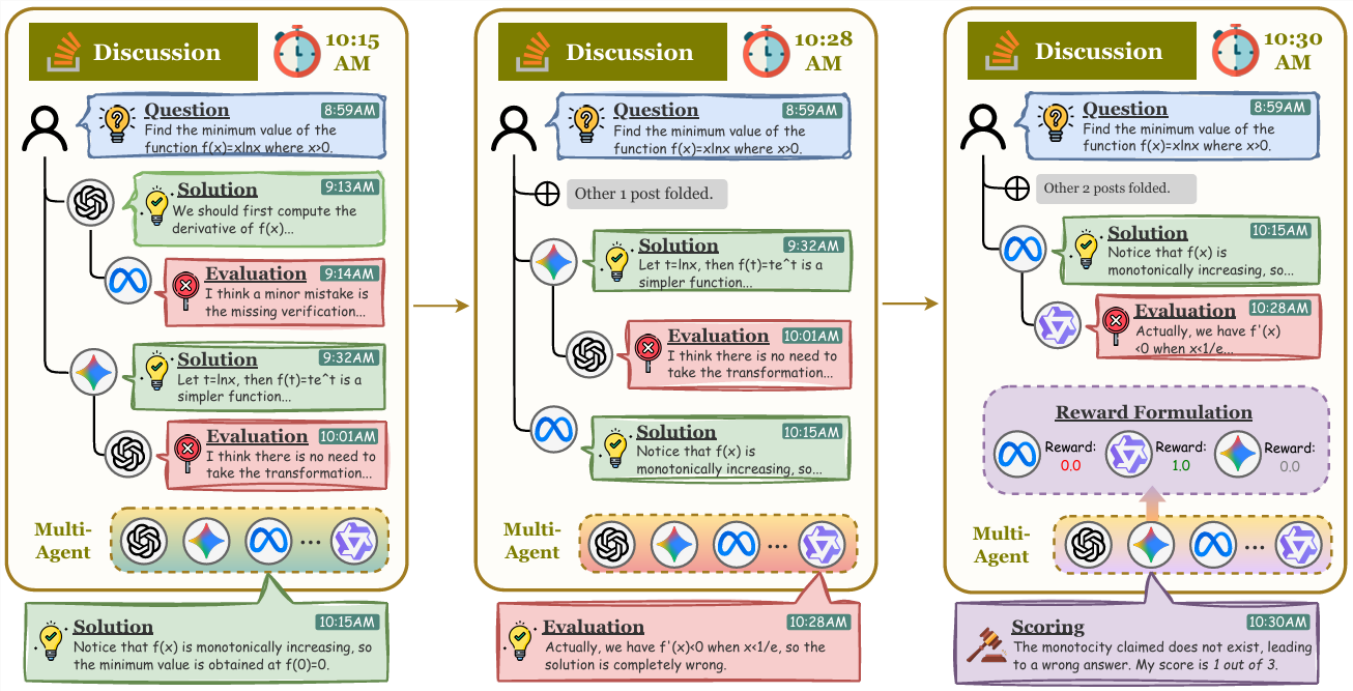
Method Comparison



- To address this question, we propose **Co-Evolving Multi-Agent Systems (CoMAS)**, where agents autonomously improve by learning from rewards generated through interactions.

CoMAS Framework

Framework Overview



- Our CoMAS framework is built on three core components.
  - a community-like **interaction pattern**.
  - an adversarial zero-sum **reward formulation**.
  - an RL-based **policy optimization** process.

Interaction Pattern

- The environment contains l agents $\mathcal{U} = \{u_1, \ldots, u_l\}$.
  - The policy of each agent u_k is π_{θ_k}, parameterized by θ_k.
  - Each agent u_k owns **a unique model weight**.
- The action of u_k can be defined in two equivalent ways.
  - From the **action-level perspective**, we have $o = \pi_{\theta_k}(p)$.
  - From the **token-level perspective**, we have $o_t \sim \pi_{\theta_k}(\cdot|p, o_{<t})$.
- We define three primary interaction patterns in our framework.
  - **Solution**: given a specific question q and its discussion history h_q, the agent generates a solution $s_i = u_k(q, h_q)$.
  - **Evaluation**: given a question q, its history h_q, and a solution s_i, the agent provides a critical evaluation $e_{i,j} = u_k(q, h_q, s_i)$.
  - **Scoring**: given a question q, a solution s_i, and an evaluation $e_{i,j}$, the agent scores the solution $\tau_{i,j} = u_k(q, s_i, e_{i,j})$.

Interaction Pattern

- Given a question q, the discussion **unfolds over m rounds**.
  - In round i, a solution s_i is generated by the active agent.
  - Then n evaluations $\{e_{i,j}\}_{j=1}^{n}$ are generated by n different active agents to figure out the mistakes in the solution.
  - Each $(s_i, e_{i,j})$ pair is scored by the active agent to form $\tau_{i,j}$.
- The entire discussion finally yields **a total of m solutions, $m \cdot n$ evaluations, and $m \cdot n$ scores**.
  - The active agent in each interaction step is selected uniformly at random from the agent pool, i.e., $u_k \sim \mathrm{Uniform}(\mathcal{U})$.
  - The history h_q will be compressed to the last κ rounds to avoid the number of tokens exceeding the context limitation.

Reward Formulation

- We employ **an LLM-as-a-judge approach**, where the rewards are computed from the scores generated in the discussion.
- Each $\tau_{i,j}$ is parsed to extract the score value

$$\hat{\tau}_{i,j} = \mathrm{Extract}(\tau_{i,j}),$$

  where $\mathrm{Extract}(\cdot)$ is a predefined extraction function.
- The score value $\hat{\tau}_{i,j}$ has **three possible choices**.
  - **3**: the solution is correct and the evaluation is unhelpful.
  - **2**: the solution is mostly correct but contains some minor flaws that are pointed out by the evaluation.
  - **1**: the solution contains fatal mistakes identified by the evaluation.

Reward Formulation

- The score value is normalized to the range of $[0, 1]$ and used to compute rewards for the solution and evaluations

$$r(s_i) = \frac{\hat{\tau}_{i,j} - 1}{2}, \quad r(e_{i,j}) = 1 - r(s_i) = \frac{3 - \hat{\tau}_{i,j}}{2}.$$

- The score itself only receives a format reward

$$r(\tau_{i,j}) = \begin{cases} 0, & \text{if } \tau_{i,j} \in \{1, 2, 3\} \text{ correctly extracted} \\ -1, & \text{otherwise} \end{cases}.$$

- Such a reward design indicates **a zero-sum game**, which encourages **both correct solutions and critical evaluations**.

Policy Optimization

- We adopt **REINFORCE++ instead of GRPO** considering the prompts in a batch may be different.
- Let $\mathcal{D}_k = \{(p, o, r(o))\}$ denote the replay buffer collecting all the experiences that are generated by u_k.
- For each sample, the advantage consists of **the trajectory-level reward** penalized by **a cumulative KL-divergence term**

$$A_t = r(o) - \beta \sum_{\lambda=t}^{|o|} \log \frac{\pi_{\theta_k}(o_\lambda | p, o_{<\lambda})}{\pi_{\text{ref}}(o_\lambda | p, o_{<\lambda})},$$

where π_{ref} is a fixed reference policy (i.e., the initial pretrained model), and β controls the strength of KL penalty.

Policy Optimization

- The advantages are then **normalized across the batch** from the replay buffer to stabilize updates

$$\hat{A}_t = \frac{A_t - \mathrm{Mean}(\{A_t\})}{\mathrm{Std}(\{A_t\}) + \epsilon}.$$

- We then use the **surrogate objective** to improve the policy

$$J(\theta_k) = \mathbb{E}_{(p,o,r(o))\sim\mathcal{D}_k}\left[\sum_{t=1}^{|o|} \min\left(\rho_t(\theta_k)\hat{A}_t,\ \mathrm{clip}\big(\rho_t(\theta_k),\, 1-\epsilon,\, 1+\epsilon\big)\hat{A}_t\right)\right],$$

where $\rho_t(\theta_k) = \frac{\pi_{\theta_k}(o_t|p,o_{<t})}{\pi_{\mathrm{old}}(o_t|p,o_{<t})}$ is the importance sampling ratio.

Experimental Result

Experimental Setup

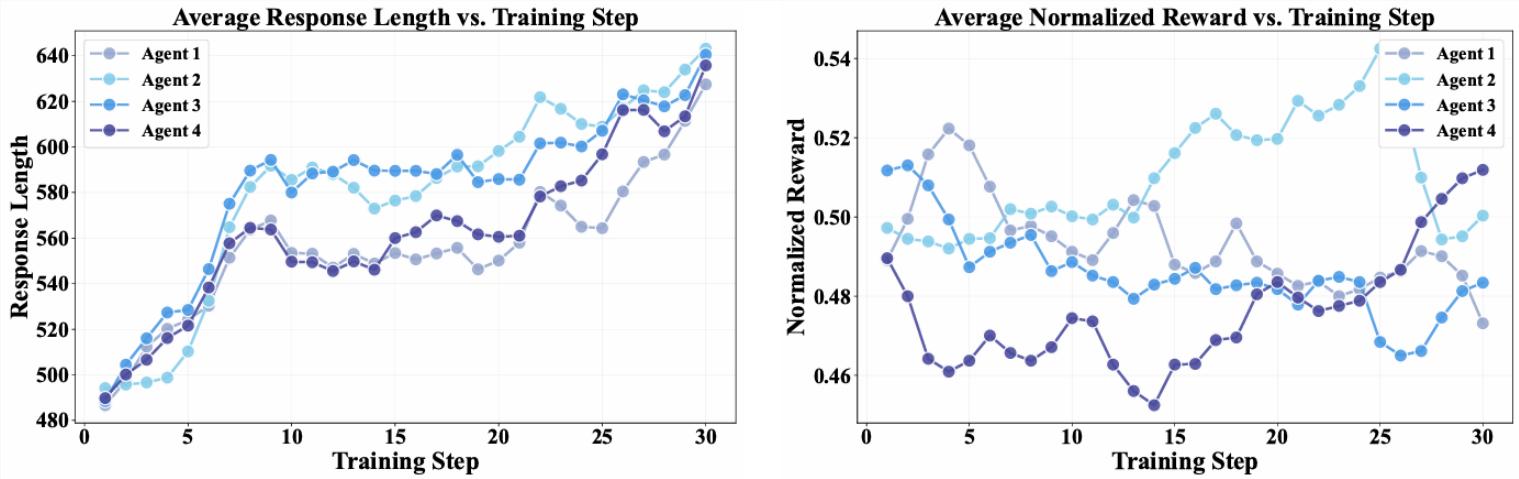
- We employ **Qwen2.5-3B-Instruct** as the base model, with the parameters of $l = 4$, $m = 8$, $n = 1$, and $\kappa = 2$.
- The dataset focuses on reasoning for math, coding, and science.
  - **Training**: 2000 samples collected by mixing non-trivial tasks from **MATH**, **KodCode**, and **WebInstruct-verified**.
  - **Evaluation**: comprehensive coverage over **GSM8K**, **MATH-500**, **HumanEval**, **MBPP**, **SciBench**, **GPQA**, and **MMLU**.
- Besides the untrained base model, we adopt **SRLM**, **MAPoRL**, and **TTRL** as three representative baselines.
- We conduct evaluation on both single-agent (i.e., **Vanilla** and **Consistency**) and multi-agent (i.e., **AutoGen** and **Debate**) setups.

Main Result

	Method	Dataset						
		GSM8K	MATH-500	HumanEval	MBPP	SciBench	GPQA	MMLU
Vanilla	Untrained	84.00	51.40	68.90	54.00	32.67	26.79	61.40
	SRLM	83.40 (-0.60)	52.20 (+0.80)	68.29 (-0.61)	53.80 (-0.20)	32.67 (+0.00)	27.01 (+0.22)	61.00 (-0.40)
	MAPoRL	84.80 (+0.80)	52.60 (+1.20)	69.51 (+0.61)	56.00 (+2.00)	34.07 (+1.40)	**28.12** (+1.34)	61.40 (+0.00)
	TTRL	84.40 (+0.40)	**53.40** (+2.00)	68.29 (-0.61)	**57.40** (+3.40)	34.47 (+1.80)	25.45 (-1.34)	61.60 (+0.20)
	CoMAS (Ours)	**85.40** (+1.40)	52.80 (+1.40)	**70.73** (+1.83)	56.20 (+2.20)	**34.67** (+2.00)	27.46 (+0.67)	**62.40** (+1.00)
Consistency	Untrained	85.40	55.00	73.78	55.80	36.47	28.79	63.20
	SRLM	86.40 (+1.00)	55.40 (+0.40)	75.00 (+1.22)	56.20 (+0.40)	36.67 (+0.20)	29.24 (+0.45)	65.20 (+2.00)
	MAPoRL	85.80 (+0.40)	55.40 (+0.40)	75.61 (+1.83)	57.00 (+1.20)	**39.08** (+2.61)	**31.47** (+2.68)	63.20 (+0.00)
	TTRL	**88.20** (+2.80)	**56.80** (+1.80)	73.78 (+0.00)	59.00 (+3.20)	38.48 (+2.00)	27.23 (-1.56)	63.80 (+0.60)
	CoMAS (Ours)	87.20 (+1.80)	55.80 (+0.80)	**77.44** (+3.66)	**59.20** (+3.40)	37.68 (+1.20)	29.69 (+0.89)	**65.60** (+2.40)
AutoGen	Untrained	52.60	38.40	39.63	29.80	20.24	16.29	37.40
	SRLM	58.00 (+5.40)	41.80 (+3.40)	44.51 (+4.88)	32.00 (+2.20)	21.24 (+1.00)	17.86 (+1.56)	42.40 (+5.00)
	MAPoRL	50.00 (-2.60)	37.40 (-1.00)	39.63 (+0.00)	34.60 (+4.80)	20.64 (+0.40)	21.65 (+5.36)	40.40 (+3.00)
	TTRL	41.00 (-11.60)	37.80 (-0.60)	23.17 (-16.46)	22.80 (-7.00)	19.64 (-0.60)	14.06 (-2.23)	34.00 (-3.40)
	CoMAS (Ours)	**72.40** (+19.80)	**45.80** (+7.40)	**50.61** (+10.98)	**38.00** (+8.20)	**22.85** (+2.61)	**22.99** (+6.70)	**50.60** (+13.20)
Debate	Untrained	84.60	55.00	71.34	54.80	38.68	28.35	62.80
	SRLM	84.60 (+0.00)	54.80 (-0.20)	72.56 (+1.22)	53.60 (-1.20)	38.68 (+0.00)	28.57 (+0.22)	64.60 (+1.80)
	MAPoRL	85.40 (+0.80)	53.60 (-1.40)	74.39 (+3.05)	55.60 (+0.80)	**39.88** (+1.20)	**31.47** (+3.12)	64.80 (+2.00)
	TTRL	**86.20** (+1.60)	55.20 (+0.20)	73.78 (+2.44)	**58.00** (+3.20)	37.88 (-0.80)	29.02 (+0.67)	64.00 (+1.20)
	CoMAS (Ours)	85.20 (+0.60)	**55.40** (+0.40)	**77.44** (+6.10)	55.60 (+0.80)	39.08 (+0.40)	29.91 (+1.56)	**65.20** (+2.40)

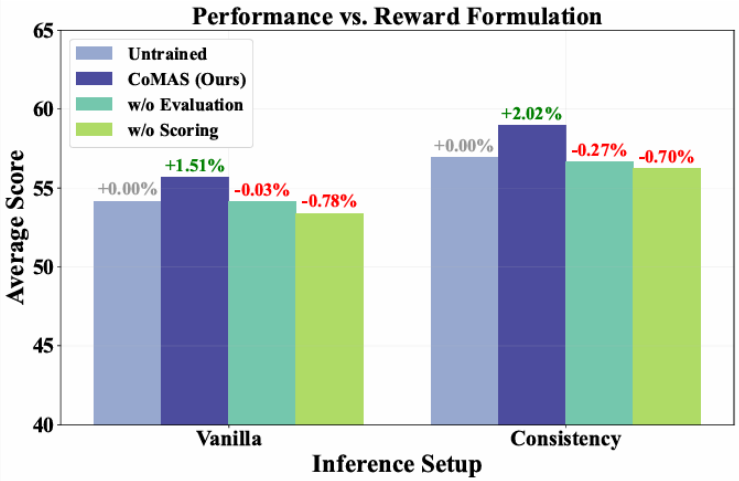
- CoMAS delivers **consistent, and in many cases, state-of-the-art** performance, while baselines show instability and degradation.

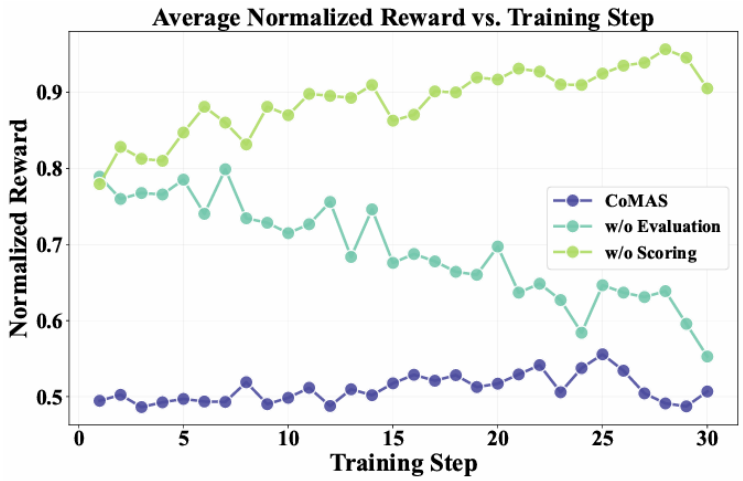
Training Dynamic



- We track the entire **training dynamics** of CoMAS.
  - The **response length grows consistently**, indicating that the agents produce more elaborate solutions and evaluations over time.
  - The **normalized reward converges around** 0.5 for all agents, indicating that the training process is stable and balanced.

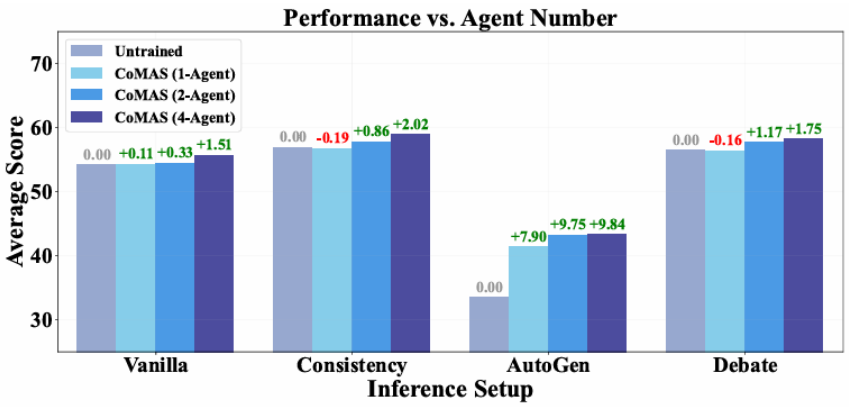
Reward Formulation

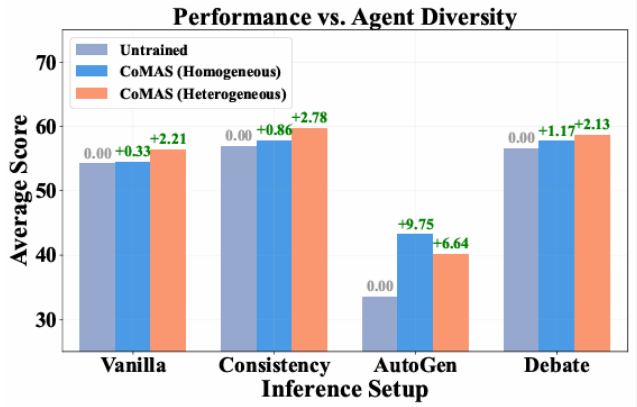




- We **remove evaluation and scoring** from CoMAS to investigate the effectiveness of the adversarial reward design.
  - **w/o evaluation**: agents **become overly strict** when judging, and the normalized reward keeps decreasing.
  - **w/o scoring**: agents **learn to collude** when scoring, and the normalized reward keeps increasing and remains around 1.0.

Framework Scalability





- We increase **the number and diversity** of agents to investigate the scalability of our CoMAS framework.
  - **Number**: with the number of agents increasing from 1 to 4, the performance of CoMAS **consistently improves**.
  - **Diversity**: the performance of CoMAS with heterogeneous agents **consistently outperforms** that with homogeneous agents.

Takeaway Message

Takeaway Message

- We propose CoMAS, which enables agents to **co-evolve via interaction rewards** without any external supervision.
- CoMAS is built upon **three core components**: interaction pattern, reward formulation, and policy optimization.
- CoMAS **consistently outperforms** untrained agents and is **competitive with or surpasses** representative baselines.
- The design of **adversarial rewards** in CoMAS is critical for preventing training collapse and reward hacking.
- CoMAS demonstrates **promising scalability** with more agents and diverse agents both improving its performance.

Thank You for Listening!