MAPoRL: Multi-Agent Post-Co-Training for Collaborative Large Language Models with Reinforcement Learning

Chanwoo Park[♡♮] **Seungju Han**[♠] **Xingzhi Guo**[♢]
Asuman Ozdaglar[♡] **Kaiqing Zhang**[♣] **Joo-Kyung Kim**[♢]
[♡]MIT [♠]Stanford [♢]Amazon [♣]UMD
{cpark97,asu}@mit.edu, seungju@stanford.edu,
{guxingzh, jookyk}@amazon.com, kaiqing@umd.edu

Abstract

Leveraging multiple large language models (LLMs) to build collaborative *multi-agentic* workflows has demonstrated significant potential. However, most previous studies focus on *prompting* the out-of-the-box LLMs, relying on their innate capability for collaboration, which may not improve LLMs' performance as shown recently. In this paper, we introduce a new *post-training* paradigm **MAPoRL** (**M**ulti-**A**gent **P**ost-c**o**-training for collaborative LLMs with **R**einforcement **L**earning), to explicitly elicit the collaborative behaviors and further unleash the power of multi-agentic LLM frameworks. In **MAPoRL**, multiple LLMs first generate their own responses independently and engage in a multi-turn discussion to collaboratively improve the final answer. In the end, a **MAPoRL** verifier evaluates both the answer and the discussion, by assigning a score that verifies the correctness of the answer, while adding incentives to encourage corrective and persuasive discussions. The score serves as the co-training reward, and is then maximized through multi-agent RL. Unlike existing LLM post-training paradigms, **MAPoRL** advocates the *co-training* of multiple LLMs together using *RL* for better generalization. Accompanied by analytical insights, our experiments demonstrate that training individual LLMs alone is insufficient to induce effective collaboration. In contrast, multi-agent co-training can boost the collaboration performance across benchmarks, with generalization to unseen domains. The code is available at https://github.com/chanwoo-park-official/MAPoRL.

1 Introduction

Recent advances in large language models (LLMs) have highlighted their potential for collaboration, particularly within the *multi-agentic* framework (Du et al., 2024; Li et al., 2023; Kim et al., 2024b). The shift from single-agent to multi-agent systems introduces new dimensions and challenges in enabling effective collaboration among LLM agents. Recent approaches to multi-LLM collaboration mostly rely on *prompting pre-trained* models. However, such approaches struggle with achieving genuine collaboration among the agents. For example, multi-agent debate does not consistently lead to improved performance with additional turns (Huang et al., 2024).

This limitation may be somewhat expected – while LLMs are able to *simulate* collaboration procedures, they were *not* explicitly *trained* to achieve effective cooperation. In theory, it is not hard to imagine that single-agent training is insufficient for collaboration – an *untrained* and *non-strategic* opponent can fail to act in a way that promotes collaboration. Instead, achieving collaborative behaviors requires interactive training environments where each agent actively engages with others, and dynamically optimizes the strategy (Gagne, 1974; Macy, 1991; Hertz-Lazarowitz et al., 2013). Moreover, conventional approaches such as supervised fine-tuning (SFT), as we will show, are inadequate for this purpose, either: merely mimicking multi-agent interactions from training data may not lead to effective collaboration.

To develop more effective collaborative agents, we propose **M**ulti-**A**gent **P**ost-c**o**-training for collaborative LLMs with **R**einforcement **L**earning (**MAPoRL**), a *co-training* paradigm for multiple LLMs using multi-agent reinforcement learning (MARL). In **MAPoRL**, within the pre-defined frameworks for multi-agent collaboration (e.g., the debate framework (Du et al., 2024)), each agent receives rewards for their responses during collaboration, based on the quality of their answers and interactions. The objective for each agent in **MAPoRL** is to maximize their own value function, defined as the expected cumulative sum of rewards over the

[♮]This work was initiated during an internship at Amazon AGI.

course of the collaboration.

To further encourage cooperation in **MAPoRL**, we incorporate incentives for successful interactions and penalties for collaboration failures, steering the LLMs toward more effective and aligned behaviors. Through a simplified game-theoretic example, we validate the following insights: 1) single-agent training alone is insufficient to produce genuinely cooperative agents, and 2) co-trained agents can reach an equilibrium that exhibits cooperative behavior.

To assess the effectiveness of **MAPoRL**, we conduct experiments across diverse tasks and evaluation strategies. Specifically, we train multi-agent LLMs for tasks such as mathematical reasoning (GSM8k (Cobbe et al., 2021)) and natural language inference (ANLI (Nie et al., 2020)), comparing their performance against baseline approaches. Additionally, we evaluate the robustness of our method by testing agents on out-of-domain tasks (e.g., training on a NLI task and evaluating on a math dataset), demonstrating the generalization capabilities of our approach. We also explore the collaboration among agents of varying capabilities, by analyzing the impact of training *heterogeneous* LLMs together.

To the best of our knowledge, this study is *among the first works to explore the training of multi-LLM systems as a whole*[1], using RL, for multi-LLM collaboration.

2 Analytical Insights: Collaborate to Solve Hard Questions

In this section, we present a simplified model of multi-LLM collaboration and explain (a) why *co-training* multiple LLMs is necessary compared to training a single agent, and (b) the role of incentives to further enhance collaboration during training. We validate both aspects through experiments in Section 4.

2.1 Problem Setup

We consider questions that inherently require collaboration for a successful solution. For instance, solving complex mathematical problems often requires collaboration among multiple agents (Liang et al., 2024; Du et al., 2024). Beyond mathematics, collaboration can also enhance the performance on tasks related to privacy, factuality, and reliability (Feng et al., 2025). We model the interaction among LLMs as a repeated game with T turns. For simplicity, we assume that in each turn, each agent chooses between two actions: *Collaborate* (a_0) or *Act Independently* (a_1). For a given question q, we define $C(q)$ as a non-negative integer representing the *collaboration threshold*. The agents achieve *collaborative synergy* if, over the course of the T-turn interactions, the total number of collaborative actions (a_0) of all the agents meets or exceeds $C(q)$. When collaborative synergy is achieved, each agent receives a reward $R_{\text{syn}}(q) = 1$, representing a (near-)guaranteed correct solution. Prior to achieving synergy, agents receive rewards based on their chosen actions: a reward of $R_{\text{col}}(q)$ for choosing to collaborate (a_0) and $R_{\text{ind}}(q)$ for acting independently (a_1), where $R_{\text{col}}(q) < R_{\text{ind}}(q)$ (see Remark 3 for a detailed justification on the setup). This reward structure creates a tradeoff between short-term accuracy and long-term collaborative success. This setup is related to the classical Coordination Games (Cooper, 1999) in game theory if R_{syn} is large. We introduce a new collaboration threshold and synergy mechanism that shapes the transition from independent actions to collaborative behavior in *multiple turns*, to better model the collaboration procedure among multiple LLMs.

Remark 1 (Rationale Behind the Setup)**.** This formalization captures several key aspects of complex problem-solving dynamics. Choosing to collaborate (a_0) represents contributing *exploratory ideas* or *partial solutions*. While these contributions have a lower immediate probability of correctness $R_{\text{col}}(q)$, they are essential building blocks towards the complete solution. Acting independently (a_1) represents using conventional approaches that may yield a higher *immediate probability* of correctness $R_{\text{ind}}(q)$, but may contribute less to solving particularly challenging problems. The collaboration threshold $C(q)$ represents the minimum amount of collaboration efforts and idea generation needed to solve complex problems. Once this threshold is reached (i.e., achieving collaborative synergy), the agents can combine their insights to solve the challenging problem, with a higher reward $R_{\text{syn}}(q)$.

2.2 Analytical Observations

To provide intuition for why co-training is necessary and single-agent training may be inadequate,

[1]Together with the contemporaneous works Subramaniam et al. (2025) and Zhao et al. (2025), both of which were released within the past month while preparing this paper. In contrast to **MAPoRL**, the algorithms therein were based on (iterative) SFT, instead of RL. Also, Motwani et al. (2024) provided a method to train verifier-generation-refiner system with DPO.

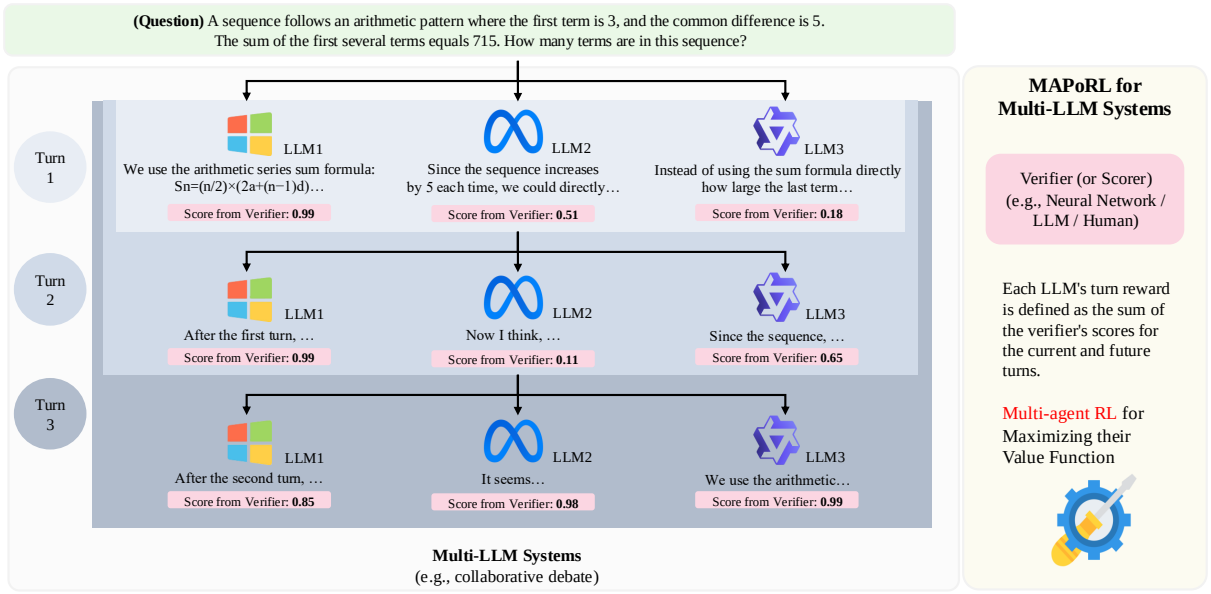

Figure 1: **MAPoRL** can be applied to any multi-LLM system with a scorer/verifier. In the illustrated example, it is integrated into a collaborative debate system for mathematical problem-solving. LLMs generate responses based on the multi-agent system pipeline, and a scorer/verifier evaluates their outputs. The reward for each LLM is determined based on these scores, which may include both current and future pipeline evaluations. Multi-Agent RL is employed to maximize each agent's value function.

we analyze the simplest case with $T = 2$ and $C(q) = 1$ to illustrate the fundamental differences between single and multi-agent training. We provide formal statements and proofs in Appendix C.

Observation 1. *Suppose that the opponent selects action a_0 with probability $\pi(q)$ for each question q. Then, the optimal strategy for the agent is as follows: if $(R_{syn}(q) - R_{ind}(q))\pi(q) \geq R_{ind}(q) - R_{col}(q)$, then the optimal strategy for question q is to collaborate (a_0). Otherwise, the optimal strategy is to act independently (a_1).*

This shows the dependence of the agent's strategy on the opponent's behavior. If the opponent is *not collaborative* enough and *non-strategic*, then $\pi(q)$ will be small, leading the trained model to behave in a *non-collaborative* way.

Observation 2 (Informal)**.** *If both agents are trained to maximize their individual cumulative rewards with an entropy regularization term scaled by τ, then as $\tau \to 0$, they will collaborate if:*

$$R_{syn}(q) > \max(3R_{col}(q) - 2R_{ind}(q), 2R_{ind}(q) - R_{col}(q)).$$

Observation 2 can be proved by adapting the results of Zhang and Hofbauer (2016), and transforming our setup with $T = 2$ into a matrix game. This observation implies that when both agents optimize their own cumulative reward, they will naturally choose collaboration when $R_{\text{syn}}(q)$ is high enough, which emphasizes the importance of additional incentives to promote collaborative synergy. Due to this observation, in Section 3.3, we incentivize collaboration by providing a higher $R_{\text{syn}}(q)$.

2.3 Toy Experiments with $T = 10, 20$ Turns

We illustrate the benefit of *jointly optimized* (multi-agent) policies over those obtained from a *single-agent* approach in our setting, with longer $T = 10, 20$ turns. Each question q is associated with the rewards $R_{\text{col}}(q)$, $R_{\text{ind}}(q)$, and $R_{\text{syn}}(q)$, along with a collaboration threshold $C(q)$. Further details on the choices of these quantities be found in Appendix D.

We first consider a *single* agent interacting with a fixed opponent whose probability of collaborating, $\pi(q)$, is set at $\{0.5, 0.6, 0.7\}$. Despite the relatively high likelihood of collaboration from the opponent, the single-agent policy, which optimizes its response to the fixed opponent, does not result in effective collaboration (Figure 5). Instead of learning to strategically engage with the opponent's behavior, the single-agent policy, which follows a best-response strategy to the fixed opponent, tends to *avoid* collaboration, highlighting the limitations of a single-agent framework when facing a fixed, non-strategic opponent.

Next, we consider two *jointly optimizing* (multi-agent) learners who adapt their policy based on the other's actions. Concretely, we compute an