Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. 2021. Multi-agent reinforcement learning: A selective overview of theories and algorithms. Handbook of reinforcement learning and control, pages 321–384.

Jun Zhao, Can Zu, Hao Xu, Yi Lu, Wei He, Yiwen Ding, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Longagent: Scaling language models to 128k context through multi-agent collaboration. In EMNLP.

Wanjia Zhao, Mert Yuksekgonul, Shirley Wu, and James Zou. 2025. Sirius: Self-improving multi-agent systems via bootstrapped reasoning. arXiv preprint arXiv:2502.04780.

Banghua Zhu, Michael Jordan, and Jiantao Jiao. 2023. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In International Conference on Machine Learning, pages 43037–43067. PMLR.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. arXiv preprint arXiv:1909.08593.

A Detailed Related Work Discussion

Multi-Agent Reinforcement Learning. Various algorithms have been proposed to address multi-agent reinforcement learning (MARL) (Hernandez-Leal et al., 2019; Zhang et al., 2021), including multi-agent Proximal Policy Optimization (PPO) (Yu et al., 2022), and value function factorization techniques such as QMIX and VDN (Rashid et al., 2020; Sunehag et al., 2018). In the context of language models and collaborative debating we focus on, MARL takes on a particular and unique form. Here, each agent's state is represented by the sequence of previous responses from all the agents, with each agent deciding the next token based on this history. LLMs provide compact state representations through their hidden layers, enabling the use of long debate histories.

Multi-Agent Collaboration with LLMs. An array of studies have explored effective collaboration frameworks among multiple large language model agents to solve complex tasks (Wu et al., 2023; Li et al., 2024; Zhao et al., 2024). For example, "role-playing"-based approaches utilized multi-agent LLMs by assigning a specific role to each LLM (Li et al., 2023), and "multi-agent debate"-based approaches prompted each LLM agent to solve the task independently and then discuss (Du et al., 2024; Khan et al., 2024). In a debate, the agents reason through each other's answers to converge on a consensus response, which may improve the factual accuracy, mathematical ability, and reasoning capabilities of the LLM (Du et al., 2024; Liang et al., 2024; Kim et al., 2024b). Similar mult-agentic frameworks include voting (Wang et al., 2023), group discussions (Chen et al., 2024), and negotiating (Fu et al., 2023). However, all of these frameworks rely heavily on prompt engineering, which may lead to sub-optimal results (Huang et al., 2024), and do not consider *training* LLMs specifically for collaboration. Therefore, while multi-LLM systems seem promising at the first glance, their performance may be limited when using the out-of-the-box (pretrained) LLM with only prompt tuning, which highlights the need for *training* for better multi-agent collaboration. Recently, Stengel-Eskin et al. (2025) introduced a training framework for accepting or rejecting persuasion in multi-agent systems. Additionally, very recently, Subramaniam et al. (2025) and Zhao et al. (2025) focused on training the entire multi-agent systems using iterative SFT. In contrast, **MAPoRL** employs (multi-agent) RL to train the whole multi-LLM system. Recently, after **MAPoRL**was released, Liao et al. (2025) provided a similar training system of multi-agents with reinforcement learning.

RL for LLM Training. RL has been widely used in post-training LLMs, e.g., for improving factuality (Tian et al., 2024), code generation (Le et al., 2022), and more recently and significantly, reasoning (Guo et al., 2025). One prevalent approach of RL for LLM training is RL from human feedback (RLHF) (Ziegler et al., 2019; Ouyang et al., 2022; Bai et al., 2022; Ahmadian et al., 2024). RL offers a smooth generalization to the *multi-turn* setting based on the Markov decision process (MDP) model, and there have been attempts to apply multi-turn RL for LLM training, such as RLHF for multi-turn model training to enhance the dialogue abilities (Shani et al., 2024), or deriving multi-turn RL objective for the improvement of mathematical reasoning (Xiong et al., 2025). However, the major difference from our work is that, these works did not consider multi-agent settings for collaboration. Recently, Kumar et al. (2025) enhanced LLMs' ability to self-correct using an RL-based approach. Our framework can accommodate this case by using a single agent in **MAPoRL**.

B Additional Literature Review

Multi-Agent RL. Multi-agent reinforcement learning (MARL) has achieved significant advancements, particularly in cooperative games and their real-world applications, such as coordinating robot swarms (Hüttenrauch et al., 2017) and self-driving vehicles (Shalev-Shwartz et al., 2016). (A comprehensive overview of MARL can be found in Zhang et al. (2021)). The primary challenge in MARL lies in the exponentially large action space, making it difficult to optimize the policy for each agent. Various approaches have been proposed to address this issue, including multi-agent Proximal Policy Optimization (PPO) (Yu et al., 2022), value function factorization methods (QMIX, VDN) (Rashid et al., 2020; Sunehag et al., 2018), and network-based formulations for multi-agent learning (Park et al., 2023). These methods

aim to make MARL more scalable with a large number of agents, mostly focusing on the classical models of stochastic/Markov games.

In the context of language models and collaborative debate systems, MARL takes on a unique form. Here, each agent's state is represented by the sequence of previous responses from all agents, with each agent deciding the next token based on this history. The detailed mathematical formulation for reinforcement learning in language models can be found in several theoretical and empirical studies on reinforcement learning with human feedback (RLHF) (*e.g.,* Ouyang et al. (2022); Zhu et al. (2023); Park et al. (2024)). LLMs provide high-quality state representations through their hidden layers, enabling the consideration of long debate histories. Moreover, the sequential nature of these interactions inherently captures non-Markovian policies due to the extended sequence of responses.

Teaching LLM Self-Correction. As mentioned in the main paper, single-agent self-correction and multi-agent collaboration has a very interesting relationship. Single-agent self-correction and multi-agent collaboration rely on multi-turn interactions—either internally, within a single agent, or collaboratively, among multiple agents—to improve results by challenging initial outputs and refining them through iteration. In single-agent systems, self-correction functions like an internal debate. The agent evaluates its own output over multiple turns, identifying potential mistakes and proposing alternative solutions. This process mirrors human reflection, where reconsideration often leads to improved conclusions. Meanwhile, in multi-agent systems, different agents engage in a collaborative debate, questioning and refining each other's answers. By interacting in multiple rounds, these agents combine their individual perspectives to correct errors and arrive at more accurate solutions.

There are several prior works aiming to improve LLMs' ability to self-correct. First line of work is using prompting technique, which guides LMs via prompting to iteratively correct the model outputs (Madaan et al., 2024). However, some works use the ground-truth labels to determine when to stop the self-correction (Kim et al., 2024a; Shinn et al., 2024; Yao et al., 2023), which is not applicable in the real-world scenarios where answer is not available for the tasks, and it is shown that under such scenarios the models can not do self-correct effectively (Huang et al., 2024).

Another line of works train LLMs to *learn* self-correction; Qu et al. (2024) introduced an approach using stronger LLMs to obtain multi-turn trajectories that have better responses through the iteration, and uses this data to fine-tune LLMs to learn self-correction. Different from this work, our approach do not require stronger LLMs for demonstrations, relying solely on the reward for training. Welleck et al. (2023) proposed supervised fine-tuning to train a corrector model that can edit the model response iteratively, but this is specified the type of collaboration in the generate-then-refine pattern, which can be sub-optimal to learned by the models. Kumar et al. (2025) employed an RL-based approach for the self-improvement of language models.

Multi-Agent LLMs with Game Theory. Recent work has actively explored the strategic interactions of LLM agents within game-theoretic frameworks, as demonstrated in studies such as Park et al. (2025); Brookins and DeBacker (2023); Akata et al. (2023); Lorè and Heydari (2023); Fan et al. (2024). Our paper can be viewed as training LLMs as solvers of cooperative games such as solving mathematical problems together.

C Deferred Content of Section 2

Remark 3 (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 $R_{\text{syn}}(q)$.