Proceedings of the 4th International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/2026.36623

A Survey on Reinforcement Learning Optimization Methods for Multi-Agent Collaboration of Large Language Models

Qianling Zhang

School of International Exchange, Xiangtan University, Xiangtan, China
zql111@outlook.com

Abstract. The integration of reinforcement learning (RL) into the optimization of multi-agent collaboration for Large Language Models (LLMs) is an important combination of two advanced areas, Multi-Agent Systems (MAS) and LLMs. This paper thoroughly examines the main approaches, evaluation standards, recent progress and existing problems in this new field. The research categorizes the existing solutions into three types. The first type is the individual reward-based fine-tuning which optimizes each agent individually. The second one is the joint reward-based centralized training which uses a common reward signal for collaborative learning. The third type is the interaction during the inference phase which enhances cooperation through real-time communication during the generation process. Among these, the joint reward methods demonstrate greater efficiency and effectiveness in cooperation, but they are still restricted by the difficulty in data generalization and model heterogeneity. After careful analysis of these problems, this paper proposes two possible directions for further research. One is to establish dynamic evaluation criteria to improve the evaluation accuracy, and the other is to adapt to various model architectures. The paper's objective is to provide a comprehensive guide to assist future studies on RL-supported multi-agent LLM collaboration.

Keywords: Multi-Agent Reinforcement Learning, Large Language Models, Collaboration Optimization, Joint Reward

1. Introduction

Large Language Models (LLMs) have become the main technological basis in the area of artificial intelligence. From GPT-4 to DeepSeek-R1 [1], individual models have shown remarkable abilities in tasks like logical reasoning, code generation and complicated question answering. Nevertheless, the drawbacks of "uniform thinking" and "capability limitations" in single models prevent them from dealing with the complex real-life problems which need cross-validation from different viewpoints and division of roles. Multi-Agent Systems (MAS) offer a suitable approach to solve these problems. By assigning multiple LLM agents to work in cooperation, it is anticipated that the emergent capabilities will be better than those of single models [2].

Concerning the present research field, scholars have done considerable investigations. The early multi-agent debate methods enhanced the factual accuracy by means of multi-round debates and majority voting procedures [3]. AutoGen is a comprehensive multi-agent conversation system which

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.36623

permits flexible division of roles [4]. MetaGPT imitates the whole software development procedure, allowing specific LLMs to carry out coding jobs in turn. Likewise, ChatDev has illustrated the applicability of multi-agents in particular areas by simulating the software development processes [5]. These LLM-based Multi-Agent Systems possess great potential in fields like software engineering and social simulation.

Although great progress has been made in previous studies, the main difficulty is to make the agents really learn cooperation instead of just following the fixed dialogue rules. Reinforcement Learning (RL) can provide an effective way to solve this problem. In contrast to methods which depend on manually constructed prompts, RL enables agents to acquire cooperative strategies through interactions with the environment independently, having better generalization ability and adaptability [6]. From 2024, a great deal of research has used RL techniques to improve the multi-agent cooperation of LLMs. For example, SPIRAL improves the reasoning ability based on a zero-sum game self-play method. CoRL introduces conditional rewards for different roles to promote role differentiation. MAFT self-optimizes by means of various reasoning processes. MAGRPO represents the LLM cooperation as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP), and achieves efficient training under the Centralized Training with Decentralized Execution (CTDE) framework by estimating the relative advantage within the group. Though MAPPO was initially developed for game situations, its CTDE framework has been extensively applied in the later studies of LLM's cooperation.

2. Overview of mainstream methods

Currently, applying Multi-Agent Reinforcement Learning (MARL) to enhance the collaboration optimization of LLMs includes three main categories of methods.

2.1. Individual reward-based fine-tuning methods

The main concept of this category is to divide the multi-agent cooperation problem into several independent optimization sub-problems. It establishes particular individual reward functions for each LLM agent, enabling them to learn cooperation strategies by means of trial and error according to their own optimization objectives.

In regard to the representative works, SPIRAL develops a self-play framework based on a zero-sum game. Through continuous self-play of LLM agents in adversarial conditions, it improves the model's robustness in complicated reasoning tasks without the requirement of manually labelled data. CoRL presents a conditional reward system which takes into account both the task completion and explicitly regulates the agent's behaviour, thus effectively guiding the differentiation and special cooperation among the heterogeneous agents [7]. MAFT puts forward an improvement method based on various reasoning sequences, encouraging the agents to search for better solving strategies while keeping the logical consistency by choosing the good reasoning routes as positive examples [8].

Though this category possesses the feature of flexible reward setting and can satisfy the personalized optimization goals for various positions, there are also some disadvantages. Firstly, it is very hard to construct individual reward functions and generally requires deep understanding of the field to avoid reward cheating or deviation from the objectives. Secondly, as each agent changes its own policy independently, the environment becomes extremely unstable for any single agent. Theoretical studies by Tan show that in the absence of coordination measures, these separate

DOI: 10.54254/2755-2721/2026.36623

learning approaches may not ensure convergence in complicated cooperative tasks and are probably led to local optima [9].

2.2. Joint reward-based centralized training methods

This category adheres to the Centralized Training with Decentralized Execution (CTDE) approach. The main principle is that all agents have a common global team reward, using global information integration during the centralized training stage to coordinate their strategies, and then making independent distributed decisions in the execution stage.

In regard to the representative works, MAGRPO represents the cooperation of LLMs as a Dec-POMDP and puts forward the technology of estimating intra-group relative advantage in order to significantly decrease the variance of policy gradients, thus accomplishing efficient and stable joint training under the CTDE framework [10]. Although MAPPO was originally used in multi-agent game situations, its principle of introducing a centralized value network to aid in policy updates has provided an important algorithmic standard for the later research on LLM cooperation, demonstrating the feasibility of controlling the distributed actors through a centralized critic network under the condition of common reward signals [11].

The main advantage of this type is to avoid the difficult task of designing individual rewards. The joint reward signals naturally motivate the agents to cooperate for the benefit of the whole team, thus producing more advanced coordination strategies. However, its disadvantage is the very high requirement for sample efficiency. The policy gradient estimation using Monte Carlo sampling frequently experiences high variance, causing oscillations during the training process. In addition, most successful examples are limited to the verification of short-term models with few parameters, which still encounter great difficulties in memory and computational resources when dealing with long-term relationships and collaborative work among different large models.

2.3. Inference-stage multi-agent interaction methods

This category does not deal with the adjustment or training of parameter weights of LLMs. It concentrates on the coordination of the behavioral interactions among several fixed-parameter LLM agents during the inference stage by means of precise prompt engineering.

In regard to the representative works, multi-agent debate bases on the assumption of "truth wins by majority", arranging several agents in multi-round debates and discussions, and applying a majority voting system to effectively reduce the errors in single models and enhance the factual accuracy. AutoGen, as a comprehensive multi-agent conversation structure, incorporates a "human-in-the-loop" interaction method and adjustable dialogue networks, allowing users to specify the complex distribution of roles and interaction procedures by using natural language. ChatDev realistically imitates the software development process by establishing a complete virtual corporate organization consisting of the CEO, CTO and programmers, and enables the specific Language Large Models (LLMs) to carry out the whole procedure from requirement analysis to code production in a sequential way according to the predetermined Standard Operating Procedures (SOPs) [12].

2.4. Comparison of mainstream methods

As presented in Table 1, this table evaluates the individual reward fine-tuning, joint reward fine-tuning and inference-stage interaction in terms of five important aspects, indicating that joint reward

DOI: 10.54254/2755-2721/2026.36623

fine-tuning obtains the optimal balance between high collaborative ability and inference efficiency without needing much reward design.

Table 1. Comparison of three types of methods

Comparison Dimension	Individual Reward Fine-Tuning	Joint Reward Fine-Tuning	Inference-Stage Interaction
Model Training Required	Yes	Yes	No
Reward Design Cost	High	Low	N/A
Collaboration Depth	Medium	High	Low
Convergence Guarantee	None	Partial	N/A
Inference Efficiency	Medium	High	Low

3. Datasets and evaluation metrics

3.1. Commonly used datasets

In collaborative code generation evaluation, HumanEval is the first widely used standard, including 164 manually written programming problems. But these problems are only suitable for single agent design, so they are not appropriate for direct collaborative evaluation [13]. Therefore, researchers have developed CoopHumanEval (CHE) which chooses problems having "collaboration potential" from HumanEval and adds some new ones, forming a collaborative code generation benchmark consisting of 82 problems. For evaluating the collaborative ability of multi-agents in text generation, the TLDR summarization dataset (obtained from Reddit posts) and the arXiv paper expansion dataset are applied in the writing collaboration tasks [10].

On the contrary, Collab-Overcooked overcomes the restrictions of fixed code generation by offering more than 30 open tasks in the Overcooked-AI interactive game environment, assessing the collaboration abilities of LLMs from a procedural viewpoint [14]. MultiAgentBench investigates the cooperation and confrontation performance of multi-agent systems using different coordination patterns like star, chain and tree structures [15]. A comprehensive study by Zeng et al. on 142 papers about code generation and 156 benchmarks shows that the main problems in this area are the dataset bias and the lack of coverage for real-world situations.

3.2. Evaluation metrics

Evaluation criteria in this field are divided into three aspects. For the capability of task execution, code jobs commonly employ pass@k (the ratio of passed tests in k generations), the grammatical correctness and the test success rate; however, for the structure of tasks, the principal considerations are the structural completeness, the consistency of the style and the logical coherence. Regarding the cooperative ability, the important indices include the cooperation reward values (indicating the interaction between agents), the communication efficiency (the information utilization rate in multiple rounds of interactions) and the decline ratio of the policies under new conditions (a reflection of their generalization ability). As for the efficiency indicators, the main characteristics are

DOI: 10.54254/2755-2721/2026.36623

the inference speed (tokens/second), the number of rounds required for the training to finish and the sample efficiency (the quantity of data required to obtain the intended result).

4. Current state-of-the-art in this direction

Taking the code generation collaboration tasks as an example, the results of the mainstream methods on HumanEval/CHE are presented in Table 2 (Data sources: [10, 13]). As shown in Table 2, this table indicates the performance of the mainstream methods on the HumanEval/CHE benchmark, indicating that MAGRPO (Dual 3B) performs much better than other methods with the highest test pass rate (71.2%) and collaboration reward (83.7%), at the same time keeping a faster inference speed.

Table 2. Typical performance of mainstream methods on HumanEval/CHE

Method	Test Pass Rate (%)	Collaboration Reward (%)	Inference Speed (tokens/s)
Single Model (7B)	64.8	—	73.1
Non-collaborative Parallel (Dual 3B)	40.1	24.0	189.4
Sequential Communication (Dual 3B)	55.2	35.2	97.4
Single-Round Discussion (Dual 3B)	41.9	34.8	78.3
MAGRPO (Dual 3B)	71.2	83.7	192.4

Existing experimental results show three important key findings which clearly indicate the real effectiveness of various optimization methods in multi-agent cooperation.

First, the collaboration strategies learned by reinforcement learning perform much better than the untrained collaboration methods which depend entirely on prompt engineering. Experimental results indicate that the MAGRPO method obtains a test set pass rate of 71.2%, which is substantially higher than the 55.2% of the best untrained baseline. The considerable difference in performance suggests that using fixed prompts to control multiple LLM agents is usually not enough to meet the changing interactive needs in complicated tasks. On the other hand, training through reinforcement learning allows the agents to explore and learn more effective collaboration strategies continuously by interacting with the environment, thus showing stronger generalization ability and adaptability when dealing with unfamiliar task situations.

Furthermore, the training techniques which employ joint rewards show a higher efficiency in collaborative work compared with those using individual rewards. This result shows that it is important to have a common objective for avoiding contradiction of policies in cooperative activities involving several participants. Although individual reward systems attempt to adjust the optimization objectives for each agent, contradictions often occur in their goals when these agents work in reality, leading to one agent sacrificing the overall benefit of the team for its own local advantage and ultimately resulting in the failure of cooperation. The joint reward mechanism makes all the agents optimize in a unified way by communicating a common return signal, which effectively prevents the conflict of goals and motivates the agents to cooperate in an advantageous manner, thus achieving a better team collaboration.

DOI: 10.54254/2755-2721/2026.36623

5. Discussion on problems and solutions

Although great advances have been made, there still exist three important points to be considered in this field. Concerning the doubtful generalization resulting from the bias in data set selection, the CHE data is obtained by human selection of problems with cooperative capability from HumanEval and the detailed selection criteria have not been disclosed. Thus, this method only verifies the effectiveness of the algorithm on a selected subset, which is insufficient to show its generalization ability on the original HumanEval or in practical situations. Studies on EvoEval indicate that the performance of the models scoring highly on HumanEval decreases by an average of 39.4% on its evolutionary versions, which reveals the weakness of the manual benchmarks. In order to solve this problem, the academic world may adopt the method of dynamic benchmark construction, which generates various collaborative task variations automatically by evolutionary mutation driven by the Large Language Model (LLM). This approach avoids the subjective judgment of human on the cooperation potential of a problem, so as to test the generalization ability of the models systematically.

In regards to the applicability in heterogeneous situations, current studies mainly focus on the verification of homogeneous small models. For instance, all the experiments in MAGRPO were based on two same basic models (such as Qwen3-1.7B or Qwen2.5-Coder-3B). But in practical application scenarios, the models used by different groups differ naturally. Hetero-Designer has clearly indicated that the homogeneous assumption seriously restricts the intelligence limit of multi-agent systems, and the MLC framework has shown that guiding various large language models (LLMs) to develop complementary strengths can greatly improve the cooperative efficiency. To deal with this issue, future research may borrow the method of automatic role allocation and differentiated learning strategies to extend the joint reward methods to the heterogeneous model combinations which include different scales or different basic models, thus determining their performance ranges and suitable conditions in heterogeneous environments.

The variance in Monte Carlo estimation affects the sample efficiency. To save memory, MAGRPO eliminates the value network (Critic) and employs the intra-group average as a baseline to estimate the advantage values; the authors point out that this approach has a high estimation variance. When the tasks change from single-round to multi-round, the variance problem becomes more serious, leading to a slight increase of the test pass rate of multi-round MAGRPO from 71.2% in single-round to 75.0%, with only little improvement. Therefore, a practical way to solve this technical problem is to incorporate a lightweight shared Critic network which can reduce the variance while keeping the benefits of CTDE. Moreover, it may be beneficial to use Generalized Advantage Estimation (GAE) to achieve a better balance between bias and variance, thus enhancing the overall sample efficiency.

6. Conclusion

This paper comprehensively examines the development of RL-based multi-agent collaboration optimization in LLMs. Motivated by the need to overcome the limitation of single LLMs and to achieve autonomous cooperation learning, this field has developed three technical approaches: individual reward adjustment, joint reward centralized training and interaction at inference stage. Among these, the joint reward method (such as MAGRPO) shows the best performance in both collaboration efficiency and inference speed, currently obtaining a test pass rate of 71.2% and an inference rate of 192.4 tokens/s on the CHE data set. Nevertheless, this area still has some obvious shortcomings in data generalization, heterogeneous model compatibility and sample efficiency. In

DOI: 10.54254/2755-2721/2026.36623

the future, more research can be focused on constructing dynamic benchmarks, collaborative mechanisms for heterogeneous models and introducing lightweight critics, which will help to extend the practical applications of Multi-Agent Reinforcement Learning (MARL) in the collaboration of LLMs.

References

[1] Achiam, J., Adler, S., Agarwal, S., et al. (2023). GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.
[2] Tran, N., Nguyen, T., Le, T., et al. (2025). A survey on LLM-empowered agentic systems. *arXiv preprint arXiv:2502.14674*.
[3] Du, Y., Li, S., Torralba, A., et al. (2024). Improving factuality and reasoning in language models through multiagent debate. In *International Conference on Machine Learning (ICML)* (pp. 11982-12017). PMLR.
[4] Wu, Q., Bansal, G., Zhang, J., et al. (2024). AutoGen: Enabling next-gen LLM applications via multi-agent conversation. In *Conference on Language Modeling (COLM)*.
[5] Hong, S., Zhuge, M., Chen, J., et al. (2024). MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations (ICLR)*.
[6] Ouyang, L., Wu, J., Jiang, X., et al. (2022). Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS), 35*, 27730-27744.
[7] Slumbers, O., Mguni, D. H., Shao, K., et al. (2024). Leveraging large language models for optimised coordination in textual multi-agent reinforcement learning. arXiv preprint arXiv:2405.13904.
[8] Subramaniam, V., Du, Y., Tenenbaum, J. B., et al. (2025). Multiagent finetuning: Self improvement with diverse reasoning chains. arXiv preprint arXiv:2501.05707.
[9] Tan, M. (1993). Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the 10th International Conference on Machine Learning (ICML)* (pp. 330-337).
[10] Liu, S., Chen, T., Liang, Z., Lyu, X., Amato, C. (2025). LLM collaboration with multi-agent reinforcement learning. arXiv preprint arXiv:2508.04652.
[11] Yu, C., Velu, A., Vinitsky, E., et al. (2022). The surprising effectiveness of PPO in cooperative multi-agent games. In *Advances in Neural Information Processing Systems (NeurIPS), 35*, 24611-24624.
[12] Qian, C., Liu, W., Liu, H., et al. (2024). ChatDev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 15174-15186).
[13] Chen, M., Tworek, J., Jun, H., et al. (2021). Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374.
[14] Sun, H., Zhang, S., Niu, L., et al. (2025). Collab-Overcooked: Benchmarking and evaluating large language models as collaborative agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
[15] Zhu, K., Du, H., Hong, Z., et al. (2025). MultiAgentBench: Evaluating the collaboration and competition of LLM agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*.