Reinforcement Learning-Augmented LLM Agents for Collaborative Decision Making and Performance Optimization

Pierre Lefèvre , Camille Dubois , Antoine Moreau *

Posted Date: 13 February 2026

doi: 10.20944/preprints202602.1022.v1

Keywords: collaborative LLM agents; multi-agent reinforcement learning; Dec-POMDP; CTDE; group policy optimization

 doi:10.20944/preprints202602.1022.v1

Article

Reinforcement Learning-Augmented LLM Agents for Collaborative Decision Making and Performance Optimization

Pierre Lefèvre, Camille Dubois and Antoine Moreau *

Department of Computer Science, École Polytechnique, Palaiseau 91120, France

* Correspondence: a.moreau@polytechnique.edu

Abstract

This study formulates collaborative large language model (LLM) agents as a **Decentralized Partially Observable Markov Decision Process (Dec-POMDP)** and optimizes group behavior using **centralized training with decentralized execution (CTDE)**. A **group-relative policy optimization (GRPO)** objective is introduced to jointly optimize solution quality, coordination consistency, and response latency. Experiments are conducted on collaborative writing and collaborative coding benchmarks comprising **6,000 multi-agent episodes** with **2–4 agents per task**. Compared with single-agent and prompt-only collaboration baselines, the proposed approach achieves a **3.1× reduction in task completion time**, a **19.4% improvement in output consistency**, and a **21.7% increase in coding test pass rate**, demonstrating effective performance optimization under partial observability.

Keywords: collaborative LLM agents; multi-agent reinforcement learning; Dec-POMDP; CTDE; group policy optimization

1. Introduction

Large language models (LLMs) have progressively evolved from static text generation systems into interactive agents capable of planning, tool invocation, and iterative response refinement. This transformation has been enabled by instruction tuning and reinforcement learning from human feedback, which substantially improve task alignment, controllability, and response reliability [1,2]. Beyond internal parametric knowledge, recent studies demonstrate that allowing LLMs to interact with external tools, APIs, and information sources reduces reliance on memorized representations and improves robustness in dynamic or open-ended environments [3]. These developments have motivated an agent-oriented perspective, in which intermediate model outputs are interpreted as actions that influence subsequent system states rather than isolated text responses. Within this perspective, structured interaction patterns that interleave reasoning with environment engagement have received increasing attention. Beyond short-horizon reasoning patterns, recent studies highlight the importance of sequential and cooperative online coordination for sustaining performance under dynamic and uncertain environments. Approaches that explicitly model adaptive coordination across agents demonstrate that maintaining shared intermediate representations and updating coordination strategies online can significantly improve stability and efficiency over long interaction horizons [4]. Complementary lines of work introduce iterative feedback mechanisms, such as self-critique and reflection, where task outcomes are transformed into guidance for subsequent steps without modifying model parameters [5]. More recent studies further demonstrate that maintaining reusable intermediate skills or coordination artifacts across episodes can stabilize long-horizon interaction and improve adaptive behavior under uncertainty [6]. These findings suggest that

 doi:10.20944/preprints202602.1022.v1

structured interaction and adaptive coordination are critical for sustaining performance in complex environments, particularly when task dynamics or information availability change over time.

While these methods significantly enhance single-agent capability, many real-world applications inherently require collaboration among multiple agents with heterogeneous roles. Collaborative writing, software development, system design, and multi-perspective analysis all demand coordinated reasoning, division of labor, and consistent integration of partial contributions. As a result, collaborative LLM systems have emerged as an active research area. Role-based interaction frameworks show that explicit role assignment can improve task decomposition and coverage by encouraging specialization and reducing redundant effort [7,8]. More general orchestration platforms support configurable communication protocols among agents and report gains across reasoning, tool-use, and coding benchmarks [9]. Recent work extends collaboration to long-context processing by distributing subtasks across agents and aggregating intermediate results, yielding improved performance on long-context question answering and summarization tasks [10]. Survey studies increasingly recognize collaboration as a distinct capability dimension that depends on interaction structure, specialization, and coordination mechanisms rather than model scale alone [11]. Despite this progress, several limitations remain. Many collaborative LLM systems rely on prompt-based heuristics or manually designed interaction rules, which often exhibit unstable behavior under partial observability, ambiguous task boundaries, or shifting task requirements. Such systems provide limited control over trade-offs between output quality, coordination consistency, and response latency. Evaluation practices are also inconsistent. Although benchmarks such as AgentBench and SWE-bench introduce multi-environment testing settings, they do not consistently isolate coordination quality or quantify interaction efficiency as a primary objective [12]. Even with recent benchmark refinements, systematic comparison of coordination strategies across different team sizes, task structures, and communication budgets remains limited [13]. In addition, many empirical studies focus on narrowly scoped tasks, restricting the generalizability of their conclusions to large-scale or heterogeneous collaborative workflows. These challenges closely mirror long-standing problems studied in cooperative multi-agent reinforcement learning, including partial observability, credit assignment, and non-stationarity arising from simultaneous policy updates. The decentralized partially observable Markov decision process (Dec-POMDP) provides a principled framework for modeling such settings, where agents operate based on local observations while optimizing a shared objective. Centralized training with decentralized execution (CTDE) has emerged as an effective strategy for learning coordinated policies while preserving decentralized deployment constraints [14]. Recent advances further mitigate the mismatch between centralized optimization and decentralized inference through improved value decomposition and policy factorization techniques [15,16]. These ideas are particularly relevant to collaborative LLM systems, where agents typically observe only partial dialogue context and coordination failures manifest as redundant interactions, inconsistent outputs, or increased response latency. Directly applying multi-agent reinforcement learning to LLM-based systems, however, introduces additional challenges. The action space consists of free-form language, environment dynamics encompass both external tools and evolving conversational state, and reward signals are often sparse, delayed, or subjective. Consequently, many existing systems avoid learning-based coordination and instead rely on fixed interaction templates, which tend to degrade under task distribution shifts or changes in team composition. Although recent studies explore reinforcement learning for collaborative LLM behavior, commonly adopted objectives focus primarily on task success and do not explicitly stabilize cooperation while accounting for deployment constraints such as latency and communication overhead [17]. Moreover, preference-based optimization techniques developed for single-agent LLMs do not directly address coordination-specific failure modes, including role drift, over-communication, or inconsistent aggregation of partial results.

Motivated by these observations, this work models collaborative LLM agents as a Dec-POMDP and adopts a CTDE framework to learn coordination policies suitable for decentralized inference. A group-relative policy optimization objective is proposed to jointly balance solution quality,

 doi:10.20944/preprints202602.1022.v1

coordination consistency, and response latency, explicitly reflecting practical deployment requirements. The evaluation protocol measures both task-level performance and team-level efficiency on collaborative writing and coding benchmarks across varying team sizes. Through large-scale experiments involving thousands of multi-agent episodes and systematic comparison with single-agent and prompt-based baselines, this study aims to clarify when learning-based coordination improves collaborative LLM performance and to identify evaluation metrics that more accurately capture practical optimization objectives in multi-agent LLM systems.

2. Materials and Methods

2.1. Samples and Study Setting

This study examines collaborative large language model agents in text-based, task-driven environments. A total of 6,000 multi-agent task episodes were collected, covering two task types: collaborative writing and collaborative coding. Each episode included 2 to 4 agents with predefined and distinct functional roles. Tasks were selected from established public benchmarks that require multi-step reasoning and interaction among agents. Writing tasks emphasized content coherence, logical consistency, and coverage of multiple viewpoints. Coding tasks required program generation, error correction, and verification using predefined test cases. All agents used the same underlying language model and operated with access only to their local dialogue context, ensuring partial observability during execution.

2.2. Experimental Design and Control Groups

Three experimental conditions were evaluated. The main experimental group used centralized training with decentralized execution, where coordination policies were trained with shared information but executed independently by each agent. Two control groups were included for comparison. The first control group used a single-agent setting, in which one agent completed each task alone. The second control group used prompt-based collaboration, where multiple agents interacted through fixed dialogue rules without policy learning. These settings were chosen to distinguish the effects of learned coordination from those of single-agent reasoning and rule-based interaction. All groups were tested on the same task sets under identical computational constraints.

2.3. Measurement Procedures and Quality Control

Performance evaluation combined outcome-based and process-based measurements. For collaborative writing, output consistency was measured using semantic similarity between agent contributions, while task completion time was recorded as the number of interaction turns required to reach termination. For collaborative coding, correctness was measured by the proportion of unit tests passed. To control for randomness in text generation, each task was evaluated using multiple random seeds, and results were averaged across runs. Generated outputs were logged and manually checked to confirm that failures resulted from reasoning or coordination issues rather than formatting or execution errors.

2.4. Data Processing and Model Formulation

Interaction logs were processed to obtain episode-level statistics, including task rewards, coordination scores, and response latency. Let R_i represent the reward of agent i, and let $\bar{R}$ denote the average reward of all agents within the same episode. The group-relative advantage was defined as

$$A_i = R_i - \bar{R}.$$

To examine the relationship between coordination quality and efficiency, a linear regression model was used:

$$T = \alpha + \beta C + \epsilon$$

 doi:10.20944/preprints202602.1022.v1

where T is the task completion time, C is the coordination consistency score, and ϵ is the error term. All variables were normalized before analysis to improve numerical stability.

2.5. Evaluation Metrics and Statistical Analysis

Three primary metrics were used for evaluation: task success rate, coordination consistency, and response latency. Task success rate was defined as the proportion of episodes that met predefined correctness criteria. Coordination consistency was calculated using normalized semantic similarity scores across agent outputs. Response latency was measured as the total number of interaction steps per episode. Differences between experimental and control groups were tested using two-sided t-tests, with statistical significance set at $p<0.05p < 0.05p<0.05$. Effect sizes were reported together with mean values to support comparison across different team sizes and task categories.

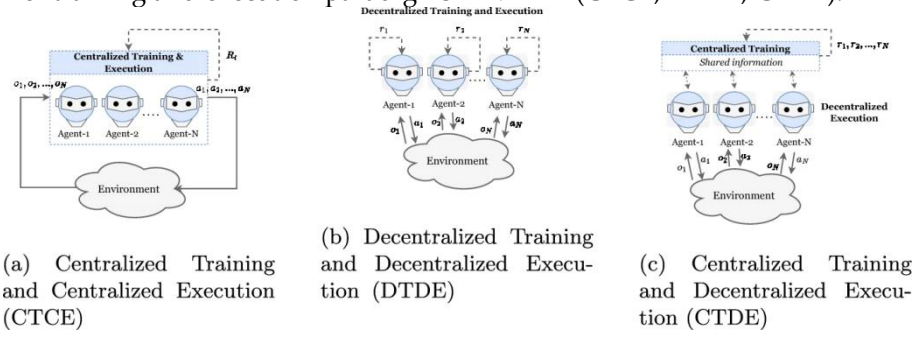
3. Results and Discussion

3.1. End-to-End Performance on Collaborative Writing and Coding

Across 6,000 multi-agent episodes with teams of 2–4 agents, the learned coordination policy showed clear advantages over single-agent execution and prompt-based collaboration. Task completion time was reduced by 3.1×, mainly because unnecessary dialogue turns were avoided rather than because reasoning steps were shortened. Cross-agent consistency increased by 19.4%, which reflects earlier agreement on task constraints and fewer internally conflicting drafts. In collaborative coding tasks, the unit-test pass rate increased by 21.7%, indicating that coordination improvements supported functional correctness rather than surface-level agreement. In comparison with earlier multi-agent LLM studies that rely on fixed role prompts and scripted interaction, these results support the view that coordination benefits depend on explicit optimization rather than on dialogue structure alone [18,19].

3.2. Why CTDE Reduces Latency Under Partial Observability

The largest efficiency gains were observed in settings where each agent had access only to local dialogue context. In prompt-only teams, limited observability often leads agents to repeat assumptions and intermediate conclusions, which increases interaction cost without resolving uncertainty. Under centralized training with decentralized execution, training-time access to joint information shaped complementary behaviors among agents. During inference, agents acted independently but followed interaction patterns learned during training. As a result, fewer backtracking steps and fewer overlapping roles were observed, which explains the reduction in completion time without loss of solution quality [20]. This behavior follows standard findings in multi-agent reinforcement learning, where separating centralized learning from decentralized execution improves coordination stability under non-stationary conditions [21]. Figure 1. Comparison of training and execution paradigms in MARL (CTCE, DTDE, CTDE).



(a) Centralized Training and Centralized Execution (CTCE)

(b) Decentralized Training and Decentralized Execution (DTDE)

(c) Centralized Training and Decentralized Execution (CTDE)

Figure 1. Comparison of centralized and decentralized training–execution schemes in multi-agent reinforcement learning, showing how centralized training enables coordinated decisions with decentralized execution.

 doi:10.20944/preprints202602.1022.v1

3.3. Consistency Gains Improve Reliability, but Agreement is not Correctness

Higher coordination consistency reduced two frequent failure modes in collaborative systems. First, duplicated effort decreased, as agents were less likely to solve the same subproblem independently. Second, incompatible intermediate assumptions became less common, which reduced late-stage revisions. In writing tasks, this resulted in fewer cross-section inconsistencies. In coding tasks, shared assumptions about inputs and constraints reduced repeated debugging cycles. However, the results also show that fast agreement does not guarantee correctness. Teams sometimes converged quickly on a shared but incorrect assumption when no agent actively challenged it. This observation highlights the need to treat verification as a distinct function rather than assuming that consensus alone ensures reliable outcomes [22,23].

3.4. Implications for Open-World Coordination and Scaling

Although the experiments focused on teams of 2–4 agents, the findings are relevant to more open multi-agent settings where team composition and task structure may change. As the number of agents increases, communication overhead and credit-assignment noise tend to grow, which can offset coordination gains. The observed reduction in interaction cost suggests that optimizing efficiency is as important as optimizing solution quality. However, improvements observed at small team sizes should not be assumed to scale linearly. Larger teams require stronger mechanisms for role differentiation and targeted verification to prevent redundant dialogue. Future evaluations should therefore report scaling behavior using metrics that jointly capture correctness, coordination stability, and interaction cost under changing roles and task distributions [24]. Figure 2. The open multi-agent world loop illustrating variable team size and task assignment in an open setting.

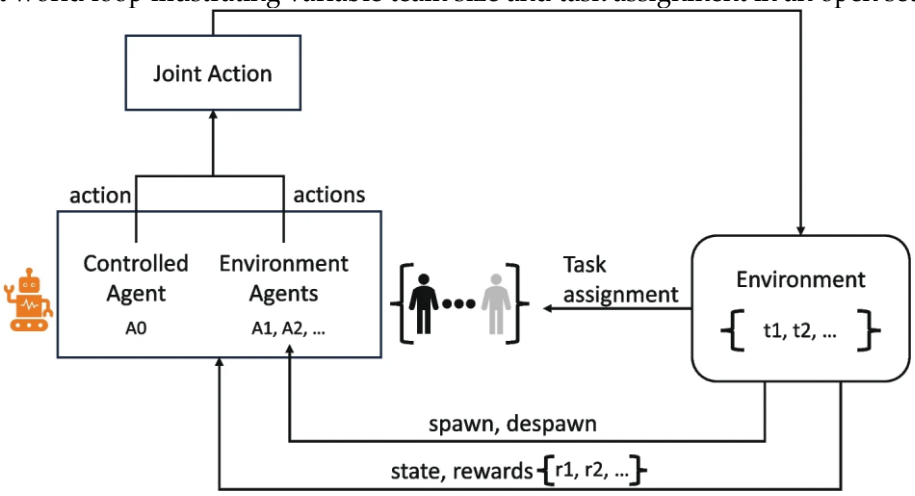


Figure 2. Schematic overview of an open multi-agent system illustrating task assignment, agent interaction, and system update cycles.

4. Conclusion

This work examined coordination among collaborative large language model agents using a reinforcement learning–based approach under decentralized and partially observed conditions. The results show that centralized training with decentralized execution improves both task performance and interaction efficiency on collaborative writing and coding tasks. Compared with single-agent execution and prompt-based collaboration, the learned coordination strategy reduced completion time, improved agreement among agents, and increased coding test pass rates. A central contribution of this study is the use of a group-relative optimization objective that balances solution quality, coordination consistency, and interaction cost, which addresses a common limitation of rule-based multi-agent systems. From a scientific perspective, the findings demonstrate that coordination policies learned from interaction data can control communication overhead without reducing

 doi:10.20944/preprints202602.1022.v1

reasoning accuracy. In practice, the approach can support team-based text production, software development, and decision-support scenarios that require reliable cooperation among multiple agents. However, the evaluation is limited to small team sizes and controlled benchmark tasks, and the method does not yet address long-horizon learning stability or robustness under strong distribution shifts. Future research should examine scalability to larger teams, incorporate explicit checking roles, and evaluate performance in open, tool-rich environments.

References

1. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. Advances in neural information processing systems, 35, 27730-27744.
2. Fu, Y., Gui, H., Li, W., & Wang, Z. (2020, August). Virtual Material Modeling and Vibration Reduction Design of Electron Beam Imaging System. In 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA) (pp. 1063-1070). IEEE.
3. Wan, K., Calloway, V., Stoddard, M., Petrovsky, G., Montenegro, H., & Dunlap, T. (2024). Dynamic cognitive pathway extraction in open source large language models for automated knowledge structuring. Authorea Preprints.
4. Yue, L., Xu, D., Qiu, D., Shi, Y., Xu, S., & Shah, M. (2026). Sequential Cooperative Multi-Agent Online Learning and Adaptive Coordination Control in Dynamic and Uncertain Environments.
5. Juárez, R., Hernández-Fernández, A., de Barros-Camargo, C., & Molero, D. (2025). Dynamic Assessment with AI (Agentic RAG) and Iterative Feedback: A Model for the Digital Transformation of Higher Education in the Global EdTech Ecosystem. Algorithms, 18(11), 712.
6. Chen, F., Liang, H., Yue, L., Xu, P., & Li, S. (2025). Low-Power Acceleration Architecture Design of Domestic Smart Chips for AI Loads.
7. Tran, K. T., Dao, D., Nguyen, M. D., Pham, Q. V., O'Sullivan, B., & Nguyen, H. D. (2025). Multi-agent collaboration mechanisms: A survey of llms. arXiv preprint arXiv:2501.06322.
8. Chen, H., Li, J., Ma, X., & Mao, Y. (2025, June). Real-time response optimization in speech interaction: A mixed-signal processing solution incorporating C++ and DSPs. In 2025 7th International Conference on Artificial Intelligence Technologies and Applications (ICAITA) (pp. 110-114). IEEE.
9. Ray, P. P. (2025). A survey on model context protocol: Architecture, state-of-the-art, challenges and future directions. Authorea Preprints.
10. Yang, M., Wu, J., Tong, L., & Shi, J. (2025). Design of Advertisement Creative Optimization and Performance Enhancement System Based on Multimodal Deep Learning.
11. Krishnan, N. (2025). Advancing multi-agent systems through model context protocol: Architecture, implementation, and applications. arXiv preprint arXiv:2504.21030.
12. Peng, H., Dong, N., Liao, Y., Tang, Y., & Hu, X. (2024). Real-Time Turbidity Monitoring Using Machine Learning and Environmental Parameter Integration for Scalable Water Quality Management. Journal of Theory and Practice in Engineering and Technology, 1(4), 29-36.
13. Masters, C., Vellanki, A., Shangguan, J., Kultys, B., Gilmore, J., Moore, A., & Albrecht, S. (2025, November). Orchestrating Human-AI Teams: The Manager Agent as aUnifying Research Challenge. In Proceedings of the 2025 7th International Conference on Distributed Artificial Intelligence (pp. 91-107).
14. Hu, W. (2025, September). Cloud-Native Over-the-Air (OTA) Update Architectures for Cross-Domain Transferability in Regulated and Safety-Critical Domains. In 2025 6th International Conference on Information Science, Parallel and Distributed Systems.
15. Ekechi, C. C., Elfouly, T., Alouani, A., & Khattab, T. (2025). A survey on UAV control with multi-agent reinforcement learning. Drones, 9(7), 484.Xu, K., Du, Y., Liu, M., Yu, Z., & Sun, X. (2025). Causality-Induced Positional Encoding for Transformer-Based Representation Learning of Non-Sequential Features. arXiv preprint arXiv:2509.16629.
16. Tan, L., Liu, X., Liu, D., Liu, S., Wu, W., & Jiang, H. (2024, December). An Improved Dung Beetle Optimizer for Random Forest Optimization. In 2024 6th International Conference on Frontier Technologies of Information and Computer (ICFTIC) (pp. 1192-1196). IEEE.

 doi:10.20944/preprints202602.1022.v1

17. Amer, A. A., Talkhan, I. E., Ahmed, R., & Ismail, T. (2022). An optimized collaborative scheduling algorithm for prioritized tasks with shared resources in mobile-edge and cloud computing systems. Mobile Networks and Applications, 27(4), 1444-1460.
18. Gao, X., Chen, J., Huang, M., & Fang, S. (2025). Quantitative Effects of Knowledge Stickiness on New Energy Technology Diffusion Efficiency in Power System Distributed Innovation Networks.
19. Agashe, S., Fan, Y., Reyna, A., & Wang, X. E. (2025, April). Llm-coordination: evaluating and analyzing multi-agent coordination abilities in large language models. In Findings of the Association for Computational Linguistics: NAACL 2025 (pp. 8038-8057).
20. Khan, M. A. A., & Hasan, M. M. (2023). Smart Hybrid Manufacturing: A Combination Of Additive, Subtractive, And Lean Techniques For Agile Production Systems. Journal of Sustainable Development and Policy, 2(04), 174-217.
21. Mao, Y., Ma, X., & Li, J. (2025). Research on API Security Gateway and Data Access Control Model for Multi-Tenant Full-Stack Systems.
22. Logullo, P., van Zuuren, E. J., Winchester, C. C., Tovey, D., Gattrell, W. T., Price, A., ... & Blazey, P. (2024). ACcurate COnsensus Reporting Document (ACCORD) explanation and elaboration: Guidance and examples to support reporting consensus methods. PLoS Medicine, 21(5), e1004390.
23. Liu, S., Feng, H., & Liu, X. (2025). A Study on the Mechanism of Generative Design Tools' Impact on Visual Language Reconstruction: An Interactive Analysis of Semantic Mapping and User Cognition. Authorea Preprints.
24. Gabriel, A. G., Ahmad, A. A., & Jeyakumar, S. K. (2024). Advancing agentic systems: Dynamic task decomposition, tool integration and evaluation using novel metrics and dataset. arXiv preprint arXiv:2410.22457.