ISSN: 3079-644X

Adaptive Role Allocation via Hierarchical Reinforcement Learning in Collaborative Agent Systems

Michael Johnson[1], Emily Carter[2], David Thompson[3*]

Department of Computer Science, Stanford University, Stanford, CA 94305, USA

Corresponding author: d.thompson@stanford.edu

Abstract

Collaborative agent systems often suffer from inefficient coordination due to static role assignment in dynamic environments. This study investigates adaptive role allocation using a hierarchical reinforcement learning (HRL) framework, where a high-level controller assigns roles and low-level policies execute task-specific actions. The approach is trained using proximal policy optimization (PPO) with a role-transition regularization term to stabilize switching behavior. Experiments are conducted on a benchmark of 9,200 multi-step decision tasks, including scheduling and distributed planning scenarios. Results show that the proposed method improves task success rate from 73.6% to 86.8% and reduces redundant interactions by 24.5% compared to fixed-role baselines. In addition, convergence speed is accelerated by 19%, indicating more efficient policy learning. The findings suggest that hierarchical role modeling is effective for improving coordination efficiency in complex decision workflows.

Keywords

Hierarchical reinforcement learning; Role allocation; Multi-agent systems; PPO; Task coordination

1. Introduction

Collaborative agent systems have been widely studied in scheduling, distributed planning, and other tasks that require coordination across multiple decision steps. In such settings, fixed role assignment often constrains system performance because it cannot respond effectively to changing task demands. When the environment, workload, or interaction structure shifts during execution, static role designs may leave some agents underused while others repeat overlapping actions, ultimately reducing coordination efficiency and wasting computational or operational resources [1]. Recent advances in multi-agent reinforcement learning (MARL) have improved cooperative decision-making and inter-agent communication, yet important challenges remain, including training instability, large joint action spaces, and weak adaptability in dynamic environments [2,3]. These difficulties become more pronounced in tasks with long decision chains, delayed rewards, and evolving interaction patterns, where effective coordination depends not only on local action quality but also on timely adjustment of agent responsibilities [4,5]. Hierarchical reinforcement learning (HRL) offers a useful framework for handling such long-horizon coordination problems. In this setting, a high-level policy determines abstract decisions such as subgoals, coordination modes, or role assignments, while low-level policies execute concrete actions under those decisions. By decomposing decision-making across levels, HRL can reduce learning complexity, improve credit assignment, and support more structured coordination in complex environments [6,7].

ISSN: 3079-644X

Recent studies have applied hierarchical learning to emergency response, resource allocation, distributed control, and related multi-step decision tasks, showing that a separation between strategic coordination and local execution can improve both learning efficiency and policy robustness [8,9]. This line of research suggests that hierarchical structures are especially valuable when collaborative tasks require agents to adapt their functions over time rather than follow a fixed behavioral template [10]. A closely related research direction concerns role allocation. Instead of assigning permanent roles to agents before learning begins, adaptive role allocation methods allow the role structure itself to evolve with the task state, environmental feedback, and coordination demands [11]. This improves flexibility and can reduce redundant interactions across agents [12,13]. Such ideas have been explored in dynamic environments, traffic coordination, resource systems, scheduling, and distributed decision-making problems, where the division of labor must change as tasks unfold [14,15]. Existing results indicate that role allocation should not be treated as a separate preprocessing step, but as an integral component of the learning process. Even so, several limitations remain. Many existing methods optimize role assignment and action learning in a loosely coupled manner, which weakens coordination between strategic and operational decisions. Role switching is also often unstable, and excessive switching may interrupt behavioral consistency, increase exploration noise, and slow convergence [16]. In addition, empirical validation is frequently confined to relatively small task sets or isolated scenarios, making it difficult to judge whether the reported gains can generalize to more complex collaborative environments [17,18]. These gaps point to the need for a framework that can jointly model role adaptation and action execution while maintaining stable learning dynamics. For collaborative multi-step tasks, the value of adaptive role allocation lies not only in assigning different functions to different agents, but also in determining when role changes are truly beneficial and when they merely introduce unnecessary coordination overhead. A practical method should therefore balance flexibility and stability: it should allow agents to reconfigure their responsibilities when task conditions change, while discouraging frequent or uninformative role transitions that weaken cooperation. From this perspective, hierarchical learning provides a natural basis for adaptive coordination, but its effectiveness depends on how tightly the high-level role policy is linked to low-level execution and how well transition behavior is regularized during training. This study addresses that need by developing a hierarchical reinforcement learning framework for adaptive role allocation in collaborative agent systems. A high-level controller is used to assign agent roles according to the current task state, and low-level policies execute role-conditioned actions. Proximal Policy Optimization (PPO) is adopted for policy learning, and a role-transition regularization term is introduced to reduce unnecessary switching and improve coordination stability during training. The framework is evaluated on 9,200 multi-step tasks spanning scheduling and distributed planning scenarios, with the aim of improving coordination efficiency, reducing redundant interactions, and accelerating policy learning. The significance of this study lies in showing that adaptive role modeling can be integrated into hierarchical decision-making in a stable and scalable manner. By strengthening the connection between role assignment and action execution, the proposed approach contributes to the design of collaborative agent systems that are better suited to complex, changing task environments and offers empirical evidence for more effective coordination mechanisms in long-horizon multi-agent problems.

ISSN: 3079-644X

2. Materials and Methods

2.1. Task Samples and Environment Description

Experiments were carried out in a simulated multi-agent environment with role-dependent coordination. A total of 9,200 tasks were generated. Tasks covered scheduling and distributed planning settings with different levels of dependency and workload. Each task involved 4–6 agents. Agents acted based on local observations and did not access the full system state. The environment included stochastic transitions and changing task conditions. Task settings varied in resource levels, dependency depth, and interaction frequency. The dataset was divided into training, validation, and test sets at a ratio of 70%, 15%, and 15%. Task parameters changed across episodes to avoid fixed patterns.

2.2. Experimental Design and Baseline Setting

Two settings were compared: a fixed-role baseline and a hierarchical role allocation model. In the baseline, each agent kept a fixed role throughout the task. In the hierarchical model, a high-level policy selected roles, and a low-level policy selected actions. Both settings used the same network structure and training configuration. PPO was used in both cases. This ensured that differences in performance came from role allocation. Each experiment was repeated five times with different random seeds. Results were averaged across runs. The comparison focused on success rate, redundant interactions, and convergence speed.

2.3. Measurement Methods and Quality Control

Three metrics were used: task success rate, redundant interaction count, and convergence speed. Task success rate was defined as the ratio of tasks completed within a fixed step limit. Redundant interactions referred to repeated actions that did not improve task progress. Convergence speed was measured by the number of episodes needed to reach stable performance. Results were averaged over repeated runs, and standard deviations were reported. All experiments used the same hardware and software setup. Hyperparameters were kept constant across settings. Early stopping based on validation results was applied to limit overfitting.

2.4. Data Processing and Model Formulation

Training data consisted of state–role–action–reward trajectories collected during interaction. Observations were normalized before input. Rewards were scaled to a fixed range to stabilize training. Policy learning followed PPO. The objective function is

$$L^{PPO}(\theta)=E_t\left[\min\left(r_t(\theta)\widehat{A}_t, clip\left(r_t(\theta),1\text{-}\epsilon,1\text{+}\epsilon\right)\widehat{A}_t\right)\right]$$

Where $r_t(\theta)$ is the probability ratio and $\widehat{A}_t$ is the advantage estimate.

To reduce frequent role changes, a penalty term was added:

ISSN: 3079-644X

$$L_{role}=\lambda\cdot E_t[\mathbf{1}(z_t\neq z_{t-1})]$$

Where z_t is the role at step t, and λ controls the penalty strength.

2.5. Training Procedure and Implementation Details

Training was carried out for 1.3 million interaction steps. Trajectories were stored during training, and updates followed PPO with mini-batch sampling. The learning rate was 3×10^{-4}, and the discount factor was 0.99. The clipping parameter was set to 0.2. The role policy and action policy were updated in an alternating manner. Parallel environments were used to improve sampling efficiency. The model was implemented in PyTorch and trained on a GPU. After training, the policy was tested on unseen tasks with different dependency levels to evaluate generalization performance.

3. Results and Discussion

3.1. Overall Coordination Performance

The hierarchical role allocation model performed better than the fixed-role baseline on the full test set. The improvement appeared in both scheduling and distributed planning tasks. It was more obvious in tasks with deeper dependency and uneven workload. The success rate increased from 73.6% to 86.8%. This result shows that adaptive role assignment helped agents adjust to changing task demands more effectively. As shown in **Fig. 1**, the proposed method kept a clear advantage as task difficulty increased, while the fixed-role baseline showed a larger drop at later stages. Similar trends have been reported in recent studies, where hierarchical control improved coordination by separating global decisions from local actions [19,20].

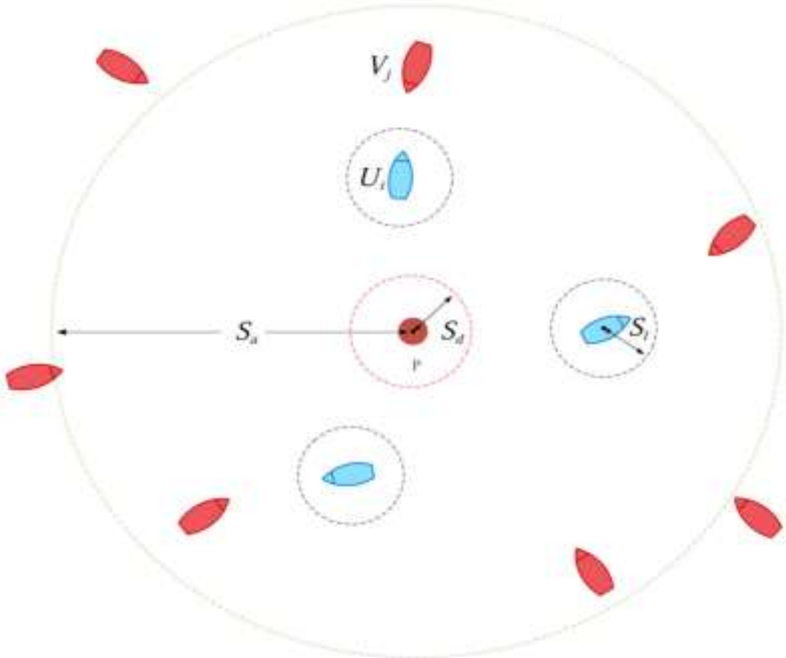

Figure 1 Task success rates for hierarchical and fixed-role methods across task levels.

ISSN: 3079-644X

3.2. Reduction in Redundant Interactions

The proposed method also reduced redundant interactions. Under the fixed-role setting, agents often repeated actions or produced unnecessary exchanges when the assigned roles no longer matched the current task state. The hierarchical model reduced redundant interactions by 24.5%. This reduction is important because repeated coordination increases system cost and slows task completion. The role-transition penalty appears to be useful in this process. By limiting frequent switching, it reduced unstable reassignment and kept local behavior more consistent. Earlier MARL studies have also noted that coordination quality depends not only on action selection, but also on whether agent responsibilities remain stable during execution [21,22].

3.3. Convergence Behavior and Training Stability

Training efficiency improved under the hierarchical setting. The proposed method reached stable performance 19% faster than the fixed-role baseline. This result suggests that adaptive role assignment reduced the difficulty of flat policy learning. Instead of learning one broad policy for all situations, the model learned role-specific behaviors under guidance from the high-level controller. As shown in **Fig. 2**, the learning curve of the proposed method increased more smoothly and entered a stable stage earlier than the baseline. This result is in line with recent HRL studies, where hierarchical structure improved sample use and reduced instability in tasks with complex interaction patterns [23].

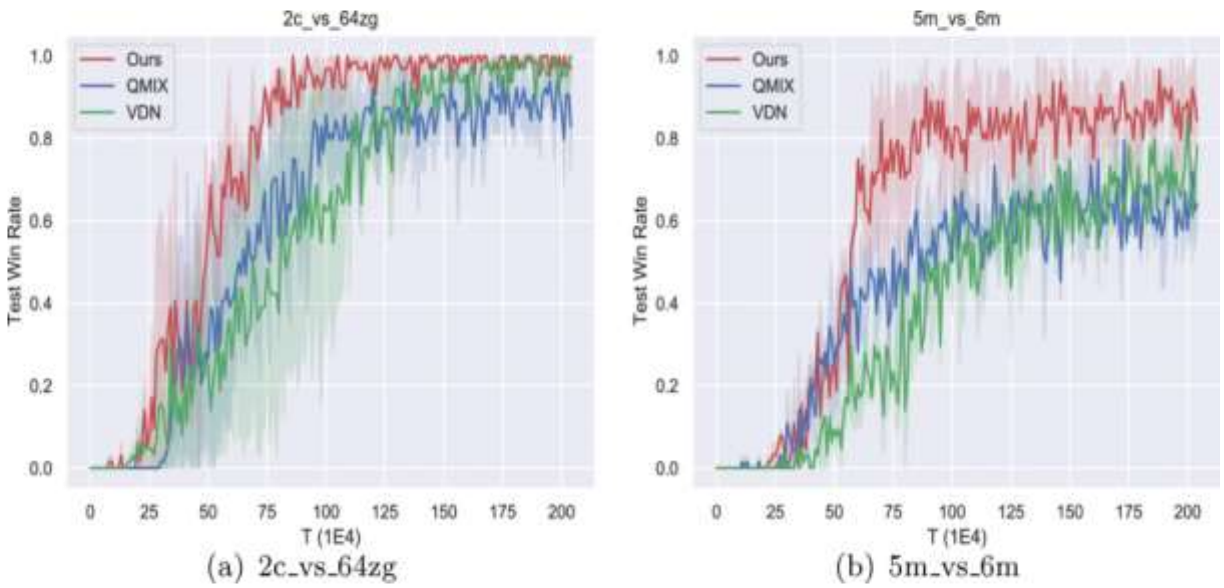

Figure 2 Learning curves for hierarchical and fixed-role training.

3.4. Comparison with Existing Studies and Practical Implications

Compared with earlier studies, this framework places more focus on the link between role selection and action execution during learning. Many existing methods focus on communication design, graph-based coordination, or task-specific control. Fewer studies examine how role reassignment itself should be learned and stabilized in one framework. The

ISSN: 3079-644X

present results suggest that explicit role modeling is useful in multi-step tasks with uneven workload and changing team demands. However, some limits remain. The experiments were conducted in a simulated environment, and the team size was still moderate. The switching penalty also used a fixed coefficient. Future work may consider larger teams, richer communication settings, and adaptive penalties for role change. Even with these limits, the results support hierarchical role modeling as a practical way to improve coordination efficiency in collaborative decision tasks [24].

4. Conclusion

This study examined adaptive role allocation in collaborative agent systems using a hierarchical reinforcement learning framework. A high-level policy selected roles, and low-level policies executed actions. This setting improved coordination in multi-step tasks with changing demands. Compared with fixed-role methods, the approach achieved a higher success rate, reduced redundant interactions, and reached stable performance in fewer training steps. These results show that learning role structure together with action policies improves coordination efficiency and reduces unnecessary communication. The role-transition penalty helped limit frequent switching and kept behavior more stable during execution. The method can be applied to tasks such as scheduling and distributed planning, where workload and dependencies change over time. However, some limitations remain. The experiments were conducted in a simulated environment, and the number of agents was limited. The switching penalty used a fixed coefficient. Future work may consider larger systems, richer communication settings, and adaptive role transition methods to improve stability and scalability.

References

[1] Xu, D., Liu, H., Qiu, D., & Ma, Q. (2026). Structured Modeling and Representation Methods for Post-Retrieval Inference Processes in Large Video Language Models.

[2] Hady, M. A., Hu, S., Pratama, M., Cao, Z., & Kowalczyk, R. (2025). Multi-agent reinforcement learning for resources allocation optimization: a survey. Artificial Intelligence Review, 58(11), 354.

[3] Zhang, Y., & Wang, J. (2026). Design and Implementation of a Computer-Aided Full Lifecycle Quality Management System for Wind Farms in Upgrades, Renovations, and Subcontractor Supervision.

[4] Mahanta, U. (2025). Beyond Automation: Guidelines for a Human-Centered Multi-Agent System for Coordinated Decision Making under Industrial Settings.

[5] Jiao, Y., Zhao, B., Wang, A., & Shi, T. (2026). Construction and Empirical Study of a Modularized Teaching System for Art Courses Based on a Unified Training Pathway.

[6] Ramakrishnan, D., & Radhakrishnan, K. (2025). Hierarchical Brain-Inspired Deep Learning for Autonomous Decision-Making in Complex Dynamic Environments. In Artificial Intelligence and Applications.

ISSN: 3079-644X

[7] Xu, D., Gui, H., & Chen, H. (2026). Research on Layered Control and Fault Recovery Mechanisms for Fast Charging Safety Diagnosis of High Voltage Battery Systems Under Charging Network Interoperability Conditions.

[8] Gao, T., Goins, D., Ballotti, C., Liu, J., & Qu, C. (2025). Learning-Based UAV Swarm Video Analytics Orchestration in Disaster Response Management. SN Computer Science, 6(5), 537.

[9] Wang, Y., Chen, J., Wang, Y., & Yin, X. (2026). Application of Obtainable Biological Agent Characteristics in Efficacy Stratification of Oral Anti-Obesity Drugs.

[10] Crowley, J. L., Coutaz, J., Grosinger, J., Vazquez-Salceda, J., Angulo, C., Sanfeliu, A., ... & Cohn, A. G. (2022). A hierarchical framework for collaborative artificial intelligence. IEEE pervasive computing, 22(1), 9-18.

[11] Zhang, Y., Gu, W., & Wang, J. (2026). Construction of Wind Farm Asset Health Index Based on Multi-Dimensional Indicators and Analytic Hierarchy Process and Its Correlation with Operational Performance. Authorea Preprints.

[12] Qasim, A., Ghouri, A., & Munawar, A. (2024). An effective approach for reducing data redundancy in multi-agent system communication. Multiagent and Grid Systems, 20(1), 69-88.

[13] Liu, S., & Yim, J. (2025). Research on Generative AI Creation Systems Based on Visual Language Modeling: Human-Machine Collaboration and Cognitive Feedback Mechanisms. Available at SSRN 6139770.

[14] Szabo, C., Baker, R., Pearce, G., Teffera, E., & Perry, A. (2025). Overview and challenges of distributed decision making in resource contested and dynamic environments. ACM Computing Surveys, 57(9), 1-34.

[15] Gao, G., Gao, R., Gao, R., & Zhou, H. (2026). Engineering Analysis and Quantitative Research on the Platform-Based Evolution of Enterprise Communication Systems.

[16] Rahman, R., & Nguyen, D. C. (2025). Escaping Barren Plateaus in Variational Quantum Algorithms Using Negative Learning Rate in Quantum Internet of Things. IEEE Internet of Things Journal.

[17] Liu, H., Xu, D., Ma, Q., Xu, S., & Qiu, D. (2026). Memory Poisoning Propagation and Repair Mechanism in Multi-Agent Collaborative Environments.

[18] Raza, A., Hanif, F., & Mohammed, H. A. (2025). Analyzing the enhancement of CNN-YOLO and transformer based architectures for real-time animal detection in complex ecological environments. Scientific Reports, 15(1), 39142.

[19] Qiu, D., Xu, D., & Yue, L. (2025, December). Reinforcement Learning-Augmented LLM Agents for Collaborative Decision Making and Performance Optimization. In 2025 7th International Conference on Frontier Technologies of Information and Computer (ICFTIC) (pp. 1337-1342). IEEE.

[20] Pathak, V., & Deshkar, S. (2026). Optimising resources and localising sustainability through hierarchical village clustering in Nagpur Metropolitan Region, India. Scientific Reports, 16(1), 5749.

[21] Gao, G., Ma, X., Lu, C., & Gao, R. (2026). Reliability Analysis and Application Research of SMS Communication Systems in Medical Notification Scenarios.

ISSN: 3079-644X

[22] Mason, F., Chiariotti, F., Zanella, A., & Popovski, P. (2024). Multi-agent reinforcement learning for coordinating communication and control. IEEE Transactions on Cognitive Communications and Networking, 10(4), 1566-1581.

[23] Xu, D., Chen, H., & Gui, H. (2026). Unified Online Estimation Method for SOC, SOH, and Power Capacity Considering Safety Boundary Consistency in Battery Management Systems.

[24] Cooper, R. E., Saunders, K. R., Greenburgh, A., Shah, P., Appleton, R., Machin, K., ... & Johnson, S. (2024). The effectiveness, implementation, and experiences of peer support approaches for mental health: a systematic umbrella review. BMC medicine, 22(1), 72.