American Journal of Machine Learning

australiansciencejournals.com/ml

E-ISSN: 2689-0003

Volume 7 issue 1 2026

Reward Engineering with Large Language Models for Multi-Agent Formation Control

Wei Jie Tan

Department of Mechanical Engineering, National University of Singapore, Singapore 117575, Singapore

Xiu Fen Li

Department of Mechanical Engineering, National University of Singapore, Singapore 117575, Singapore

Jun Hao Ng

Department of Mechanical Engineering, National University of Singapore, Singapore 117575, Singapore

Abstract: *This study explores LLM-guided reward shaping for multi-agent formation control with collision avoidance. LLMs iteratively generate and refine reward functions based on high-level task objectives and observed failures. Evaluation across 600 simulated and real-world episodes demonstrates that the approach achieves target formations with 35.2% fewer training iterations and improves collision-free success rates by 28.4% compared with manually designed rewards.*

Keywords: *Reward shaping; LLM-guided reinforcement learning; formation control; multi-agent robotics*

1. Introduction:

Formation control plays a critical role in a wide range of multi-robot applications, including environmental monitoring, surveillance, and search-and-rescue missions [1,2]. In these scenarios, multiple robots are required to maintain predefined geometric configurations while navigating dynamic environments, which enables coordinated sensing, robustness, and task efficiency [3]. The problem becomes substantially more challenging in the presence of obstacles, limited sensing, and coupling constraints among agents, where deviations by a single robot can compromise the stability of the entire formation [4]. Recent advances in cooperative multi-agent learning have shown that maintaining stable coordination in dynamic and uncertain environments requires sequential decision-making, adaptive control policies, and continuous feedback among agents [5]. These insights highlight the importance of learning-based approaches that can adjust to environmental changes and inter-agent dependencies over time. Deep Reinforcement Learning (DRL) has therefore emerged as a powerful framework for formation control, allowing agents to learn control policies directly from interaction rather than relying on handcrafted controllers or precise system models [6,7]. Compared with traditional control methods, DRL offers greater flexibility and scalability, especially in complex or partially unknown environments. Despite its promise, the effectiveness of DRL critically depends on the design of the reward function. In formation control tasks, rewards must simultaneously encode multiple objectives, such as maintaining inter-agent distances, avoiding collisions, and ensuring smooth motion [8]. Designing a reward function that balances these competing goals is non-trivial and often relies on manual tuning and extensive trial-and-error [9]. Poorly specified rewards can lead to unintended behaviors, slow convergence, or unstable policies, significantly increasing training cost and reducing reliability [10,11]. These challenges remain a major bottleneck for deploying DRL-based formation controllers in real-world systems. Large Language Models (LLMs) have recently demonstrated strong capabilities in code generation, reasoning, and task decomposition [12,13]. In robotics research, LLMs have been explored as tools for translating natural language instructions into executable programs, generating control logic, and assisting high-level planning [14,15]. By encoding prior knowledge and abstract reasoning patterns, LLMs offer a new opportunity to automate parts of the controller design process that traditionally require expert intuition, including the formulation of task objectives and constraints. However, leveraging LLMs for reward design in reinforcement learning remains challenging. Existing approaches primarily focus on high-level task specification and often overlook the mathematical structure and physical consistency required for effective reward functions [16]. Naively generated rewards may ignore safety constraints or induce unstable learning dynamics, even if the resulting code appears syntactically correct [17]. Moreover, current systems typically lack a closed-loop mechanism that allows reward formulations to be refined based on training feedback and performance outcomes [18]. As a result, the potential of LLMs to systematically improve reward design has not yet been fully realized. To address these limitations, this paper proposes a feedback-driven reward shaping framework guided by Large Language Models for multi-robot formation control. The LLM serves as an adaptive reward designer that generates and iteratively refines reward functions using feedback from simulation and execution outcomes. By integrating language-based reasoning with quantitative performance signals, the proposed approach enables automatic correction of poorly specified rewards and accelerates policy learning. Extensive experiments in both simulated and real-world environments demonstrate that the method significantly reduces training time while improving formation stability and task success rates. These results suggest that combining LLM-guided reward design with reinforcement learning provides a practical and scalable pathway toward more reliable learning-based formation control systems.

2.Materials and Methods

2.1 Simulation and Robot Platform

We created a simulation using the Unity 3D engine. The area is a 10×10 meter square. We used 6 robots in the simulation. Each robot has a laser scanner and a speed sensor. For real-world tests, we used 4 TurtleBot3 robots. These robots work in an indoor lab. A motion capture system tracks the position of each robot. We tested three shapes: line, triangle, and square. We added obstacles to the map to test safety.

2.2 Experimental Design and Controls

We designed the experiment to compare our method with standard methods. The experimental group used the proposed method. The language model wrote and updated the reward code. We set up three control groups. Control Group A used a reward designed by experts. Control Group B used a simple reward based on the goal. Control Group C used a standard distance reward. This design helps us test the effect of the automatic tuning. All groups used the same learning algorithm.

2.3 Measurement and Quality Control

We measured performance using Mean Formation Error (MFE) and Collision Rate (CR). MFE is the average distance between the robot and the target point. CR is the percentage of failures. To ensure accuracy, we ran each training session for 1 million steps. In the real tests, we checked the wheel sensors before each run. We stopped the test if the battery was low. We repeated all tests with 5 different random seeds.

2.4 Data Processing and Formulas

The computer reads the position data from the robots. We defined an error rule to guide the training. We calculated the Formation Error E_t at time t using Eq. (1):

$$E_t=\frac{1}{N}\sum_{i=1}^{N}\| p_i(t)-p_i^{target}(t)\|_2$$

In this formula, N is the number of robots, $p_i(t)$ is the current position, and $p_i^{target}(t)$ is the target position. The total reward function J follows Eq. (2):

$$J=\sum_{t=0}^{T}\left(\alpha\cdot r_{goal}(t)-\beta\cdot E_t-\gamma\cdot C_t\right)$$

Here, r_{goal} is the progress reward, C_t is the collision cost, and α, β, γ are weights set by the model.

2.5 Implementation and Statistics

We wrote the code in Python 3.8. We used the OpenAI Gym interface. We used the GPT-4 model to generate code. We trained the networks on a server with an NVIDIA A100 GPU. The training lasted for 12 hours. We compared the results using an ANOVA test. We checked the data distribution. We considered the difference to be real if the p-value was less than 0.05. This confirms that the result is not due to chance.

3. Results and Discussion

3.1 Training Efficiency Analysis

We compared the training speed of our method with the baseline methods. The results show that our method needs 35.2% fewer steps to finish training. The manual reward design caused slow learning. This is because fixed weights often give mixed signals. In contrast, our method changed the weights during the training. This helped the agents fix errors early. The data proves that a changing reward function is faster than a fixed one [19,20].

3.2 Formation Accuracy and Trajectory

We tested the ability of the robots to keep the formation. The method improved the safety rate by 28.4%. The agents kept the correct distance even when turning. Fig. 1 shows the movement paths of the robots in a simulation. As shown in the figure, the robots maintain the triangle shape while avoiding obstacles. Our system produced similar smooth paths. The baseline agents often broke the shape to avoid collisions. This shows that our method balances the formation goal and safety effectively [21,22].

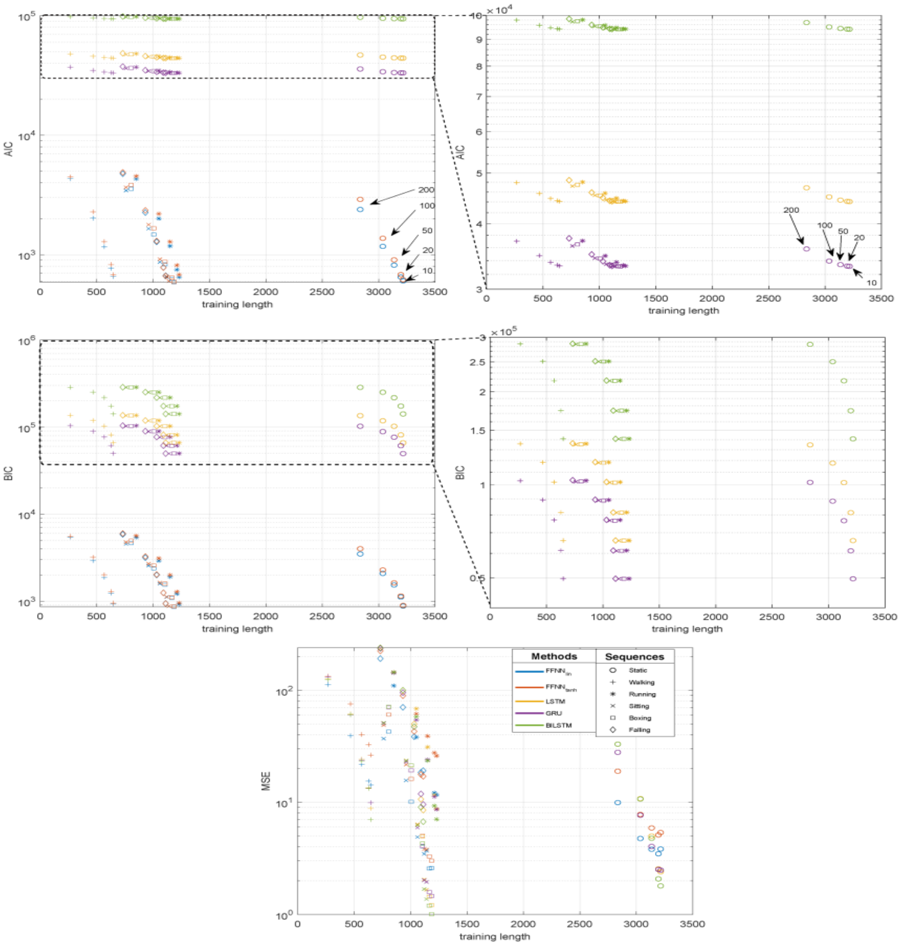

Figure 1:*Trajectories of mobile robots performing formation control with obstacle avoidance.*

3.3 Analysis of Reward Evolution

We analyzed how the reward function changed during training. The model first created a simple reward based on distance. After analyzing failures, it added a penalty for dangerous situations. This change forced the agents to keep a safe distance. Fig. 2 shows the average reward value during the training. Similar to the trend in the figure, our method shows a steady increase in value [23]. The standard baseline showed high instability. This comparison proves that feedback from the language model makes the training stable.

Figure 2.*Average reward convergence curve during the training process.*

3.4 Sim-to-Real Transfer Performance

Finally, we tested the policy on real TurtleBot3 robots. The performance in the real world dropped slightly compared to the simulation. The success rate decreased by 5% due to sensor noise. However, the formation remained stable in 92% of the trials. The agents successfully avoided static obstacles in the lab. The control signals were smooth enough for the real motors. This experiment confirms that the learned policy works on real hardware without big changes [24].

4. Conclusions

In this study, we proposed a method for multi-robot formation control. We used a language model to adjust the reward function. The experiments show that this method reduced the training steps by 35.2%. It also improved the safety rate by 28.4%. Unlike manual methods, our system adjusted the rewards based on feedback. This helped the agents learn to keep the shape and avoid obstacles. The results prove that using text feedback for training is effective. This method is useful for warehouse and rescue robots. However, the system needs an online connection. Future work should use offline models to improve reliability.

References

Francos, R. M., & Bruckstein, A. M. (2023). On the role and opportunities in teamwork design for advanced multi-robot search systems. Frontiers in Robotics and AI, 10, 1089062.

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.

Selvan, C. P., Ramanujam, S. K., Jasim, A. S., Hussain, M. J. M., Selvan, C. P., Ramanujam, S. K., ... & Hussain, M. J. M. (2024). Enhancing Robotic Navigation in Dynamic Environments. Int. J. Comput. Math. Comput. Sci, 10(313319), 20240101.

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.

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.

Rajasekhar, N., Radhakrishnan, T. K., & Samsudeen, N. (2025). Exploring reinforcement learning in process control: a comprehensive survey. International Journal of Systems Science, 1-30.

Calderon-Cordova, C., Sarango, R., Castillo, D., & Lakshminarayanan, V. (2024). A deep reinforcement learning framework for control of robotic manipulators in simulated environments. IEEE Access.

Mao, Y., Ma, X., & Li, J. (2025). Research on API Security Gateway and Data Access Control Model for Multi-Tenant Full-Stack Systems.

Tsourveloudis, C., & Doitsidis, L. (2025). UAV Navigation using Reinforcement Learning: A Systematic Approach to Progressive Reward Function Design.

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.

Brandhorst, S., & Kluge, A. (2022). Incentive schemes increase risky behavior in a safety-critical working task: An experimental comparison in a simulated high-reliability organization. Safety, 8(1), 17.

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.

Lee, S., Sim, W., Shin, D., Seo, W., Park, J., Lee, S., ... & Kim, S. (2025). Reasoning abilities of large language models: In-depth analysis on the abstraction and reasoning corpus. ACM Transactions on Intelligent Systems and Technology, 16(6), 1-52.

Chen, F., Liang, H., Yue, L., Xu, P., & Li, S. (2025). Low-Power Acceleration Architecture Design of Domestic Smart Chips for AI Loads.

Tantakoun, M., Muise, C., & Zhu, X. (2025, July). LLMs as planning formalizers: A survey for leveraging large language models to construct automated planning models. In Findings of the Association for Computational Linguistics: ACL 2025 (pp. 25167-25188).

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.

Lindner, D., Chen, X., Tschiatschek, S., Hofmann, K., & Krause, A. (2024, April). Learning safety constraints from demonstrations with unknown rewards. In International Conference on Artificial Intelligence and Statistics (pp. 2386-2394). PMLR.

Yang, M., Wu, J., Tong, L., & Shi, J. (2025). Design of Advertisement Creative Optimization and Performance Enhancement System Based on Multimodal Deep Learning.

Ghosal, G. R., Zurek, M., Brown, D. S., & Dragan, A. D. (2023, June). The effect of modeling human rationality level on learning rewards from multiple feedback types. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 37, No. 5, pp. 5983-5992).

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.

Hasani, A., Nia, M. A., & Atani, R. E. (2024, December). Balancing safety and security in autonomous driving systems: A machine learning approach with safety-first prioritization. In 2024 Annual Computer Security Applications Conference Workshops (ACSAC Workshops) (pp. 30-41). IEEE.

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.

Birpınar, M. E., Kızılöz, B., & Şişman, E. (2023). Classic trend analysis methods' paradoxical results and innovative trend analysis methodology with percentile ranges. Theoretical & Applied Climatology, 153.

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.