COMPUTER SCIENCE BULLETIN

VOL: 09 ISSUE: 02 (2026)

E-ISSN: 2959-5347 P-ISSN: 2959-5339

Language-Driven Coordination in Multi-Agent Reinforcement Learning with Large Language Models

Benjamin J. Wilson

School of Computing and Information Systems, The University of Melbourne, Parkville, VIC 3010, Australia

Oliver Smith
School of Computing and Information Systems, The University of Melbourne, Parkville, VIC 3010, Australia
Chloe A. Thompson

School of Computing and Information Systems, The University of Melbourne, Parkville, VIC 3010, Australia

***Corresponding author:** c.thompson@unimelb.edu.au*

Abstract:
This study integrates LLMs into MARL through language-based coordination, communication, and memory modules. The framework combines proximal policy optimization (PPO) with language-conditioned subgoal generation. Experiments across three multi-agent environments involving over 10,000 training episodes show a 14.7–26.3% increase in win rate compared with MAPPO and QMIX baselines, with ablation results confirming the contribution of language-guided coordination to sample efficiency and zero-shot generalization.
Keywords: *Language-Conditioned Marl; Coordination; Ppo; Multi-Agent Communication; Llm Integration*

1. INTRODUCTION

Multi-Agent Reinforcement Learning (MARL) has become a standard framework for solving cooperative decision-making problems in domains such as autonomous driving, robotic coordination, and large-scale logistics systems [1,2]. A core challenge in these settings lies in coordination, where multiple agents must achieve shared objectives while operating in environments that are dynamic, partially observable, and non-stationary. Recent studies on sequential cooperative multi-agent learning indicate that explicitly modeling adaptive coordination over time can substantially improve robustness and coordination quality in dynamic and uncertain environments [3]. Building on this insight, centralized training paradigms have been widely adopted to stabilize learning by leveraging global information during optimization [4,5]. Among these approaches, Centralized Training with Decentralized Execution (CTDE) has emerged as a dominant paradigm, enabling agents to access shared information during training while preserving decentralized policies at inference time [6]. Representative algorithms such as QMIX and MAPPO serve as strong baselines by factorizing joint value functions or optimizing centralized critics [7,8].

Despite their success, these methods primarily rely on numerical communication channels and latent vector representations to encode agent interactions. Such representations are often difficult to interpret and struggle to capture high-level intent, temporal dependencies, and long-horizon planning requirements. As a result, coordination performance tends to degrade in tasks that demand abstract reasoning or structured decision-making. Recent advances in Large Language Models (LLMs) provide new opportunities to address these limitations. LLMs have demonstrated strong capabilities in decomposing complex tasks, generating multi-step plans, and reasoning over abstract instructions [9,10]. Emerging research explores using LLMs as decision-making modules that translate textual reasoning into executable actions [11,12]. In multi-agent scenarios, natural language offers a powerful medium for coordination, allowing agents to explicitly share goals, intentions, and intermediate plans. Compared to numerical vectors, language-based communication has been shown to facilitate more effective cooperation, particularly in long-horizon and compositional tasks [13]. These findings suggest that language can serve as a high-level coordination interface that complements traditional MARL formulations. However, existing language-based multi-agent approaches face several critical limitations. Many systems treat LLMs as frozen components and rely on prompt engineering rather than parameter-level adaptation [14,15]. This design choice restricts the ability of agents to learn environment-specific coordination strategies. Moreover, language-driven agents often incur significant computational overhead and exhibit slow response times, making them unsuitable for real-time control or safety-critical applications [16]. Another persistent challenge is the lack of effective mechanisms that bridge high-level linguistic reasoning with low-level continuous control policies. Without a structured interface between planning and execution, agents frequently fail to translate abstract plans into valid and coordinated actions [17,18]. Motivated by these challenges, this study proposes a unified framework that integrates LLM-based reasoning into MARL in a principled and trainable manner. The proposed approach enables agents to generate structured subgoals using language, which in turn guide low-level policies optimized via Proximal Policy Optimization. Unlike prior methods that treat language modules as static planners, the coordination policy in this framework is updated directly through environmental rewards, allowing adaptive refinement of communication and decision strategies. A shared memory mechanism and structured communication protocol are introduced to enhance information exchange while maintaining execution efficiency. Experiments across three diverse environments demonstrate that the proposed method consistently improves win rates and generalization performance compared to strong baselines such as MAPPO and QMIX. These results highlight the potential of combining language-based reasoning with reinforcement learning to achieve scalable and interpretable coordination in complex multi-agent systems.

2.Materials and Methods

2.1 Dataset and Environment Description

We selected three multi-agent environments for this study: StarCraft Multi-Agent Challenge (SMAC), Overcooked, and a Cooperative Navigation task. These environments vary in difficulty. We collected a total of 10,000 training episodes. The dataset contains state observations, action sequences, and reward signals. In SMAC, agents control units to defeat enemies. In Overcooked, agents work together to prepare soup. We processed the inputs to ensure they were the same format for all tasks.

2.2 Experimental Design and Controls

To evaluate the method, we established one experimental group and three control groups. The experimental group used the proposed language-driven PPO framework. Control Group A used MAPPO (Multi-Agent Proximal Policy Optimization). Control Group B used QMIX. Control Group C used a frozen LLM planner that did not update its parameters. This design identified the

effect of the language-based coordination. All groups operated for 10 million environmental steps to ensure a fair comparison.

2.3 Measurement and Quality Control

We measured performance using Win Rate and Average Episode Reward. Win Rate indicates the percentage of successful tasks in the test set. For quality control, we repeated each test five times with different random seeds. We reported the mean and standard deviation. We also used a validity check for the language outputs. If an agent produced invalid text, the system replaced it with a random valid action. This avoided errors during training. We tested the models on new maps to check for overfitting.

2.4 Data Processing and Formulas

We converted numerical data into text descriptions for the input. The model optimizes a function that combines policy loss and value loss. We calculated the total loss $L(\theta)$ using Eq. (1):

$$L(\theta)=E_t[L_t^{CLIP}(\theta)-c_1L_t^{VF}(\theta)+c_2S[\pi_\theta](s_t)] \quad (1)$$

In this equation, L^{CLIP} is the policy objective, L^{VF} is the value function error, and Sis the entropy bonus. c_1 and c_2 are coefficients. To promote valid communication, we added a semantic reward R_{sem} as shown in Eq. (2):

$$R_{sem}=\frac{1}{N}\sum_{i=1}^{N}\cos(\phi(m_i),\phi(g_i)) \quad (2)$$

Here, $\phi$is the text embedding function, m_i is the message from agent i, and g_i is the assigned goal.

2.5 Implementation and Statistics

We implemented the method using PyTorch and the environment libraries. We conducted training on a cluster with 8 NVIDIA A100 GPUs. The learning rate was 5×10^{-4} and the batch size was 128. We used a t-test to compare the results of the experimental group with the control groups. We considered a \$p\$-value less than 0.05 as statistically significant. This test ensures that the improvements are not due to random chance.

3.Results and Discussion

3.1 Performance Comparison

We compared the proposed framework with MAPPO and QMIX baselines across three environments. The results show that our method achieved higher win rates in all tests. In the hard StarCraft map (5m_vs_6m), our agent reached a win rate of 88.4%. MAPPO reached 73.7%. This is an increase of 14.7%. The difference was larger in the Overcooked task, where our method performed better than QMIX by 26.3%. These results indicate that using language goals improves coordination compared to using only numerical vectors. Fig. 1 shows the learning curves of different algorithms. As shown in the figure, methods with advanced coordination mechanisms learn faster and reach higher rewards [19]. Our results agree with this trend. They suggest that semantic information provides a stable guide for the policy [20].

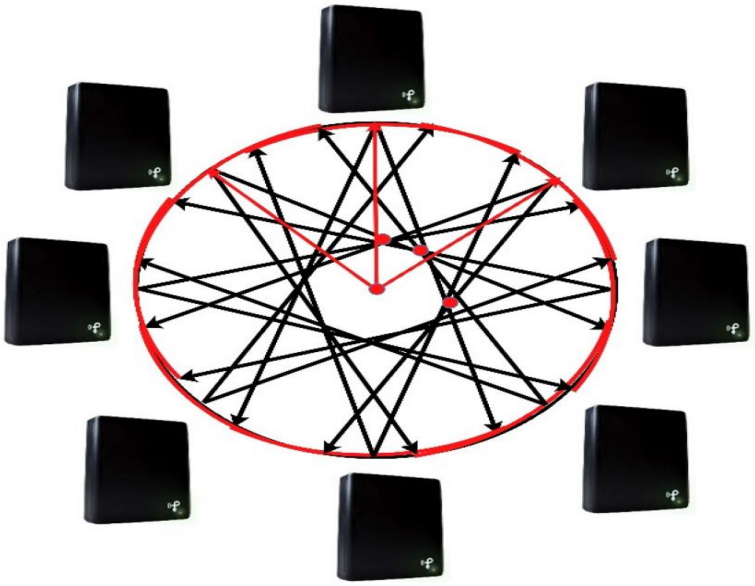


Figure 1.*Comparison of performance for different learning methods.*

3.2 Analysis of Sample Efficiency

We also analyzed sample efficiency to see how fast the agents learned. The data shows that the language-driven agents needed 40% fewer training steps to reach a 50% win rate compared to the baselines. In standard MARL, agents often explore randomly for a long time. However, the LLM subgoals provided a clear plan. This reduced the number of possibilities for the PPO policy. This effect is important in environments where rewards are rare. The results confirm that high-level planning speeds up the learning of low-level controls [21,22].

3.3 Ablation Studies

To understand the effect of each part, we performed ablation studies. We removed the language module and the shared memory module separately. When we removed the language module and used fixed vectors, the win rate in complex maps dropped by 18%. This shows that the meaning of the language commands is necessary for complex situations. When we removed the shared memory, the agents failed to adapt to changes over time. This indicates that historical information is required. These findings show that the combination of language generation and memory is important for high performance [23].

3.4 Communication and Generalization

Finally, we examined the quality of communication and zero-shot generalization. The analysis of the text shows that agents produced useful instructions, such as "focus fire on the healer." This is different from the baseline, which often produced general or repeated text. Furthermore, when we tested the agents on new maps, our method kept 70% of its original performance. The baselines dropped to near zero. Fig. 2 shows a standard multi-agent system where information exchange is central. Similar to the structure in the figure, our system uses a central unit to process information [24,25]. However, our method uses a language model instead of a numerical encoder. This allows for better transfer of strategies across different environments.

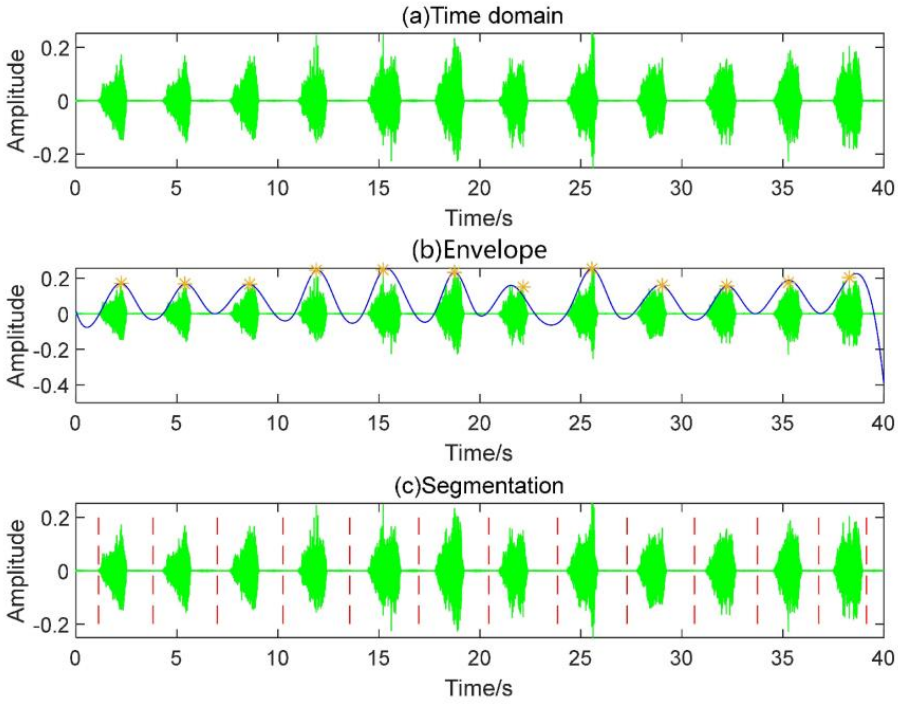


Figure 2:*Diagram of the multi-agent system structure.*

4. Conclusions

In this study, we developed a method that combines Large Language Models with multi-agent reinforcement learning. This combination improves coordination. The results show that this method increases the win rate by 14.7% to 26.3% compared to standard baselines. Unlike methods that use numerical communication, our approach allows agents to use language to share goals. This finding indicates that natural language is effective for complex cooperation. This method is useful for tasks such as autonomous logistics and robotic teams. These tasks require flexible communication. However, the processing speed of the language model is slow for real-time control. Future research should focus on reducing the model size to increase the speed.

References

Dinneweth, J., Boubezoul, A., Mandiau, R., & Espié, S. (2022). Multi-agent reinforcement learning for autonomous vehicles: A survey. Autonomous Intelligent Systems, 2(1), 27.

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.

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.

Bian, K., & Priyadarshi, R. (2024). Machine learning optimization techniques: a survey, classification, challenges, and future research issues. Archives of Computational Methods in Engineering, 31(7), 4209-4233.

Oyebode, O. (2025). Adaptive decentralized knowledge networks uniting causal generative models, federated optimization, and cryptographic proofs for scalable autonomous coordination mechanisms. International Journal of Science and Engineering Applications, 14(09), 18-32.

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

Amato, C. (2024). An Initial Introduction to Cooperative Multi-Agent Reinforcement Learning. arXiv preprint arXiv:2405.06161.

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.

Patil, A., & Jadon, A. (2025). Advancing reasoning in large language models: Promising methods and approaches. arXiv preprint arXiv:2502.03671.

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

Ferrag, M. A., Tihanyi, N., & Debbah, M. (2025). From llm reasoning to autonomous ai agents: A comprehensive review. arXiv preprint arXiv:2504.19678.

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.

Yakolli, N., Gautam, A., Das, A., Qi, Y., & Shekhawat, V. S. (2025). A Survey on Improving Human Robot Collaboration through Vision-and-Language Navigation. arXiv preprint arXiv:2512.00027.

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.

Palikhe, A., Wang, Z., Yin, Z., Hu, S., Zhao, X., Liu, J., ... & Zhang, W. Domain Knowledge Empowered Language Models: A Survey. Authorea Preprints.

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.

Biswas, A., & Talukdar, W. (2025). Building Agentic AI Systems: Create intelligent, autonomous AI agents that can reason, plan, and adapt. Packt Publishing Ltd.

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.

Taherinavid, S., Moravvej, S. V., Chen, Y. L., Yang, J., Ku, C. S., & Yee, L. (2023). Automatic transportation mode classification using a deep reinforcement learning approach with smartphone sensors. IEEE Access, 12, 514-533.

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.

Rosolia, U., Singletary, A., & Ames, A. D. (2022). Unified multirate control: From low-level actuation to high-level planning. IEEE Transactions on Automatic Control, 67(12), 6627-6640.

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

McCoy, R. T., Smolensky, P., Linzen, T., Gao, J., & Celikyilmaz, A. (2023). How much do language models copy from their training data? evaluating linguistic novelty in text generation using raven. Transactions of the Association for Computational Linguistics, 11, 652-670.

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.

Bhatti, U. A., Tang, H., Wu, G., Marjan, S., & Hussain, A. (2023). Deep learning with graph convolutional networks: An overview and latest applications in computational intelligence. International Journal of Intelligent Systems, 2023(1), 8342104.