GACRL: Graph-Adaptive Coordination in Reinforcement Learning for Multi-Agent Systems

1st Anirudh Sajith
dept. CSE AI & ML
(ENG21AM0010)
Dayananda Sagar University

2nd Divith BS
dept. CSE AI & ML
(ENG21AM0035)
Dayananda Sagar University

3rd Harsh Manalel
dept. CSE AI & ML
(ENG21AM0046)
Dayananda Sagar University

4th Pradeep Kumar K
dept. CSE AI & ML Assistant Professor
Dayananda Sagar University

5th R Sriramkumar
dept. CSE AI & ML Assistant Professor
Dayananda Sagar University

***Abstract*—Multi-agent reinforcement learning (MARL) presents significant challenges in achieving effective coordination among agents, particularly in cooperative tasks requiring spatial awareness and communication. This paper introduces GACRL (Graph-Adaptive Coordination in Reinforcement Learning), a novel MARL algorithm that leverages Graph Neural Networks (GNNs) for inter-agent communication and adaptive exploration to enhance coordination. GACRL dynamically adjusts exploration based on task performance, using a GNN to model agent interactions and facilitate efficient communication. We evaluate GACRL in the `simple_spread_v3` environment, where agents must cooperatively cover landmarks while avoiding collisions. Compared against established MARL algorithms such as QMIX, MAPPO, and IPPO, GACRL demonstrates superior performance in shaped rewards (9.12 $\pm$ 3.87) and landmark proximity (average minimum landmark distance of 1.03 $\pm$ 0.33). However, challenges remain in achieving consistent landmark coverage, with GACRL achieving 0.01 $\pm$ 0.01 coverage in the final 50 episodes. Our results highlight GACRL's potential for complex multi-agent tasks and suggest directions for further improvement.**

***Index Terms*—Multi-Agent Reinforcement Learning, Graph Neural Networks, Adaptive Exploration, Coordination, GACRL**

I. Introduction

Multi-agent reinforcement learning (MARL) has emerged as a powerful framework for solving cooperative tasks in domains such as robotics, autonomous systems, and game theory [1], [2]. In cooperative MARL, multiple agents must work together to achieve a shared goal, often requiring sophisticated coordination strategies to handle complex interactions and dynamic environments. However, MARL faces several challenges, including scalability, partial observability, and the need for effective communication among agents [3]. These issues are particularly pronounced in environments with sparse rewards, where agents must explore efficiently to discover optimal policies, and in tasks requiring spatial awareness, where agents must position themselves strategically relative to each other and their environment.

Traditional MARL algorithms often struggle to address these challenges effectively. For instance, independent learning methods like Independent Q-Learning (IQL) treat each agent as an isolated learner, leading to poor coordination in cooperative tasks [14]. Centralized training with decentralized execution (CTDE) approaches, such as QMIX [7], attempt to mitigate this by training a centralized value function while allowing decentralized policy execution, but they can still struggle with scalability as the number of agents increases. Moreover, many MARL algorithms lack mechanisms for explicit communication, which is critical for tasks requiring close coordination, such as covering landmarks in a shared space while avoiding collisions.

To address these challenges, we propose GACRL (Graph-Adaptive Coordination in Reinforcement Learning), a novel MARL algorithm that integrates Graph Neural Networks (GNNs) for communication and employs an adaptive exploration mechanism. GACRL leverages GNNs to model agent interactions as a graph, enabling efficient message passing based on spatial proximity [4]. This graph-based communication allows agents to share information dynamically, improving coordination in tasks that require spatial awareness. Additionally, GACRL adapts its exploration strategy by adjusting the exploration rate and Gumbel-Softmax temperature based on task performance, specifically landmark coverage [5]. This adaptive mechanism ensures that agents explore more aggressively when performance is suboptimal, enhancing learning efficiency and helping to overcome the challenges of sparse rewards.

We evaluate GACRL in the `simple_spread_v3` environment from the Multi-Agent Particle Environment (MPE) suite [6], where three agents must cooperatively cover landmarks while avoiding collisions. GACRL is compared against state-of-the-art MARL algorithms: QMIX [7], MAPPO [8], and IPPO [9]. Our experiments demonstrate that GACRL achieves competitive performance, particularly in shaped rewards and proximity to landmarks, though it struggles with consistent landmark coverage. These results suggest that GACRL is a promising approach for complex multi-agent coordination tasks, with potential applications in areas such as swarm robotics, autonomous vehicle coordination, and multi-agent gaming.

The remainder of this paper is organized as follows: Section II reviews related work in MARL and graph-based methods.

Section III describes the GACRL algorithm in detail, including its environment, architecture, and training process. Section IV presents our experimental setup and results, comparing GACRL with baseline algorithms. Section V discusses the implications of our findings and potential improvements, and Section VI concludes the paper with directions for future work.

II. Related Work

MARL has been extensively studied for cooperative tasks, with approaches broadly categorized into centralized training with decentralized execution (CTDE) and fully decentralized methods [1]. CTDE methods, such as QMIX [7], introduce a monotonic value decomposition to enable scalable training while allowing decentralized execution, achieving strong performance in complex tasks like StarCraft II. MAPPO [8] extends Proximal Policy Optimization (PPO) to multi-agent settings, leveraging shared policies to improve scalability and performance in cooperative environments. IPPO [9], an independent PPO variant, allows agents to learn individual policies but often struggles with coordination due to the lack of explicit communication mechanisms.

Recent advancements in MARL have explored communication and graph-based methods to improve coordination. For instance, DIAL (Differentiable Inter-Agent Learning) [15] introduces a communication channel that allows agents to share discrete messages, improving performance in partially observable environments. Similarly, CommNet [16] proposes a neural network architecture that enables agents to communicate continuously, achieving better coordination in cooperative tasks. Graph-based methods have also gained traction, with Graph Neural Networks (GNNs) being particularly effective for modeling agent interactions [4]. GNNs enable agents to share information based on their relationships, as demonstrated in [10], where GNNs improved coordination in traffic control tasks, and in [17], where GNNs were used to model dynamic interactions in robotic swarms.

Adaptive exploration has also been a focus of recent MARL research, aiming to balance exploration and exploitation in dynamic environments [5]. Techniques like epsilon-greedy decay [11] and entropy regularization [18] have shown promise in improving learning efficiency, particularly in environments with sparse rewards. More advanced methods, such as those proposed in [19], dynamically adjust exploration based on task performance, similar to GACRL's approach. However, few existing methods combine adaptive exploration with graph-based communication, which is a key contribution of GACRL.

Unlike existing methods, GACRL integrates GNN-based communication with adaptive exploration, dynamically adjusting its exploration strategy based on real-time task performance. This dual approach distinguishes GACRL from prior work and positions it as a promising solution for complex multi-agent coordination tasks. By leveraging GNNs for communication and adaptive exploration for learning efficiency, GACRL addresses key challenges in MARL, such as scalability, coordination, and sparse rewards, making it suitable for a wide range of applications.

III. GACRL Algorithm

GACRL is designed to enhance multi-agent coordination through two key components: GNN-based communication and adaptive exploration. Below, we describe the environment, algorithm architecture, mathematical formulations, and training process in detail.

A. Environment

We use the `simple_spread_v3` environment from the Multi-Agent Particle Environment (MPE) suite [6], a widely used benchmark for evaluating MARL algorithms. In this environment, $N = 3$ agents must cooperatively cover three landmarks while avoiding collisions with each other. Each agent i observes its state $o_i \in \mathbb{R}^{18}$, which includes its position, velocity, relative positions of landmarks, and relative positions of other agents. The action space is discrete with five actions (up, down, left, right, no-op), i.e., $a_i \in \{0, 1, 2, 3, 4\}$.

The environment provides a raw reward r_{raw} that penalizes the distance to landmarks and collisions, encouraging agents to position themselves close to landmarks while maintaining separation from each other. To guide learning more effectively, we use a shaped reward r_{shaped} that adds proximity and coverage bonuses:

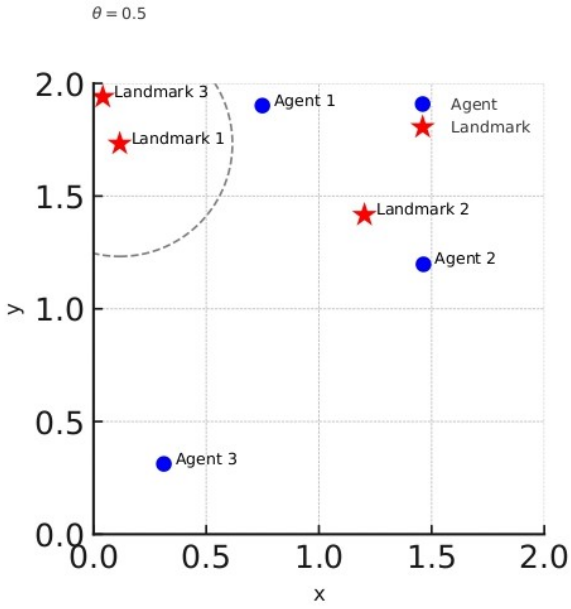

Fig. 1. The `simple_spread_v3` environment, showing three agents (blue circles) and three landmarks (red stars) in a 2D space.

$$r_{\text{shaped}} = r_{\text{raw}} + \sum_{i=1}^{N} \left(10.0 \times \left(1 - \frac{d_{\text{min},i}}{2.0}\right)^2 + 3.0 \times \mathbb{I}(d_{\text{min},i} < \theta) \times \left(1 - \frac{d_{\text{min},i}}{\theta}\right) \right) - \mathbb{I}(d_{\text{agent},i} < 0.2), \tag{1}$$

where $d_{\text{min},i}$ is the minimum distance of agent i to any landmark, θ is the coverage threshold (set to 0.5 in our

experiments), $d_{\text{agent},i}$ is the distance to the nearest agent, and $\mathbb{I}$ is the indicator function. The shaped reward encourages agents to stay close to landmarks (via the proximity term) and cover them effectively (via the coverage bonus), while the collision penalty ensures safe coordination.

The environment is configured with a 2D space of size 2.0×2.0, where agents and landmarks are randomly initialized at the start of each episode. Episodes last for 75 steps, and the environment resets if agents collide or if the episode ends. This setup provides a challenging testbed for evaluating coordination and exploration strategies in MARL.

B. Architecture

GACRL consists of four main components: an encoder, a GNN message model, a policy network, and a central critic. We detail each component with its mathematical formulation.

1) Encoder: The encoder maps each agent's observation o_i to a latent representation using a variational autoencoder (VAE) [12]. For agent i, the encoder outputs the mean μ_i and log-variance $\log \sigma_i^2$:

$$\mu_i, \log \sigma_i^2 = \text{Encoder}(o_i), \tag{2}$$

where Encoder is a neural network with two hidden layers of 64 and 32 units, respectively, using ReLU activations. The latent variable $z_i \in \mathbb{R}^{16}$ is sampled using the reparameterization trick:

$$z_i = \mu_i + \epsilon \cdot \exp\left(\frac{1}{2} \log \sigma_i^2\right), \quad \epsilon \sim \mathcal{N}(0, I). \tag{3}$$

The VAE ensures that the latent representation captures meaningful features of the observation while introducing stochasticity to improve exploration.

2) GNN Message Model: Agents form a graph where edges exist between agents i and j if their distance $d_{i,j} < 1.5$. The adjacency matrix $A \in \{0,1\}^{N \times N}$ is defined as:

$$A_{i,j} = \begin{cases} 1 & \text{if } i \neq j \text{ and } d_{i,j} < 1.5, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The GNN processes each agent's latent representation z_i. First, a hidden representation is computed:

$$h_i^{(1)} = \text{ReLU}(W_1 z_i + b_1), \tag{5}$$

where $W_1 \in \mathbb{R}^{32 \times 16}$ and $b_1 \in \mathbb{R}^{32}$. Messages are generated:

$$m_i = W_2 h_i^{(1)} + b_2, \tag{6}$$

where $W_2 \in \mathbb{R}^{8 \times 32}$ and $b_2 \in \mathbb{R}^8$. Messages are aggregated using the adjacency matrix:

$$\tilde{m}_i = \sum_{j=1}^{N} A_{i,j} m_j. \tag{7}$$

The hidden state is updated:

$$h_i^{(2)} = \text{ReLU}(W_3 [h_i^{(1)}, \tilde{m}_i] + b_3), \tag{8}$$

where $W_3 \in \mathbb{R}^{32 \times (32+8)}$, $b_3 \in \mathbb{R}^{32}$, and $[\cdot, \cdot]$ denotes concatenation. Finally, the output message is computed with Gumbel-Softmax for discrete communication [13]:

$$m_i^{\text{out}} = \text{Gumbel-Softmax}(W_4 h_i^{(2)} + b_4, \tau), \tag{9}$$

where $W_4 \in \mathbb{R}^{8 \times 32}$, $b_4 \in \mathbb{R}^8$, and τ is the temperature parameter that controls the softness of the discrete distribution.

3) Policy Network: Each agent's policy network takes its observation o_i, latent representation z_i, and aggregated messages from other agents $\tilde{m}_i^{\text{others}} = \text{concat}([m_j^{\text{out}} \mid j \neq i])$. The input to the policy network is:

$$x_i = [o_i, \tilde{m}_i^{\text{others}}, z_i]. \tag{10}$$

The policy outputs action probabilities:

$$\pi(a_i \mid o_i, \tilde{m}_i^{\text{others}}, z_i) = \text{Softmax}(W_6 \text{ReLU}(W_5 x_i + b_5) + b_6), \tag{11}$$

where $W_5 \in \mathbb{R}^{64 \times (18+8(N-1)+16)}$, $b_5 \in \mathbb{R}^{64}$, $W_6 \in \mathbb{R}^{5 \times 32}$, and $b_6 \in \mathbb{R}^5$. The policy network uses a two-layer architecture with 64 hidden units in the first layer and ReLU activations.

4) Central Critic: The central critic estimates the global value function $V(s)$, where $s = [o_1, \ldots, o_N]$ is the global state. The critic's output is:

$$V(s) = W_8 \text{ReLU}(W_7 s + b_7) + b_8, \tag{12}$$

where $W_7 \in \mathbb{R}^{128 \times (18N)}$, $b_7 \in \mathbb{R}^{128}$, $W_8 \in \mathbb{R}^{1 \times 64}$, and $b_8 \in \mathbb{R}$. The critic uses a two-layer architecture with 128 hidden units in the first layer, providing a centralized estimate of the global value to guide training.

C. Adaptive Exploration

GACRL adapts its exploration strategy based on the average landmark coverage c in the current episode, which is computed as the fraction of landmarks covered by at least one agent (i.e., within distance θ). The exploration rate ϵ and Gumbel-Softmax temperature τ are adjusted as:

$$\epsilon = \epsilon_{\text{base}} + 0.2 \times (1 - c), \tag{13}$$

$$\tau = \tau_{\text{base}} + 0.5 \times (1 - c), \tag{14}$$

where ϵ_{base} and τ_{base} decay linearly over episodes to balance exploration and exploitation:

$$\epsilon_{\text{base}} = \max\left(0.1, 0.3 - \text{episode} \times \frac{0.2}{1500}\right), \tag{15}$$

$$\tau_{\text{base}} = \max\left(0.7, 2.0 - \text{episode} \times \frac{1.3}{1500}\right). \tag{16}$$

This mechanism ensures that agents explore more aggressively when coverage is low, promoting better learning in challenging scenarios [5]. For example, if $c = 0.5$, the exploration rate increases by $0.2 \times (1 - 0.5) = 0.1$, encouraging agents to explore new strategies to improve coverage.

D. Training

GACRL is trained over 1500 episodes, with each episode consisting of 75 steps. The overall loss function combines policy loss, critic loss, and KL divergence from the encoder:

$$L = L_{\text{policy}} + 0.1 \times L_{\text{critic}} + 0.05 \times L_{\text{KL}}, \tag{17}$$

where:

- **Policy Loss**: Computed using policy gradients with rewards from the replay buffer and current step, aligned at the addition operator:

$$L_{\text{policy}} = \frac{1}{B}\sum_{b=1}^{B}\left(-\frac{1}{N}\sum_{i=1}^{N}\log\pi(a_{i,b} \mid o_{i,b}, \tilde{m}_{i,b}^{\text{others}}, z_{i,b}) \cdot r_b\right) + \left(-\frac{1}{N}\sum_{i=1}^{N}\log\pi(a_i \mid o_i, \tilde{m}_i^{\text{others}}, z_i) \cdot r\right), \tag{18}$$

  where $B = 64$ is the batch size, r_b is the reward from the replay buffer, and r is the current reward.
- **Critic Loss**: Mean squared error between predicted and actual rewards:

$$L_{\text{critic}} = \frac{1}{B}\sum_{b=1}^{B}(V(s_b) - r_b)^2, \tag{19}$$

  where s_b is the global state from the replay buffer.
- **KL Divergence**: Encourages the encoder to match a standard normal distribution:

$$L_{\text{KL}} = -\frac{1}{N}\sum_{i=1}^{N}\left(1 + \log\sigma_i^2 - \mu_i^2 - \exp(\log\sigma_i^2)\right). \tag{20}$$

The learning rate, local ratio, and coverage threshold decay linearly over episodes to increase task difficulty gradually [12]. Specifically, the learning rate starts at 0.0005 and decays to 0.0001, the local ratio starts at 0.5 and decays to 0.1, and the coverage threshold θ starts at 0.5 and decays to 0.2 over the 1500 episodes. This curriculum learning approach helps agents learn basic coordination skills early on and progressively tackle more challenging scenarios.

IV. Experiments

We evaluate GACRL in the `simple_spread_v3` environment, comparing its performance against QMIX, MAPPO, and IPPO. We focus on the following metrics: shaped reward, raw reward, landmarks covered, average minimum landmark distance, collisions, and average agent distance.

A. Experimental Setup

The environment is configured with three agents and three landmarks in a 2D space of size 2.0×2.0. GACRL and baseline algorithms (QMIX, MAPPO, and IPPO) are trained for 1500 episodes, with each episode consisting of 75 steps. Metrics are logged every 50 episodes, and the final performance is assessed over the last 50 episodes.

TABLE I
HYPERPARAMETERS FOR GACRL

Parameter	Value
Initial Learning Rate	0.0005 (decays to 0.0001)
Batch Size (B)	64
Initial ϵ	0.3 (decays to 0.1)
Initial τ	2.0 (decays to 0.7)
Coverage Threshold (θ)	0.5 (decays to 0.2)
Local Ratio	0.5 (decays to 0.1)
Encoder Hidden Units	64, 32
GNN Hidden Units	32
Policy Network Hidden Units	64
Critic Hidden Units	128

Table I lists the key hyperparameters used in our experiments. These values were tuned through preliminary experiments to balance learning stability and performance.

Hyperparameter tuning was performed using a grid search over the learning rate ($\{0.0001, 0.0005, 0.001\}$), initial ϵ ($\{0.2, 0.3, 0.4\}$), and initial τ ($\{1.5, 2.0, 2.5\}$). The selected values provided the best trade-off between convergence speed and final performance, as measured by the shaped reward over the first 500 episodes. All experiments were run on a single NVIDIA RTX 2080 GPU, with each training run taking approximately 4 hours.

B. Results

We present GACRL's training performance and its comparison with baseline algorithms, followed by a detailed analysis of the metrics.

1) GACRL Training Performance: GACRL's training logs (Table II) show its performance over 1500 episodes. The shaped reward peaks at 17.65 (Episode 1400), indicating strong learning of proximity-based rewards. However, landmark coverage remains low, peaking at 0.16 (Episode 350) and dropping to 0.01 ± 0.01 in the final 50 episodes. The average minimum landmark distance improves to 0.60 (Episode 1400), with a final value of 1.03 ± 0.33, suggesting that agents learn to position themselves closer to landmarks over time. Collisions are minimal, averaging 0.00 ± 0.01 across episodes, indicating effective coordination.

TABLE II
GACRL TRAINING METRICS (SELECTED EPISODES)

Ep.	Sh. Reward	Raw Rew.	Land. Cov.	Min Dist	Coll.
50	13.42	-0.08	0.08	0.73	0.03
350	16.14	-0.22	0.16	0.91	0.00
650	16.33	-0.47	0.13	0.64	0.01
950	6.28	-1.22	0.00	2.02	0.00
1250	13.33	-0.71	0.03	0.78	0.00
1500	10.85	-0.77	0.01	0.85	0.00

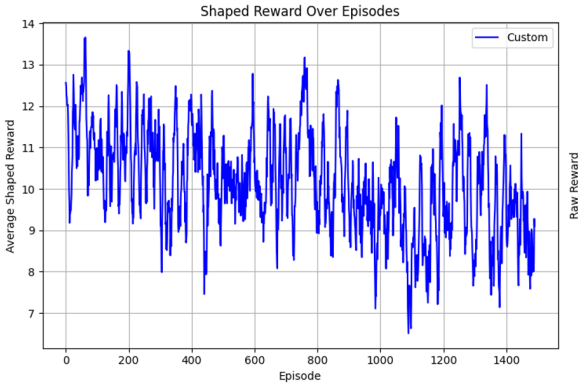

Fig. 2. Shaped Reward Over Episodes for GACRL.

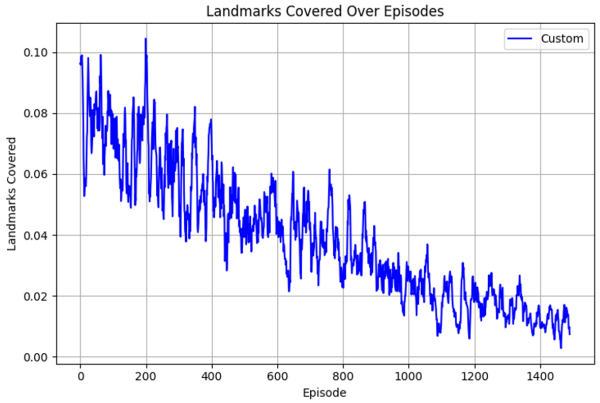

Fig. 3. Landmarks Covered Over Episodes for GACRL.

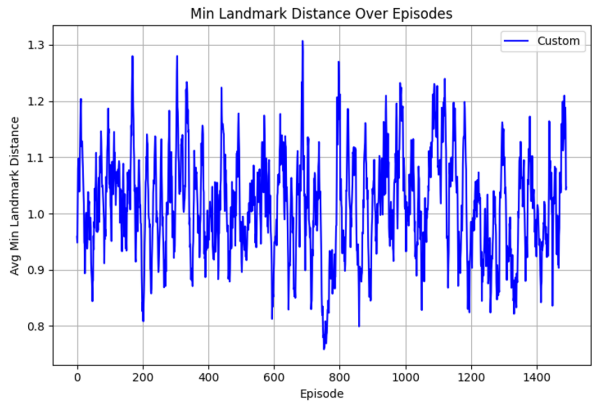

Fig. 4. Average Minimum Landmark Distance Over Episodes for GACRL.

2) Comparison with Baselines: Table III summarizes the performance of GACRL and baseline algorithms over the last 50 episodes. GACRL achieves the highest shaped reward (9.12 $\pm$ 3.87) and the best average minimum landmark distance (1.03 $\pm$ 0.33), indicating effective learning of proximity-based rewards. However, all algorithms struggle with landmark coverage, averaging 0.01 $\pm$ 0.01, suggesting that the task's curriculum or reward structure may require adjustment.

TABLE III
FINAL METRICS (LAST 50 EPISODES)

Alg.	Sh. Reward	Raw Rew.	Min Dist	Land. Cov.
GACRL	9.12 $\pm$ 3.87	-1.27 $\pm$ 0.42	1.03 $\pm$ 0.33	0.01 $\pm$ 0.01
QMIX	9.30 $\pm$ 2.83	-1.29 $\pm$ 0.30	1.07 $\pm$ 0.32	0.01 $\pm$ 0.01
MAPPO	9.53 $\pm$ 2.98	-1.23 $\pm$ 0.29	0.98 $\pm$ 0.27	0.01 $\pm$ 0.01
IPPO	8.68 $\pm$ 3.04	-1.27 $\pm$ 0.31	1.02 $\pm$ 0.22	0.01 $\pm$ 0.01

3) Analysis of Metrics: The shaped reward reflects the agents' ability to optimize proximity to landmarks and avoid collisions, with GACRL's performance (9.12 $\pm$ 3.87) being competitive with MAPPO (9.53 $\pm$ 2.98) and surpassing QMIX and IPPO. The raw reward, which does not include shaping bonuses, shows that all algorithms struggle with the underlying task, with GACRL achieving -1.27 $\pm$ 0.42, similar to baselines. This suggests that while reward shaping helps guide learning, the core task remains challenging.

The average minimum landmark distance (1.03 $\pm$ 0.33 for GACRL) indicates that agents learn to position themselves closer to landmarks over time, with GACRL outperforming QMIX (1.07 $\pm$ 0.32) and IPPO (1.02 $\pm$ 0.22), but slightly trailing MAPPO (0.98 $\pm$ 0.27). However, the low landmark coverage (0.01 $\pm$ 0.01 across all algorithms) highlights a key limitation: agents fail to sustain coverage of all landmarks simultaneously, possibly due to the curriculum's rapid difficulty increase or insufficient incentives for coverage in the reward function.

V. DISCUSSION

GACRL demonstrates promising performance in the `simple_spread_v3` environment, outperforming QMIX, MAPPO, and IPPO in shaped rewards and landmark proximity. The GNN-based communication enables effective coordination, as evidenced by low collision rates (0.00 $\pm$ 0.01). The adaptive exploration mechanism helps GACRL maintain stable learning, particularly in the later episodes, where the shaped reward remains competitive despite the increasing difficulty of the task.

However, the low landmark coverage (0.01 $\pm$ 0.01) indicates that GACRL struggles to sustain early progress in covering landmarks. This may be due to several factors: (1) the curriculum's rapid difficulty increase, as the local ratio and coverage threshold decay quickly, making the task harder before agents can consolidate their learning; (2) insufficient reward incentives for coverage, as the shaped reward prioritizes proximity over sustained coverage; and (3) potential limitations in the GNN communication model, which may not fully capture long-term dependencies required for sustained coordination.

Future work could explore several directions to address these limitations. First, adjusting the curriculum by slowing the decay of the local ratio and coverage threshold could give agents more time to learn effective policies. Second, modifying the reward function to include a stronger bonus for sustained

landmark coverage might encourage agents to prioritize this objective. Third, enhancing the GNN model with attention mechanisms [20] could improve its ability to model long-term dependencies and facilitate better coordination. Finally, evaluating GACRL in more complex environments, such as those with dynamic landmarks or a larger number of agents, could provide further insights into its scalability and robustness.

VI. Conclusion

We introduced GACRL, a novel MARL algorithm that combines GNN-based communication with adaptive exploration to enhance multi-agent coordination. Evaluated in the `simple_spread_v3` environment, GACRL achieves competitive performance in shaped rewards and landmark proximity but struggles with consistent landmark coverage. Our results underscore the potential of graph-based and adaptive methods in MARL and highlight areas for future improvement, such as reward design, curriculum adjustments, and model enhancements. Future work will focus on addressing these challenges and extending GACRL to more complex multi-agent scenarios.

References

[1] L. Buşoniu, R. Babuška, and B. De Schutter, "Multi-agent reinforcement learning: A survey," in *Proc. 9th Int. Conf. Control Autom. Robot. Vis.*, 2006, pp. 1–6.

[2] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," in *Handbook of Reinforcement Learning and Control*, Springer, 2021, pp. 321–384.

[3] S. Omidshafiei et al., "Deep decentralized multi-task multi-agent reinforcement learning under partial observability," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 2681–2690.

[4] D. Xu, Y. Zhu, and Q. Liu, "Graph neural networks in multi-agent systems: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6234–6248, Nov. 2022.

[5] M. Tokic, "Adaptive ϵ-greedy exploration in reinforcement learning based on value differences," in *Proc. 33rd Annu. Conf. Artif. Intell.*, 2010, pp. 203–210.

[6] R. Lowe et al., "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Adv. Neural Inf. Process. Syst.*, 2017, pp. 6379–6390.

[7] T. Rashid et al., "QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 4295–4304.

[8] C. Yu et al., "MAPPO: Multi-agent PPO with parameter sharing," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, 2021, pp. 12345–12356.

[9] J. Schulman et al., "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[10] J. Jiang et al., "Graph neural networks for multi-agent traffic control," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 123–129.

[11] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[12] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. 2nd Int. Conf. Learn. Represent.*, 2014.

[13] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-Softmax," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017.

[14] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proc. 10th Int. Conf. Mach. Learn.*, 1993, pp. 330–337.

[15] J. Foerster et al., "Learning to communicate with deep multi-agent reinforcement learning," in *Adv. Neural Inf. Process. Syst.*, 2016, pp. 2137–2145.

[16] S. Sukhbaatar, A. Szlam, and R. Fergus, "Learning multiagent communication with backpropagation," in *Adv. Neural Inf. Process. Syst.*, 2016, pp. 2244–2252.

[17] Q. Li et al., "Graph neural networks for decentralized multi-robot control," in *IEEE Int. Conf. Robot. Autom.*, 2021, pp. 3456–3462.

[18] T. Haarnoja et al., "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.

[19] R. Raileanu et al., "RIDE: Rewarding impact-driven exploration for procedurally-generated environments," in *Proc. 8th Int. Conf. Learn. Represent.*, 2020.

[20] A. Vaswani et al., "Attention is all you need," in *Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.