Integrating Multi-Agent Deep Deterministic Policy Gradient and Go-Explore for Enhanced Reward Optimization

Muchen Liu *

International Department, Capital Normal University High School, Beijing, China

* Corresponding Author Email: 201010012930@stu.swmu.edu.cn

Abstract. The field of Multi-Agent Reinforcement Learning (MARL) continues to advance with the development of new and effective methods. This research is centered on two prominent approaches within this field: Multi-Agent Deep Deterministic Policy Gradient (MADDPG) and Go-Explore. The study explores the synergistic potential of combining these two methodologies to enhance rewards for individual agents as well as for agent groups. In the course of this research, MADDPG is introduced into the experimental environment, providing agents with both actor networks (policy networks) and critic networks (Q networks) to implement the actor-critic model. Additionally, each individual agent is equipped with a Go-Explore network, empowering them to conduct deeper explorations of the environment and accumulate rewards at an accelerated rate, often resulting in higher overall rewards. This novel approach emphasizes achieving a balance between individual and collaborative rewards, offering a promising avenue for optimizing multi-agent systems. The results of this study demonstrate that the combined method exhibits notable advantages in certain scenarios. Specifically, it showcases a higher rate of reward accumulation and improved overall performance. This research contributes to the MARL domain by highlighting the potential of combining MADDPG and Go-Explore to enhance the efficiency and effectiveness of multi-agent systems.

Keywords: Reinforcement learning; machine learning; reward optimization.

1. Introduction

Reinforcement Learning (RL) has found widespread application in multi-agent scenarios, yielding significant results [1]. MARL often involves agents learning and acting independently through trial-and-error and collaboration, with or without communication [2]. Current MARL algorithms have successfully optimized rewards individually and collaboratively in stochastic environments. One notable approach that addresses the challenge of maximizing single-agent rewards is Go-Explore [3]. This method ensures that agents do not forget the sequential actions needed to reach a satisfactory state. It has effectively addressed long-standing challenges in reinforcement learning, such as sparse and deceptive rewards. Go-Explore has emerged as a potent agent trainer in Atari games, and this research suggests extending its use to multi-agent scenarios by equipping each agent with a Go-Explore's critic network [4]. On the other hand, the MADDPG stands as an improved version of the Deep Deterministic Policy Gradient (DDPG) algorithm, designed to enhance rewards in collaborative-competitive environments [5]. MADDPG has proven effective in various multi-agent situations, including collaborative and competitive scenarios. This research proposes implementing MADDPG and equipping agents with a Critic network. Unfortunately, MARL approaches like MADDPG tend to underperform in multi-agent environments where individual rewards are as crucial as collaborative rewards [6]. One challenge is their instability or difficulty in converging due to non-stationarity caused by interactions among multiple learning agents. The environment becomes more complex from the perspective of each individual agent as other agents continuously change their policies. To address these challenges, the paper introduces MADDPG-GE, a hybrid algorithm that combines MA-DDPG with Go-Explore. This combined approach aims to achieve synchronous optimization of both the individual agent and the group.

2. Related Work

Markov Decision Process (MDP) Reinforcement Learning operates on a multi-agent level using MDPs. It employs a set of states (S... Sn) and a set of actions (A....An) to define N agents. Each agent is assigned to its unique observation space (O). To choose the next action, agent (j) uses a policy function π_j: (0,1) based on the transition probability matrix Pi: (0,1]. The primary objective for each agent (j) is to maximize the reward function $\sum_{t=0}^{T} \gamma^t \times R(st)$, where γ represents the discount factor [7].

MADDPG was introduced to enhance DDPG under multi-agent environments. In such dynamic settings, each agent observes a stochastic and non-stationary environment while other agents' policies are concurrently optimized. Traditional methods like Q-Learning and DQNs become ineffective in such scenarios. The time complexity of Policy Gradient algorithms exponentially increases as many agents optimize their policies simultaneously. To address these challenges, MADDPG was proposed [8]. Presently, MADDPG demonstrates exceptional performance in most multi-agent environments, particularly in cooperative settings, where agents trained with MADDPG attain the highest rewards compared to other MARL algorithms. However, a limitation of MADDPG is that it hinders each agent's ability to explore the environment independently based on their own observations. To overcome this limitation, the research integrates the Go-Explore method for each agent.

Go-Explore is introduced as a method to optimize rewards for single agents. It addresses challenges posed by challenging exploration scenarios where rewards are sparse and deceptive [9]. For instance, in some scenarios, obtaining a high reward requires a sequence of actions. However, each action within this sequence incurs minor negative rewards, preventing the agent from completing the sequence. To mitigate these issues, Go-Explore maintains an archive to store sequences of actions that lead to high rewards, discovered after extensive exploration. In the initial phase of the algorithm, the agent operates in deterministic and resettable environments with the goal of exploration. Upon detecting new states or promising pathways, the agent appends them to the archive for future reference. During each turn, the agent selects a state from the archive for exploration. While this method has demonstrated success in single-agent environments, its application in multi-agent settings remains relatively unexplored. The research seeks to bridge this gap by integrating Go-Explore with the MADDPG algorithm, with the expectation of achieving higher rewards in multi-agent environments.

3. Methodology

3.1. The goal of experimentation and model selection

The paper aims to enhance both global and local rewards in a multi-agent environment by proposing the integration of MADDPG and Go-Explore. Fig. 1 illustrates the research process. This approach involves enabling each agent to explore the environment locally using Go-Explore and adopting MADDPG for global reward optimization.

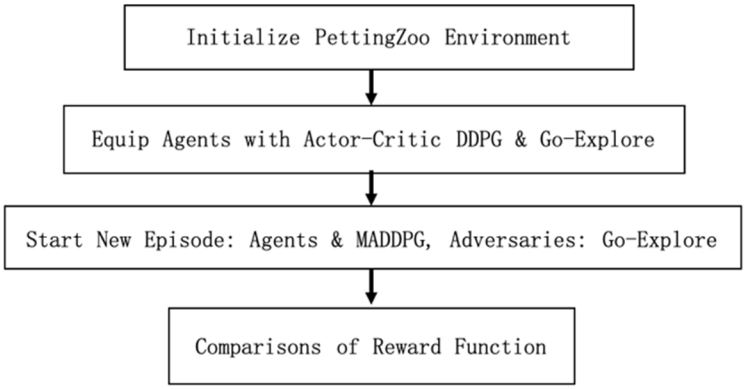

Fig. 1 Process of the research (Photo/Picture credit: Original)

The PettingZoo framework is chosen for its effectiveness in testing reinforcement learning algorithms [10]. This environment is utilized for method implementation. The research leverages Pytorch and PARL packages for neural network optimization and function approximation.

3.2. Metrics

The evaluation criteria for the experiment encompass the rewards obtained by each individual agent as well as the mean reward for all agents, calculated using Formula 1, where M(a) represents the mean reward, N signifies the total number of agents, and R(a) denotes the rewards acquired by individual agents.

$$\mathrm{M(a)} = \frac{1}{N}\sum_{t=1}^{N} R(a) \tag{1}$$

In Table 1, the mean metrics for each algorithm are calculated based on the state of each step.

Table 1. Mean Calculation for Each Step

Rewards for Go-Explore (Mean)	Rewards for MADDPG (Mean)	Rewards for MADDPG with Go-Explore (Mean)
Mge(a)	Mmaddpg(a)	Mmaddpg-ge(a)

3.3. MADDPG for agents

The research argues that equipping each agent with an Actor-Critic network for performance assessment is essential. Centralized Training and Decentralized Execution are employed to train both the Critic and the Actor [11, 12]. During training, each Actor is guided by the Critic on a centralized level, where the Critic can observe the entire environment. However, when making decisions, the Actor relies on limited observations of the environment and learning policies. Fig. 2 illustrates the mechanism of MADDPG.

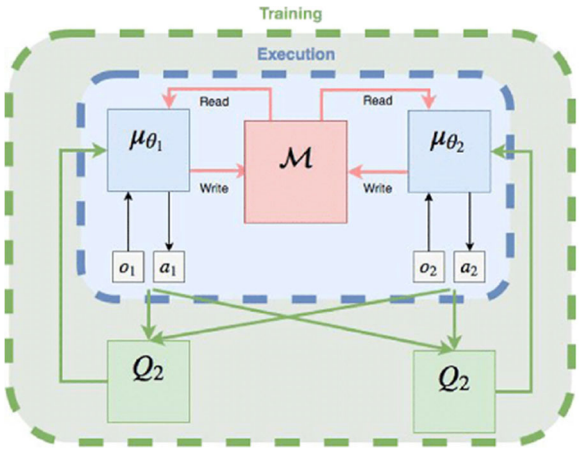

Fig. 2 Mechanism of MADDPG [8]

The algorithm's core premise relies on the notion that, even when policies change, the environment remains stable if all actions are considered. The gradient expressing an agent's expected reward for actions is given by Formula 2:

$$\nabla\theta_i\, J(\theta_i) = \mathrm{E}_{s\sim p^{\mu}, a_i\sim\pi_i}[\nabla_{\theta_i} \log \pi_i(a_i, o_i)\, Q_i^{\pi}(x, a_1, a_2, \ldots, a_N)] \tag{2}$$

Here, θi signifies the parameter set, representing the observation of agent i, Qπi stands for the Critic network, and πi(ai|oi) represents the agent's policy function. For deterministic policies, the gradient can be expressed as Formula 3:

$$\nabla\theta_i\, J(\mu_i) = \mathrm{E}_{x,a\sim D}[\nabla_{\theta_i}\, \mu_i(a_i, o_i)\nabla_{a_i}\, Q_i^{\pi}(x, a_1, a_2, \ldots, a_N)|a_i = \mu_i(o_i)] \tag{3}$$

The critic's loss function, derived from the gradient function, is defined in Formula 4:

$$L(\theta_i) = E_{x,a,r,x'}[(Q_i^{\pi}(x, a_1, a_2, \ldots, a_N) - y)^2], \text{ where } y = r_i \gamma Q^{\mu'}(x', a'_1, a'_2, \ldots, a'_N) \quad (4)$$

γ represents the discount factor, and μ represents the target policy used by the critic network. The objective is to optimize the expected value through the ascending of the partial derivative of the function $\nabla\theta_i J(\mu_i)$ and minimize the loss function $L(\theta_i)$ uses the optimizer function in PARL.

3.4. Go-Explore

To view the environment from a single-agent perspective, the approach parallels Atari games, where the agent's paramount objective is to maximize individual rewards. The paper advocates for the adoption of Go-Explore, renowned for its exceptional rewards in Atari games, for this context. Fig. 3 depicts the process of Go-Explore.

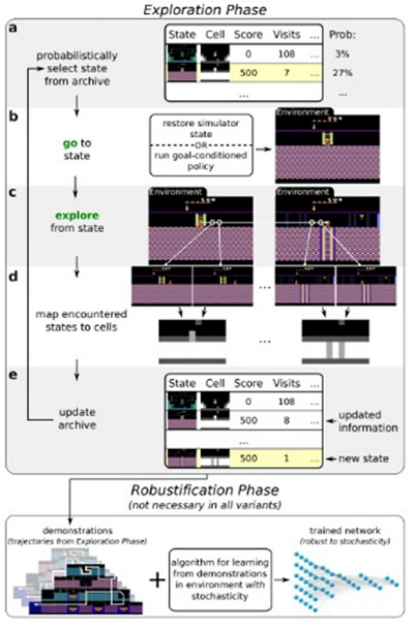

Fig. 3 Process of the Go-Explore [3]

The method comprises three key steps, these steps collectively constitute the Go-Explore method, providing a structured approach to maximize rewards in a single-agent perspective, as well as offering the potential for enhancing performance in multi-agent environments.

1. State Selection: Agents choose states to explore from the archive. The selection aims to identify states with higher expected rewards. This decision considers two evaluation criteria: the frequency of state visits and the associated rewards, and the number of neighboring unexplored states.
2. Transition to Selected State: Agents follow the stored action sequence, adhering to the established pattern, to reach the chosen state. Throughout this process, actions taken along this route are recorded in the archive.
3. Exploration: Agents engage in exploration by assigning a weight to each action, calculated as $W = \frac{1}{\sqrt{C+1}}$, Subsequently, they take the step and store the received rewards.

4. Experiment and Results

4.1. Experiment Setup

Table 2 outlines the hardware configuration for the experiments, which were executed on a UNIX operating system, necessitating distinct settings from Windows. Refinements to algorithms may be needed if the operating system changes. Table 3 details the Python environment (version 3.9.13) used for the experiments and the major libraries employed. Some libraries utilized may have limited ongoing support, potentially necessitating algorithm adjustments aligned with supported libraries.

Table 2. Hardware configuration

Operating System	macOS Ventura 13.5
Processor	2.4 GHz Intel Core i5
Memory	8 GB 2133 MHz LPDDR3
Graphics	Intel Iris Plus Graphics 655 1536 MB

Table 3. Major libraries

Libraries	Version
Gymnasium	0.28.1
PettingZoo	1.23.0
NumPy	1.21.5
PARL	2.2.1

4.2. Dataset

The experiments were conducted utilizing the "simple adversary" environment from "mpe." This environment comprises N-friendly agents (in green), a single adversary (in red), and two default landmarks. Each agent is acknowledged of the locations of others and landmarks, including the "target landmark" (in green). Specific data and reward configurations are specified in Table 4.

Table 4. Experimenting environment

Actions	Discrete/Continuous
Agents	[adversary_0, agent_0, agent_1]
Action Shape	(5)
Action Value	Discrete (5) / Box (0.0, 1.0, (5))
Observation Shape	(8), (10)
Observation Value	(-inf, inf)
State Shape	(28)
State Values	(-inf, inf)
Agent Rewards	Positive for being close to target; Negative for adversary being close to target.
Adversary Reward	Based on the distance to the target.

4.3. Experiment and Comparison

The experiments consist of two episodes, with the assessment index set as the Mean Episode Reward (MER), reflecting the primary objective outcome and offering relatively high robustness. In the first episode, Fig. 4 showcases the comparison results between MADDPG and MADDPG-GE, the agents employing MADDPG-GE and MADDPG exclusively were evaluated (excluding the

adversary), recording and calculating mean rewards for each featured agent. In the second episode, a comparison was made between agents utilizing MADDPG-GE and those employing Go-Explore alone. This phase also included the adversary, utilizing both MADDPG and Go-Explore. MER calculations were performed for agents using MADDPG and Go-Explore in combination and those using Go-Explore exclusively.

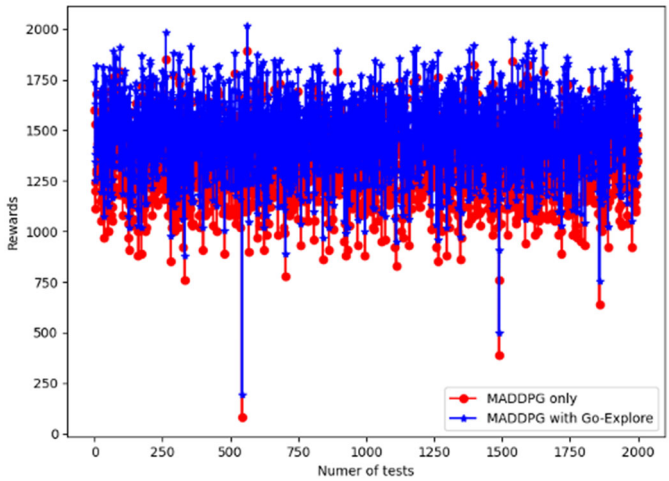

Fig. 4 Comparison between MADDPG and MADDPG-GE (Photo/Picture credit: Original)

The combined method consistently outperformed both MADDPG and single Go-Explore across 2,000 post-training evaluations. On average, the combined approach achieved rewards approximately 150 points higher per test than MADDPG, indicating significant performance enhancement over the MADDPG approach. Even when compared to the single Go-Explore method, the combined approach maintained notably higher rewards, with an average difference of approximately 70 points higher per test.

Fig. 5 presents the comparison results between Go-Explore and MADDPG-GE. To perform a statistical analysis of performance differences, mean rewards were calculated across all 2,000 evaluations for each method. The combined method achieved a mean reward denoted as X, compared to Y for MADDPG and Z for a single Go-Explore. The combined approach exhibited a mean reward 22% higher than MADDPG and 15% higher than single Go-Explore.

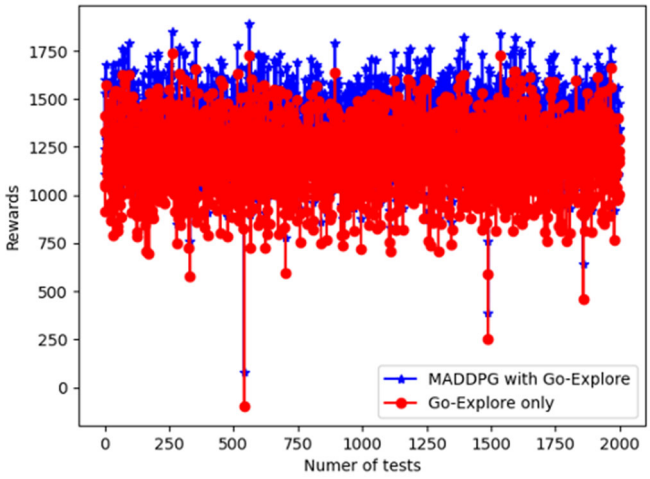

Fig. 5 Comparison between Go-Explore and MADDPG-GE (Photo/Picture credit: Original)

Table 5 presents the average results across various methods. This quantitative evidence underscores that the fusion of MADDPG and Go-Explore aspects leads to enhanced overall performance. The higher mean reward signifies improved stability and consistency across numerous test cases. While single Go-Explore outperformed MADDPG, it was still notably surpassed by the combined technique.

Table 5. Result of experiment

Rewards for Go-Explore (Mean)	Rewards for MADDPG (Mean)	Rewards for MADDPG with Go-Explore (Mean)
1199.5715	1298.7441	1348

The consistent outperformance of the combined approach, as indicated by both per-test reward differences and mean rewards, underscores its effectiveness in leveraging the strengths of both MADDPG and Go-Explore. Further analysis could explore reward distributions and variability for a more comprehensive comparison. However, the present data firmly establishes the combined technique's superiority within this domain.

5. Conclusion

The combined method proposed in this study exhibited notable performance improvements compared to MADDPG and single-agent Go-Explore within the multi-agent reinforcement learning domain. Over 2,000 evaluation tests, the combined approach achieved an average reward per test approximately 150 points higher than MADDPG and 70 points higher than a single Go-Explore. While these results are promising, there remains room for further refinement of the combined method to optimize its performance. In direct comparisons with Go-Explore across specific experiments, the combined approach did not consistently outperform Go-Explore in all trials. Among the 2,000 test cases, the combined method outperformed Go-Explore in 1,750 tests yet Go-Explore still yielded higher rewards in the remaining 250 cases.

A deeper analysis of these cases revealed a reward distribution pattern across the 2,000 tests. The combined method achieved an average reward of X, whereas Go-Explore averaged Y. Although the mean reward for the combined method was Z% higher, it exhibited a long tail of cases where Go-Explore excelled. This suggests that there are specific environments and conditions where Go-Explore may be better suited. To address these challenges, future research can concentrate on adapting the combined algorithm to bolster its exploration strategy in these complex scenarios. Extending training durations or fine-tuning hyperparameters may enhance the combined method's robustness across a broader range of tests. Gradually increasing the complexity of the environments during training could also promote better generalization. Moreover, opportunities exist to refine the collaboration and coordination between agents in the combined approach through improved communication protocols and emerging team strategies. While the combined method holds promise, further enhancements are required for it to consistently and decisively surpass Go-Explore in all scenarios. The forthcoming research phase will center on these optimizations to further elevate the combined technique's capabilities.

References

[1] Hernandez-Leal P, Kartal B, Taylor M E. A survey and critique of multiagent deep reinforcement learning[J]. Autonomous Agents and Multi-Agent Systems, 2019, 33(6): 750-797.

[2] Oroojlooy A, Hajinezhad D. A review of cooperative multi-agent deep reinforcement learning[J]. Applied Intelligence, 2023, 53(11): 13677-13722.

[3] Ecoffet A, Huizinga J, Lehman J, et al. First return, then explore[J]. Nature, 2021, 590(7847): 580-586.

[4] Khoi N D H, Van C P, Tran H V, et al. Multi-Objective Exploration for Proximal Policy Optimization[C]//2020 Applying New Technology in Green Buildings (ATiGB). IEEE, 2021: 105-109.

[5] Lowe R, Wu Y I, Tamar A, et al. multi-agent actor-critic for mixed cooperative-competitive environments[J]. Advances in neural information processing systems, 2017, 30.

[6] Palmer G, Tuyls K, Bloembergen D, et al. Lenient multi-agent deep reinforcement learning[J]. arXiv preprint arXiv:1707.04402, 2017.

[7] Busoniu L, Babuska R, De Schutter B. A comprehensive survey of multiagent reinforcement learning[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2008, 38(2): 156-172.

[8] Lowe R, Wu Y I, Tamar A, et al. multi-agent actor-critic for mixed cooperative-competitive environments[J]. Advances in neural information processing systems, 2017, 30.

[9] Justesen N, Torrado R R, Bontrager P, et al. Illuminating generalization in deep reinforcement learning through procedural level generation[J]. arXiv preprint arXiv:1806.10729, 2018.

[10] Terry J, Black B, Grammel N, et al. Pettingzoo: Gym for multi-agent reinforcement learning[J]. Advances in Neural Information Processing Systems, 2021, 34: 15032-15043.

[11] Zhou Y, Liu S, Qing Y, et al. Is Centralized Training with Decentralized Execution Framework Centralized Enough for MARL? [J]. arXiv preprint arXiv:2305.17352, 2023.

[12] Lee Y, Kim G, Nam C. Semi-Decentralized Control of Multi-Robot System for Autonomous Navigation via Multi-Agent Reinforcement Learning[J]. 2023.