Multi-Agent Reinforcement Learning for Sparse Reward Tasks using Incremental Goal Enhanced Method

Minglei Han[1,2], Zhentao Guo[1,2], Licheng Sun[1,2], Ao Ding[1,2], Tianhao Wang[1,2], Guiyu Zhao[1,2], and Hongbin Ma*[1,2]

[1] School of Automation, Beijing Institute of Technology, 100081, Beijing, China
[2] National Key Lab of Autonomous Intelligent Unmanned Systems, 100081, Beijing, China

Abstract. As the application of artificial intelligence continues to expand, complex decision-making problems such as multi-player gaming, multi-robot planning and multi-vehicle controlling have become new challenges for machine intelligence. Multi-Agent Reinforcement Learning (MARL) which concentrates on learning the optimal strategies of multiple agents that coexist in a shared environment, is a valid method to solve multi-agent decision-making challenges. Among MARL Algorithms, the MAPPO algorithm has won the favor of machine learning community due to its superb performance. However, the original MAPPO algorithm suffers from sparse reward issues. To overcome the sparse rewards problem and achieve sufficient learning in complex task, this paper proposes a IGE-MAPPO which uses a IGM that generates a variable-density and bi-domain reward signal, and conducts experiments on SMAC. The results show that the IGE-MAPPO algorithm can adapt to a variety of complex environment and has improved performance compared with other typical MARL algorithms.

Keywords: Multi-Agent Proximal Policy Optimization · Sparse Rewards · Incremental Goals

1 Introduction

As the application of artificial intelligence continues to expand, multi-agent decision-making problems such as multi-player gaming, multi-robot planning and multi-vehicle controlling have become new challenges for machine intelligence [1]. These years, with DeepMind's AlphaStar [2] surpassing professional level performance in Go and StarCraft II and OpenAI Five [3] beating world champions in Dota II, machine intelligence has made remarkable progress in handling complex tasks and effectively responded to the multi-agent decision-making challenges [4]. These milestones and successes are largely powered by *Multi-Agent Reinforcement Learning* (MARL) algorithms. MARL is a type of *Reinforcement Learning* (RL) which concentrating on learning the optimal strategies of multiple agents that coexist in a shared environment. In the MARL, the *Multi-Agent Proximal*

Policy Optimization (MAPPO) is an on-policy MARL algorithm which uses importance sampling to perform off-policy correction [5]. This algorithm maintains stable and robust learning, and performs well in cooperative tasks [6]. However, despite the success of the MAPPO, there are still some challenges when dealing with sparse reward problems [7] [8].

This paper focuses on sparse reward problem and present a *incremental goal enhanced MAPPO* (IGE-MAPPO) algorithm which deploys a *incremental goal model* (IGM) that generates a variable-density and bi-domain reward signal. This rewarding model can not only improve exploration efficiency, but also avoid sub-optimal policy from carefully shaped reward. The proposed algorithm are evaluated on *StarCraft Multi-Agent Challenge* (SMAC) [9] platform which is a benchmark problem in MARL [10]. Compared with other typical MARL algorithms, the efficiency of the IGE-MAPPO algorithm is demonstrated and verified.

2 Preliminaries

2.1 Dec-POMDP

The *decentralized partially observable markov decision process* (Dec-POMDP), a generalization of *markov decision process* (MDP), is tailored to describe a MARL system which multiple agents collaborates to accomplish tasks with only the access of their own local observation [11]. Such model can provide a more accurate description of the multi-agent team decision problems in the real world. In the Dec-POMDP [12], the key elements can be formulated as a 8-tuple $< N, S, \{b^0\}, \{A_i\}, \{O_i\}, P, R, \gamma >$ where N is the finite set of agents' indices and n is the number of agents, S is the finite set of states, $b^0 \in \Delta(S)$ is the initial state distribution, A_i is the finite set of available actions for agent i and $A = \times_{i \in N} A_i$ is the set of joint actions, where $\times$ denotes the Cartesian product operator and $\boldsymbol{a}$ represents a joint action, O_i is the finite set of observations for agent i and $O = \times_{i \in N} O_i$ is the set of joint observations, where $\boldsymbol{o}$ denotes a joint observation, P is the state transition and observation probability function, where $P(s', \boldsymbol{o} \mid s, \boldsymbol{a})$ denotes the probability which the agents take a joint action $\boldsymbol{a}$ under state s and receive a joint observation $\boldsymbol{o}$ under a new state s', R is the reward function and $R : S \times A \times S \to \mathbb{R}$ and $\boldsymbol{r}$ denotes the immediate reward of agents, γ denotes the discount factor of return. The objective of Dec-POMDP is to learn a joint policy $\pi(\boldsymbol{a} \mid \boldsymbol{o}) = \prod_{i=1}^{n} \pi_i(a_i \mid o_i)$ that maximizes the expected discounted cumulative reward $J(\boldsymbol{\pi}) = \hat{\mathbb{E}}_t \left[\sum_{t=0}^{T} \gamma^t r_t \right]$.

2.2 CTDE Paradigm

In the MARL, there are typically three learning paradigms [13]: fully centralized learning, fully decentralized learning and *centralized training with decentralized execution* (CTDE). The fully centralized learning which uses a center controller

to manage all the agents' action is suffering from the huge amount of computational demand in the central controller [14] [15]. On the other hand, the fully decentralized learning which isolates the agents to make their own decision both can't reach the global optimal of the coordinated agents [16] [17]. To solve these problems, CTDE [18] allows agents to access the global information and learn the coordinated policy in the training phase and lets agents to act based on local situation in the executing phase. Such learning framework can not only encourage the agents to learn the optimal policy, but also speed up the decision process of individual agents, which has made the CTDE a major scheme for MARL [19].

2.3 PPO

The *proximal policy optimization* (PPO) [13], the core of MAPPO, is a popular policy gradients based algorithm in deep reinforcement learning algorithm. This algorithm is built on *trust region policy optimization* (TRPO) algorithm [14] and is able to avoid large policy updates that may slow down training performance by limiting the deviation of new policies from old ones [15].

There are typically two types of PPO algorithm: PPO-Penalty and PPO-Clip [16]. Among them, PPO-Clip uses clipping technique to balance the improvement of the new strategy with maintaining stability. The objective function of PPO-Clip can be written as Eq. 1.

$$J_{PPO}^{CLIP}(\theta) = \mathbb{E}\Big[\min\left(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t\right)\Big] \tag{1}$$

where θ denotes the parameter associated with the policy network, the variable t denotes the specific time step under consideration, the function A_t denotes the advantage function at time step t, ϵ controls the degree of clipping, $r_t(\theta)$ denotes the likelihood ratio between the new policy and the old policy at time step t, as defined by Eq. 2.

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \tag{2}$$

where $\pi_\theta(a_t|s_t)$ denotes the probability of selecting action a_t in state s_t according to the new policy.

The optimization of this objective function can not only maximize the expected reward of the RL agent, but also prevent the policy from tremendous changing, which ensures the stability of learning. In the objective function, the first term encourages the policy to move in the direction of higher advantage, while the second term restricts the size of the policy update by clipping the likelihood ratio.

3 Method

The sparse reward task [17] [18] seldom gives reward feedback to the MARL agent until the mission is done, which has been a major problem in RL. With the joint action space growing exponentially in MARL, such learning problem can

pose a severe challenge to the exploration ability of MARL algorithms. MAPPO algorithm uses a clipping mechanism to prevent huge policy updating, which could suffer from the inadequate exploration and low convergence rate during a sparse reward scenario [7] [8].

3.1 Complete Algorithm Flow

The major difficulty of sparse reward scenario is that the agents cannot be motivated and guided by a goal far away from the initial state. Such scenario is hard even for humans. A direct way of solving sparse reward problem is to decompose the goal into multiple sub-goals that closer to the initial state. Following this concept, the proposed IGE-MAPPO algorithm deploys a IGM that generates a variable-density and bi-domain reward signal. This rewarding model can not only improve exploration efficiency, but also avoid sub-optimal policy from carefully shaped reward. The principle and structure diagram is shown in Figure 1. The pseudocode of the IGE-MAPPO algorithm is shown in Algorithm 1.

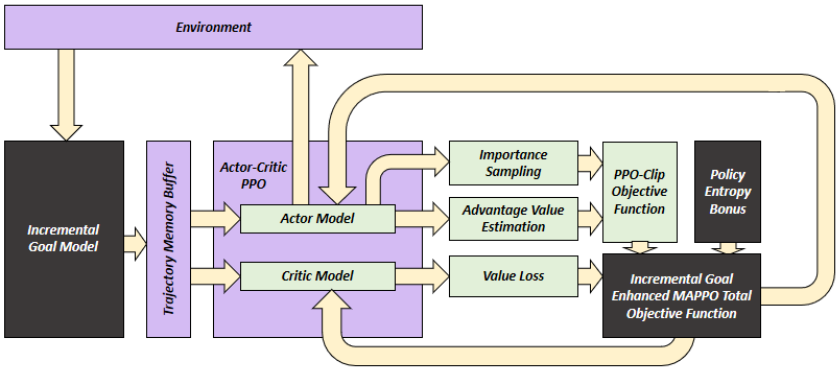

Fig. 1. Principle and structure diagram of IGE-MAPPO

Algorithm 1 IGE-MAPPO

Initialize: policy network with parameter θ and value network with parameter ϕ, maximum episodes M, maximum timesteps T, number of agents N, learning rate α, experience replay buffer D, mini-batch size K, incremental goal set IG and incremental goal numbers Q

1: **for** $i = 1$ to M **do**
2: Reset training environment;
3: Reset incremental goal set IG for each agent;
4: **for** $t = 0$ to T **do**

for $a = 0$ to N **do**
  Observe the environment o_t^a, execute the action a_t^a based on the policy $\pi_{\theta_{old}}^a$;
  Calculate the distance between the current observation and the incremental goal $d_c(o_t^a, \boldsymbol{g})$;
  for $q = 1$ to Q **do**
    if $d_c(o_t^a, \boldsymbol{g}) == d_q^a$ **then**
      Get incremental goal completeness reward $(r_t^{completeness})^a$;
      Delete d_q^a from incremental goal set IG^a;
    end if
  end for
  Get effectiveness reward $(r_t^{effectiveness})^a$;
  Get environment reward $(r_t^{environment})^a$;
  Calculate the entire reward $\widetilde{r}_t^a$ for agent a;
  Collect a transition of all agent τ_t+=$\{s_t^a, a_t^a, \widetilde{r}_t^a, s_{t+1}^a\}$;
end for
Store trajectories of all agent τ_t into buffer D;
Compute reward-to-go $\hat{R}_t$ and IGE return G_t^{IGE};
Estimate advantages $\hat{A}_t$ via GAE based on both the value function and the IGE value function $V_\pi^{IGE}(s)$;
for $k = 1$ to K **do**
  Random sample a transition data τ_r of all agent from buffer D;
  Optimize $J_{IGE-MAPPO}^{total}$ using stochastic gradient ascent;
end for
Update θ and ϕ;
end for
end for

3.2 Incremental Goal Model

The IGM, the core of IGE-MAPPO, is designed to provide extra bi-dimensional rewards in the sparse reward tasks based on the incremental goals. To characterize incremental goals in a certain task, one has to define what the goal is. In MARL, the goal of learning agents represents a family of global state vectors $\boldsymbol{s}$ that satisfy certain conditions. In these vectors, there are 3 type of terms: the term $\boldsymbol{s_C}$ that directly determine the completeness of a task, the term $\boldsymbol{s_E}$ that directly affect the efficiency of a task and the term $\boldsymbol{s_I}$ that indirectly affect the task. Under such setting, the definition of incremental goals in this paper can be characterized as a series of global state vectors $\boldsymbol{s}$ that have certain distance d_c with the goal, which is described as follows:

Definition 1. *The incremental goals of MARL agents in a task is a set of global state vectors that satisfy*

$$\boldsymbol{ig} := \{\boldsymbol{s}(\boldsymbol{s_C}, \boldsymbol{s_E}, \boldsymbol{s_I}) \mid d_c(\boldsymbol{s}, \boldsymbol{g}) \in IG, IG = \{d_1, d_2, ..., d_Q\}\} \tag{3}$$

$$d_c(\boldsymbol{s_{initial}}, \boldsymbol{g}) = \sum_{q=0}^{Q} x^q = \underbrace{x^0}_{d_Q} + \underbrace{x^1}_{d_{Q-1}} + ... + \underbrace{x^Q}_{d_1} \tag{4}$$

where $d(\cdot,\cdot)$ represents the L_2 norm of the difference between certain component of two global state vectors, $d_c(\cdot,\cdot)$ denotes the L_2 norm based on the $\boldsymbol{s_C}$ component, Q is the number of incremental goals, and IG is the distance set used to control the density of incremental goals.

After the incremental goals being defined, multiple incremental goals can be generated to guide the learning process of MARL agents. In order to prevent the MARL agents from local optimal caused by incremental goals, the distance value from IG are generated using a geometrical progression which can create incremental goals with decreasing density. This decreasing density setting of subgoals performs a "sports-training" style which provides a intensive guidance to the agents in the early stage and gives adequate exploration to the agents in the later stage of learning.

Following the incremental goals produced above, the learning rewards can be presented in a bi-dimensional view which includes the completeness of the task and the effectiveness of the task. In the completeness view, positive rewards $r^{completeness}$ are given whenever the agents achieve a incremental goal or the final goal in the whole learning process. In the effectiveness view, negative rewards $r^{effectiveness}$ are given to encourage agents' exploration in the late training stage. Concretely, the bi-dimensional rewards can be calculated through Eq. 5-7:

$$r_t^{IGE} = r_t^{completeness} + r_t^{effectiveness} \tag{5}$$

$$r_t^{completeness} = e_1 Q, \quad \text{if } d_c(\boldsymbol{s_t}, \boldsymbol{subg_q}) \leqslant \delta \tag{6}$$

$$r_t^{effectiveness} = -e_2 d_e(\boldsymbol{s_t}, \boldsymbol{s_{initial}}), \quad \text{if } t \geqslant T/2 \tag{7}$$

where $r^{completeness}$ denotes the rewards of task completeness, $r^{effectiveness}$ represents the rewards of task effectiveness, T is the total steps of training episode, e_1 and e_2 are the .

After the reward r^{IGE} is defined, the entire rewards $\widetilde{r}$ that the agents can receive is determined by $\widetilde{r}_t = r_t^{environment} + r_t^{IGE}$, the total discounted return of agents and the value function based on certain policy π can be calculated through Eq. 8 and Eq. 9.

$$G_t^{IGE} = \sum_{k=0}^{T-t-1} \gamma^k \widetilde{r}_{t+k+1} \tag{8}$$

$$V_\pi^{IGE} = \mathbb{E}\left[G_t^{IGE} \mid \boldsymbol{s_t} = \boldsymbol{s}\right] \tag{9}$$

where G_t^{IGE} represents the discount cumulative rewards from step $t+1$ to the final step, γ is the discount factor and $\gamma \in [0, 1]$).

In order to perform proper training efficiency and ensure adequate exploration, a total IGE-MAPPO objective function is formulated and aimed to be

maximized. Such a objective function can not only concerns the policy objective and the value loss, but also deploys a policy entropy bonus, which is determined by Eq. 10 and Eq. 11.

$$(J_{\phi}^{Value})^{(i)} = (V_{\phi}(s^{(i)}) - (V_{target})^{(i)})^2 \tag{10}$$

$$J_{IGE-MAPPO}^{total}(\theta) = \sum_{i=1}^{N} \mathbb{E}\left[(J_{PPO}^{Clip}(\theta))^{(i)} - c_1(J_{\phi}^{Value})^{(i)} + c_2 H(s^{(i)}, \pi_{\theta})\right] \tag{11}$$

where N is the number of agents, $(V_{target})^{(i)} = V^{IGE}(s^{(i)}) + \hat{A}^{(i)}$ represents value target which is the sum of IGE value and advantage value of each agent, $H(s^{(i)}, \pi_{\theta})$ is the policy entropy regularization term, c_1 and c_2 are coefficients hyperparameters.

4 Experiments

4.1 Environment

SMAC [9] is a MARL test bed based on the famous real-time strategy game StarCraft II, integrating various team fighting scenes where units are manipulated by individual agents that act with their local observations. These scenes naturally satisfy the Dec-POMDP properties and the CTDE paradigm, which poses cooperative challenges to the multi-agent learning systems. Due to the representativeness of these scenario, SMAC has been regarded as a universal benchmark problem to test and evaluate the state-of-art algorithms in the MARL community. According to the configuration of SMAC, scenarios are divided into 3 categories: easy, hard and super-hard, while the fairness and the types of controlled units can also be set. In this paper, the experiment focuses on the most challenging scenarios that are heterogeneous and asymmetric, and have difficulties of hard and super-hard. Additionally, the reward using in the training process is sparse reward setting. Specifically, the experimental scenarios chosen in this paper are 5m_vs_6m, 27m_vs_30m, MMM2, and 3s5z_vs_3s6z, as shown in Figure 2 (a) (b) (c) (d). The configuration like ally units, enemy units and type of the chosen scenarios are listed in Table 1. The specific scenario settings are presented in Table 2

Table 1. The configuration of the chosen scenarios in SMAC

Scenario Name	Ally Units	Enemy Units	Type
5m_vs_6m	5 Marines	6 Marines	homogeneous&asymmetric
27m_vs_30m	27 Marines	30 Marines	homogeneous&asymmetric
MMM2	1 Medivac, 2 Marauders & 7 Marines	1 Medivac, 3 Marauders & 8 Marines	heterogeneous&asymmetric
3s5z_vs_3s6z	3 Stalkers & 5 Zealots	3 Stalkers & 6 Zealots	heterogeneous&asymmetric

Fig. 2. Visualization of different types of multi-agent environment. (a) (b) (c) (d) are the 5m_vs_6m scenario, 27m_vs_30m scenario, MMM2 scenario, and 3s5z_vs_3s6z scenario in SMAC, respectively.

Table 2. The settings of the chosen scenarios

Scenario Name	Number of Actions	Number of Agents	State Dimension	Observation Dimension	Time-step limit
5m_vs_6m	12	5	98	55	70
27m_vs_30m	36	27	526	285	180
MMM2	16	10	246	110	180
3s5z_vs_3s6z	15	8	227	136	170

4.2 Performance Evaluation

Due to the representativeness and effectiveness [19] [20], this paper chose MAPPO, *Independent PPO* (IPPO) [21] and *Multi-Agent Deep Deterministic Policy Gradient* (MADDPG) [22] as benchmark algorithms for comparative experiment. MAPPO is a popular on-policy MARL algorithm that takes the advantages of importance sampling method and perform well in both continuous and discrete action spaces. IPPO is a simple on-policy MARL algorithm and a decentralized variant of PPO where each agent independently optimizes its own policy without explicit communication or coordination. Such a MARL algorithm can handle both discrete and continuous action spaces. MADDPG is a famous off-policy MARL algorithm which deploys a centralized Q-function taking observations and actions from all the agents to alleviate the non-stationarity issue and stabilize multi-agent training. Specially, in order to make MADDPG solve discrete tasks, this paper uses a MADDPG with softmax output settings. For each algorithm, this paper uses recommended hyperparameters, and trains the MARL agents parallelly. The performance data of algorithms for the selected tasks are provided in Figure 3, which can presents a clear view on the efficiency of each algorithms across different challenges.

In Figure 3, MAPPO achieved a proper performance on all tasks, providing a relatively stable learning process on all the tasks and converging to good policies in all challenges. However, MAPPO had a low convergence rate and suffered from high computational overhead due to its frequent sampling of the environment. IPPO showed some convergence, but its efficiency was limited due to the lack of effective communication and collaboration among agents, especially in the environment with a large number of controlled agents. MADDPG with softmax outputs showed some performance and achieved acceptable results, but had an unstable learning behavior and training process, and act poorly in super hard

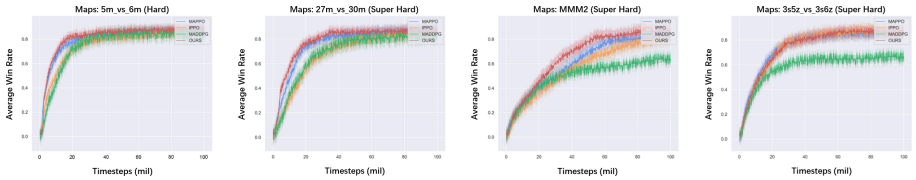

Fig. 3. The figure showcases the training results of multiple algorithms across various multi-agent environment, including MAPPO, IPPO, MADDPG, and PG-MAPPO.

tasks. The problem of slow convergence rate and low peak reward in large-scale environment within MADDPG were also exposed in the experiment. In comparison,

From the performance data, it can be seen that the proposed IGE-MAPPO has a explicit advantage over existing MARL aLgorithms, and its performance and efficiency surpasses the others in multiple aspects. Specifically, IGE-MAPPO presents faster convergence rates and achieves the average win rates of 82%, 84% and 85% in super hard tasks. These results shows the remarkable learning ability in handling large-scale multi-agent coordinated tasks.

5 Conclusions

This paper improves the MAPPO algorithm's capabilities of overcoming sparse reward problem, adds a IGM which generates a variable-density and bi-domain reward signal to the typical MAPPO algorithm, introduces hyperparameters in the cumulative return and the value function of each agent training process and the upper and lower bounds of clipping function. Further more, the policy objective, value loss and the policy entropy bonus is integrated to the IGE-MAPPO total objective function. In this paper, the efficiency of the IGE-MAPPO algorithm is verified by different virtual environment. The simulation results show that the IGE-MAPPO algorithm has strong learning ability and strong generalization ability, and can well complete the tasks in various environment. The convergence and stability of the algorithm are outstanding. In summary, this paper proposes a model that can guide the MAPPO to overcome sparse reward problems and improve training efficiency.

Acknowledgements

This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0101000 and the National Natural Science Foundation of China under Grant 62076028.

Declarations

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. X. Zhang, Y. Liu, H. Mao, and C. Yu, "Common belief multi-agent reinforcement learning based on variational recurrent models," *Neurocomputing*, vol. 513, pp. 341–350, 2022.
2. O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev *et al.*, "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.
3. OpenAI, :, C. Berner, G. Brockman, B. Chan, V. Cheung, P. Dbiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, R. Józefowicz, S. Gray, C. Olsson, J. Pachocki, M. Petrov, H. P. d. O. Pinto, J. Raiman, T. Salimans, J. Schlatter, J. Schneider, S. Sidor, I. Sutskever, J. Tang, F. Wolski, and S. Zhang, "Dota 2 with large scale deep reinforcement learning," 2019.
4. Z. Ning and L. Xie, "A survey on multi-agent reinforcement learning and its application," *Journal of Automation and Intelligence*, vol. 3, no. 2, pp. 73–91, 2024.
5. C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of ppo in cooperative multi-agent games," in *Advances in Neural Information Processing Systems 35 - 36th Conference on Neural Information Processing Systems, NeurIPS 2022*, ser. Advances in Neural Information Processing Systems. Neural information processing systems foundation, 2022.
6. H. Kang, X. Chang, J. Mii, V. B. Mii, J. Fan, and Y. Liu, "Cooperative uav resource allocation and task offloading in hierarchical aerial computing systems: A mappo-based approach," *IEEE Internet of Things Journal*, vol. 10, no. 12, pp. 10 497–10 509, 2023.
7. Wang, L., Zhang, Y., Hu, Y., Wang, W., Zhang, C., Gao, Y., Hao, J., Lv, T. & Fan, C. Individual Reward Assisted Multi-Agent Reinforcement Learning. *Proceedings Of The 39th International Conference On Machine Learning*. **162** pp. 23417-23432 (2022,7,17)
8. Liu, Z., Wan, L., Yang, X., Chen, Z., Chen, X. & Lan, X. Imagine, Initialize, and Explore: An Effective Exploration Method in Multi-Agent Reinforcement Learning. *Proceedings Of The AAAI Conference On Artificial Intelligence*. **38**, 17487-17495 (2024)
9. M. Samvelyan, T. Rashid, C. S. de Witt, G. Farquhar, N. Nardelli, T. G. J. Rudner, C. Hung, P. H. S. Torr, J. N. Foerster, and S. Whiteson, "The starcraft multi-agent challenge," *CoRR*, vol. abs/1902.04043, 2019.
10. W. Du and S. Ding, "A survey on multi-agent deep reinforcement learning: from the perspective of challenges and applications," *Artificial Intelligence Review*, vol. 54, no. 5, pp. 3215–3238, 2021.
11. L. Canese, G. C. Cardarilli, L. Di Nunzio, R. Fazzolari, D. Giardino, M. Re, and S. Spanò, "Multi-agent reinforcement learning: A review of challenges and applications," *Applied Sciences*, vol. 11, no. 11, p. 4948, 2021.

12. "Decentralized graph-based multi-agent reinforcement learning using reward machines," *Neurocomputing*, vol. 564, p. 126974, 2024.
13. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
14. J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 1889–1897.
15. Y. Wang, H. He, X. Tan, and Y. Gan, "Trust region-guided proximal policy optimization," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
16. N.-C. Huang, P.-C. Hsieh, K.-H. Ho, and I.-C. Wu, "Ppo-clip attains global optimality: Towards deeper understandings of clipping," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 11, 2024, pp. 12 600–12 607.
17. B. Liu, Z. Pu, Y. Pan, J. Yi, Y. Liang, and D. Zhang, "Lazy agents: A new perspective on solving sparse reward problem in multi-agent reinforcement learning," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 23–29 Jul 2023, pp. 21 937–21 950.
18. S. Chen, Z. Zhang, Y. Yang, and Y. Du, "Stas: Spatial-temporal return decomposition for solving sparse rewards problems in multi-agent reinforcement learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 17 337–17 345, 03 2024.
19. X. Pan, M. Liu, F. Zhong, Y. Yang, S.-C. Zhu, and Y. Wang, "Mate: Benchmarking multi-agent reinforcement learning in distributed target coverage control," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 862–27 879, 2022.
20. B. Ellis, J. Cook, S. Moalla, M. Samvelyan, M. Sun, A. Mahajan, J. Foerster, and S. Whiteson, "Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
21. C. S. D. Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. S. Torr, M. Sun, and S. Whiteson, "Is independent learning all you need in the starcraft multi-agent challenge?" *ArXiv*, vol. abs/2011.09533, 2020.
22. R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Curran Associates Inc., 2017, p. 63826393.