Knowledge Transfer from Situation Evaluation to Multi-agent Reinforcement Learning

Min Chen[1,2], Zhiqiang Pu[2(✉)], Yi Pan[2], and Jianqiang Yi[2]

[1] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[2] Institute of Automation, Chinese Academy of Sciences, Beijing, China
chenmin161@mails.ucas.ac.cn

Abstract. Recently, multi-agent reinforcement learning (MARL) has achieved amazing performance on complex tasks. However, it still suffers from challenges of sparse rewards and contradiction between consistent cognition and policy diversity. In this paper, we propose novel methods for transferring knowledge from situation evaluation task to MARL task. Specifically, we utilize offline data from a single-agent scenario to train two situation evaluation models for: (1) constructing guiding dense rewards (GDR) in multi-agent scenarios to help agents explore real sparse rewards faster and jump out of locally optimal policies without changing the global optimal policy; (2) transferring a situation comprehension network (SCN) to multi-agent scenarios that balances the contradiction between consistent cognition and policy diversity among agents. Our methods can be easily combined with existing MARL methods. Empirical results show that our methods achieve state-of-the-art performance on Google Research Football which brings together above challenges.

Keywords: Multi-agent reinforcement learning · Transfer learning · Football

1 Introduction

Deep reinforcement learning (DRL) has achieved super-human performance on complex games [2,16,28]. However, in the multi-agent setting, task complexity grows exponentially with the number of agents. In addition, some tasks have only sparse rewards and require agents to emerge diversity while cooperating. The above challenges make it difficult for agents to learn a robust and satisfactory policy.

For the challenge of sparse rewards, dense rewards are designed with three major ways to guide learning process of agents. Human knowledge rewards [13,25] give agents a numerical reward when they complete sub-goals that are helpful for the original task in terms of designers' experience. Intrinsic rewards [3,30,35]

M. Tanveer et al. (Eds.): ICONIP 2022, CCIS 1791, pp. 3–14, 2023.
https://doi.org/10.1007/978-981-99-1639-9_1

encourage agents to perform certain characteristics, such as curiosity and diversity. However, these two ways of reward shaping will change the optimal policy of agents. [17] proposes potential-based reward shaping (PBRS) that guarantees the optimal policy unchanging in single-agent setting. [31] additionally considers the effect of agent actions on PBRS and [7] considers dynamic potential situation. [9] gives a method to translate an arbitrary reward function into PBRS. [6] extends PBRS to multi-agent setting and proves the Nash Equilibria of underlying stochastic game is not modified. However, the above three reward shaping methods require fine-tuning hyper-parameters, otherwise they possibly bring worse results. A commonly adopted solution is to consider the hyper-parameter adjustment of reward shaping as the second optimization objective in addition to the reinforcement learning task. [11] uses population optimization to learn the optimal human knowledge reward size by assigning different parameters to agents and improving the optimal reward iteratively. [34] proposes a scalable meta-gradient framework for learning useful intrinsic reward functions. [10] formulates the utilization of PBRS as a bi-level optimization problem to adjust the weight of various rewards. However, the above methods are difficult to deploy on complex multi-agent tasks due to the exponentially growing complexity.

As for the contradiction between consistent cognition and policy diversity, policy decentralization with shared parameters (PDSP) has been widely used [15,26] to speed up training process and endow agents with consistent cognition of their task. However, it also brings the difficulty of emerging personalities of agents. Some approaches stimulate player roles [30], and diversity [4] by introducing intrinsic rewards. As mentioned above, these approaches changes the original optimization objective, which results in agents less being eager to complete their original tasks.

Transfer learning is to study how to utilize the knowledge learned from source domain to improve the performance on target domain. Since deep learning has strong representation learning capability and transferability, many achievements are obtained in supervised learning [5,20,23,27]. As for RL tasks, they are also expected to benefit from related source tasks to improve the performance of agents.

Based on the above discussion, for the challenges of sparse rewards and contradiction between consistent cognition and policy diversity in MARL, we transfer the knowledge learned from offline dataset of a single-agent scenario to multi-agent scenarios. Specifically, we (1) construct guiding dense rewards (GDR) in multi-agent scenarios to help agents explore real sparse rewards faster and jump out of locally optimal policies without changing the global optimal policy; (2) transfer a situation comprehension network (SCN) to balances the contradiction between consistent cognition and policy diversity among agents. Empirical results show that our methods achieve state-of-the-art performance on Google Research Football (GRF) [12] which brings together the above challenges.

2 Related Works

The advanced deep MARL approaches include value-based [21,24,29] algorithms and policy-gradient-based [14,33] algorithms. Theoretically, our methods can be

combined with any of above approaches, we choose to validate our methods on MAPPO because of its better performance in solving sparse reward task than others.

Though the constructing of GDR is close to the idea of inverse reinforcement learning (IRL) [1,8,19] for both constructing dense rewards from offline data, GDR has fundamental difference with IRL. The goal of IRL is to imitate the offline policy. It always assumes that the offline policy is already optimal. We still maintain the original task goal and do not have hard requirements on the performance of offline policies. The transfer of the SCN is similar to the pre-training and fine-tuning paradigm [32]. They both reuse the network from upstream tasks in downstream tasks.

3 Background

Multi-agent reinforcement learning can be formulated as a Dec-POMDP [18], which is defined as a tuple $\langle N, S, A, P, R, O, \Omega, n, \gamma \rangle$. At each step, environment is characterized as $s \epsilon S$. The set of n agents N receives the local observation $\mathbf{o} \epsilon \Omega$ according to the observation function O, then agents choose the joint action $\mathbf{a} \epsilon A$ according to their joint policy π. The environment transfers to a new state $s^{'}$ according to the state transition function P and gives agents a reward $r \epsilon R$. The task of agents is to learn the optimal joint policy to maximum the accumulated reward $E\left(\sum_{t=0} \gamma^t r_t\right)$ with discount factor γ.

4 Methods

In this section, we will introduce the details of GDR and SCN. We illustrate and verify our methods with a football game. It needs to be emphasized that, our methods are also applicable for other tasks with sparse reward character and requiring cooperation and diversity among agents.

4.1 Situation Evaluation Task

The situation evaluation task is defined as:

Given: A vector representing the game state s_t.

Do: Estimate value $\phi(s_t)$: the value of s_t

Once the form of s_t, $\phi(s_t)$ and learning algorithm are determined, a corresponding model will be obtained. To collect data for this task, we train an agent with RL algorithm PPO [22] for single-agent 11 vs.11 scenario and test it with easy baseline, hard baseline and self to collect 3000 episodes respectively for each team combination. For each episode, we record all raw observation, which is composed of:

- `player information`: the two-dimensional position, two-dimensional speed, fatigue value, and role of all players in the game.
- `ball information`: the position, speed, rotation speed, and ball possession which indicates the team and player controlling the ball.

– `game information`: the score, remaining time, and game mode which includes normal, kickoff, goal-kick, free-kick, corner, throw-in, and penalty.

where the score information is used to label samples and other information can be combined as s_t.

4.2 Construction of Guiding Dense Reward

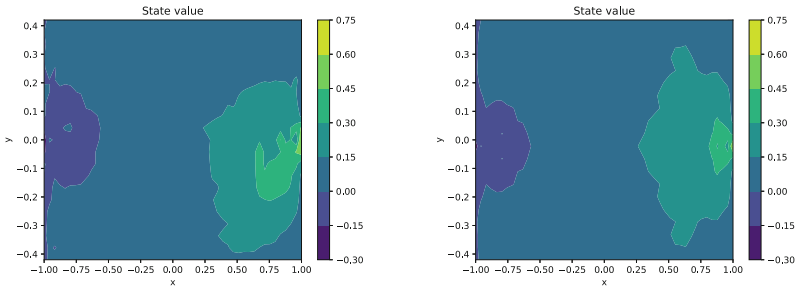
The first situation evaluation model is used to construct guiding dense reward. Firstly, we label samples as:

$$\phi\left(s_t\right) = \gamma^{t-t_g} Sign\left(t_g\right) \tag{1}$$

where t_g represent the most recent score time after t, $Sign\left(t_g\right)$ is the signature function. If it is a home team score at t_g, $Sign\left(t_g\right)$ will be 1, otherwise it will be -1. That means, the absolute value of a game state is positively related the number of steps required to scoring. Then we choose two-dimensional position of ball to represent game state s_t and divide the pitch into grids to calculate the mean value of each grid as its expected state value:

$$\phi\left(G\right) = \frac{1}{n} \sum_{s_t \epsilon G} \phi\left(s_t\right) \tag{2}$$

where G is the grid, and n is the number of samples in G. In addition, it can be seen in Fig. 1 that we make the situation evaluation symmetrical in horizontal direction, because the agent of offline data excels at scoring in the bottom half of the pitch, which is agent's biased knowledge that needs to be removed.



(a) Unsymmetrical state value. (b) Symmetrical state value.

Fig. 1. State value visualization (the boundary of GRF is equal to the limitation of axis). The color of each point represents the state value of dribbling the ball there.

At each step, agents will obtain a PBRS reward $F\left(s_t, s_{t+1}\right)$ for transferring game state from s_t to s_{t+1}:

$$F\left(s_t, s_{t+1}\right) = \gamma\phi\left(G^{'}\right) - \phi\left(G\right) \tag{3}$$

where s_t represent the game state at time t, $\phi\left(s_t\right)$ the potential of s_t, γ the discount factor, G and $G^{'}$ the grids that s_t and s_{t+1} in.

4.3 Transfer of Situation Comprehension Network

The second situation evaluation model is used to transfer situation comprehension network. The input of model s_t concludes the position, velocity and tired factor of 22 players; the position, velocity and rotation of ball; ball possession, scores of two teams and game mode. The model contains five fully connected layers and the output is the probability of scoring before the changing of ball possession.

We divided the observations of agents into two parts. One part is the information related to game situation, which is the same as the input of the situation evaluation network above mentioned, and the other part is private information. The transfer method of deep network structure among similar tasks has been widely used, which benefits from the strong representation learning ability of deep network. Much researches claim that the output of low layers of deep network is general semantics, which of medium layers is hidden variable, and the output of high layers is specific semantics. That means if the situation evaluation network is properly extracted, a situation comprehension network that processes the raw situation observation into situation comprehension information can be transferred to RL task. Hence, we make agents share the situation comprehension network (the first several layers of situation evaluation network). Once agents obtain their raw observation, the situation comprehension information given by SCN and private information will be spliced as the new observation. Then agents train their actor separately to balance the contradiction between consistent cognition and policy diversity among them.

5 Experiments

In this section, we will validate our methods on multi-agent scenarios of GRF to illustrate the effectiveness of GDR and SCN.

5.1 Performance on Google Research Football

We validate our methods on following seven multi-agent scenarios: (a) academy 3 vs 1 with keeper, (b) academy pass and shoot with keeper, (c) academy run pass and shoot with keeper, (d) academy counterattack hard, (e) academy corner, (f) 5 vs 5, (g) 11 vs 11 easy stochastic (we control four agents and the other seven agents are controlled by built-in rules of GRF). In (a)–(e), agents need to learn offensive skills that score a goal quickly in a short time span. In (f) and (g), agents need to learn offensive and defensive skills that score more goals than opponents in a whole match. To verify the effectiveness of our methods, we experiment and compare the following four methods, in which SCN is only used in the scenarios with 22 players on the pitch (d, e, g):

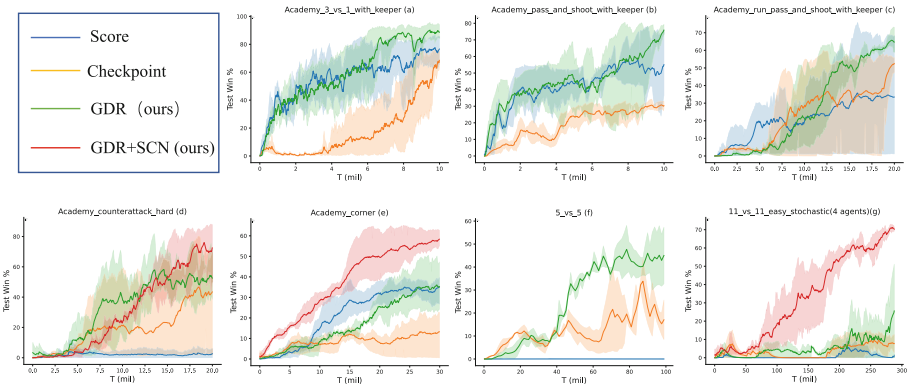


Fig. 2. Comparison of our approach against baseline algorithms on Google Research Football.

Score: Agents obtain a sparse reward +1 when scoring a goal and −1 when opponents scoring.

Checkpoint: Checkpoint is built-in human knowledge reward of GRF, it divides pitch into 10 zones based on the Euclidean distance from the goal. Agents will obtain a 0.1 reward when they possess the ball for the first time in each zone to encourage them to run towards opponents' goal with the ball.

GDR: Agents obtain a dense reward that constructed according to Sect. 4.2 at each step.

GDR+SCN: Agents share SCN to process situation information and obtain GDR at each step.

As can be seen in Fig. 2, the winning rate of our methods significantly higher than which of baselines (Score and Checkpoint) in all scenarios.

For Score, agents tend to fall into the first local optimal policy on account of sparse reward in simple tasks (a, b, c, e), because agents find that sparse rewards can only be obtained by adopting the local optimal policy. Without the guidance of dense reward, agents will gradually strengthen the local optimal policy and eventually converge. In complex tasks (d, f), agents hardly continuously sample good actions to explore sparse rewards. In these cases, most of samples are invalid. As for Checkpoint, it plays a guiding role in some scenarios (d, f, g). However, it changes the optimization objective, which leads to ordinary convergent policies. Moreover, due to the bias and incompleteness of human knowledge, in some scenarios (a, b, e), Checkpoint even hinders the learning process of agents.

The role of GDR can be summarized as: (1) helping agents jump out of random local optimal policies in simple tasks (a, b, c, e); (2) gradually guiding agents to explore real sparse rewards, then continuously stimulate agents to explore better policies in difficult tasks (d, f, g). As showed in Fig. 3, in the academy pass and shoot with keeper, which is a simple scenario, the learning process of Score-agents and GDR-agents is divided into two stage. In the

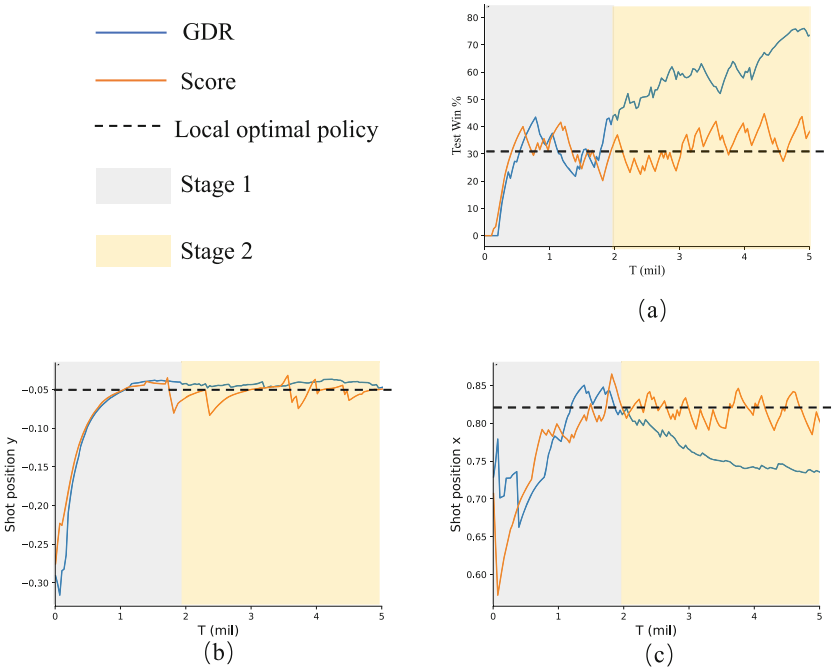


Fig. 3. The test results of the same seed trained with Score and GDR in academy pass and shot with keeper. (a) the curve of winning rate over time. (b), (c) the curve of average shot position over time.

first stage, the Score-agents obtain sparse rewards through randomly sampling actions. At the end of the first stage, the Score-agents find a local optimal policy of 30% test winning rate by strengthening the state-action pairs that obtain real sparse rewards. The effect of GDR on agents is approximately equivalent to specifying different paths for agents to obtain sparse rewards. Therefore, unlike the Score-agents, GDR-agents learn to approximate GDR with their critics, instead of entirely relying on randomly sampling actions when exploring sparse rewards. Hence, it appears that both agents learn the same policy at the end of the first stage. In reality, GDR-agents collect much more valid samples than Score-agents. In the second stage, GDR-agents quickly jump out of the local optimal policy, while Score-agents fall into the local optimal policy. Academy counterattack hard is a typical difficult scenario, which requires more steps and cooperation among players than easy tasks. For Score-agents, it is difficult to explore real sparse rewards by randomly sampling actions. Therefore, it can be seen in Fig. 4, the winning rate of Score-agents is less than 10% during the entire learning process. As for GDR-agents, their learning process is naturally divided into two stages under the guidance of GDR. In the first stage, GDR-agents mainly focus on dribbling skills to pursue higher dense rewards. They consciously learn to approach opponents' goal. Although shot is the key action of this task, GDR-agents

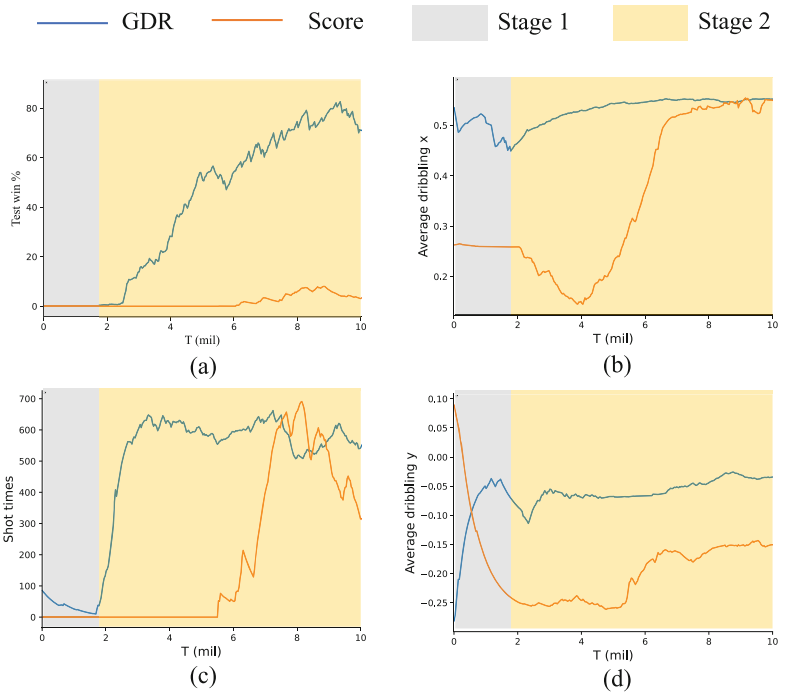


Fig. 4. The test results of the same seed trained with Score and GDR in academy counterattack hard. (a) the curve of winning rate over time. (b), (d) the curve of average dribbling position over time. (c) the curve of the frequency of performing shot action over time. The shot action is blocked when agents are much far from the goal. Score-agents do not perform shot at the beginning of training, because they are always far from the goal.

gradually weaken the frequency of performing shot action. They find that the positions where they dribble the ball are not good enough to obtain real sparse rewards, but make the ball out of control and lead to lower dense rewards. At the end of the first stage, GDR-agents have found good shot positions, where they possibly obtain real sparse rewards when performing shot action. Hence, the frequency of performing shot action increases sharply. In the second stage, GDR-agents mainly focus on shot skills to pursue higher sparse rewards. The winning rate increases rapidly.

5.2 Parametric Sensitivity of GDR

We study the parameter sensitivity of GDR in two scenarios to illustrate that GDR does not require precisely adjusting its parameter. It can be seen in Fig. 5 that regardless of the size of GDR, the winning rate will not be significantly

affected. The relative size of potentials in different states is more important than absolute size. The role of GDR is to guide agents from lower potential states to higher ones. The parameter only affects the absolute value, while the relative relationship is learned from offline data.

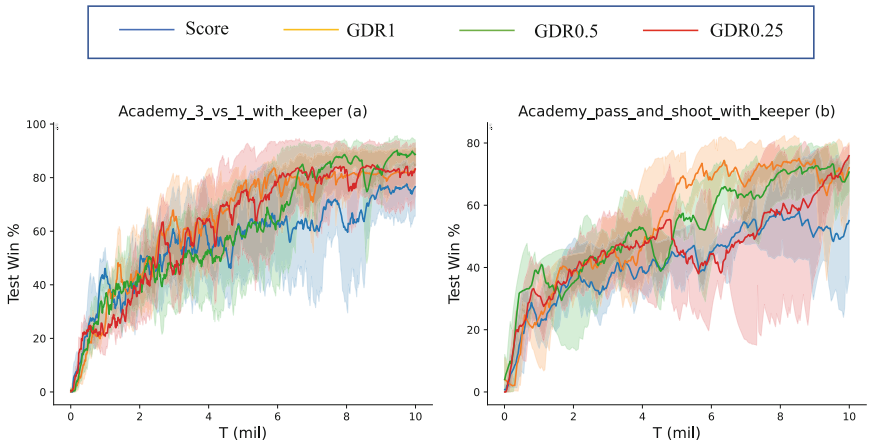


Fig. 5. We train agents respectively with 1, 0.5 and 0.25 times GDR in (a) academy 3 vs 1 with keeper and (b) academy pass and shoot with keeper, and contrast with Score-agents.

5.3 Transfer Layers Study

As can be seen in Fig. 6, when one or two layers are transferred, the performance in both academy counterattack hard and academy corner scenarios is better than no transferring, because the low layers of deep networks is to learn general features. On the one hand, these features have deeper semantic information than raw observation. On the other hand, they are not limited to specific task. Therefore, the transfer of low layers is effective and universal. However, for deep layers, the third and fourth layers in our task, learn abstract features related to specific task. When transferring three or four layers, agents perform exceptionally well in academy corner scenario, but poorly in academy counterattack scenario. The former is a fast-paced task. Hence the state space is smaller than the last. The deep-layer features of the situation evaluation network happen to suitable for solving this specific MARL task. As for counterattack, it is difficult for these features to remain unbiased in such a large state space.

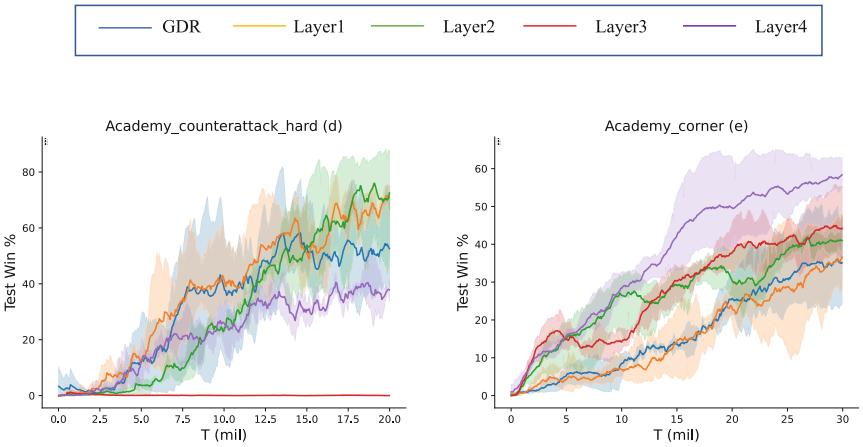


Fig. 6. We respectively transfer 1, 2, 3, 4 layers from situation evaluation network as situation comprehension network of GDR-agents in (a) academy counterattack hard and (b) academy corner, and contrast with GDR-agents.

6 Conclusion

In this paper, we study viable solutions to the challenges of (1) sparse rewards and (2) contradiction between consistent cognition and policy diversity in MARL. On the one hand, the output of situation evaluation models can be utilized to construct guiding dense reward. On the other hand, situation evaluation network can be transferred as situation comprehension network to MARL task. Our method can be easily combined with existing MARL methods.

Acknowledgment. This work was supported by the National Key Research and Development Program of China under Grant 2020AAA0103404 and the National Natural Science Foundation of China under Grant 62073323.

References

1. Arora, S., Doshi, P.: A survey of inverse reinforcement learning: challenges, methods and progress. Artif. Intell. **297**(C), 103500 (2021)
2. Berner, C., et al.: Dota 2 with large scale deep reinforcement learning. arXiv preprint arXiv:1912.06680 (2019)
3. Burda, Y., Edwards, H., Pathak, D., Storkey, A., Darrell, T., Efros, A.A.: Large-scale study of curiosity-driven learning. In: International Conference on Learning Representations (2018)
4. Chenghao, L., Wang, T., Wu, C., Zhao, Q., Yang, J., Zhang, C.: Celebrating diversity in shared multi-agent reinforcement learning. In: Advances in Neural Information Processing Systems, vol. 34 (2021)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

6. Devlin, S., Kudenko, D.: Theoretical considerations of potential-based reward shaping for multi-agent systems. In: The 10th International Conference on Autonomous Agents and Multiagent Systems. pp. 225–232. ACM (2011)
7. Devlin, S.M., Kudenko, D.: Dynamic potential-based reward shaping. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, pp. 433–440. IFAAMAS (2012)
8. Finn, C., Levine, S., Abbeel, P.: Guided cost learning: deep inverse optimal control via policy optimization. In: International Conference on Machine Learning, pp. 49–58. PMLR (2016)
9. Harutyunyan, A., Devlin, S., Vrancx, P., Nowé, A.: Expressing arbitrary reward functions as potential-based advice. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29 (2015)
10. Hu, Y., et al.: Learning to utilize shaping rewards: a new approach of reward shaping. Adv. Neural Inf. Process. Syst. **33**, 15931–15941 (2020)
11. Jaderberg, M., et al.: Human-level performance in 3d multiplayer games with population-based reinforcement learning. Science **364**(6443), 859–865 (2019)
12. Kurach, K., et al.: Google research football: a novel reinforcement learning environment. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 4501–4510 (2020)
13. Lample, G., Chaplot, D.S.: Playing fps games with deep reinforcement learning. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
14. Lowe, R., Wu, Y.I., Tamar, A., Harb, J., Pieter Abbeel, O., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
15. Ma, X., Yang, Y., Li, C., Lu, Y., Zhao, Q., Jun, Y.: Modeling the interaction between agents in cooperative multi-agent reinforcement learning. arXiv preprint arXiv:2102.06042 (2021)
16. Mnih, V., et al.: Playing Atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602 (2013)
17. Ng, A.Y., Harada, D., Russell, S.: Policy invariance under reward transformations: theory and application to reward shaping. In: ICML, vol. 99, pp. 278–287 (1999)
18. Oliehoek, F.A., Amato, C.: A concise introduction to decentralized POMDPs. BRIEFSINSY, Springer, Cham (2016). https://doi.org/10.1007/978-3-319-28929-8
19. Peng, X.B., Kanazawa, A., Toyer, S., Abbeel, P., Levine, S.: Variational discriminator bottleneck: improving imitation learning, inverse RL, and GANs by constraining information flow (2020)
20. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018)
21. Rashid, T., Samvelyan, M., Schroeder, C., Farquhar, G., Foerster, J., Whiteson, S.: QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In: International Conference on Machine Learning, pp. 4295–4304. PMLR (2018)
22. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
24. Son, K., Kim, D., Kang, W.J., Hostallero, D.E., Yi, Y.: Qtran: learning to factorize with transformation for cooperative multi-agent reinforcement learning. In: International Conference on Machine Learning, pp. 5887–5896. PMLR (2019)

25. Song, S., Weng, J., Su, H., Yan, D., Zou, H., Zhu, J.: Playing fps games with environment-aware hierarchical reinforcement learning. In: IJCAI, pp. 3475–3482 (2019)
26. Sunehag, P., et al.: Value-decomposition networks for cooperative multi-agent learning. arXiv preprint arXiv:1706.05296 (2017)
27. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
28. Vinyals, O., Babuschkin, I., Czarnecki, W.M., Mathieu, M., Dudzik, A., Chung, J., Choi, D.H., Powell, R., Ewalds, T., Georgiev, P., et al.: Grandmaster level in starcraft ii using multi-agent reinforcement learning. Nature **575**(7782), 350–354 (2019)
29. Wang, J., Ren, Z., Liu, T., Yu, Y., Zhang, C.: Qplex: Duplex dueling multi-agent q-learning. arXiv preprint arXiv:2008.01062 (2020)
30. Wang, T., Dong, H., Lesser, V., Zhang, C.: Roma: multi-agent reinforcement learning with emergent roles. In: International Conference on Machine Learning, pp. 9876–9886. PMLR (2020)
31. Wiewiora, E., Cottrell, G.W., Elkan, C.: Principled methods for advising reinforcement learning agents. In: Proceedings of the 20th International Conference on Machine Learning (ICML-2003), pp. 792–799 (2003)
32. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems, vol. 27 (2014)
33. Yu, C., Velu, A., Vinitsky, E., Wang, Y., Bayen, A., Wu, Y.: The surprising effectiveness of PPO in cooperative, multi-agent games. arXiv preprint arXiv:2103.01955 (2021)
34. Zheng, Z., et al.: What can learned intrinsic rewards capture? In: International Conference on Machine Learning, pp. 11436–11446. PMLR (2020)
35. Zheng, Z., Oh, J., Singh, S.: On learning intrinsic rewards for policy gradient methods. In: Advances in Neural Information Processing Systems, vol. 31 (2018)