# Appendix: Independent Deep Deterministic Policy Gradient Reinforcement Learning in Cooperative Multiagent Pursuit Games

Shiyang Zhou, Weiya Ren*, Xiaoguang Ren, Yanzhen Wang, and Xiaodong Yi

[1] Artificial Intelligence Research Center, Defense Innovation Institute, Beijing 100072, China
[2] Tianjin Artificial Intelligence Innovation Center, Tianjin 300457, China
weiyren.phd@gmail.com

**Abstract.** This article is the additional experimental details of paper 'Independent Deep Deterministic Policy Gradient Reinforcement Learning in Cooperative Multiagent Pursuit Games'. In this article, we mainly tests and compares the hyperparameter of the algorithm, and the performance of the algorithms in the case of different preys. The results show that PGDDPG, we proposed, is insensitive to hyperparameter and has stronger robustness which Has a good performance against different levels of prey.

## 1 Experimental Results

### 1.1 Predator-Prey Game with MPE

To evaluate the effectiveness of the proposed approach, we introduce a predator-prey game based on OpenAI's Multi-Agent Particle Environments (MPE) [2]. The goal of the predator is to catch prey **simultaneously** as quickly as possible, while the goal of the prey is to survive as long as possible. The game ends when the predator catches the prey or reaches the maximum number of steps. The horizontal and vertical coordinates of the map are limited to [-1, 1] for the predators and [-0.8, 0.8] for the prey. A predator's maximum speed is 0.5, while the prey's maximum speed is 0.7. To highlight a larger map, the areas of the predators and prey are relatively small. The environmental rewards are sparse (a reward of +10 is earned for success) and depend only on the terminal state of each episode.

We consider that N predators (3 predators in this paper) chase one prey in a randomly generated environment. Each predator independently learns to capture the prey, without knowing the others' policies or actions, in both training and testing. When all predators capture the prey **simultaneously**, each predator will receive a reward of +10. As long as any predator fails to catch the prey, no reward will be given to all predators. Because of the large search space and intelligence of the prey, this situation is a difficult learning problem that requires

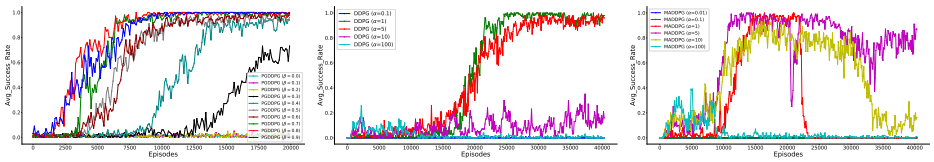
* Shiyang.Z and Weiya.R contributed equally to this work.

excellent tacit cooperation. Both the simple DDPG algorithm and the simple MADDPG algorithm completely fail (success rates = 0.0). In accordance with [1,3,4], by experimental comparison, we choose the best reward-shaping method, as shown in equation (1)[3], which includes the change in distance and the angle between the velocity vector and the vector that points to the prey.

$$r_{rs} = (d_{last} - d_{now}) * \alpha * cos\theta \tag{1}$$

where $r_{rs}$ represents the shaped reward and $d_{last}$ and $d_{now}$ represent the distance to the prey in the last episode and the distance to the prey in the current episode, respectively. $\theta$ is the angle between the velocity vector and the vector that points to the prey. $\alpha$ is a hyperparameter.

Different hyperparameter values yield different performances. Via experiments, we selected the hyperparameters with the best performance for each algorithm: $\alpha = 1$ for DDPG, $\alpha = 5$ for MADDPG and $\beta = 0.8$ for PGDDPG. The differences in performance with different hyperparameters will be subsequently discussed.



(a) PGDDPG. (b) DDPG. (c) MADDPG.

**Fig. 1.** The rates of successful capture with different hyperparameter values for a) PGDDPG, b) DDPG (reward shaping) and c) MADDPG (reward shaping) in the predator-prey game with MPE.

**Hyperparametric Test** As shown in equations (1) and the original paper, the PGDDPG algorithm and reward-shaping function each have a hyperparameter. To obtain the best values, we tested the results obtained with different values of these hyperparameters. Based on the resulting success rate curves, which are shown in Figure 1, we chose $\alpha = 1$ for DDPG, $\alpha = 5$ for MADDPG and $\beta = 0.8$ for PGDDPG in this paper.

As the experimental results show, different values of the hyperparameters will have different performances. The hyperparameters of reward-shaping are only effective within a small range; when the parameters are too small, reward-shaping has no effect, and when they are too large, reward-shaping will drown out the final reward [3]. In our method, the hyperparameter ($\beta$) can work well over a wide range of values. Therefore, hyperparameter insensitivity is an important feature.

[3] We have also tested many other methods, including methods based on the minimum distance to the prey, in an attempt to design an effective reward-shaping function, which has been utilized in the original MPE environment. However, none can perform better than equation (1)

**Prey With Different Speeds** Furthermore, we set the maximum speed of prey-00 to 0.7, 0.5 and 0.3; in these cases, the prey is faster than, equally as fast as, and slower than the predators, respectively. The experiments were carried out again with all other settings being the same. The success rates are plotted in Figure 2.

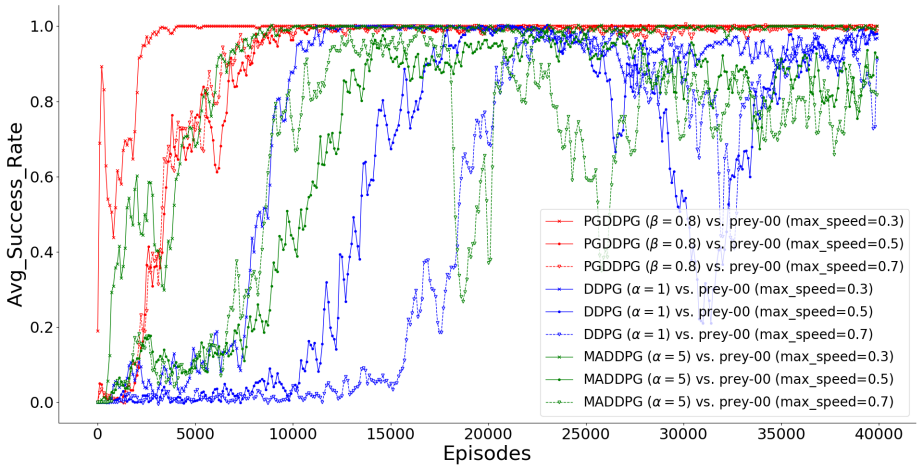



**Fig. 2.** The rates of successful capture for each algorithm when prey-00 has different maximum speeds (fast = 0.7, normal = 0.5, or slow = 0.3) in the predator-prey game with MPE. Different colors represent different algorithms, and different line types represent different maximum speeds of prey-00. The legend shows the details.

## References

1. Harutyunyan, A., Devlin, S., Vrancx, P., Nowé, A.: Expressing arbitrary reward functions as potential-based advice. In: AAAI. pp. 2652–2658 (2015)
2. Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. NIPS (2017)
3. 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)
4. Xie, L., Miao, Y., Wang, S., Blunsom, P., Wang, Z., Chen, C., Markham, A., Trigoni, N.: Learning with stochastic guidance for navigation. arXiv preprint arXiv:1811.10756 (2018)