Modeling and Simulation of UAV Autonomous Obstacle Avoidance Based on DQN

Jiayong Fang
Equipment Management and Unmanned Aerial Vehicle Engineering College, Air Force Engineering University Xi'an 710051, China
fjylike@163.com

Zhongliang Zhou
Equipment Management and Unmanned Aerial Vehicle Engineering College, Air Force Engineering University Xi'an 710051, China
zzl_1978@163.com

Huanbin Wang
Equipment Management and Unmanned Aerial Vehicle Engineering College, Air Force Engineering University Xi'an 710051, China
wanghb11@163.com

Sheng Sheng
Equipment Management and Unmanned Aerial Vehicle Engineering College, Air Force Engineering University Xi'an 710051, China
ss_123@163.com

Shitao Chen
Equipment Management and Unmanned Aerial Vehicle Engineering College, Air Force Engineering University Xi'an 710051, China
stcting@163.com

ABSTRACT

During the obstacle avoidance process of UAV, the route planning and obstacle avoidance decision-making is dynamic and sequential due to the change of obstacles in real time, so it is difficult to build a dynamic and accurate route planning model. This paper breaks through the traditional research thinking of route modeling and optimization solution for UAV obstacle avoidance, and applies deep reinforcement learning to UAV autonomous obstacle avoidance decision-making; through designing the environment model, autonomous decision-making model and DQN algorithm model for UAV obstacle avoidance, four simulation experimental environments for UAV obstacle avoidance are constructed to verify the superiority and effectiveness of deep reinforcement learning in solving the decision-making problems related to dynamic model and timing sequence, and provide a new solution for future UAV route planning.

CCS CONCEPTS

• **Computing methodologies**; • **Modeling and simulation**;

KEYWORDS

Deep Q Network, Unmanned Aerial Vehicle, Autonomous Obstacle Avoidance

ACM Reference Format:
Jiayong Fang, Zhongliang Zhou, Huanbin Wang, Sheng Sheng, and Shitao Chen. 2021. Modeling and Simulation of UAV Autonomous Obstacle Avoidance Based on DQN. In *2021 4th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI'21), December 22–24, 2021, Sanya, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3508546.3508620

1 INTRODUCTION

With the development of modern science and technology, UAV autonomous combat has gradually become an indispensable mode of combat in the future battlefield, and the combating tasks implemented by UAV have developed from the traditional combat supporting such as reconnaissance, early warning and surveillance into the integrated combat operation of autonomous reconnaissance, jamming, attack and evaluation in the whole process, which greatly demonstrates the operational advantages of UAV in its original combat capability [1-5]. The rapid development of artificial intelligence and the extensive application of deep reinforcement learning theory provide powerful conditions for creating unmanned aerial vehicles with high autonomous capability. Deep reinforcement learning is to combine the perception ability of deep learning with the decision-making ability of reinforcement learning in a general form to enable the intelligent agents to perceive information from high-dimensional space, establish models and make decisions through end-to-end learning, and realize direct control from original input until output [6-8]. In this paper, autonomous obstacle avoidance of UAV is taken as the research object, and by the constructing of the environment model, intelligent agent control model and obstacle model for UAV obstacle avoidance and through the utilization of DQN algorithm, the reinforcement learning model and deep learning model of UAV obstacle avoidance is designed and the obstacle avoidance algorithm flow is constructed. Finally, the effectiveness of the algorithm model designed in this paper is verified by four UAV obstacle avoidance Scenes, which provides technical development direction for future UAV intelligent operations in the battlefield.

2 DQN MODEL FOR AUTONOMOUS OBSTACLE AVOIDANCE OF UAV

2.1 UAV obstacle avoidance reinforcement learning model

2.1.1 UAV obstacle avoidance action value function. Every obstacle avoidance action of the UAV is realized through selecting the action based on the state and strategy of the last time period, and at the same time, the UAV is updated to a new state. In this process, the unmanned aerial vehicle needs to choose the action with best anticipated result according to the strategy of$\varepsilon - greedy$, and needs to establish the action value function of the UAV state which is expressed by state-action pair, its mathematical model is as follows:

$$Q\left(s\_state, a\_action\right) = r_{t+1} + \gamma Q\left(s'\_state, a'\_action\right) \quad (1)$$

For the determination of UAV state, two parameters are set up, and the modeling is:

$$s\_state = s\left(UAV\_x, UAV\_y\right) \quad (2)$$

2.1.2 Model of UAV obstacle avoidance exploration strategy. In the process of updating the UAV state, a suitable action must be determined for every step based on the function of the current state and $Q(s\_state, a\_action)$. If the action of avoiding obstacles is taken completely according to the past experience, that is, the UAV chooses the action with the biggest corresponding value in$Q(s_state, a_action)$, it is possible to be restricted in the existing experience and unable to find out the new obstacle avoidance behavior with more value; However, if the UAV only focuses on exploring new obstacle avoidance routes, i.e., completely random action, the majority of actions will be worthless, which leads to very slow learning speed of $Q(s\_state, a\_action)$ function.

The method of striking a balance between "experience" and "exploration" is to adopt the strategy $\varepsilon - greedy$to select suitable actions [9-10].A relatively small $\varepsilon$is set in advance(for instance, $\varepsilon = 0.1$), the UAV selects obstacle avoidance action by learning $Q(s\_state, a\_action)$ based on $1 - \varepsilon$ probability and the random actions of the remainingε probability are used to explore new experience

The exploration strategy modeling based on the above is:

$$a\_action_{\pi(a|s)} = \begin{bmatrix} 1-\varepsilon & a\_action\_\max_a Q\left(s\_state, a\_action\right) \\ \varepsilon & a\_action\_random \end{bmatrix} \quad (3)$$

2.1.3 Updating model of UAV value function. In the process of continuous motion and learning, the UAV needs to continuously update its learning experience and obtain continuous return, thus the state action value function is required to be continuously updated and iterated. Since the state transition probability and return value of the UAV are unknown, a search method similar to LMS (Least Mean Square) algorithm is adopted [11]. Starting from the initial estimation of the value function of each state (where the value function is initialized to 0), the recursive form of the value function can be expressed as the error form:

$$\varsigma = r_{t+1} + \gamma V^{\pi}\left(s', a'\right) - V^{\pi}\left(s, a\right) \quad (4)$$

The updating process of value function in the process of practical application is to increase the product of learning rate $\alpha$and prediction error $\varsigma$to the state value of the previous action. The introduction of learning rate α is aimed at making the updating process of value function tend to be gentle. The modeling for value function of action state is:

$$Q_{k+1}\left(s\_state, a\_action\right) = Q_k\left(s\_state, a\_action\right) + \alpha\left[r_{t+1} + \gamma Q\left(s_{t+1}', a\_action_{\pi(a|s)}\right) - Q\left(s\_state, a\_action\right)\right] \quad (5)$$

2.1.4 UAV motion decision-making model. For an unmanned aerial vehicle (UAV) moving horizontally and vertically, it is needed to judge the action state within a certain range to determine whether it can continue to move in a certain direction. Only moving towards the left, right and upward directions can the UAV pass through obstacles. According to the conditions of obstacles, the conditional model that the UAV can continue to move towards the left at a certain time is:

$$UAV\_x > UAV\_WIDTH/2 \quad (6)$$

In the same way, the conditional model that the UAV can continue to move towards the right at a certain time is:

$$UAV\_x < WINDOW\_WIDTH - UAV\_WIDTH/2 \quad (7)$$

Whether can the UAV move upward? It is needed to determine whether the upward movement will collide with obstacles. If the collision happens, the punishment measure should be taken. Secondly, confirm whether meeting the requirements of the horizontal boundary limit of obstacles, the established conditional model for judgement is:

$$UAV\_y < WINDOW\_HIGHT - UAV\_WIDTH/2 \quad (8)$$

2.2 UAV obstacle avoidance deep learning model

2.2.1 Value network. Value network is equivalent to value function and belongs to nonlinear approximation model. The main tasks of the value network are, on the one hand, to carry on training according to the samples generated by the interaction between UAV intelligent agent and the environment, adjust the network weight and improve the value function; on the other hand, after the completion of the network training, according to the current state of the UAV, the numerical value of the value required by the UAV action is output [12-14]. A three-layer feed forward neural network is constructed, and the established model is:

$$EVALUATE\_NET = Q(s, a, \theta)_{[(l_1, \omega_1, b_1), (l_2, \omega_2, b_2), (l_3, \omega_3, b_3)]} \quad (9)$$

The specific forward calculation model is:

$$l_1 = f\left([s]_{IN} \times [\omega_1] + [b_1]\right) \quad (10)$$

$$l_2 = f\left([l_1] \times [\omega_2] + [b_2]\right) \quad (11)$$

$$Q\big([s]_{IN}, [action]\big)_{OUT} = f\left([s] \times [\omega_1] + [b_1]\right) \quad (12)$$

2.2.2 Target network. The structure of target network is exactly the same as that of value network. In order to alleviate the fluctuation caused by instability of data samples when Q-Learning updates the model, DQN algorithm introduces a method of updating the parameters of the value function, which prevents the network training from falling into the feedback loop of target value and predicted

Table 1: Obstacle Parameters Setting in Scene 1

Name of Parameter	Name of Parameter Variables	Parameter Configuration
Number of obstacles	OBSTACLE_NUM	20
Obstacle height	OBSTACLE_HEIGHT	2
Obstacle interval width	OBSTACLE_WIDTH	10
Initial coordinates of the obstacle in $y$direction	OBSTACLE_START	100
Number of channels on obstacles	OBSTACLE_DOOR_NUM	1
Movement mode in channel mouth	ENTRABCE_MED	Motionless

value [15]. It mainly completes the decoupling of the return value of UAV at the current moment and the value estimation at the next moment. The established model is as follows:

$$TARGET\_NET = Q(s, a, \theta^{-})_{[(l_1,\omega_1,b_1),(l_2,\omega_2,b_2),(l_3,\omega_3,b_3)]} \tag{13}$$

2.2.3 Updating of network parameters. The completion of parameters updating of value network depends on the target network. The specific updating process has two stages: the first stage is to compare the target value of the target network with the estimated value of the value network, and get the target value and update the value network; what is updated here is the network weight of the value network, which is equivalent to the value function parameters [16-17]. The established model is:

$$LOSS\left[Q^{-}(s,a,\theta^{-})_{TARGET\_NET}, Q(s,a,\theta)_{EVALUATE\_NET}\right] \rightarrow UPDATE\left[\theta_{it} \rightarrow \theta_{it+1}\right] \tag{14}$$

The second stage is to update the parameters of the target network by the value network. When the value network is iterated and updated several times, the parameters are transmitted to the target network to update it; in other circumstances, the parameters of the target network remain unchanged, so that the parameters of the target network are frozen in a certain period of time, which reduces the volatility of the model. The established model is:

$$UPDATE_j\left[\theta_{it} \rightarrow \theta_{it+1}\right] \rightarrow UPDATE\left[\theta_i \rightarrow \theta_i^{-}\right] \tag{15}$$

2.3 Algorithm Flow of UAV Obstacle Avoidance

The algorithm flow of UAV obstacle avoidance based on DQN is shown in Table.

Algorithm 1 of UAV Obstacle Avoidance

```
Input
Initialize Replay Buffer sample container D with a capacity of N
Initialize state action value network $Q$and parameters$\theta$
Initialize state action value network $Q^{-}$and parameters$\theta^{-}$
Start
for episode = 1, M do
  Initialize the obstacle avoidance environment, the initial state of the UAV$s_1$is obtained, and after preprocessing, $\phi_1 = \phi(s_1)$ is obtained.
  for t = 1, T do
The UAV randomly chooses an action$a_t$by$\varepsilon$probability, or chooses the best action at present according to the model,
      $a_t = \max_a Q*(\phi(s_t), a; \theta)$
Execute the action$a_t$, get a new round of state $s_{t+1}$and return$r_{t+1}$
After preprocessing, $\phi_{t+1} = \phi(s_{t+1})$ is obtained.
Save $\{\phi_t, a_t, r_{t+1}, \phi_{t+1}\}$in D
Take a batch of samples $(\phi_j, a_j, r_{j+1}, \phi_{j+1})$ from D
Calculate $y_i =$
$\{ \begin{matrix} r_{j+1} & \text{for} & ter\min al\ \phi_{j+1} \\ r_{j+1} + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta^{-}) & \text{for} & non-ter\min al\ \phi_{j+1} \end{matrix}$
The method of gradient descent is used to get the solution,
according to the target function $(y_j - Q(\phi_j, a_j; \theta))^2$
The parameter $\theta \rightarrow \theta^{-}$update is completed for every several rounds
  end for
end for
```

3 SIMULATION EXPERIMENT OF UAV AUTONOMOUS OBSTACLE AVOIDANCE

3.1 Setting of the simulation Scenes

Four simulation Scenes are set up for UAV autonomous obstacle avoidance. Scene 1: Static single-channel obstacle avoidance Scene, its parameter configuration is shown in Table 1, that is, there is one channel on each layer of obstacle and the obstacle is stationary; Scene 2: Dynamic single-channel obstacle avoidance Scene, its parameter configuration is shown in Table 2, that is, there is one channel on each layer of obstacle and the obstacle is moving; Scene 3: Static two-channel obstacle avoidance Scene, its parameter configuration is shown in Table 3, that is, there are two channels on each layer of obstacle and the obstacles are stationary; Scene 4: Dynamic two-channel obstacle avoidance Scene, its parameters are shown in Table 4, that is, there are two channels on each layer of obstacle, and the obstacles are moving. The DQN algorithm settings corresponding to the Scenes are shown in Table 5

3.2 Analysis of simulation results

For Scene 1 and Scene 2, the UAV cannot come across the decision-making problem of multiple routes selection due to only one channel on each layer of obstacle, and it is OK for the UAVs to learn and pass-through multi-layer obstacles in the most direct route without hitting any obstacle. For Scene 3 and Scene 4, because there are two channels on each layer of obstacle in them, the UAVs should not only learn not to hit the obstacles, but also learn how to select one channel from the two channels on each layer of obstacle, so as to realize the passing through of multi-layer obstacles in the shortest route. For this reason, in order to validate the algorithm model adopted in this paper, the simulation results of Scene 3 and

Table 2: Obstacle Parameters Setting in Scene 2

Name of Parameter	Name of Parameter Variables	Parameter Configuration
Number of obstacles	OBSTACLE_NUM	20
Obstacle height	OBSTACLE_HEIGHT	2
Obstacle interval width	OBSTACLE_WIDTH	10
Initial coordinates of the obstacle in $y$direction	OBSTACLE_START	100
Number of channels on obstacles	OBSTACLE_DOOR_NUM	1
Movement mode in channel mouth	ENTRABCE_MED	Dynamic loop

Table 3: Obstacle Parameters Setting in Scene 3

Name of Parameter	Name of Parameter Variables	Parameter Configuration
Number of obstacles	OBSTACLE_NUM	20
Obstacle height	OBSTACLE_HEIGHT	2
Obstacle interval width	OBSTACLE_WIDTH	10
Initial coordinates of the obstacle in $y$direction	OBSTACLE_START	100
Number of channels on obstacles	OBSTACLE_DOOR_NUM	2
Movement mode in channel mouth	ENTRABCE_MED	Motionless

Table 4: Obstacle Parameters Setting in Scene 4

Name of Parameter	Name of Parameter Variables	Parameter Configuration
Number of obstacles	OBSTACLE_NUM	20
Obstacle height	OBSTACLE_HEIGHT	2
Obstacle interval width	OBSTACLE_WIDTH	10
Initial coordinates of the obstacle in $y$direction	OBSTACLE_START	100
Number of channels on obstacles	OBSTACLE_DOOR_NUM	2
Movement mode in channel mouth	ENTRABCE_MED	Dynamic loop

Table 5: Algorithm Parameters

Name of Parameter		Name of Parameter Variables	Parameter Configuration
Maximum number of training		MAX_EPISODES	1000
Maximum stride length of single training		MAX_EP_STEPS	10000
Number of samples for experience buffer zone		MEMORY_SIZE	5000
ε*-greedy* strategy increment		e_greedy_increment	0.0005
Learning rate		learning_rate	0.005
Discount factor for value return		reward_decay	0.9
Maximum value action probability for $\varepsilon - greedy$strategy		e_greedy	0.99
Score setting for UAV movement	Towards left	Reward	-1
	Towards right	Reward	-1
	Hit obstacles	Reward	-1
	Pass through obstacles	Reward	+10

Scene 4 are mainly analyzed to determine whether the UAVs can select the optimum route.

The simulation for Scene 3 is shown in Figure 1, in which from Stage A to Stage F the obstacles are stationary without any change occurrence, and two channels are set on each layer of obstacle. As indicated from the simulation results, the UAV, through learning under DQN algorithm decision-making model, can successfully complete the task of obstacle avoidance by selecting one of the channels on each layer of obstacle and moving upward. However, in order to meet the requirement of being the shortest route, it needs to select one optimum channel from two obstacles of each layer. With a view of verifying whether the UAV has selected one better route, two algorithms are chosen here to make comparative analysis of routes: one algorithm is the Monte Carlo method, that

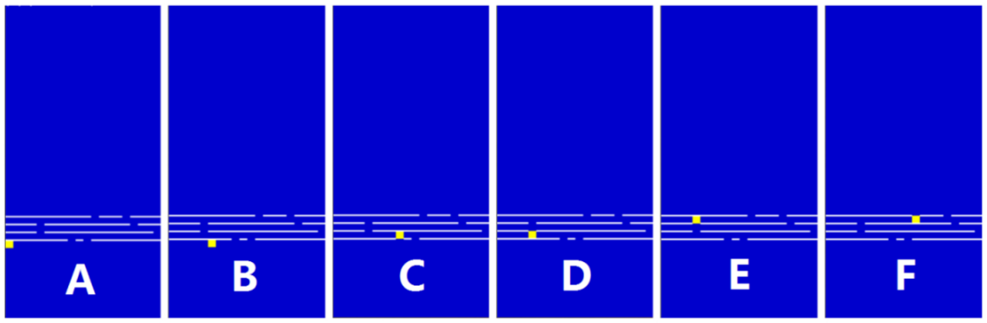

Figure 1: UAV Obstacle avoidance simulation in Scene 3

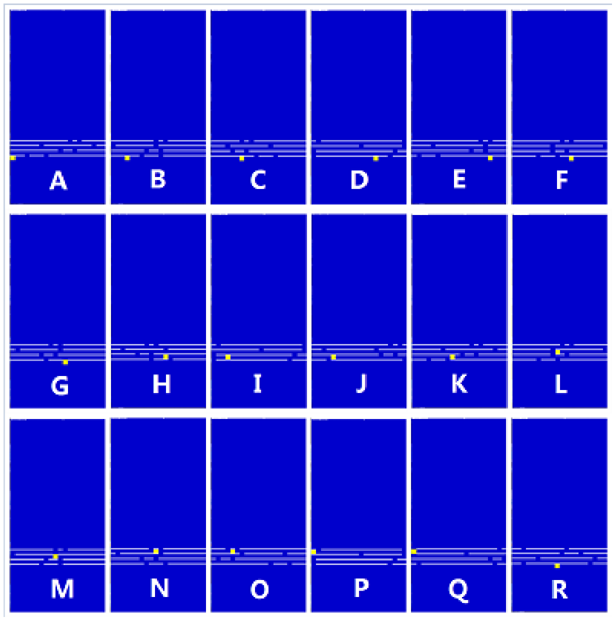

Figure 2: UAV Obstacle avoidance simulation in Scene 4

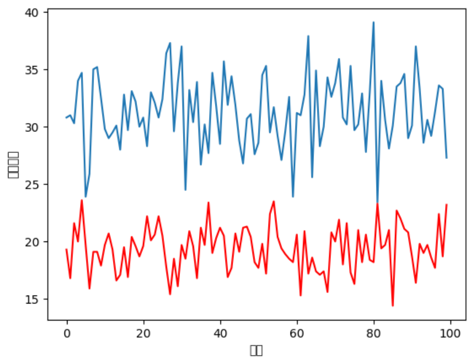

Figure 3: Routes under two different methods adopted in Scene 3 adopted

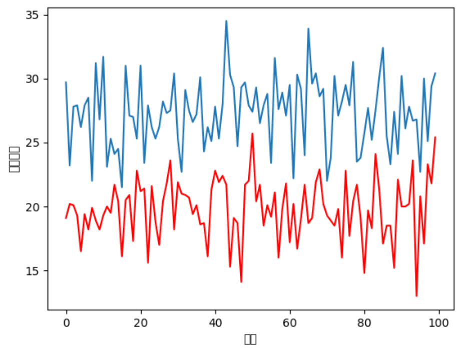

Figure 4: Routes under two different methods in Scene 4

is, the UAV randomly select one channel from the two channels on each layer of obstacle; another is the DQN decision-making model adopted in this paper to let the UAV select the route. In order to carry on algorithm validation, firstly, randomly generate 1000 Scenes 3, that is to say, the locations of channel are generated randomly, secondly, with the two algorithms calculate out the steps the UAV take to pass through all obstacles in each Scene, and then, calculate out the mean value of steps according to the cumulative steps for every 10 Scenes; the result is shown in Figure 3, the abscissa is the number of Scenes for simulation, the vertical ordinate is the steps the UAV passes through all obstacles, the blue signifies the route of simulation by applying the Monte Carlo method and the red signifies the route of decision-making by applying the DQN algorithm. It can be obviously seen from the results that the UAV obstacle avoidance decision-making algorithm based on DQN has advantage over the Monte Carlo method.

The setting of Scene 4 is shown in Figure 2, from Stage A to Stage R the UAV has completed an obstacle avoidance simulation process based on DQN. In Scene 3, because obstacles are stationary, the problem about the shortest path can be solved by using optimization algorithms such as dynamic programming. But in Scene 4, obstacles in each layer move at every moment, that is, the optimization model needs to be solved changes in real time, which limits the using of dynamic programming method to solve the shortest path problem of dynamically changing models. However, the artificial intelligence method of deep reinforcement learning can solve this problem well, through the perception and trial of obstacle avoidance environment, the motion decision is made by using the estimated value. After several trial and error studies, the memory is formed, and the multi-channel obstacle avoidance task in dynamic environment is completed. Similarly, in order to verify whether the UAV has chosen a better route, select Monte Carlo method and DNQ method to simulate and verify 1000 Scenes 4 (randomly generated), and the results are shown in Figure 4. It can be clearly seen from Figure 4 that the obstacle avoidance route based on DQN is shorter than that simulated by Monte Carlo, which proves that, under the learning of DQN algorithm, the UAV has the well-qualified ability to pass the preset obstacle in the shortest route after several trials and errors according to its own reward situation.

4 CONCLUSION

The autonomous obstacle avoidance modeling and simulation of UAV based on DQN algorithm effectively applied and solved the problems of UAV autonomous movement and optimal route selection in dynamic obstacle environment. Deep reinforcement learning is used to solve the decision-making control problem of UAV obstacle avoidance learning, especially to solve the decision-making problems related to dynamic model and time series. By combining the global optimization of dynamic programming method with the value estimation of timing sequence difference method, the future value is estimated by the present state, and the future return is used for decision-making. Compared with other route optimization algorithms, it can better solve the shortest route optimization problem in dynamic timing sequence environment.

REFERENCES

[1] Ling,ZHANG Zhu-feng,WU Wei.Coordinated test allocation in multi-UCAV based on distributed constrained optimization[J].Journal of Naval University of Engineering,2018,12(6) :64-68.

[2] Hu Teng, LIU Zhangjun, LIU Yang,et al.3D surveillance path planning for multi-UAVs[J].Systems Engineering and Electronics[J].2019,7)7(:1551-1559.

[3] Lucian Bu, Robert Babu, Bart De Schutter, *et al.* A comprehensive survey of multiagent reinforcement learning. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2008, 38(2):156–172,.

[4] Matignon, Laurent Jeanpierre, and Abdel-Illah Mouaddib. Coordinated multi-robot exploration under communication constraints using decentralized markov decision processes. In Twenty-sixth AAAI conference on artificial intelligence, 2012.

[5] Peng, Quan Yuan, Ying Wen, Yaodong Yang, Zhenkun Tang, Haitao Long, and Jun Wang. Multiagent bidirectionally-coordinated nets for learning to play starcraft combat games. arXiv preprint arXiv:1703.10069, 2, 2017.

[6] M L. Markov games a framework for multi-agent reinforcement learning[M].New Brunswick: Machine Learning Proceedings,1994:157-163.

[7] Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershel-vam, Marc Lanctot, *et al.* Mastering the game of go with deep neural networks and tree search. nature, 529(7587):484, 2016.

[8] Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Grae-pel, *et al.* Mastering chess and shogi by self-play with a general reinforcement learning algorithm. arXiv preprint arXiv:1712.01815, 2017.

[9] S, HOLLY E,LILLICRAP T,et a1. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates[C]//IEEE International Conference on Robotics and Automation. Singapore:IEEE Press, 2017:3389-3396.

[10] J, ASSAEL I, DE FREITAS N, et a1. Learning to communicate with deep multi-agent reinforcement learning[C]//Advances in Neural Information Processing Systems.Spain: NIPS Press,2016:2137-2145.

[11] WU Y, et a1. Mu1ti-agent actor-critic for mixed cooperative-competitive Environments[C]//advances in Neural Information Processing Systems. Los Angeles :NIPS Press, 2017:6379-6390.

[12] M, ZAMBALDI V,GRUSLYS A, et a1.A unified game-theoretic approach to multi-agent reinforcement learning[C]//Advances in Neural Information Processing Systems. Los Angles : NIPS Press,2017:4190-4203.

[13] J, ZAMBALDI V, LANCTOT M, *et al.* Multi-agent reinforcement learning in sequential social dilemmas[C]//Proceedings of the 16th Conference on Autonomous Agents and Multi-agent Systems, Singapore: AAMAS Press, 2017:464-473.

[14] SHALEV-SHWARTZ S, SHAMMMAH S, SHASHUA A. Safe, multi-agent reinforcement learning for autonomous driving [J].https://arxiv.org/abs/1610.03295.

[15] JIN J, SONG C, LI H, et a1. Rea1-Time Bidding with Multi-Agent Reinforcement Learning in Display Advertising[J/OL].https//arxiv.org/abs/1802. 09756.

[16] L, CHEV J, HUANG Y, et a1. Smart generation control based on multi -agent reinforcement learning with the idea of the time tunnel[J].Energy, 2018, 153:977-987.

[17] J,LEIBO JZ,ZAMBALDI V, et a1.A multi-agent reinforcement learning model of common-pool resource appropriation[C] Advances in pleural Information Processing Systems. Los Angeles: NIPS Press, 2017:3643-3652.