Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.041253

ARTICLE

An Intelligent Algorithm for Solving Weapon-Target Assignment Problem: DDPG-DNPE Algorithm

Tengda Li, Gang Wang, Qiang Fu*, Xiangke Guo, Minrui Zhao and Xiangyu Liu

Air Defense and Antimissile College, Air Force Engineering University, Xi'an, 710051, China

*Corresponding Author: Qiang Fu. Email: fuqiang_66688@163.com

Received: 16 April 2023 Accepted: 04 July 2023 Published: 08 October 2023

ABSTRACT

Aiming at the problems of traditional dynamic weapon-target assignment algorithms in command decision-making, such as large computational amount, slow solution speed, and low calculation accuracy, combined with deep reinforcement learning theory, an improved Deep Deterministic Policy Gradient algorithm with dual noise and prioritized experience replay is proposed, which uses a double noise mechanism to expand the search range of the action, and introduces a priority experience playback mechanism to effectively achieve data utilization. Finally, the algorithm is simulated and validated on the ground-to-air countermeasures digital battlefield. The results of the experiment show that, under the framework of the deep neural network for intelligent weapon-target assignment proposed in this paper, compared to the traditional RELU algorithm, the agent trained with reinforcement learning algorithms, such as Deep Deterministic Policy Gradient algorithm, Asynchronous Advantage Actor-Critic algorithm, Deep Q Network algorithm performs better. It shows that the use of deep reinforcement learning algorithms to solve the weapon-target assignment problem in the field of air defense operations is scientific. In contrast to other reinforcement learning algorithms, the agent trained by the improved Deep Deterministic Policy Gradient algorithm has a higher win rate and reward in confrontation, and the use of weapon resources is more efficient. It shows that the model and algorithm have certain superiority and rationality. The results of this paper provide new ideas for solving the problem of weapon-target assignment in air defense combat command decisions.

KEYWORDS

Weapon-target assignment; DDPG-DNPE algorithm; deep reinforcement learning; intelligent decision-making; GRU

1 Introduction

Weapon-target assignment (WTA) is the core link in the command and control of air defense operations, which has a significant impact on improving combat effectiveness. Its connotation refers to the efficient use of its own multi-type and multi-platform weapon resources based on battlefield situational awareness, rational allocation and interception of multiple incoming targets, avoiding the omission of key targets, repeated shooting, and other phenomena, to achieve the best combat effect [1–3]. This problem has been proven to be a non-deterministic polynomial complete (NP-complete)

problem [4,5]. The efficient WTA is more than 3 times more effective than free fire [6], acting as a force multiplier.

The experts and scholars at home and abroad regard WTA as a class of mathematical problems and have carried out related model construction and solution work from multiple levels [7–13]. Davis et al. [14] focused on ballistic missile defense with multiple rounds of launches under dynamic conditions from the perspective of maximizing the protection of strategic points; Xu et al. [15] fully considered the communication and cooperation between sensor platforms and weapon platforms, and carried out research on the optimization goals of maximizing the target threat and minimizing the total cost of interception for the multi-stage air defense WTA problem. Guo et al. [16] mainly studied the problem of many-to-many missile interception and proposed two strategies: fixed and adaptive grouping strategies, which were solved by artificial swarm algorithm; Aiming at the WTA problem of regional defense, Kim et al. [17] proposed two new algorithms: rotation fixed strategy and rotation strategy to deal with multi-target attacks, and the effectiveness of the algorithm was verified by experiments. Li et al. [18] summarized and analyzed the static WTA problem, and introduced an improved Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D framework) to solve it. Based on analyzing the difficulties of regional air defense decision-making, Severson et al. [19] adopted the idea of multi-layer defense to establish a distribution model to maximize the interference effectiveness index. Ouyang et al. [20] proposed a distributed allocation method to effectively manage radar resources and use probabilistic optimization algorithms to allocate radar targets for limited radar early warning resources. To maximize the probability of killing, Feng et al. [21] divided the fire units into groups, considered the situation of compound strikes in the same group of fire units, and constructed a dynamic WTA model with multiple interception timing. Bayrak et al. [22] studied how to efficiently solve the firepower cooperative allocation problem using the genetic algorithm (GA) with good convergence and search speed; Li et al. [23] designed a particle swarm optimization (PSO) algorithm with perturbation by attractors and inverse elements for the anti-missile interceptor-target allocation problem; to realize the rapid solution of WTA, Fan et al. [24] introduced the variable neighborhood search factor into the solution equation, mimics the natural behavior of bees, and proposes a memory mechanism that improves the global search efficiency of the artificial bee colony (ABC) algorithm. Fu et al. [25] proposed a multi-target PSO algorithm based on the co-evolution of multiple populations to construct a model of co-evolution.

However, the above studies are based on traditional analytical models and algorithms. Due to the rapid changes and difficult quantification of the battlefield environment and the opponent's strategies, there are bottlenecks in the uncertainty and nonlinearity of traditional analytical models and algorithms in dealing with decision-making, and it is difficult to adapt to the changing battlefield environment. Facing the needs of air defense operation, decision-making advantage is the core, so it is urgent to study new methods for WTA to improve the level of intelligent decision-making. WTA is a typical sequential decision-making process oriented to incomplete information games, which can be boiled down to solving the Markov decision process (MDP) problem; deep reinforcement learning (DRL) provides an efficient solution to this problem: DRL can realize the end-to-end learning process from perception to action, and its learning mechanism and method are in line with the experience learning and decision-making thinking mode of combat commanders, which has obvious advantages for solving sequential decision-making problems under game confrontation conditions. Some good results have been achieved in the application of Go [26–28], real-time strategy games [29,30], automatic driving [31], intelligent recommendation [32], and other fields.

In summary, this paper aims to apply the theory and algorithm of DRL to the WTA problem of air defense operation command decision-making. By introducing an event-based reward mechanism

(EBR), multi-head attention mechanism (MHA), and gated cyclic unit (GRU), a new deep neural network framework for intelligent WTA is constructed, which is solved by the improved Deep Deterministic Policy Gradient algorithm with dual noise and prioritized experience replay (DDPG-DNPE) algorithm with dual noise and prioritized experience replay to improve the auxiliary decision-making ability of intelligent WTA for air defense operation in highly dynamic, uncertain and complex battlefield environments, transform information advantages into decision-making advantages, and provide more accurate WTA decision support for commanders. Finally, the red and blue sides are designed on the simulation deduction platform to verify the network architecture and algorithm proposed in this paper. The experimental results prove the practicability and effectiveness of the method used in this paper.

2 Related Theories

2.1 DRL

The goal of reinforcement learning (RL) is to enable the agent to obtain the maximum cumulative reward value during interaction with the environment, and to learn the optimal control method of its actions. RL introduces the concept of agent and environment, expanding the optimal control problem into a more general and broader sense of sequential decision-making problems, and the agent can autonomously interact with the environment and obtain training samples, rather than relying on a limited number of expert samples. The RL model consists of five key parts: agent, environment, state, action, and reward. Each interaction between the agent and environment produces corresponding information, which is used to update the agent's knowledge, and this perception-action-learning cycle is shown in Fig. 1.

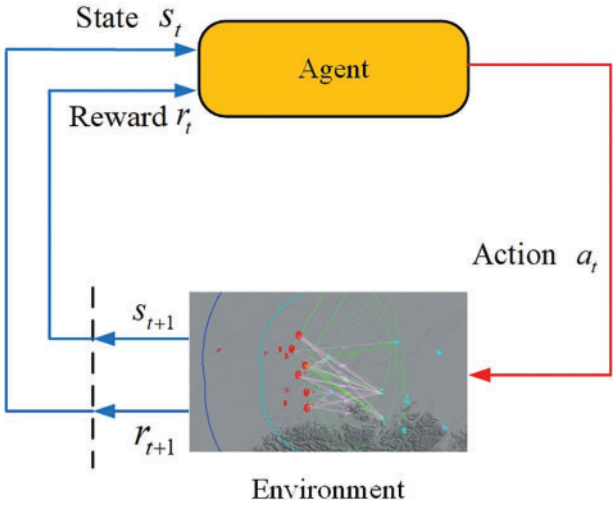


Figure 1: Schematic diagram of RL

DRL is the reinforcement learning method of using a deep neural network to express the agent strategy. In the field of air defense intelligent WTA, the perception ability of deep learning (DL) can be used for battlefield situation recognition, and the RL algorithm can be used to assist decision-making to improve the efficiency of WTA and gain competitive advantages.

2.2 Markov Decision Process

MDP is represented by five tuples $\langle S, A, P, R, \gamma \rangle$: $S = (s_1, s_2, \cdots, s_n)$ is the set of states; $A = (a_1, a_2, \cdots, a_m)$ is the action set; P is the state transition matrix; R is the reward. γ is the discount factor.

In the process of MDP, the agent is in the initial state s in the environment, at which time it will execute an action a, and then the environment will output the next state s' and the reward r obtained by the current action a. The agent is constantly interacting with the environment.

R_t is the cumulative reward, which is the sum of rewards after the time step t:

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{1}$$

The policy π is the probability of selecting an action a in the state s:

$$\pi(a|s) = P[a_t = a \,|s_t = s] \tag{2}$$

The state-value function $V^{\pi}(s)$ is the expected total reward of taking strategy π in the initial state s:

$$V^{\pi}(s) = E_{\pi}[R_t \,|s_t = s] = E_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,|s_t = s\right] \tag{3}$$

The action-value function $Q^{\pi}(s, a)$ is the expected total reward obtained when action a is performed in state s and subsequent actions follow strategy π:

$$Q^{\pi}(s, a) = E_{\pi}[R_t \,|s_t = s, a_t = a] = E_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,|s_t = s, a_t = a\right] \tag{4}$$

The goal of RL is to solve the optimal policy function π^* of MDP to maximize the return, while the optimal state value function $V^*(s)$ and the optimal action value function $Q^*(s, a)$ are expressions of the optimal policy function π^*:

$$\max_{\pi} E\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}\right] \tag{5}$$

$$V^*(s) = \max_{\pi} V^{\pi}(s) \tag{6}$$

$$Q^*(s, a) = \max_{\pi} Q^{\pi}(s, a) \tag{7}$$

2.3 DDPG Algorithm

2.3.1 Introduction of the DDPG Algorithm

Aiming at the dimensionality disaster problem existing in traditional RL algorithms, a better DRL algorithm can be found for solving large-scale decision-making tasks by combining the representation advantages of DL and the decision-making advantages of RL.

DDPG algorithm has strong deep neural network fitting ability and generalized learning ability, and its sequential decision-making ability is strong, which is very consistent with the decision-making thinking of air defense operations, so this paper considers the use of DDPG algorithm in air defense intelligent decision-making.

The expectation that defines the cumulative reward is the objective function of the DDPG algorithm:

$$J_\beta(\mu) = \mathrm{E}_\mu\left[r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots + \gamma^n r_n\right] \tag{8}$$

To find the optimal deterministic behavior policy μ^*, is equivalent to maximizing the policy in the objective function $J_\beta(\mu)$:

$$\mu^* = \arg\max_\mu J(\mu) \tag{9}$$

The Actor network is updated as follows:

$$\begin{aligned}\nabla_{\theta^\mu} J &\approx \mathrm{E}_{s_t\sim\rho^\beta}\left[\nabla_{\boldsymbol{\theta}^\mu} Q_\mu\left(s_t, \mu\left(s_t\right)\right)\right] \\ &= \mathrm{E}_{s_t\sim\rho^\beta}\left[\nabla_{\boldsymbol{\theta}^\mu} Q\left(s, a; \boldsymbol{\theta}^Q\right)\Big|_{s=s_t, a=\mu\left(s_t;\boldsymbol{\theta}^\mu\right)}\right] \\ &= \mathrm{E}_{s_t\sim\rho^\beta}\left[\nabla_a Q\left(s, a; \boldsymbol{\theta}^Q\right)\big|_{s=s_t, a=\mu(s_t)} \nabla_{\boldsymbol{\theta}^\mu}\mu\left(s_t; \boldsymbol{\theta}^\mu\right)\big|_{s=s_t}\right]\end{aligned} \tag{10}$$

$Q_\mu(s, \mu(s))$ is the action state value that can be generated when the action is selected according to the deterministic policy μ under the state s; $\mathrm{IE}_{s\sim\rho^\beta}$ represents the expectation of Q if the state s conforms to the ρ^β distribution. The gradient ascent algorithm is used to optimize the above equation to continuously improve the expectation of discount cumulative reward. Finally, the algorithm updates the parameters $\boldsymbol{\theta}^\mu$ of the policy network in the direction $Q(s, a; \boldsymbol{\theta}^Q)$ of increasing the action value.

To update the critic network by Deep Q Network (DQN) updating value network, the gradient of the value network is as follows:

$$\nabla_{\boldsymbol{\theta}^Q} = \mathrm{E}_{s,a,r,s'\sim R}\left[\left(r + \nabla_{\boldsymbol{\theta}^{Q'}}\gamma Q'\left(s', \mu\left(s'; \boldsymbol{\theta}^{\mu'}\right)\right) - Q\left(s, a; \boldsymbol{\theta}^Q\right)\right)\nabla_{\boldsymbol{\theta}^Q} Q\left(s, a; \boldsymbol{\theta}^Q\right)\right] \tag{11}$$

The neural network parameters $\boldsymbol{\theta}^{\mu'}$ and $\boldsymbol{\theta}^{Q'}$ in the Target Q value are the parameters of the target policy network and target value network, respectively, and the gradient descent algorithm is used to update the parameters in the network model. The process of training the value network is to find the optimal solution of the parameters $\boldsymbol{\theta}^Q$ in the value network.

Therefore, the training objective of the DDPG algorithm is to maximize the objective function $J_\beta(\mu)$ and minimize the loss of the value network Q.

2.3.2 DDPG-DNPE Algorithm

The traditional DDPG algorithm uses an experience replay mechanism, and samples uniformly from the experience replay pool during sampling, that is, all experience importance is considered to be consistent. In the actual simulation process, it is found that the importance of the sample is different, and the data that makes the network performs poorly in the interaction process is more valuable for learning. Therefore, this paper introduces a priority experience replay mechanism, and gives different data a certain weight, so that the training network can be invested in learning high-value data as much as possible.

$$\delta_i = r\left(s_t, a_t\right) + \gamma Q'\left(s_{t+1}, a_{t+1}, w\right) - Q\left(s_t, a_t, w\right) \tag{12}$$

If $|\delta_i|$ is larger, give the experience a higher weight. The sampling probability of experience can be defined as:

$$P_i = \frac{D_i^\alpha}{\sum_k D_k^\alpha} \tag{13}$$

where, $D_i = \frac{1}{rank\,(i)}$, $rank\,(i)$ is the sequence number of experience i in the experience pool. The larger $|\delta_i|$ is, the higher the sequence number is, that is, the greater the probability of experience i being drawn. α mainly determines the order in which priorities are used.

A High-frequency sampling of experiences with high weights changes the distribution of samples, making it difficult for the model to converge. Importance sampling is often considered, and the importance sampling weight is:

$$W_i = \frac{1}{S^\beta \cdot P_i^\beta} \tag{14}$$

S is the size of the experience replay pool, and β is a hyperparameter that controls the level of experience replay based on priority.

As shown in Fig. 2, in the training process, to better take into account exploration and update, OU (Ornstein-Uhlenbeck) random noise and Gaussian noise are introduced to change the decision-making process of the action from deterministic to a random process, and the differential form of OU random noise N_t is:

$$\mathrm{d}N_t = \theta\,(\mu - N_t)\,\mathrm{d}t + \delta \mathrm{d}B_t \tag{15}$$

where, μ is the mean value; θ is the speed at which noise tends to the average value; δ is the fluctuation degree of noise; B_t is the standard Brownian motion.

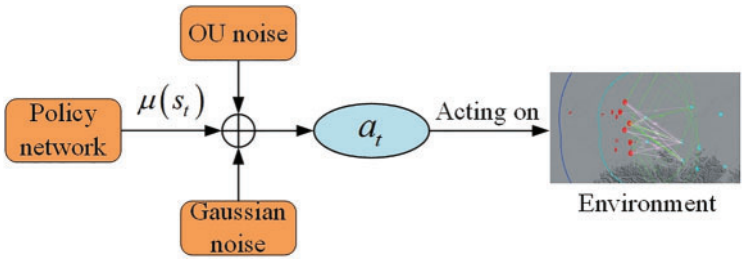


Figure 2: Exploration strategy based on OU noise and Gaussian noise

Gaussian noise is directly superimposed in the motion exploration in the form of $\varepsilon \sim N\left(0, \delta^2\right)$.

In summary, the structural block diagram of the improved DDPG algorithm is shown in Fig. 3.

The calculation flow of the DDPG-DNPE algorithm is as follows:

Algorithm 1: DDPG-DNPE algorithm

Randomly initialize critic network $Q\left(s, a|\boldsymbol{\theta}^Q\right)$ and actor $\mu\,(s|\boldsymbol{\theta}^\mu)$ with weights $\boldsymbol{\theta}^Q$ and $\boldsymbol{\theta}^\mu$

Initialize target network Q' and μ' with weights $\boldsymbol{\theta}^{Q'} \leftarrow \boldsymbol{\theta}^Q, \boldsymbol{\theta}^{\mu'} \leftarrow \boldsymbol{\theta}^\mu$

Initialize replay buffer S

(Continued)

Algorithm 1 (continued)

for episode = 1,M **do**
Initialize a random process N_t and ε for action exploration
Receive initial observation s_1
for t = 1,T do
Select action $a_t = \mu(s_t;\boldsymbol{\theta}^{\mu}) + N_t + \varepsilon$ according to the current policy and exploration noise
Execute action a_t and observe reward r and observe the new state s_{t+1}
Calculate δ_i and D_i, store transition (s_t, a_t, r_t, s_{t+1}) in S
Calculate W_i, conduct importance sampling a minibatch of N transitions (s_t, a_t, r_t, s_{t+1}) from S
Set $y_i = r_i + \gamma Q'\left(s_{i+1}, \mu'\left(s_{i+1}|\boldsymbol{\theta}^{\mu'}\right)|\boldsymbol{\theta}^{Q'}\right)$
Update critic network by minimizing the loss: $L = \frac{1}{N}\sum_i \left(y_i - Q\left(s_i, a_i|\boldsymbol{\theta}^{Q}\right)\right)^2$
Update the actor policy using the sampled policy gradient:

$$\nabla_{\boldsymbol{\theta}^{\mu}} J \approx \frac{1}{N}\sum_i \nabla_a Q\left(s, a; \boldsymbol{\theta}^{Q}\right)\Big|_{s=s_i, a=\mu(s_i)\boldsymbol{\theta}^{\mu}} \nabla_{\boldsymbol{\theta}^{\mu}} \mu\left(s;\boldsymbol{\theta}^{\mu}\right)\Big|_{s_i}$$

Update the target networks:

$$\begin{cases} \boldsymbol{\theta}^{Q'} \leftarrow \tau\boldsymbol{\theta}^{Q} + (1-\tau)\boldsymbol{\theta}^{Q'} \\ \boldsymbol{\theta}^{\mu'} \leftarrow \tau\boldsymbol{\theta}^{\mu} + (1-\tau)\boldsymbol{\theta}^{\mu'} \end{cases}$$

end for
end for

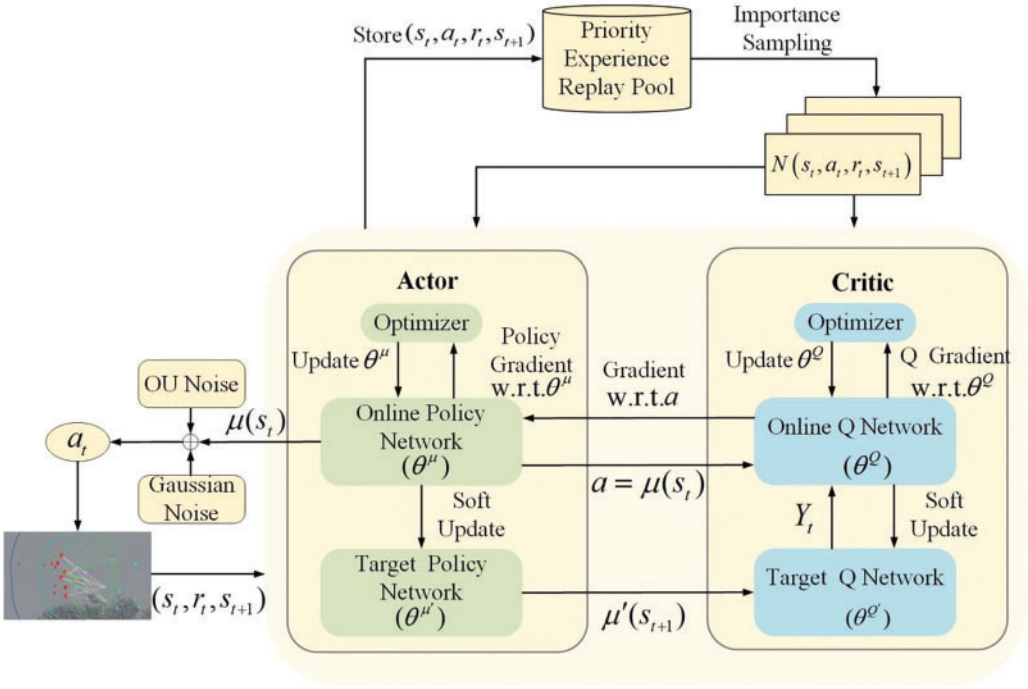


Figure 3: Block diagram of the DDPG-DNPE algorithm

2.3.3 DRL Training Framework

Before using DRL to solve the WTA problem, it is necessary to collect training samples through the interaction between the agent and the environment, and then optimize the neural network parameters through the RL algorithm, so that the agent can learn the optimal strategy. The agent training framework is shown in Fig. 4.

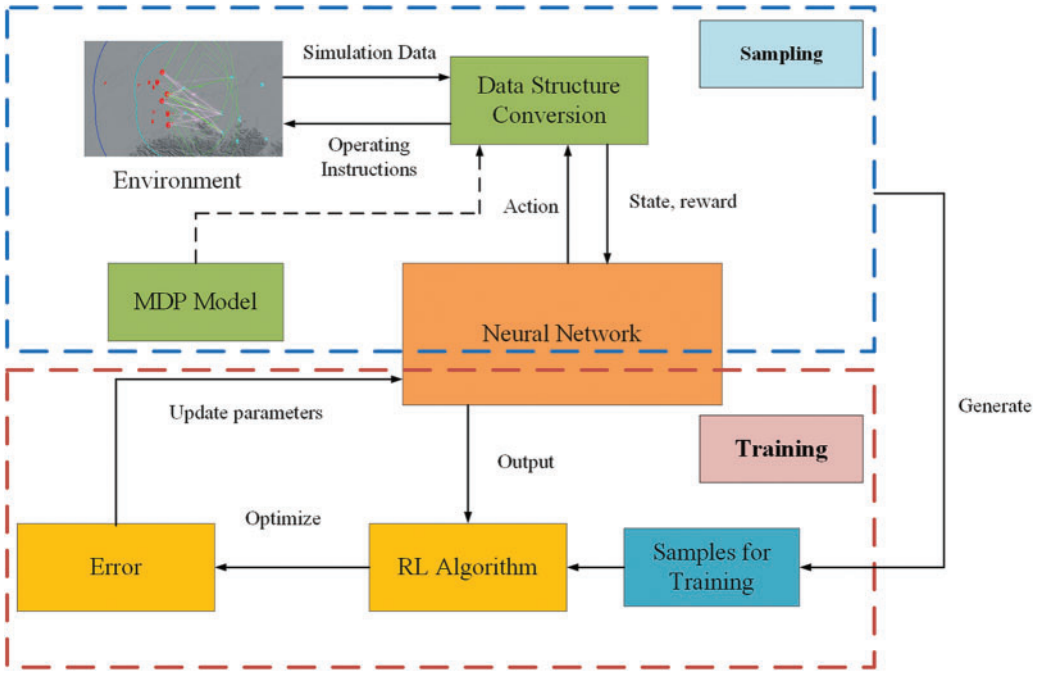


Figure 4: Schematic diagram of the agent training framework

The focus of DRL is on both sampling and training. When sampling, the input of the neural network used by the agent is state information and reward, and the output is action information, while the simulation environment requires input operating instructions and the output is battlefield situation information. Therefore, when the agent interacts with the environment for sampling, the data output from the simulation environment is transformed into state information and reward, and the action output from the neural network is transformed into action commands according to the parameters in the MDP model. During training, the collected samples are input to the RL algorithm, and the parameters are continuously trained and optimized to finally obtain the optimal strategy.

3 Intelligent Decision-Making Model and Training

3.1 Deep Neural Network Framework for WTA

Based on the key indicators such as air defense combat boundaries, engagement criteria, and physical constraints, the deep neural network structure used in this paper is shown in Fig. 5. In the red-blue confrontation, the red agent comprehensively evaluates the threat degree of the incoming blue target according to the real-time state of both sides, considers the deployment of the red side's air defense fire units, and decides which blue targets to intercept at which points in time. The input of the network model is mainly the real-time state of the red and blue sides, and the output is which interceptor weapons are used to intercept which blue targets are in the current state. The

network structure can be divided into three parts: battlefield situation input, decision-making action calculation, and decision-making action output.

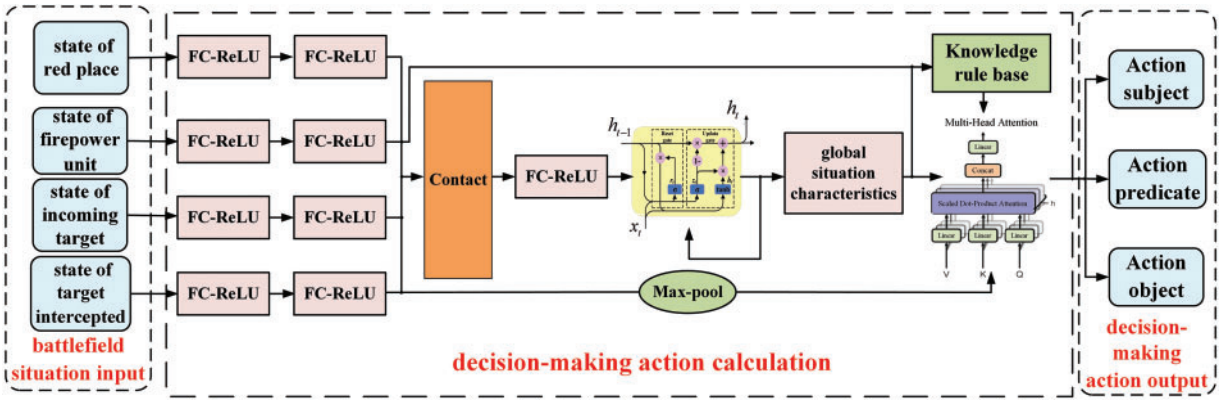


Figure 5: Deep neural network framework for WTA

3.1.1 Battlefield Situation Input

The battlefield situation input is mainly to input the network state space. The network state space is integrated and reduced by combining the air defense WTA combat elements. The battlefield situation is planned to be divided into four categories and input to the neural network in the form of semantic information. The specific classification and state information are shown in Table 1.

Table 1: Classification and information of the state

State classification	State Information
State of red place	The type, location, and attack condition of airfields and command posts
State of the Red firepower units	The number and position of the firepower units and radar vehicles, the state of the radar switch, the number of interceptors, the firepower unit that can be intercepted
State of blue incoming target that can be observed	The type, number, motion status, and threat level of the incoming target
State of blue target that can be intercepted	The type, number, and status of the target that can be intercepted

In a complex battlefield environment, there are many air defense combat entities and operational constraints, and the battlefield situation will change in space over time, so the number of each type is dynamically changing.

3.1.2 Decision-Making Action Calculation

After the state of the red key target, the state of the red fire unit, the state of the blue incoming target that can be observed, and the state of the blue target that can be intercepted are input into the neural network, each type of state data is extracted from the situation characteristics through two layers of fully connected-rectified linear unit (FC-ReLU), and then all the data are combined and connected

to the site. After a layer of FC-ReLU and GRU, the global situation characteristics are output, and then decision reasoning and action calculation are carried out.

Due to the complex battlefield environment and random disturbance, the battlefield situation presents dynamic uncertainty, and the temporal attributes of the situation and the spatial attributes of the operational nodes should be fully considered. Moreover, the red and blue adversarial data often contain the historical value, that is, the decision-making in the current state is related to historical information, and the GRU network can selectively forget unimportant historical information, which better solves the problem of gradient disappearance and gradient explosion in long-sequence training. The structure of the GRU network is shown in Fig. 6.

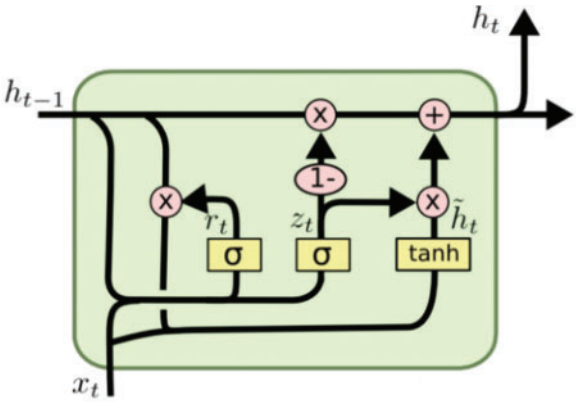


Figure 6: GRU

z_t and r_t represent the updated door and the reset door, respectively. The update gate is used to control the extent to which the state information at the previous moment is brought into the current state, and the larger the value of the update gate, the more state information is brought in at the previous moment. Resetting the gate controls how much information is written to the current candidate set $\widetilde{h}_t$ from the previous state, and the smaller the reset gate, the less information is written from the previous state. The update mechanism for each door is:

$$\begin{aligned} r_t &= \delta\left(W_r \cdot [h_{t-1}, x_t]\right) \\ z_t &= \delta\left(W_z \cdot [h_{t-1}, x_t]\right) \\ \widetilde{h}_t &= \tanh\left(W_{\widetilde{h}} \cdot [r_t \otimes h_{t-1}, x_t]\right) \\ h_t &= (1 - z_t) \otimes h_{t-1} \oplus z_t \otimes \widetilde{h}_t \\ y_t &= \delta\left(W_o \cdot h_t\right) \end{aligned} \tag{16}$$

where, x_t is the input information at the current moment, h_{t-1} denotes the hidden state of the previous moment. The hidden state acts as a neural network memory, which contains information about the data seen by the previous node, h_t represents the hidden state passed to the next moment, $\widetilde{h}_t$ is the candidate hidden state, δ stands for the sigmoid function, by which the data can be changed to a value in the range 0–1, tanh is the tanh function, by which the data can be changed to a value in the range $[-1, 1]$, W_r is the weight matrix.

After the introduction of GRU, it can effectively retain high-value historical information, so that the neural network can skillfully store and retrieve information, rationally use the effective information in the strategy to achieve cross-time correlation events, fully carry out comprehensive analysis and

judgment, and improve the prediction accuracy of the neural network strategy in the time-varying environment.

3.1.3 Decision-Making Action Output

By integrating the network action space, the action output in the WTA process can be divided into three categories: 1. Action subject: selectable red fire unit; 2. Action predicate: the timing of interception of the red fire unit and the type of weapon launched; 3. Action object: blue target that can be intercepted.

In the red-blue confrontation, the combat units of two sides change dynamically with the development of the situation, and the intention of the blue target is closely related to its state, and different characteristics of the target state have different degrees of influence on the analysis of target intention. To improve the efficiency of training, the multi-head attention mechanism is considered. It simulates the human brain's different attention to different objects in the same field of view by assigning certain correlation degree weights to the input sequence features and further analyzes the importance of different target features so that the agent can focus on the blue target with higher threat degree at some moments, give priority to important information, and make accurate decisions quickly.

According to the attention distribution relationship of input characteristics, the attention mechanism can be divided into hard attention and soft attention. The soft attention mechanism assigns attention to each input feature and continuously learns and trains to obtain the weight of each feature. At the same time, the whole mode based on the soft attention mechanism is differentiable, that is, backpropagation learning can be realized. Therefore, the soft attention mechanism is chosen in this article.

The attention variable z is used to represent the location of the selected information, and the probability of the i input information is defined as a_i, then

$$a_i = p\left(z = i|X, q\right) = \text{soft max}\left(f\left(x_i, q\right)\right) = \frac{\exp\left(f\left(x_i, q\right)\right)}{\sum_{j=1}^{N} \exp\left(f\left(x_i, q\right)\right)} \tag{17}$$

$$\boldsymbol{f}\left(x_i, q\right) = \boldsymbol{v}^{\mathrm{T}} \tanh\left(Wx_i + Uq\right) \tag{18}$$

where, $X = [x_1, \ldots, x_N]$ is the input information, which is the intercepted blue target feature vector; q is the selectable red fire unit feature vector, namely, the hidden state obtained by GRU; $f\left(x_i, q\right)$ is the attention scoring function, representing the attention score of the red fire unit to the blue target; W and U are the neural network parameters; $\boldsymbol{v}$ is the global situation feature vector. The current situation is processed by soft max function, the relative importance of each parameter information is obtained, and the focus of local situation information is realized.

After the global situation features are generated, the feature vectors of the situation of the red fire unit and the interceptable blue target are respectively scored for attention, and the score of each red fire unit about each interceptable blue target is generated. Finally, the sigmoid sampling of the score vector is carried out to generate the attack target of the red fire unit. When making decisions, the algorithm will output the command and control for each unit, collect the status and overall situation of each unit, and then call the next decision command.

3.2 Red Agent Training Method

In the simulation, the data itself is unstable, each round of iteration may produce fluctuations, and will immediately react to the next round of iterations, it is difficult to obtain a stable model. This paper intends to decouple the intelligent WTA neural network in the training process, as shown in Fig. 7, dividing it into an inference module and a training module. The two modules use the same network structure and parameters, and the inference module is responsible for interacting with the simulation environment to obtain the interaction data. Based on the interactive data, the training module continuously updates the network parameters through the improved DDPG algorithm and synchronizes the network parameters to the inference module when the training module completes N_i iteration.

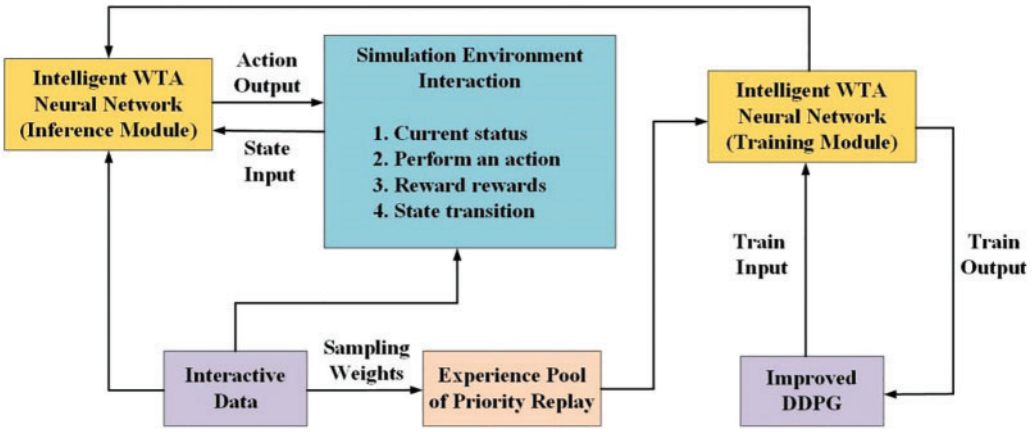


Figure 7: Schematic representation of the decoupling training

Since the parameters of the inference module are fixed in N_i iteration time, the data difference is reduced and the network fluctuation can be effectively avoided. The value of N_i is affected by the fluctuation range of the training module, and the threshold T is considered. When the fluctuation range is less than T and N_i meets the lower limit, the parameters of the inference module are updated synchronously.

4 Simulation Deduction and Verification

This paper uses the intelligent simulation deduction platform to compile the confrontation operation plan of the red and blue sides, and realize data collection and air defense command and control intelligent body verification. The deduction platform has a variety of models such as UAVs, cruise missiles, bombers, early warning aircraft, long-range and short-range fire units, radars, etc., which can realize a variety of operations such as aircraft takeoff and landing, flight along designated routes, bombing, missile launching, fire unit firing, radar switching and on, etc., and can carry out countermeasure deduction in real-time and evaluate the decision-making level of the agent.

4.1 WTA Platform Architecture

As shown in Fig. 8, the training environment and the extrapolation environment are physically divided, the corresponding training environment is constructed in the digital battlefield according to the combat idea, and the training environment and the agent are deployed on the training cloud of the large-scale data center. By training in the learning environment of the training cloud for some time, the

agent will have some real-time decision-making ability. Then a corresponding extrapolation system is constructed in the digital battlefield which runs on the extrapolation cloud composed of small-scale server clusters. The agents trained on the training cloud will also be deployed on the same extrapolation cloud, and the countermeasures learned during training will be applied to the extrapolation system. Through intuitive adversarial deduction, the decision-making level of the agent is evaluated, and the defects and deficiencies of the agent are analyzed. The hyperparameters of the neural network in the training environment are adjusted in a targeted manner, and then iteratively trained.

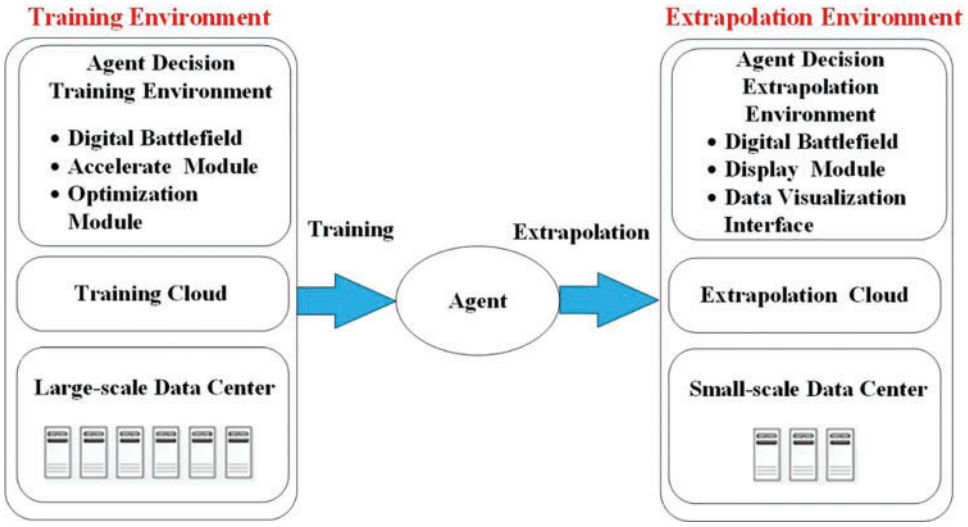


Figure 8: Platform architecture diagram

4.2 Simulation Environment

The whole simulation deduction process is based on the virtual digital battlefield close to the real battlefield, using the real elevation digital map, which can be configured with physical constraints and performance indicators of equipment, including radar detectable area, missile killing area, and killing probability, etc. At the same time, the combat damage of both sides and other confrontation results can be recorded in real-time.

The simulation environment is shown in Fig. 9. The confrontation process is divided into two camps, "red and blue". In the combat area, a certain number of blue forces attack the command post and airport of the red side, and the task of the red side is to protect strategic places such as the command post and airfield. The task of the blue side is to destroy strategic points of the red side and attack the exposed fire units of the red side. The red agent receives the battlefield situation in real-time in the battlefield environment and makes decision-making instructions according to the battlefield situation to strike at incoming blue targets, protecting important places. It uses the reward and punishment mechanism to continuously modify the behavior of the decision-making brain and finally enables it to generate correct decision-making instructions for the situation in the environment.

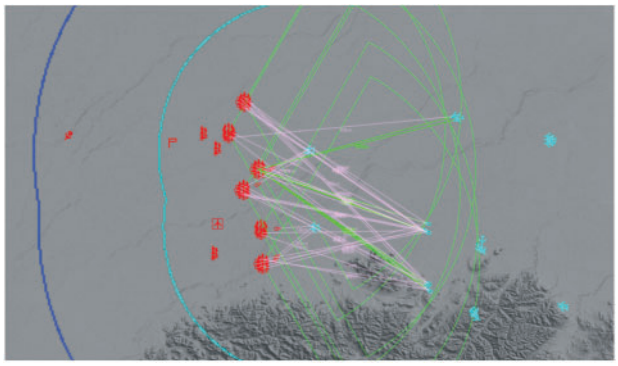


Figure 9: Simulation environment

4.3 Troop Setting

The main goal of the red side is to rationally plan the use of ammunition and defend the command post and airfield with a minimum interceptor missile resource. The main goal of the blue side is to destroy the command post and airfield of the red side while attacking the exposed fire units. The force settings and performance indicators of the red and blue sides are shown in Table 2.

Table 2: Troop setting

	Style	Number
Red	Command post	1
	Airfield	1
	Long-range interception units	6
	Short-range interception units	3
	early warning aircraft	1
Blue	Cruise missile	10
	UAV	10
	Fighter	6
	Bomber	2
	Electronic jammer	1

In the initial deployment stage, taking into account the fire connection and overlap of each air defense position while ensuring a certain depth of destruction, some troops are selected to deploy in advance. A total of nine interception units are deployed to protect the command post and airfield: among them, three long-range interception units and two short-range interception units are deployed to defend the command post of the red side; Three long-range interception units and one short-range interception unit were deployed to defend the airfield, as shown in Fig. 9.

4.4 Reward Function Setting

The reward signal is the only supervised information in RL, and whether an objective and appropriate reward function can be given is crucial for training an excellent model. The design of the

reward function is closely related to the combat mission and directly affects the update of strategy parameters. Due to the large number of units on both sides of red and blue, the state space and action space are correspondingly large. If the neural network only gets feedback according to the reward function after each round of confrontation, it will reduce the exploration efficiency, resulting in each action facing the problem of sparse feedback. That is, the neural network does a lot of "correct" actions, and a small number of "incorrect" actions lead to combat failure, while these "correct" actions are not rewarded, resulting in difficulty in strategy exploration and optimization. Due to the complexity of the air defense combat command decision-making task, the probability of the agent exploring the winning state by itself is very low, so it is necessary to reasonably design the reward function, clarify the key events that trigger the reward, formulate the final indicators of each component of the reward function, and closely associate the trigger mechanism of the reward with the air defense WTA combat process.

Considering that bombers and fighters pose a greater threat to the red side, the interception of high-value blue targets can be taken as a key reward and punishment trigger, and a one-time periodic reward will be given after the first wave of blue side attacks is successfully intercepted; when the blue side's different value targets are successfully intercepted, a certain reward value will be given; when the red side wins, a winning bonus value is given. The reward function is set as follows:

$$R = \begin{cases} 8\mathrm{i} + 5\mathrm{j} + 1\mathrm{k} + 0.5\mathrm{m} & \text{Fail} \\ 50 + 8\mathrm{i} + 5\mathrm{j} + 1\mathrm{k} + 0.5\mathrm{m} & \text{Win} \end{cases} \tag{19}$$

where, i, j, k, m respectively represent the number of intercepted bombers, fighters, cruise bombs, and UAVs. The red side will be awarded 8 points, 5 points, 1 point, and 0.5 points for intercepting a bomber, fighter, cruise bomb, and UAV respectively. Since the trigger events that reward the red agent are all the goals that the red agent must achieve to win, the reward function can gradually guide the agent to find the direction of learning.

4.5 Antagonism Criterion and Winning Condition Setting

The radar needs to be switched on throughout the guidance. The red side fire control radar will radiate electromagnetic waves, which will be captured by the blue side and then expose the position. If the red fire control radar is destroyed, the red fire unit cannot fight. The interception rate of the anti-aircraft missiles launched by the red fire unit is about 45%–75% in the kill zone, which fluctuates with different types of blue combat units. If the red radar is interfered with, the kill probability will be reduced accordingly.

When the red command post and airfield are all destroyed, or the radar loss exceeds 60%, the red side fails; When the blue team loses more than 50% of its fighters, the blue team fails.

5 Analysis of the Simulation Results

5.1 Ablation Experiment

To compare and analyze the effects of event-based reward mechanism, GRU, and multi-head attention mechanism on training effectiveness, this section designs an ablation experiment as shown in Table 3.

Table 3: Design of experiment

Algorithm	Event-based reward mechanism	GRU	Multi-head attention mechanism
DDPG + EGM	•	•	•
DDPG + E	•	◯	◯
DDPG + G	◯	•	◯
DDPG + M	◯	◯	•
DDPG	◯	◯	◯

Note: • indicates that the mechanism is included; ◯ indicates that this mechanism is not included.

The results of the ablation experiment are shown in Figs. 10 and 11.

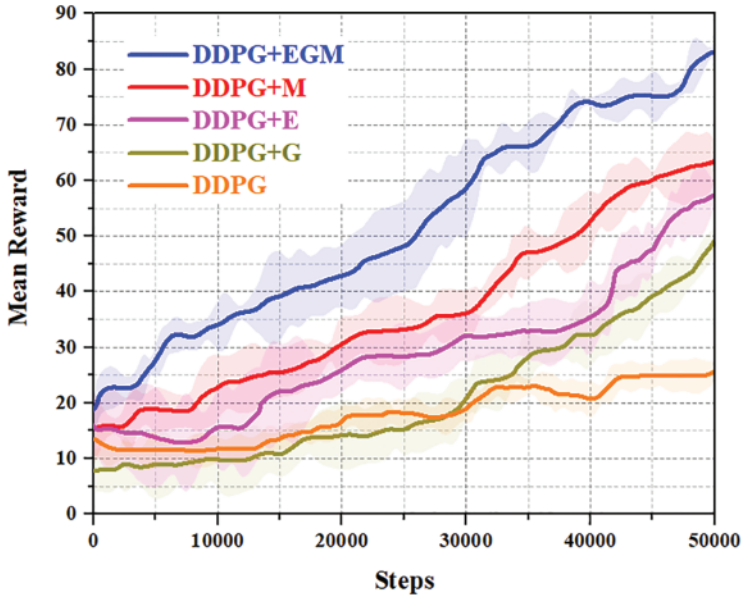


Figure 10: Average reward comparison

The horizontal axis represents the training round, and the vertical axis represents the average reward obtained. It can be seen from Fig. 10 that with the increase of training rounds, the average reward obtained by the algorithm using the three mechanisms has increased, indicating that the three mechanisms proposed have a certain role, but the degree of impact is different: among them, the DDPG algorithm has a low reward and an extremely slow rise, probably because it has not used any mechanism, and the bottleneck of training is more obvious. The average reward of DDPG + G is also low, which may be due to the lack of real-time analysis of the battlefield situation, the difficulty of grasping the battlefield dynamics in real-time, and the delay in rewards, making it difficult to obtain better training results; The higher rewards obtained by DDPG + E and DDPG + M algorithms indicate that the influence of event-based reward mechanism and multi-head attention mechanism is greater, but the effect based on event-based reward mechanism is more obvious. When the three mechanisms are used at the same time, the average reward obtained by the agent increases from 15 to about 80, an increase of 81.25%, indicating that the three mechanisms introduced can significantly

improve the performance of the algorithm, accelerate the training of the agent, and improve the quality of decision-making.

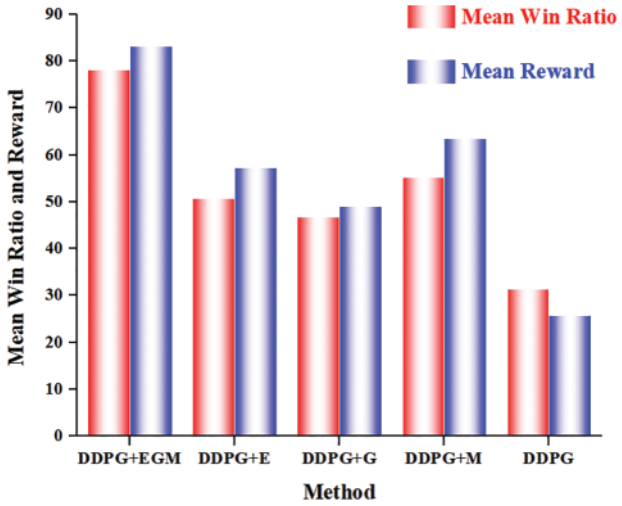


Figure 11: Comparison of ablation results

Fig. 11 shows the comparison of win rate and average reward using different mechanics. Consistent with the above analysis, when the three mechanisms are used, it ensures an effective understanding of the characteristics of the air situation and masters the implementation situation. At the same time, it can reward the agent in time, the winning rate of the red agent is the highest at this time, which can reach 78.14%; If only one machine is used, the win rate of the agent also increases, indicating that the introduction of the machine is necessary and critical for the improvement of the win rate.

5.2 Comparison of Different Algorithms

Under the neural network architecture proposed in this paper, the comparison of the improved DDPG algorithm, Asynchronous Advantage Actor-Critic (A3C) algorithm, DQN algorithm, and the RELU algorithm is shown in Fig. 12. Among them, the RELU algorithm refers to the method of expert rule base to solve the model, as a contrast between the traditional method and the agent model.

In horizontal comparison, the reward function curve and win rate curve when using the rule algorithm are relatively stable, and only fluctuate within a very limited range, indicating that the play of the expert agent is stable. However, agents trained with RL algorithms (such as DDPG, A3C, and DQN) have higher win rates and reward values than rule algorithms, which shows that it is scientific and reasonable to use deep reinforcement learning algorithms to solve WTA problems in the field of air defense operations. Due to the rapid changes in the battlefield environment and the opponent's strategy, it is difficult to deal with complex situations by relying on traditional rules alone, and it is not possible to solve such problems well. The network model trained by neural network and DRL algorithm can provide good solutions to such problems and have a strong ability to adapt to complex battlefields.

In longitudinal comparison, under the same network architecture, compared with the A3C algorithm and DQN algorithm, the use of an improved DDPG algorithm can obtain a higher win rate and reward, indicating that the improved DDPG algorithm can effectively deal with such problems, and the algorithm proposed in this paper is effective. It is worth noting that the win curve and reward

curve jitter is more intense because the scene is full of a large number of uncertainties, resulting in the overall fluctuation of the curve.

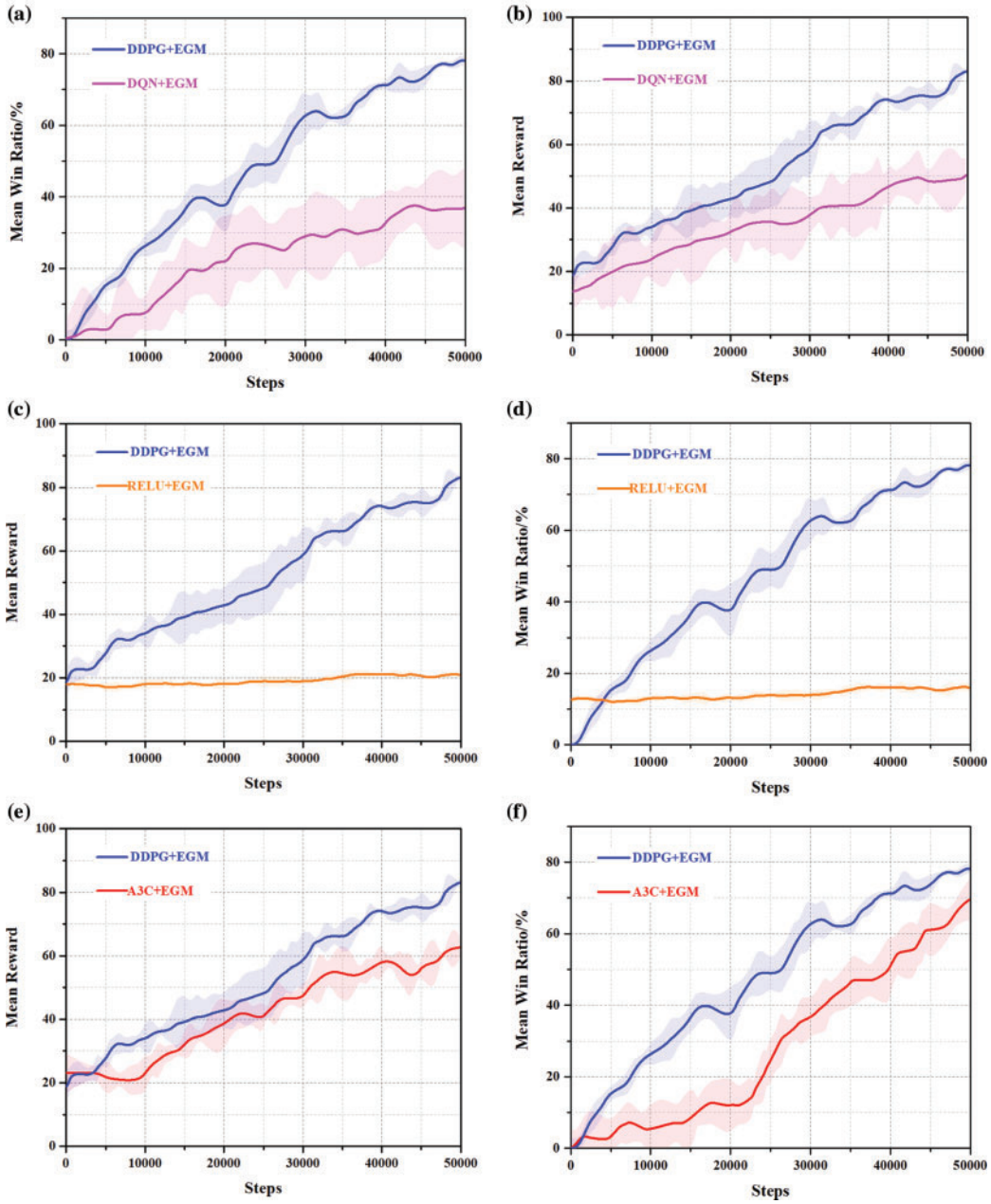


Figure 12: Performance comparison of different algorithms. (a) The mean reward of DDPG + EGM and DQN + EGM; (b) The mean win ratio of DDPG + EGM and DQN + EGM; (c) The mean reward of DDPG + EGM and RELU + EGM; (d) The mean win ratio of DDPG + EGM and RELU + EGM; (e) The mean reward of DDPG + EGM and A3C + EGM; (f) The mean win ratio of DDPG + EGM and A3C + EGM

As can be seen from Fig. 13, consistent with the above analysis, the improved DDPG algorithm has a higher win rate and average reward compared with RELU, A3C, and DQN. It shows that the improved DDPG algorithm is more suitable to solve the red-blue confrontation problem in air defense operations to a certain extent.

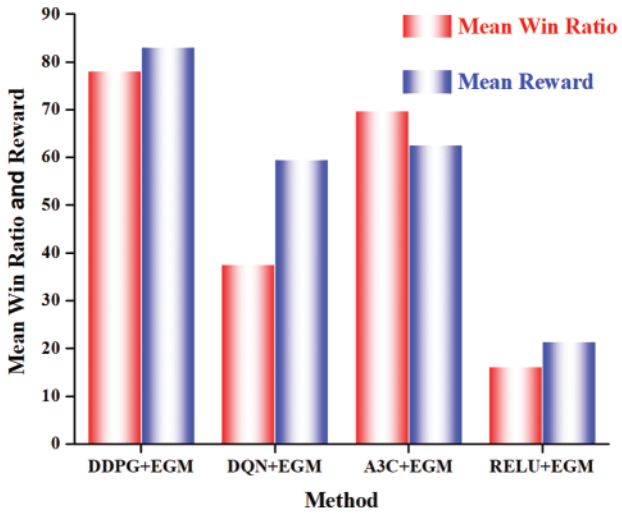


Figure 13: Comparison of the final reward and win rate for different algorithms

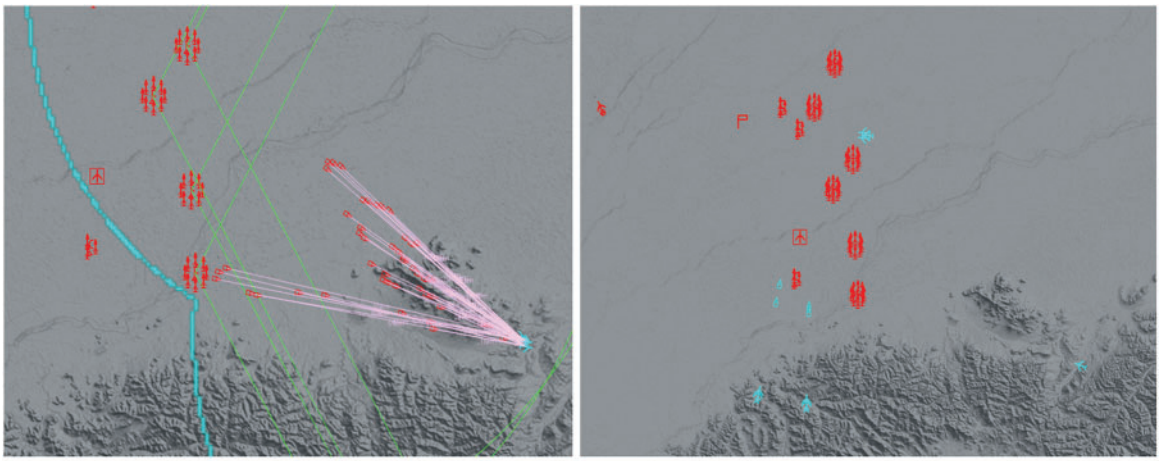
5.3 Analysis of Confrontation Details

In the process of simulation and deduction of the virtual digital battlefield, the trained red agent uses ammunition more reasonably, and tactics emerge, which can better complete the task of defending key places. This section mainly reviews and analyzes the data obtained in the process of simulation and summarizes the strategies emerging from the red agent in the process of confrontation.

(1) reasonable planning of ammunition, the first-line interception

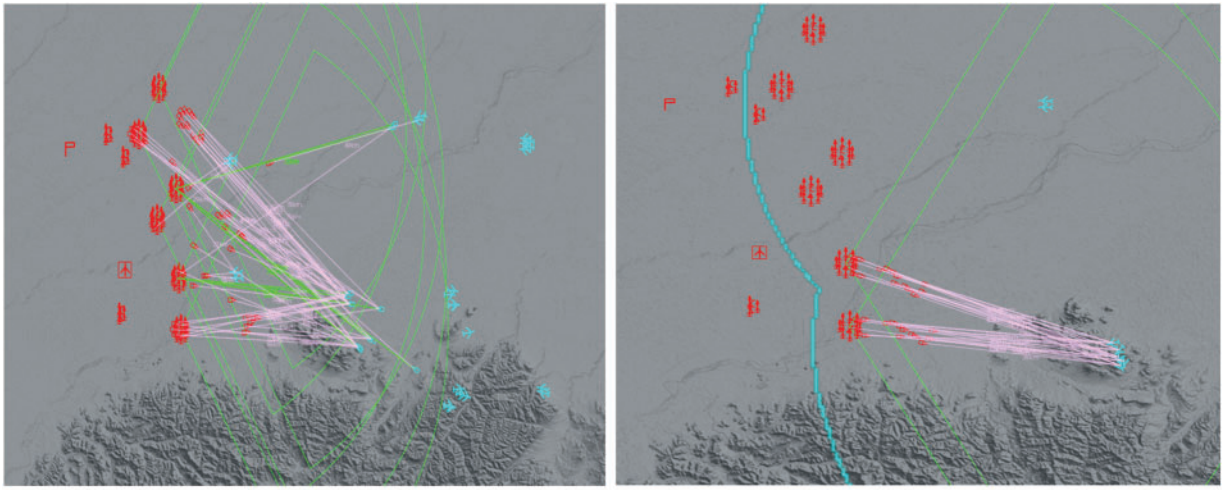
As shown in Fig. 14, in the initial stage, the red agent has almost no strategy, and each firepower unit fires freely when the firing conditions are met. The firepower units use too much ammunition to intercept incoming enemy aircraft at the same time, resulting in excessive ammunition consumption in the early stage, and the efficiency-cost ratio is extremely low, which greatly causes a waste of resources; When the blue's important and threatening incoming targets approach, the ammunition available is extremely limited, and it has to adopt a very conservative strategy to shoot and intercept, and finally the red is failed due to insufficient ammunition.

After a period of training, as shown in Fig. 15, the red agent can better adapt to the blue side's offensive rhythm, master certain rules, and correctly plan the use of ammunition. After the blue target enters the kill zone, the firepower units cooperate to complete the interception with the least ammunition, reflecting the effectiveness of the strategy; When the important and threatening incoming target of the blue side approaches, the firepower unit has a large ammunition stock at this time, and can flexibly adapt the shooting strategy to complete the defense task with low ammunition consumption. Without training, only when the blue target is about to enter the strategic hinterland, the red fire unit can complete the interception; After training, the red side firepower unit can be detected and intercepted as soon as possible, which further verifies the rationality and effectiveness of the network structure trained by the improved DDPG algorithm.



(a) (b)

Figure 14: Agent performance before training



(a) (b)

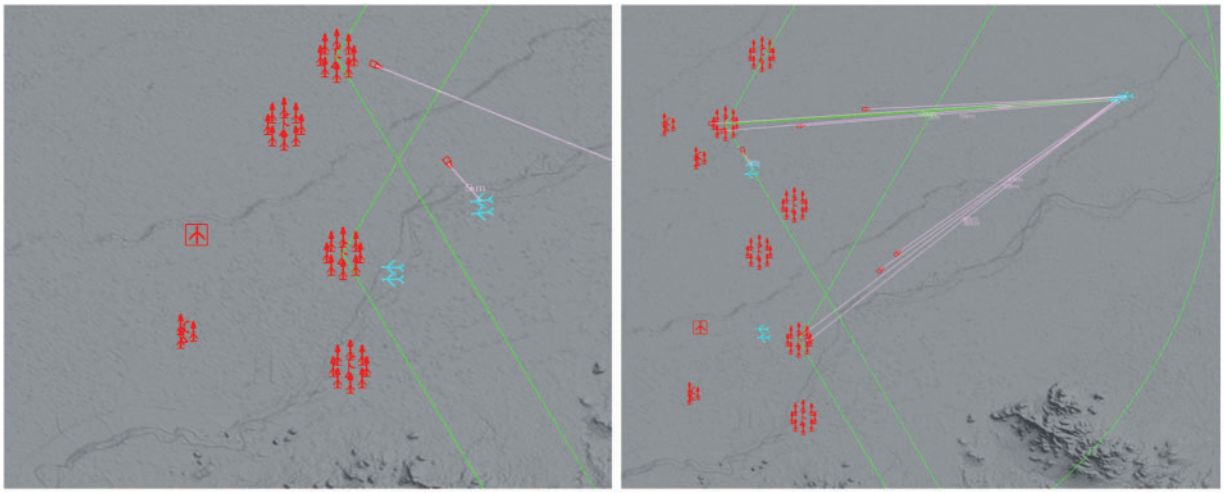
Figure 15: Agent performance after training

(2) short-range and long-range units cooperate, intercept with high efficiency

As shown in Fig. 16, when it has not been trained, the red side firepower units fight independently, and only perform their defensive tasks, and when friendly neighboring units encounter danger, they fail to respond in time to counterattack without any tactics and battle methods.

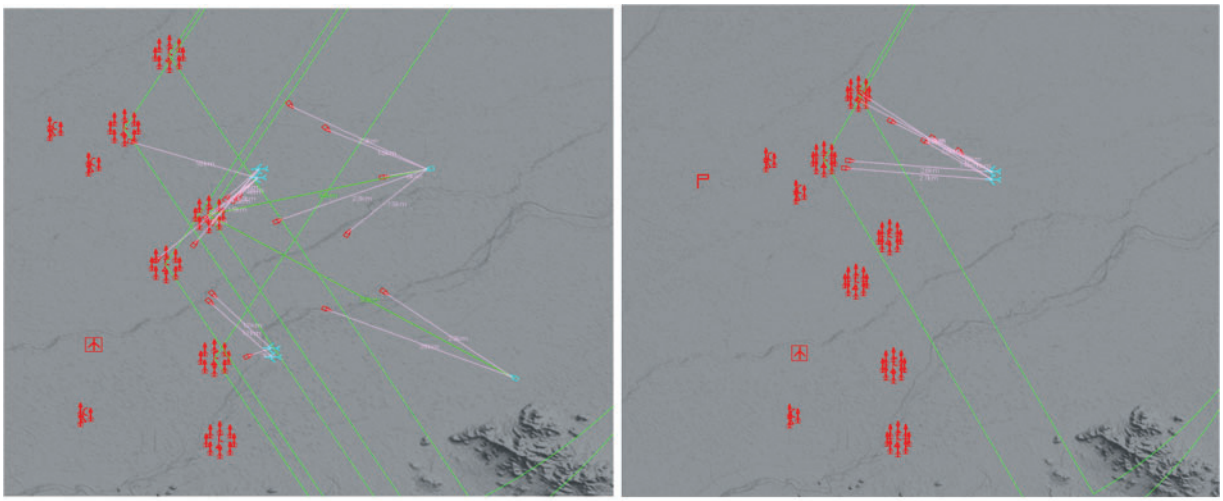
After a period of training, as shown in Fig. 17, the red agent can command the firepower units to carry out coordinated defense, and while completing its defense task, it provides timely and appropriate fire support to other neighboring units, which greatly relieves the defensive pressure of other firepower units and improves the overall defense efficiency. When the long-range firepower units are seriously damaged, they will turn off the radar in time and defend themselves in a silent state. At this time the short-range firepower units can actively react, when the blue target enters the ambush circle, they can

cooperate with the long-range firepower units to quickly and efficiently destroy the incoming blue targets. Since the blue side strategy has not been fixed, that is, the blue side strategy is random, the trained red agent strategy has a certain generalization and can be adapted to other battle scenarios.



(a) (b)

Figure 16: Agent performance before training



(a) (b)

Figure 17: Agent performance after training

6 Conclusion

Aiming at the difficulty of traditional models and algorithms to solve the uncertainty and nonlinearity problems in WTA, this paper constructs a new deep neural network framework for intelligent WTA, analyzes the network structure composition in detail, and introduces event-based reward mechanism, multi-head attention mechanism, and GRU. Then, based on the virtual digital

battlefield close to the real battlefield, real-time confrontation simulation experiments are carried out, and the improved DDPG algorithm with dual noise and priority experience playback techniques is used to solve the problem. The results show that under the new deep neural network framework, compared with the A3C algorithm, DQN algorithm, and RELU algorithm, the agent trained by the improved DDPG algorithm has a higher win rate and reward return, and the planning and use of ammunition is more reasonable, which can show a high decision-making level. The framework proposed in this paper has some reasonableness.

Acknowledgement: We thank our teachers, friends, and other colleagues for their discussions on simulation and comments on this paper.

Funding Statement: This research was funded by the Project of the National Natural Science Foundation of China, Grant Number 62106283.

Author Contributions: Study conception and design: Gang Wang, Qiang Fu; data collection: Minrui Zhao, Xiangyu Liu; analysis and interpretation of results: Tengda Li, Xiangke Guo; draft manuscript preparation: Tengda Li, Qiang Fu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets used or analysed during the current study are available from the corresponding author Qiang Fu on reasonable request.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Z. R. Bogdanowicz, A. Tolano, K. Patel and N. P. Coleman, "Optimization of weapon-target pairings based on kill probabilities," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1835–1844, 2013.

[2] A. Kline, D. Ahner and R. Hill, "The weapon-target assignment problem," *Computers and Operations Research*, vol. 105, no. 5, pp. 226–236, 2019.

[3] P. Forte, A. Mannucci, H. Andreasson and F. Pecora, "Online task assignment and coordination in multi-robot fleets," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4584–4591, 2021.

[4] Q. Y. Cao and Q. Y. Chao, "A genetic algorithm of solving WTA problem," *Control Theory and Applications*, vol. 18, no. 1, pp. 76–79, 2001.

[5] W. G. Fang and X. Y. Shi, "Swarm intelligence optimization algorithms for weapon target allocation problem in multilayer defense scenario," *Mathematics in Practice and Theory*, vol. 43, no. 7, pp. 76–84, 2013.

[6] B. Xin, J. Chen, J. Zhang and L. H. Dou, "Efficient decision makings for dynamic weapon-target assignment by virtual permutation and tabu search heuristics," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 40, no. 6, pp. 649–662, 2010.

[7] Z. W. Li, Y. Z. Chang, Y. X. Kou, H. Y. Yang, A. Xu *et al.,* "Approach to WTA in air combat using IAFSA-IHS algorithm," *Journal of Systems Engineering and Electronics*, vol. 29, no. 3, pp. 519–529, 2018.

[8] X. W. Hu, P. C. Luo, X. N. Zhang and J. Wang, "Improved ant colony optimization for weapon-target assignment," *Mathematical Problems in Engineering*, vol. 24, no. 11, pp. 1–14, 2018.

[9] Y. B. Choi, S. H. Jin, K. S. Kim and B. D. Chung, "A robust optimization approach for an artillery fire-scheduling problem under uncertain threat," *Computers & Industrial Engineering*, vol. 43, no. 11, pp. 23–32, 2018.

[10] Y. Li, Y. X. Kou, Z. W. Li, A. Xu and Y. Z. Chang, "A modified pareto ant colony optimization approach to solve biobjective weapon-target assignment problem," *International Journal of Aerospace Engineering*, vol. 10, no. 1, pp. 1–14, 2017.
[11] F. Yang, J. Wang and C. Y. Dong, "Decision-making of saturation attack for missile weapon-target assignment with multi-target," *Journal of Beijing University of Aeronautics and Astronautics*, vol. 36, no. 8, pp. 996–999, 2010.
[12] C. Y. Han, B. J. Lunday and M. J. Robbins, "A game theoretic model for the optimal location of integrated air defense system missile batteries," *INFORMS Journal on Computing*, vol. 58, no. 3, pp. 405–416, 2016.
[13] C. Leboucher, H. S. Shin, S. L. Ménec, A. Tsourdos, A. Kotenkoff *et al.,* "Novel evolutionary game based multi-objective optimisation for dynamic weapon target assignment," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 3936–3941, 2014.
[14] M. T. Davis, M. J. Robbins and B. J. Lunday, "Approximate dynamic programming for missile defense interceptor fire control," *European Journal of Operational Research*, vol. 41, no. 3, pp. 873–886, 2017.
[15] W. Y. Xu, C. Chen, S. X. Ding and P. Pardalos, "A bi-objective dynamic collaborative task assignment under uncertainty using modified MOEA/D with heuristic initialization," *Expert Systems with Applications*, vol. 31, no. 2, pp. 1–24, 2020.
[16] D. Guo, Z. Liang, P. Jiang, X. Dong, Q. Li *et al.,* "Weapon-target assignment for multi-to-multi interception with grouping constraint," *IEEE Access*, vol. 7, no. 12, pp. 34838–34849, 2019.
[17] J. E. Kim, C. H. Lee and M. Y. Yi, "New weapon target assignment algorithms for multiple targets using a rotational strategy and clustering approach," *IEEE Access*, vol. 10, no. 4, pp. 43738–43750, 2022.
[18] X. Y. Li, D. Y. Zhou, Q. Pan, Y. C. Tang and J. C. Huang, "Weapon-target assignment problem by multiobjective evolutionary algorithm based on decomposition," *Complexity*, vol. 26, no. 2, pp. 8623051:1–8623051:19, 2018.
[19] T. A. Severson and D. A. Paley, "Distributed multitarget search and track assignment with consensus-based coordination," *IEEE Sensors Journal*, vol. 15, no. 2, pp. 864–875, 2014.
[20] Z. Ouyang, L. Xue and F. Ding, "Jamming target assignment method of regional electronic air defense against electro-optical precision guided weapon," *Systems Engineering and Electronics*, vol. 40, no. 12, pp. 2621–2628, 2018.
[21] C. Feng and X. N. Jing, "Weapon target assignment at multiple interception opportunities in composite strikes," *Acta Aeronautica et Astronautica Sinica*, vol. 37, no. 25, pp. 3444–3454, 2016.
[22] A. E. Bayrak and F. Polat, "Employment of an evolutionary heuristic to solve the target allocation problem efficiently," *Information Sciences*, vol. 46, no. 3, pp. 675–695, 2013.
[23] L. Y. Li, F. X. Liu, G. Z. Long, P. S. Guo and X. F. Bie, "Modified particle swarm optimization for BMDS interceptor resource planning," *Applied Intelligence*, vol. 44, no. 3, pp. 471–488, 2016.
[24] C. L. Fan, Q. Fu, G. Z. Long and Q. H. Xing, "Novel hybrid artificial bee colony algorithm with variable neighborhood search and memory mechanism," *Journal Systems Engineering and Electronics*, vol. 29, no. 2, pp. 405–414, 2018.
[25] G. Y. Fu, C. Wang, D. Q. Zhang, J. F. Zhao and H. Q. Wang, "A multiobjective particle swarm optimization algorithm based on multipopulation coevolution for weapon-target assignment," *Mathematical Problems in Engineering*, vol. 25, no. 3, pp. 1–11, 2019.
[26] T. Osa, N. Sugita and M. Mitsuishi, "Online trajectory planning and force control for automationof surgical tasks," *IEEE Transaction on Automation Science and Engineering*, vol. 15, no. 2, pp. 675–697, 2018.
[27] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre *et al.,* "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 148, no. 2, pp. 484–489, 2016.
[28] D. Silver, J. L. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang *et al.,* "Mastering the game of Go without human knowledge," *Nature*, vol. 149, no. 40, pp. 354–359, 2017.
[29] D. H. Ye, G. B. Chen, P. L. Zhao, F. H. Qiu, B. Yuan *et al.,* "Supervised learning achieves human-level performance in MOBA games: A case study of honor of kings," *IEEE Transactions on Neural and Learning Systems*, vol. 9, no. 11, pp. 12–13, 2020.

[30] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik *et al.,* "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 151, no. 40, pp. 250–254, 2019.
[31] S. Yang, W. Wang, C. Liu and W. Deng, "Scene understanding in deep learning-based end-to-end controllers for autonomous vehicles," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 1, pp. 53–63, 2018.
[32] Y. J. Zhang, M. Kampffmeyer, X. G. Zhao and M. Tan, "Deep reinforcement learning for query-conditioned video summarization," *Applied Sciences*, vol. 9, no. 3, pp. 750–761, 2019.