A New Intelligent Decision-Making Method for Air-Sea Joint Operation Based on Deep Reinforcement Learning

SONG Xiaocheng[1], *FENG Shuting*[1], *LI Zhi*[1], *JIA Zhengxuan*[1], *ZHOU Guojin*[2], *YE Dong*[3*]

1. Beijing Institute of Electronic System Engineering, Beijing 100854, P. R. China;

2. Beijing Huashu Defense Technology Co. Ltd, Beijing 100084, P. R. China;

3. Research Center of Satellite Technology, Harbin Institute of Technology, Harbin 150080, P. R. China

(Received 9 November 2022; revised 16 December 2022; accepted 8 January 2023)

Abstract: Aiming at the difficulty of air-sea joint operation in complex multi-equipment combat with high uncertainty, a new intelligent decision-making method for air-sea joint operation based on deep reinforcement learning is proposed. To uniformly represent the input and output of complex networks and their corresponding relations, various networks are utilized, e. g., perceptron, deep long-short term memory network and actor critical structure. Aiming at the instability of policy network learning process and the defects of the proximal policy optimization(PPO) algorithm, an improved proximate policy optimization algorithm is proposed. To enhance the variability of opponent's strategy in the process of policy network self-learning, a baseline policy model selection method based on model performance and model diversity is proposed. The experiments demonstrate that the proposed method is effective and stable in air-sea joint operation decision. In the 4th Wargaming Competition hosted by Chinese Institute of Command and Control, the winning rate in more than 100 rounds against regular decision-making algorithm and human confrontation was 97%, which was about 20% higher than that of regular decision-making algorithms.

Key words: air-sea joint operation; deep reinforcement learning; proximal policy optimization; intelligent decision

CLC number: TP273 **Document code:** A **Article ID:**1005-1120(2023)01-0025-12

0 Introduction

Reinforcement learning is an important research field in machine learning. It does not require supervisory signals. It interacts with the environment with a trial-and-error mechanism, balances exploration and utilization in an environment with unknown models, and learns optimal strategies by maximizing cumulative rewards. Deep reinforcement learning combines the advantages of deep neural networks and reinforcement learning, which can be used to solve the perception decision-making problem of agents in complex high-dimensional state space.

In 2015, the Google DeepMind team[1] proposed the deep Q network (DQN), which innovatively combines convolutional neural networks with Q-learning. By using the experience replay technique and the fixed-target Q network, the instability and divergence caused by the neural network's nonlinear action value function approximator are effectively handled, greatly improving the applicability of the reinforcement learning method. At the same time, the experience replay technology increases the use efficiency of historical data, and breaks the correlation between data by using random sampling, which stabilizes the training process of action value function furtherly. In 2016, artificial intelligence AlphaGo[2] based on deep reinforcement learning and Monte Carlo tree search defeated the top professional Go players, which has attracted worldwide attention. Subsequently, DeepMind introduced artificial intelligence AlphaGoZero[3] and AlphaZero[4] based

*Corresponding author, E-mail address: yed@hit.edu.cn.

How to cite this article: SONG Xiaocheng, FENG Shuting, LI Zhi, et al. A new intelligent decision-making method for air-sea joint operation based on deep reinforcement learning[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2023, 40(1): 25-36.

http://dx.doi.org/10.16356/j.1005-1120.2023.01.003

on reinforcement learning and greatly promoted the development of reinforcement learning technology.

In recent years, deep reinforcement learning has been widely used in games[5], robots[6], recommendation systems[7-8] and other fields. In 2017, OpenAI used reinforcement learning to defeat top human players in the real-time strategy game Dota2[9]; in 2019, Google' s Deepmind AlphaStar[10] defeated StarCraft human players. In the same year, OpenAI trained the humanoid robot hand Dactyl[11] to manipulate objects flexibly; in 2018, DeepMimic[12] simulated humanoid robots which mastered difficult motion modes.

With the successful application of algorithms such as DQN, deep reinforcement learning has gradually been applied to military decision-making. For different practical applications, scholars have designed a series of methods from different technical approaches, such as military decision-making method based on game theory[13-15], military decision-making method based on optimization theory[16-18] and so on. Conde et al.[19] used genetic algorithm to solve the problem of conflict avoidance according to the characteristics of fighters. Smith et al.[16] used the bilateral model to solve the aircraft planning problem, but their algorithm is easy to fall into local optimum. Cui et al.[20] used particle swarm optimization to analyze the conflict avoidance problem, but due to the lack of time series data, its convergence accuracy is low. Burak[21] proposed a path planning method for unmanned aerial vehicle (UAV) team using reinforcement learning. Yang et al.[22] designed a short-range UAV air combat decision-making method based on DQN, which improved the decision-making efficiency and algorithm performance.

It can be seen that although deep reinforcement learning has become an effective solution to military decision-making problems and being widely used, most of the current methods focus on a single type of weapon agent. There is still a lack of research on joint decision-making of multiple types of weapons, such as air-sea joint operations. Even in the few joint operation algorithms, most of them are still rule-based, which face serious bottlenecks in terms of handling timeliness and robustness, and are far from meeting the complex confrontation requirements of modern warfare.

Compared with the traditional application, the difficulties of joint operation and decision-making mainly lie in the complex confrontation characteristics of non-omniscience, uncertainty, high confrontation and strong game, together with its non-linearity, randomness, fuzziness and other uncertainty characteristics. As well as the multi-domain complex of opposing forces, complex opposing environment, unpredictable situation evolution and many other characteristics also make a difference. For example, in the process of joint air-sea operation, the sea attack and defense confrontation of aircraft carrier battle group needs to make decisions under thousands of constraints in a very limited time against dozens of combat equipments, hundreds of combat sequence groups, such as airman, ship-based air defense weapons, and jamming equipment. The combat unit needs to respond to the autonomously generated maneuvering instructions and make tactical action selections timely and accurately by sensing the environmental state information. Its problem-solving dimension far exceeds the general two-person multi-round game problem.

The main contributions of this work are as following: First, we propose a general framework for joint operations of multi-type equipment with different functions in air-sea confrontation. Second, an improved proximal policy optimization (PPO) algorithm is proposed to mitigate the instability in case of large variance of advantage function. Third, aiming at the variability of rival strategy in the self-learning process of policy network, a baseline policy model selection method is proposed.

Combining the above points, we construct a new intelligent decision-making approach for joint operations for air-sea confrontation. The proposed new approach is numerically simulated on the simulation deduction platform. The simulation results show that the method can make effective and stable air-sea joint operation decisions in the local scale conflict scenario of single aircraft carrier combat group. The superiority and advancement of the method have been fully verified in the 4th Wargaming Competition hosted by Chinese Institute of Command and Control. The method has achieved 97% winning rate in more than 100 rounds against rule decision algorithms and human decision.

1 Intelligent Decision Method for Air-Sea Joint Operation

The objective of air-sea joint operation decision-making is to destroy the rival's combat equipment to the maximum extent and minimize our casualties through our own cooperation in the encounter of aircraft carrier battle groups. The forces of both sides are limited to a single aircraft carrier and escort destroyers, carrier-based aircraft, while the red and blue forces are completely the same. The essence of the decision-making problem of the joint air-sea operation mission is a complex game between the red (our) and blue (enemy) agents. The complexity is mainly due to the different action space and decision rules of different type forces, as well as the uncertainty and agnosticity of the game process. For example, the destroyer mainly completes the major attack mission in the assault mission, but also cooperates with the air fighters to complete the mission of air supremacy and aircraft carrier attack. At this time, the destroyer must adjust its role and action in time on the basis of maximizing the overall reward.

Faced with the complex types of forces and varied game environment, this paper constructs a unified air-sea joint operation network model by using perceptron, deep long-short term memory network and actor-critic structure. In the network structure, the action space and decision rules of different forces can be uniformly represented and flexibly configured. The joint operation network model is shown in Fig.1, which mainly includes three parts: State (network input), action (network output) and network model structure.

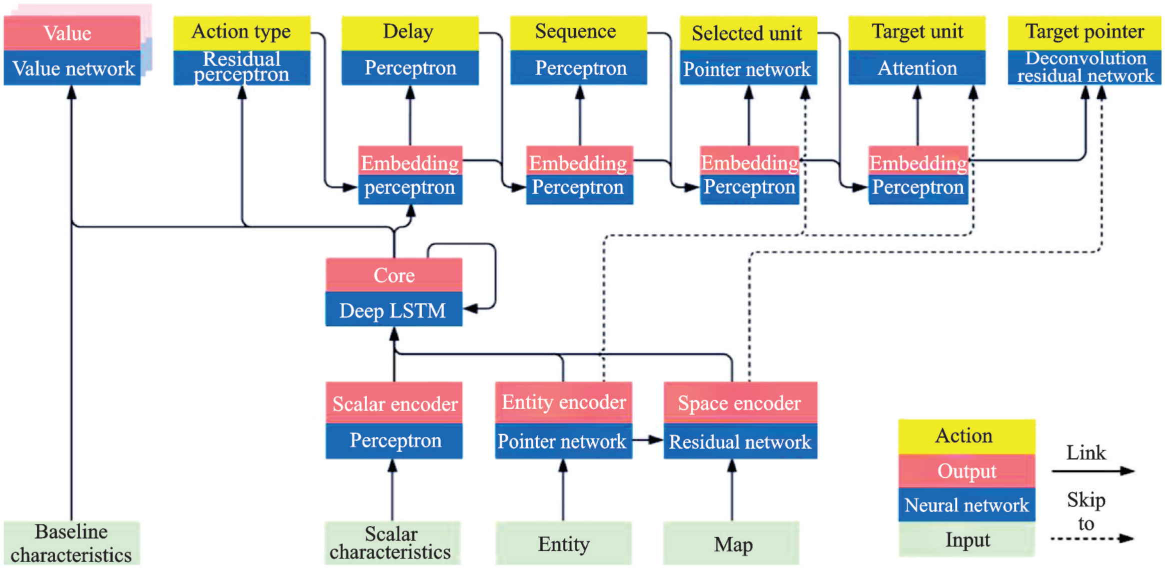


Fig.1 Overall architecture of the air-sea joint operation

1.1 Environmental state of joint operation

The environmental state of joint operation is divided into four parts: Entity information, map information, player data information, and combat statistics information, as shown in Fig.2.

(1) Entity information: Entity refers to the aircraft carrier, destroyer, aircraft, etc. in the current environment. The entity information of each entity is represented by a vector. For example, the entity information vector of an aircraft includes information such as firepower, grade, position, and flight time at the current moment. For all the entity information at the current moment, the environment will input N vectors with lengths of K_i to the neural network to represent the specific information of the N entities that the agent can see at this moment.

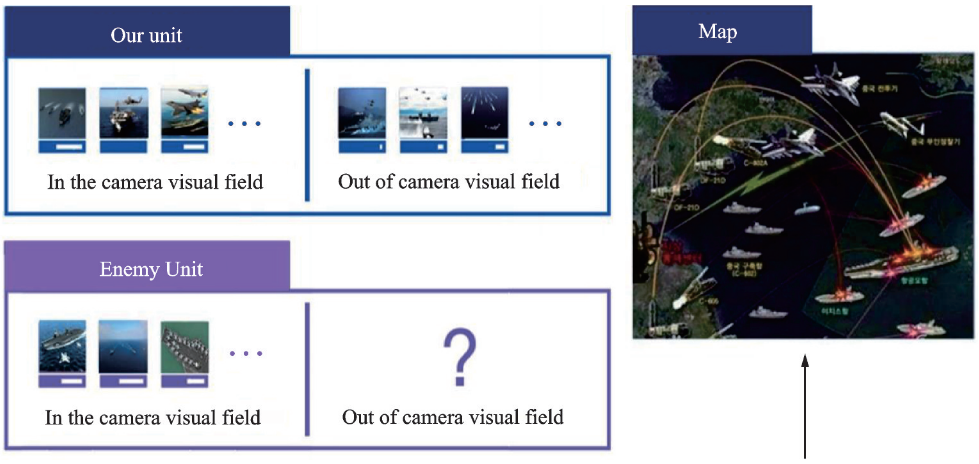

Our unit
In the camera visual field
Out of camera visual field
Enemy Unit
In the camera visual field
?
Out of camera visual field
Map

Fig.2 Environment state information

(2) Map information: The map information represents the confrontation situation in the current state, and the map information is fed into the neural network in the form of a matrix.

(3) Player data information: The player's ID and level information (scalar information) in the current state.

(4) Game statistics information: The location of the view and the start time of the current game (scalar information).

For scalar information, multi-layer perceptron (MLP) is used to get the corresponding vector, which is an embedding process. For entity information, the transformer architecture[23] is used as an encoder to get its vector. For map information, the ResNet architecture is used as the encoder to obtain a fixed-length vector.

1.2 The action information

The action information includes six parts: Action type, delay, the sequence of execution action, the selected unit, the target unit, and whether it is repeated. The output action is related back and forth.

(1) Action type: The type and attribute of action to be performed at the next moment. Such as the destroyer will go forward or turn left, and its speed. The action type uses the embedded vector of the deep long-short term memory network as an input, uses the residual multi-layer perceptron to obtain the output of the Softmax activation function, and then passes it to the next sub-model for embedding.

(2) Delay: How long to wait before receiving network input. The delay inputs the result of the embedded action type and the result of the deep long-short term memory network into the multi-layer perceptron to output, and passes it to the next sub-model for embedding.

(3) The sequence of execution action: Whether to perform the action immediately. For example, for fighter A, whether to attack directly or stand by after reaching the destination. The sequence of execution action inputs the delayed result and the embedded result together into the multilayer perceptron to obtain the output, which is passed to the next sub-model for embedding.

(4) Selected unit: The entity that performs the action in the first step. For example, the type of action we want to carry out in the first step is to control the fighter, so we should choose which fighter to control. The selected unit sends the results of the sequence, the embedded results, and all the results after the entity coding (non-average results) together into the pointer network to obtain the results and pass them to the next sub-model for embedding. The input of the pointer network is a sequence, the output is another sequence, and the elements of the output sequence come from the input sequence.

(5) Target unit: The selected destination or attack object. After the aircraft is given an action, the target unit is a certain location to go to or the ri-

val's entity to be attacked, etc. The target unit and the target area are either selected. For the target unit, the attention mechanism is used to get the optimal action to effect it. For the target area, a deconvolution residual network is used to deconvolute the embedded vector to the size of the map, thereby performing the corresponding action of moving to a certain target area.

(6) Whether to repeat: If the fighter A needs to attack continuously, then you do not need to calculate the next action through the network, just repeat the previous action.

The action types of aircraft carriers and escort destroyers are relatively simple, mainly including heading and its speed. The action types of fighters are complex. Considering the basic actions of fighters and the preferences or intentions of decision makers, the action types of fighters are to standstill, linear accelerating/decelerating, avoidance, active attack, and active defense. The mapping relationship between motivations and actions can be expressed in Table 1.

Table 1 Mapping relationship between motivation and maneuver

Number	Motivation	Action
1	Standstill	Low speed flight
2	Linear accelerating	Accelerated flight
3	Linear decelerating	Decelerated flight
4	Avoidance	Steer 90°
5	Active attack	Attack rival fighter
6	Active defense	Return voyage

1.3 Network model and training algorithm

The global architecture and network structure is shown in Fig.1, which is composed of three parts, as described above. The observation at each step are concatenated into a fixed-length vector as single step feature, which is composed of scalar characteristics, entity information and map information encoded by MLP, pointer network and ResNet network structures, respectively. The encoded vectors for consecutive frames (simulation steps) are then processed by a deep LSTM network, which maintains memory between steps, and are further processed by sequenced MLP networks before transformed into action outputs by decoders described in Section B and value estimation by another MLP network.

Besides the network architecture described above, the network training algorithm, i.e. the reinforcement learning algorithm, is yet another important element in agent training. Currently, the commonly used reinforcement learning algorithms include trust region policy optimization (TRPO)[24] and its approximation algorithms penalty-PPO and clip-PPO[25]. To balance between performance and simplicity, the clip-PPO algorithm is adopted as the baseline and is further improved as clip2-PPO in this paper. Taking the complexity of air-sea joint operation task into consideration, we also uses rule data to complete the model initialization and improves the model performance through self-learning.

1.3.1 Model initialization

When the training begins, the model is initialized by using expert rule-based data. The input of the model is the collected expert rule-based combat data, and the output is the trained neural network. The approach is to send the collected expert combat data, that is, the decoded game state at each moment, into the network to obtain the probability distribution of each action, and to calculate the output of the model and the Kullback-Leibler (KL) divergence of the expert combat data. Then use the KL divergence to optimize the network. Different loss functions need to be used in KL divergence. For example, the loss of action type, that is, the loss of classification problem uses cross entropy; calculated mean square errors are used for regression problems such as target location. After supervised learning, the probability distribution of model output can be similar to that of expert's output.

1.3.2 Improved PPO algorithm

The purpose of reinforcement learning is to maximize the expected reward by optimizing the policy based on the above initial model. This paper uses the improved PPO algorithm and actor-critic structure to construct the reinforcement learning model. The actor-critic model is used to train the value function $V_\theta(s_t)$ and the policy network $\pi_\theta(a_t|s_t)$, and the sampled data of experience replay is used to update them alternately. The loss of value function is

$$J^c(\theta)=E_t[(R_t^\gamma-V_\theta(s_t))^2] \tag{1}$$

where R_t^γ is the return of discount factor γ.

For the policy network $\pi_\theta(a_t|s_t)$, if the same policy algorithm is used, that is, the learning agent is the same as the agent interacting with the environment, once the parameter θ is updated to θ', the previously sampled data is not applicable, and the data needs to be resampled to update the parameters again, which is very inefficient. In order to improve efficiency, this paper changes the same policy algorithm into a different one, uses another policy $\pi_{\theta'}$, another actor θ' to interact with the environment, and trains θ with the same batch of sampled data. It can use the data sampled by θ', perform gradient ascent, and update parameters for multiple times.

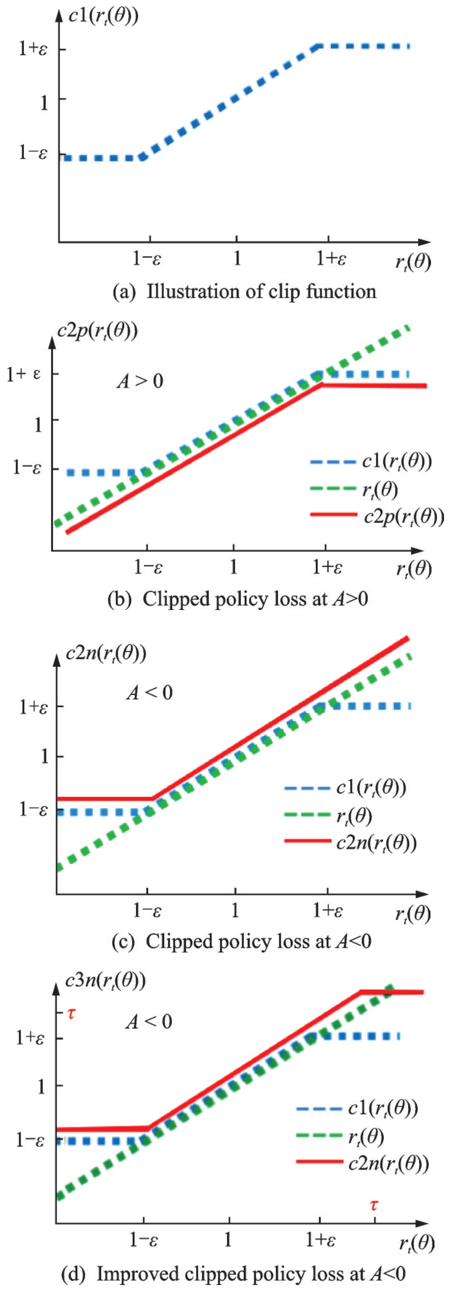
Based on the above ideas, the policy loss function is designed as

$$\begin{aligned}J^a(\theta)=E_t[&\min(r_t(\theta)A^\theta(s_t,a_t),\\&\text{clip}(r_t(\theta),1-\varepsilon,1+\varepsilon)A^\theta(s_t,a_t))]\end{aligned} \tag{2}$$

where importance sampling coefficient $r_t(\theta)=\pi_\theta(a_t|s_t)/\pi_{\theta'}(a_t|s_t)$, present policy π_θ, past policy $\pi_{\theta'}$, clip function clip(), defined by

$$\text{clip}(x,a,b)=\begin{cases}a & x<a\\ b & x>b\\ x & a<x<b\end{cases} \tag{3}$$

$l(r_t(\theta),A)=\min(r_t(\theta)A^\theta(s_t,a_t)$, and clip $(r_t(\theta),1-\varepsilon,1+\varepsilon)A^\theta(s_t,a_t))$ are used to present the policy loss at time t. Here, $A=A^\theta(s_t,a_t)$ is called the advantage function of the state-action pair (s_t,a_t), that is, the advantage of an action a_t relative to the average under state s_t. The advantage function helps to reduce variance, improve learning efficiency and make learning more stable. When $A>0$, it means that the state-action pair (s_t,a_t) is good. We hope to increase its probability, or vice versa. Fig.3(b) and Fig.3(c) show the policy loss $l(r_t(\theta),A)$ when $A>0$ and $A<0$, respectively.



(a) Illustration of clip function

(b) Clipped policy loss at A>0

(c) Clipped policy loss at A<0

(d) Improved clipped policy loss at A<0

Fig.3 Comparison of policy loss

It can be seen from Fig.3(c) that when $r_t(\theta)\gg 1$ and $A<0$, the variance of the policy loss $J^a(\theta)$ will be large, which will easily lead to the instability of the learning process. Therefore, the improved policy loss function is

$$J^a(\theta)=E_t[\tilde{l}(r_t(\theta),A)] \tag{4}$$

where

$$\tilde{l}(r_t(\theta),A)=\begin{cases}\min(r_t(\theta),\text{clip}(r_t(\theta),1-\varepsilon,1+\varepsilon))A^\theta(s_t,a_t)) & A^\theta(s_t,a_t))\geqslant 0\\ \text{clip}(r_t(\theta),\text{clip}(r_t(\theta),1-\varepsilon,1+\varepsilon),\tau)A^\theta(s_t,a_t)) & A^\theta(s_t,a_t))<0\end{cases} \tag{5}$$

$\tau>1+\varepsilon$ is the truncated upper bound. The modified policy loss when $A<0$ is shown in Fig.3(d). For the convenience of description, the original PPO algorithm is called as clip-PPO, the improved algorithm is called as clip2-PPO.

The symbols in Fig.3 are shown below.

$$\begin{cases} c1(r_t(\theta)) = \mathrm{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon) & \\ c2p(r_t(\theta)) = \min(r_t(\theta), \mathrm{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)) \cdot A & A > 0 \\ c2n(r_t(\theta)) = \min(r_t(\theta), \mathrm{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)) \cdot A & A < 0 \\ c3n(r_t(\theta)) = \mathrm{clip}(r_t(\theta), \mathrm{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon), \tau)) \cdot A & A < 0 \end{cases} \tag{6}$$

1.4 Process of self-learning

Self-learning is to fight with yourself. The key to self-learning is how to select the opponent in the training process and to archive the current strategy in what circumstance. In this paper, the purpose of self-learning is to further improve the performance and applicability of the current policy model. To facilitate the narrative, the following definitions are introduced.

Definition 1 If $r_t \in (1-\varepsilon, 1+\varepsilon)$, that is, the current learning policy π_θ is similar to the crawling policy π_u, (s_t, a_t) is called as an ordinary sampling, and recorded as $I(s_t, a_t) = 0$.

Definition 2 If $A < 0$ and $r_t \notin (1-\varepsilon, 1+\varepsilon)$, (s_t, a_t) is called as an unordinary sampling, and recorded as $I(s_t, a_t) = 1$.

Definition 3 For a round of training data $\{(s_t, a_t) | t = 1, 2, \cdots, N\}$, the more unordinary sampling is, the more diversiform this sampling round is. We use $k = 1/N \sum_t I(s_t, a_t)$ to present the sampling diversity.

In order to effectively improve the performance and sample diversity of the policy model, this paper proposes a new performance best-n (PB-n) and sampling diversity best-n (DB-n) model selection method. PB-n refers to the n models with the best performance in the historical model, and DB-n the n models with the best sampling diversity in the historical model.

Current models continually improve their performance by competing against the above $2n$ models. Compared with the traditional self-learning method, the innovation of this paper is that the adversarial model comprehensively considers the performance of historical models and the data diversity of historical models. The performance of the historical model helps to improve the performance of the current model, and the data diversity of the historical model helps to improve the adaptability and autonomy of the model. Similarly, if the current policy model can defeat the above $2n$ models, or the sampling diversity of the current model is better than that of the best-n candidate model, archive the current policy.

2 Numerical Simulation

In this paper, the simulation samples are generated based on the rules of the 4th Wargaming Competition hosted by Chinese Institute of Command and Control. In order to obtain representative state information and training sample set, the red and blue combat units are uniformly and randomly generated, and the heading angle is randomly initialized in the form of normal distribution. According to the change of threat situation, the action selection of both red and blue is in order to get minimum threat index. The threat situation degree of the combat unit at each moment involves indicators such as direction, distance, and speed, which is a mapping of reward and punishment signals.

The primary innovation of this paper is the improvement of policy loss estimation in PPO algorithm. In order to verify the effectiveness of the proposed method, Fig.4 shows the comparison of policy loss between the traditional PPO algorithm and the PPO algorithm in this paper. It can be seen from Fig.4 that compared with the traditional clip-PPO algorithm, the clip2-PPO algorithm in this paper has a smoother loss curve and less loss. The main reason is that the variance of policy loss of clip-PPO increases when $A < 0$, which leads to inaccurate esti-

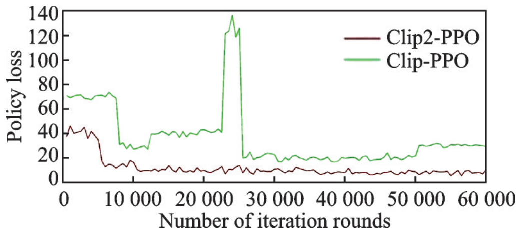
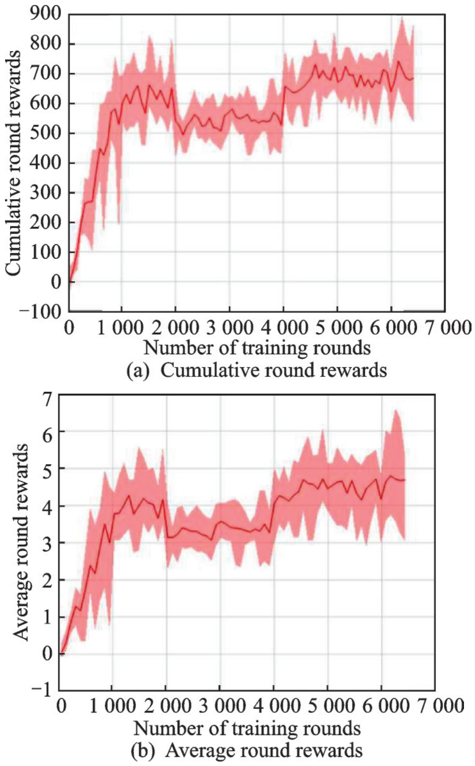


Fig.4 Comparison of policy loss curves

mation of policy loss expectation. The clip function and policy loss of the modified algorithm clip2-PPO effectively suppress the negative effect of the advantage function, and the learned policy network converges faster and more smoothly.

To verify the effectiveness of the improved PPO algorithm furtherly, Fig.5 shows the reward value distribution of the training process.



(a) Cumulative round rewards

(b) Average round rewards

Fig.5 Decision reward function

From Fig.5, it can be seen that the cumulative and average round reward functions show an upward trend as a whole during the training process, and finally basically stabilize at a higher cumulative reward value. The result shows that in the above training process, the combat unit decision-making agent can successfully achieve the combat target of quickly entering the mission area and effectively attacking the enemy unit. At the same time, from Fig.5(a), it can be seen that the cumulative reward and average reward function decreased from 1 500 to 2 000 rounds. The decrease of this reward is due to the loss caused by the agent's encounter with enemy fighters when flying to the mission area, which makes some of our fighters unable to enter the mission area. In view of this situation, the agent learned to avoid enemy fighters in the early stage, and bypassed from the side to avoid the above problems, but resulting in a decline in the reward function. After several rounds of training, the reward function rises again. As can be seen from Fig.5, the agent chooses to shoot out the air-to-air missiles it carries as soon as possible when encountering enemy fighters to protect itself, and the reward is increased by effectively destroying the enemy aircraft.

For the effect analysis of fighters in air combat, this paper mainly focuses on the implementation of fighters flying to the mission area and the micro-operation avoidance of the enemy's incoming guidance weapons during air combat. The results of the confrontation process are shown in Fig.6, and our side is the red side. In terms of the execution of fighters flying to the mission area, the octagonal area in the figure corresponds to the coordinated air combat patrol area. As can be seen from the process figure, the fighters successfully flew to the patrol mission area, which verified the ability of the agent to execute the decision of the task planning level.

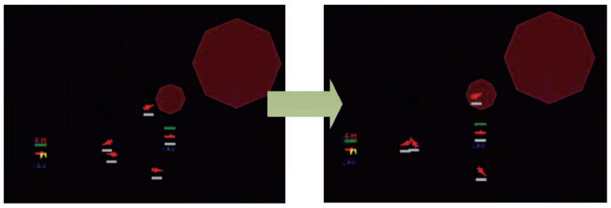


Fig.6 Execution of fighter area of responsibility

In terms of micro-operation avoidance of the enemy's incoming weapons, it can be seen from the results in Fig.7 that, through the training in the confrontation process, when the combat unit decision-making agent detects an incoming weapon, it can realize continuous tactical avoidance action and effective avoidance of the incoming weapons by actively throwing decoy jamming on

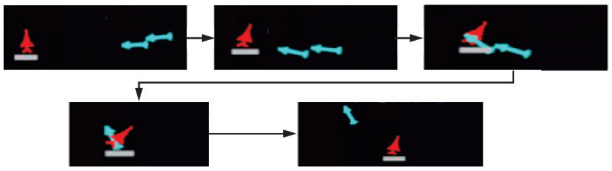


Fig.7 Incoming weapon micro-operation avoidance situation

the one hand, and continuously adjusting its own course, changing its own speed and height on the other hand. Effectively improve the survivability of our fighters.

The above results show that, through in-depth intensive learning and training, our fighters can go to the mission area to execute the combat mission designated by the mission planning layer under the condition of maximum survival.

This paper also compares two typical rule algorithms (air-to-air cooperative rule algorithm and joint air-defense rule algorithm) with the air-sea joint operation agent, and 50 kinds of rule decision algorithms are chosen to fight against over 1 000 games randomly. The combat effect is shown in Table 2 and Fig.8.

Table 2 Comparison test results of each decision-making algorithm (average value)

Algorithm	Air-to-air cooperative rule algorithm	Joint air-defense rule algorithm	Air-sea joint operation agent	PB-2n	DB-2n
Loss of fighters	2.60	3.59	3.30	3.50	3.59
Loss of ships	0.21	1.02	0.01	0.05	0.12
Units destroyed	13.48	9.06	15.37	15.12	13.90
Ships destroyed	0.00	0.00	0.96	0.60	0.40
Scores	1 125.29	−1 082.67	3 428.58	2 884.82	1 421.53
Win rate/%	79	0	97	90	84

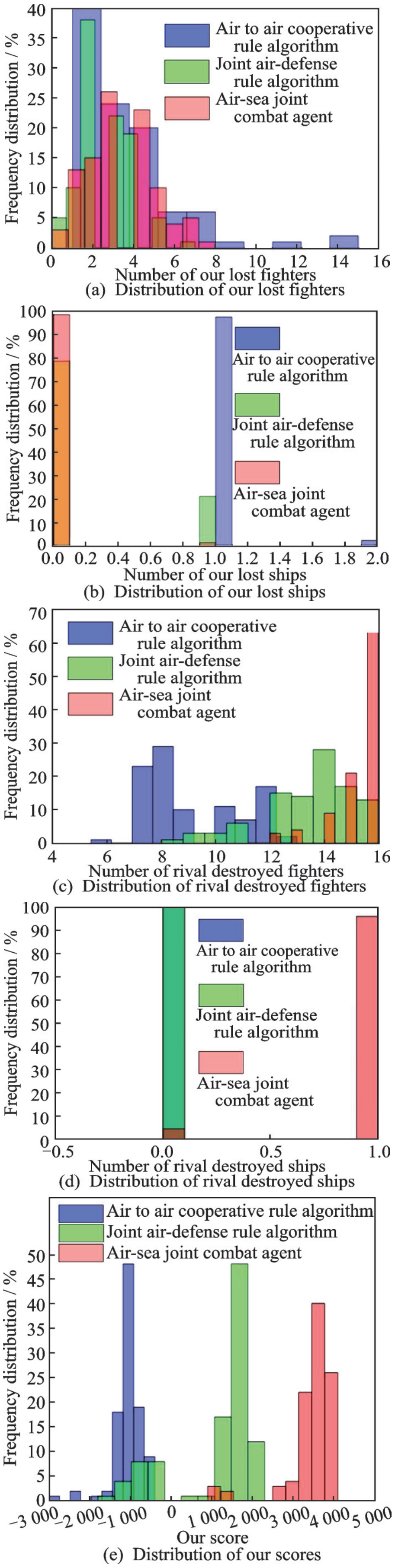
From the results of Table 2, it can be seen that the air-to-air cooperative rule algorithm effectively completes the strike and destruction of enemy fighters. Only using it can complete the destruction of an average of 13.48 of all 16 fighters, which greatly reduces the enemy's combat capability. However, the protection ability of our ship is slightly insufficient. It can be seen from the results of Fig.5(b) that in 1 000 rounds of confrontation, our destroyer has a probability of about 20% being destroyed, showing the lack of robustness of the air-to-air cooperative rule algorithm.

Due to the lack of air combat capability (air-to-air weapons of air-defense fighters are obviously insufficient compared with air-to-air fighters), the joint air defense rule algorithm has weak air combat capability, poor ability to attack the enemy and defend allied ship (almost all the destroyer is destroyed), but with the help of the destroyers, the incoming fighters can still be hit to a certain extent. At the same time, it can be seen from the loss of our fighters that this rule algorithm effectively preserves the combat capability of our air-to-surface fighters, but the short board of air combat is obvious and the robustness is insufficient.

The performance of reinforcement learning algorithm in this paper greatly exceeds the first two rule algorithms. This is mainly due to the feature representation ability of deep learning and the decision-making ability of reinforcement learning. The proposed algorithm participated in the 4th Wargaming Competition hosted by Chinese Institute of Command and Control in December 2020, and maintained a complete victory in more than 100 rounds of rule decision algorithm and human confrontation, which fully verified the effectiveness and superiority of the proposed algorithm in this paper.

Another innovation of this paper is the selection of candidate policy model in self-learning process. In order to verify the effectiveness of the proposed method, the proposed method (PB-n + DB-n) and the case only using PB-2n or DB-2n are simulated. The PPO algorithm of this method, PB-2n and DB-2n is exactly the same, and the difference is only the selection strategy of candidate models in the self-learning process. Table 2 shows the comparative performance results of the above methods.

The candidate model of PB-2n is from the 2n model with the best performance in testing the historical model during training, and the candidate model of DB-2n is from the 2n model with the best sampling diversity during training. The 2n candidate models of PB-n + DB-n are n from the best performance model and n from the best sampling diversity



(a) Distribution of our lost fighters

(b) Distribution of our lost ships

(c) Distribution of rival destroyed fighters

(d) Distribution of rival destroyed ships

(e) Distribution of our scores

Fig.8 Operation effectiveness evaluation under different algorithm configurations

model.

It can be seen from Table 2 that the result obtained by using only the best performance or only the best diversity is inferior to the method in this paper. This is because the PB-2n selection strategy only focuses on the performance of the policy model and ignores the diversity of samples during model training and vice versa. The method in this paper uses both the best performance strategy and the best sampling diversity strategy, and the performance and diversity of the model obtained by self-learning of the rules are fully considered.

Through simulation, the final winning rate of PB-2n self-learning strategy is 90%, the final winning rate of DB-2n self-learning strategy 8%, and the final winning rate of PB-n+DB-n 97%. This huge performance difference verifies the importance of model selection strategy and the high complementarity between performance best strategy and sampling diversity best strategy.

3 Conclusions

Aiming at the difficulty of high uncertainty in multi-weapon complex combat scenarios of air-sea joint operation, this paper proposes a new intelligent decision-making method for air-sea joint operation based on deep reinforcement learning. This paper innovatively proposes an improved clip2-PPO algorithm and a new strategy for selecting candidate models in the self-learning process. Aiming at the instability of the policy network learning process and the defects of the PPO algorithm, a baseline policy model based on model performance and model diversity is proposed. The strategy selection method mainly aims at the variability of the opponent strategy in the self-learning process of the policy network. Through the improvement of PPO algorithm and the new strategy of selecting candidate models in self-learning process, a policy model with faster convergence, higher precision and wider adaptability is obtained. The proposed new method is numerically verified on the simulation deduction platform. In the scene of local scale conflict of single aircraft carrier combat group, in the 4th Wargaming Competition hosted by Chinese Institute of Command and Control, this method has a winning rate of 97% in more than 100 rounds of confrontation with rule decision algorithm and human, which is about 20% higher than that of traditional rule decision algorithm. The work of this paper focuses on the scene decision-making of local scale conflict of single carrier combat group, so the follow-up research will focus on the intelligent decision-making technology and application in the scenario of increasing the number of equipment types and enhancing the heterogeneity of combat units.

References

[1] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[2] SILVER D, HUANG A, MADDISON C, et al. Mastering the game of Go with deep neural networks and tree search[J]. Nature,2016, 529: 484-489.

[3] SILVER D, SCHRITTWIESER J, SIMONYAN K, et al. Mastering the game of Go without human knowledge[J]. Nature,2017, 550: 354-359.

[4] SILVER D, HUBERT T, SCHRITTWIESER J, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play[J]. Science, 2018,362: 1140-1144.

[5] PEROLAT J, DE VYLDER B, HENNES D, et al. Mastering the game of stratego with model-free multiagent reinforcement learning[J]. Science, 2022, 378 (6623): 990-996.

[6] LIU S, LEVER G, WANG Z, et al. From motor control to team play in simulated humanoid football[J]. Sci Robot, 2022, 7(69):104-161.

[7] WANG Kai, ZOU Zhene, DENG Qilin, et al. Reinforcement learning with a disentangled universal value function for item recommendation[C]//Proceedings of the AAAI Conference on Artificial Intelligence. [S.l.]: AAAI, 2021.

[8] JI Shenggong, WANG Zhaoyuan, LI Tianrui, et al. Spatio-temporal feature fusion for dynamic taxi route recommendation via deep reinforcement learning[J]. Knowledge-Based Systems, 2020, 205: 106302.

[9] BERNER C, BROCKMAN G, CHAN B, et al. Dota 2 with large scale deep reinforcement learning[EB/OL]. (2019-12-13). http://xueshufan. com/publication/2996037775.

[10] VINYALS O, BABUSCHKIN I, CZARNECKI W M. Grandmaster level in StarCraft Ⅱ using multi-agent reinforcement learning[J]. Nature, 2019, 575: 350-354.

[11] ANDRYCHOWICZ O M, BAKER B, CHOCIEJ M, et al. Learning dexterous in-hand manipulation[J]. The International Journal of Robotics Research, 2020, 39(1): 3-20.

[12] PENG X B, ABBEEL P, LEVINE S, et al. DeepMimic: Example-guided deep reinforcement learning of physics-based character skills[J]. ACM Transactions on Graphics, 2018. DOI: 10.1145/3197517.3201311.

[13] XU G, WEI S, ZHANG H. Application of situation function in air combat differential games[C]//Proceedings of 36th Chinese Control Conference. Dalian, China: IEEE, 2017: 5865-5870.

[14] PARK H, LEE B Y, TAHK M J. Differential game-based air combat maneuver generation using scoring function matrix[J]. International Journal of Aeronautical and Space Sciences, 2016, 17(2): 204-213.

[15] XIE Rongzeng, LI Jieying, LUO Delin. Research on maneuvering decisions for multi-UAVs air combat[C]//Proceedings of the 11th IEEE International Conference on Control and Automation. Taiwan, China: IEEE, 2014: 767-772.

[16] SMITH R E, DIKE B A, MEHRA R K, et al. Classifier systems in combat: Two-sided learning of maneuvers for advanced fighter aircraft[J]. Computer Methods in Applied Mechanics and Engineering, 2016, 186(2/3/4): 421-437.

[17] HUANG Changqiang, DONG Kangsheng, HUANG Hanqiao, et al. Autonomous air combat maneuver decision using Bayesian inference and moving horizon optimization[J]. Journal of Systems Engineering and Electronics, 2018, 29(1): 86-97.

[18] PAN Q, ZHOU D, HUANG J, et al. Maneuver decision for cooperative close-range air combat based on state predicted influence diagram[C]//Proceedings of IEEE International Conference on Information & Automation. Macao, China: IEEE, 2017: 726-731.

[19] CONDE R, ALEJO D, VIGURIA A, et al. Conflict detection and resolution method for cooperation unmanned aerial vehicles[J]. Journal of Intelligent & Robotic Systems,2012,65(1/2/3/4): 495-505.

[20] CUI Liwei, SHI Weiren, LIU Xiangming, et al. Air conflict resolution based on genetic algorithm and particle swarm optimization[J]. Computer Engineering and Application,2013,49(7): 263-266.

[21] BURAK Y. Cooperative planning for an unmanned combat aerial vehicle fleet using reinforcement learn-

ing[J]. Journal of Aerospace Information Systems, 2021,18(10): 739-750.

[22] YANG Q, ZHANG J, SHI G, et al. Maneuver decision of UAV in short-range air combat based on deep reinforcement learning[J]. IEEE Access, 2020, 8: 363-378.

[23] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach, CA, USA: ACM, 2017:6000-6010.

[24] SCHULMAN J, LEVINE S, MORITZ P, et al. Trust region policy optimization[J]. Computer Science, 2015,DOI: 10.48550/arXiv.1502.05477.

[25] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal policy optimization algorithms[EB/OL]. (2021-8-28).http://arxiv.org/pdf/1707.06347.pdf.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Nos.62073102, 62203145) and the China Postdoctoral Science Foundation (No.2022M710948).

Authors Mr. **SONG Xiaocheng** is a senior engineer of Beijing Institute of Electronic System Engineering. His research interests include guidance and control, command control and combat planning design.

Prof. **YE Dong** received the B.S., M.S., and Ph.D. degrees in aerospace engineering from Harbin Institute of Technology, Harbin, China, in 2007, 2009, and 2013, respectively. Since 2021, he has been a professor with the Research Center of Satellite Technology, Harbin Institute of Technology. His research interests include hard ware-in-loop simulation technique, spacecraft dynamics, and control technique.

Author contributions Mr. **SONG Xiaocheng** designed and proposed a new intelligent decision-making method for air-sea joint operations based on deep reinforcement learning, compiled an improved approximate policy optimization algorithm and a baseline policy model selection method based on model performance and model diversity, and wrote the manuscript. Ms. **FENG Shuting** contributed data and model components for the state and action design, and carried out partial simulation verification. Prof. **LI Zhi** provided the experiment resource of this article. Mr. **JIA Zhengxuan** designed implemented the global agent architecture and conducted part of the numerical simulation analysis. Mr. **ZHOU Guojin** gave some useful suggestions and technical support for agent design and training. Prof. **YE Dong** contributed to the discussion and background of the study. All authors commented on the manuscript draft and approved the submission.

Competing interests The authors declare no competing interests.

(Production Editor: WANG Jing)

基于深度强化学习的空海联合作战智能决策新方法

宋晓程[1]，冯舒婷[1]，李陟[1]，贾政轩[1]，周国进[2]，叶　东[3]

(1.北京电子工程总体研究所,北京 100854, 中国; 2.北京华戍防务技术有限公司,北京 100084,中国;
3.哈尔滨工业大学卫星技术研究所,哈尔滨 150080,中国)

摘要:针对空海联合作战中多装备复杂作战场景不确定性高的难点,提出了一种基于深度强化学习的空海联合作战智能决策新方法。为了统一表示复杂网络的输入、输出及其对应关系,提出了综合利用感知机、深度长短时记忆网络及actor-critic结构的方法。针对策略网络学习过程中的不稳定性及近似策略优化算法的缺陷,提出了改进的近似策略优化算法;针对策略网络自学习过程中对手策略的易变性,提出了基于模型性能和模型多样性的新策略以对于基线策略模型进行选择。实验结果表明,该方法在空海联合作战决策中是有效和稳定的。在第四届中国指控学会兵棋推演专项赛中,本方法在百余轮与规则决策算法及人类的对抗中胜率达到97%,较规则决策算法提升20%左右。

关键词:空海联合作战;深度强化学习;近似策略优化;智能决策