



## OPEN ACCESS

EDITED BY  
Chen Qiao,  
Xi'an Jiaotong University, China

REVIEWED BY  
Andreas Schweiger,  
Airbus, Netherlands  
Tongle Zhou,  
Nanjing University of Aeronautics and  
Astronautics, China

\*CORRESPONDENCE  
Hongpeng Zhang  
hphzhang@126.com

RECEIVED 17 July 2022  
ACCEPTED 05 October 2022  
PUBLISHED 25 October 2022

CITATION  
Zhang H, Zhou H, Wei Y and Huang C  
(2022) Autonomous maneuver  
decision-making method based on  
reinforcement learning and Monte  
Carlo tree search.  
*Front. Neurobot.* 16:996412.  
doi: 10.3389/fnbot.2022.996412

COPYRIGHT  
© 2022 Zhang, Zhou, Wei and Huang.  
This is an open-access article  
distributed under the terms of the  
[Creative Commons Attribution License  
\(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or  
reproduction in other forums is  
permitted, provided the original  
author(s) and the copyright owner(s)  
are credited and that the original  
publication in this journal is cited, in  
accordance with accepted academic  
practice. No use, distribution or  
reproduction is permitted which does  
not comply with these terms.

# Autonomous maneuver decision-making method based on reinforcement learning and Monte Carlo tree search

Hongpeng Zhang\*, Huan Zhou, Yujie Wei and  
Changqiang Huang

Aeronautics Engineering College, Air Force Engineering University, Xi'an, China

Autonomous maneuver decision-making methods for air combat often rely on human knowledge, such as advantage functions, objective functions, or dense rewards in reinforcement learning, which limits the decision-making ability of unmanned combat aerial vehicle to the scope of human experience and result in slow progress in maneuver decision-making. Therefore, a maneuver decision-making method based on deep reinforcement learning and Monte Carlo tree search is proposed to investigate whether it is feasible for maneuver decision-making without human knowledge or advantage function. To this end, Monte Carlo tree search in continuous action space is proposed and neural networks-guided Monte Carlo tree search with self-play is utilized to improve the ability of air combat agents. It starts from random behaviors and generates samples consisting of states, actions, and results of air combat through self-play without using human knowledge. These samples are used to train the neural network, and the neural network with a greater winning rate is selected by simulations. Then, repeat the above process to gradually improve the maneuver decision-making ability. Simulations are conducted to verify the effectiveness of the proposed method, and the kinematic model of the missile is used in simulations instead of the missile engagement zone to test whether the maneuver decision-making method is effective or not. The simulation results of the fixed initial state and random initial state show that the proposed method is efficient and can meet the real-time requirement.

## KEYWORDS

autonomous air combat, maneuver decision-making, deep reinforcement learning, Monte Carlo tree search, neural networks

## Introduction

Autonomous air combat through unmanned combat aerial vehicles is the future of air combat and maneuver decision-making is the core of autonomous air combat. Therefore, it is urgent to build maneuver decision-making methods. Maneuver decision-making means that the aircraft chooses the appropriate maneuver (e.g., normal overload, tangential overload, and roll angle) to change its state according to the acquired information of the target (e.g., azimuth, velocity, height, and distance), so as to defeat the target.

Air combat can be divided into within-visual-range air combat and beyond-visual-range air combat. With the development of science and technology, the detection distance of airborne radar and the range of air-to-air missiles have been increased to hundreds of kilometers. Therefore, both sides of the air combat can discover each other and launch missiles at beyond-visual-range. Besides, the process of beyond-visual-range air combat is different from that of within-visual-range air combat because the principle and operation method between radar-guided missiles and infrared (IR) missiles are different. Radar-guided missiles are supposed to be used for beyond-visual-range and IR-guided missiles for within visual range, because the detection range of the radar is longer than that of the IR detector. The IR-guided missile does not need external equipment to provide target information after it is launched. It can obtain information about the target by means of its infrared detector and then attack the target. Therefore, the aircraft can retreat after launching missiles. However, after launching, there are two stages in the attack of radar-guided missiles, which are called the midcourse guidance stage and the terminal guidance stage. In the intermediate guidance stage, the radar of the missile is not activated. Thus, it is necessary for the aircraft radar to continuously detect the target, providing the information for the missile and guiding it to the target. During the terminal phase, the missile continues to chase the target according to the information provided by its radar until it hits the target or loses the target.

Therefore, the decision-making method in within-visual-range air combat cannot be used for beyond-visual-range air combat directly, so we need to find a new decision-making method for autonomous air combat. At the same time, the existing maneuver decision-making methods rely on human knowledge, which can also be regarded as a dense reward in reinforcement learning. Thus, sparse reward means only using the result of air combat (i.e., win or not), which does not rely on human knowledge. Moreover, if the task is complex, it is difficult to define and design human knowledge or dense reward. Therefore, it is necessary to explore maneuver decision-making methods using a sparse reward.

Recently, most of the research on maneuver decision-making is focused on within-visual-range air combat (McGrew et al., 2010; Guo et al., 2017; Du et al., 2018; Huang et al., 2018; Li et al., 2019). You et al. (2019) proposed a constrained parameter evolutionary learning algorithm for Bayesian network parameters learning with scarce data, which can be applied to unmanned aerial vehicle autonomous mission decision-making. Wu et al. (2011) proposed the situation assessment method of beyond-visual-range air combat based on missile attack area, and introduced a new angle advantage function, speed advantage function, and height advantage function into the situation assessment model. Li et al. (2020) proposed a cooperative occupation method for autonomous air combat of multiple UAVs based on weapon attack area. They used the weapon attack

area and air combat geometric description for one-to-one air combat situation assessment and established a multiple UAVs cooperative occupation model based on the encircling advantage function. Therefore, the cooperative occupation problem was transformed into a mixed integer non-linear programming problem and solved by an improved discrete particle swarm optimization algorithm. However, the flight model in this study is two-dimensional, that is, the height of both sides of air combat is always the same in air combat, and the control quantities do not include roll angle, so this study can be further improved. Wei et al. (2015) proposed a cognitive control model with three-layer structure for multi-UAVs cooperative search according to the cognitive decision-making mode of humans performing searching behavior. The mission area is carried on cognitive match, reduction, and division based on this model and the fuzzy cluster idea. The simulation experiments indicate the great performance of the fuzzy cognitive decision-making method for cooperative search. Zhang et al. (2018) proposed a maneuver decision-making method based on the Q network and Nash equilibrium strategy, and combined the missile attack area in the reward function to improve the efficiency of reinforcement learning. However, the maneuver library of this method only contains five maneuvers, which cannot meet the needs of air combat. Hu et al. (2021) proposed to use the improved deep Q network (Mnih et al., 2015) for maneuver decisions in autonomous air combat, constructed the relative motion model, missile attack model, maneuver decision-making framework, designed the reward function for training agents, and replaced the strategy network in deep Q network with the perception situation layer and value fitting layer. This method improves the winning rate of air combat, but the maneuver library is relatively simple and difficult to meet the needs of air combat.

It is worth noting that deep reinforcement learning has achieved professional performance in video games (Watkins and Dayan, 1992; Hado et al., 2016; Matteo et al., 2017), board games such as GO (Silver et al., 2016, 2017; Schrittwieser et al., 2020), real-time strategy games such as StarCraft (Oriol et al., 2019), magnetic control of tokamak plasmas (Jonas et al., 2022), data fusion (Zhou et al., 2020b), and intention prediction of aerial targets under Uncertain and Incomplete Information (Zhou et al., 2020a). Therefore, using deep reinforcement learning to improve the level of air combat maneuver decision-making is a feasible direction. AlphaStar is a multi-agent reinforcement learning algorithm based on supervised learning. It introduces league training: three pools of agents (the main agents, the league exploiters, and the main exploiters), each initialized by supervised learning, were subsequently trained with reinforcement learning. In AlphaStar, each agent is initially trained through supervised learning on replays to imitate human actions. Concretely, it uses a dataset of 971,000 replays played on StarCraft II from the top 22% of players. Therefore, it can be concluded that two features of AlphaStar are multi-agent reinforcement learning and human knowledge. However, we

mainly focus on one-on-one air combat, which means that a multi-agent algorithm is not suitable and we are supposed to use a single-agent algorithm to address this problem. Meanwhile, replays of games from top players are not difficult to obtain, but it is difficult and expensive to obtain data from human pilots, which means that we cannot use supervised learning as the first phase of AlphaStar.

Ma et al. (2020) described the cooperative occupation decision-making problem of multiple UAVs as a zero-sum matrix game problem, and proposed a solution of double oracle algorithm combined with neighborhood search. In maneuver decision-making, at first, the position to be occupied by each aircraft is determined, and then the target to be attacked by each aircraft is determined, to reduce the threat and increase the advantage. Yang et al. (2020) studied the evasive maneuver strategy of unmanned combat aircraft in BVR air combat, and the problem was solved by the hierarchical multi-objective evolutionary algorithm. In this method, the decision variables are classified according to the physical meanings and then coded independently. Four escape maneuvers are designed, including turning maneuver, vertical maneuver, horizontal maneuver, and terminal maneuver. The evolutionary algorithm is used to find approximate Pareto optimal solutions and reduce invalid solutions, thus, the efficiency of the algorithm is improved. Ma et al. (2018) built an air combat game environment and train the agent with deep Q-learning.

Eloy et al. (2020) studied the attack against static high-value targets in air combat. It analyzed the confrontation process with game theory and put forward a differential game method of air combat combined with the missile attack area (Wu and Nan, 2013; Li et al., 2015; Wang et al., 2019). In this method, the air combat process is divided into the attack stage and retreat stage, while the attacker is divided into leader and wingman. In the attack stage, the leader enters the target area and launches missiles, and the wingman flies in formation. In the retreat stage, the wingman protects the leader from the missile attack of the other party. However, the flight model of aircraft is two-dimensional rather than three-dimensional. However, the authenticity of the two-dimensional motion model is worse than that of the three-dimensional motion model, so the three-dimensional motion model should have been used. He et al. (2017) proposed a maneuver decision-making method based on Monte Carlo tree search (MCTS), and it uses MCTS to find the action with the greatest air combat advantage among the seven basic maneuvers. This method verifies the feasibility of MCTS in maneuver decision-making.

While human knowledge or dense reward can make the algorithm achieve the goal quickly, it also limits the diversity and potential of the algorithm to the scope of human experience. For example, AlphaGo with human knowledge is defeated by AlphaGo Zero without human knowledge, and AlphaZero can defeat the world champion without human knowledge and has found several joseki that human players have never found

before. Meanwhile, AlphaGo with human knowledge was once defeated by the world champion Lee Sedol, but AlphaGo Zero without human knowledge has not been defeated by any human players ever since. Thus, it is a reasonable conjecture that human knowledge is not good enough for training purposes for autonomous weapon deployment, and we propose a method in this article for air combat to investigate whether it is feasible for maneuver decision-making without human knowledge.

To this end, an air combat maneuver decision-making method based on deep reinforcement learning and MCTS is proposed, which aims at investigating whether it is feasible for maneuver decision-making without human knowledge or dense reward. First, different from existing methods, this method does not use human knowledge to assist the agent in maneuvering decision-making, but only uses the outcome of air combat simulations. Second, existing methods often make maneuver decisions in discrete and finite action space (e.g., maneuver library consists of finite maneuvers), however, the proposed method is based on continuous action space, which is more reasonable than discrete action space. Third, to select actions in continuous space, we proposed the method of MCTS in continuous space which is different from MCTS of existing decision-making methods. Moreover, existing methods often use missile engagement zone in simulations, but the missile may miss the target even if the target is in the missile engagement zone, therefore, the kinematic model of the missile is used in simulations instead of missile engagement zone to test whether it can hit the target, which reflects whether the maneuver decision-making method is effective (Li, 2010; Zhang et al., 2015). Our research logic is: if it works well in simulations, we may consider investigating it in the real world and modifying it if it does not work well. However, if it does not work well even in simulations, we do not consider transferring it to the real world. Therefore, we do the first step here, that is, investigating the method in simulations to make sure it works well in simulations at least before transferring it to the real world.

The main contributions are as follows: (1) To investigate whether it is feasible for maneuver decision-making without human knowledge, we propose to use the algorithm of self-play and MCTS which learns to search actions in continuous action space. (2) We provide a method to address the problem of MCTS in continuous space since MCTS cannot be applied to continuous space directly. (3) The simulation results demonstrate that although maneuver decision-making without human knowledge cannot completely defeat that with human knowledge, it is still feasible in air combat. The rest of this paper is organized as follows: In Section Aircraft model and missile model, the motion dynamics model of aircraft and missile is established. In Section Maneuver decision-making method based on deep reinforcement learning and MCTS, the process of self-play and neural network training is described (Hinton and Salakhutdinov, 2006; Goodfellow et al., 2017), and the role of human knowledge in maneuver decision-making is interpreted.

In Section Experiments and results, the training results of the neural network and the simulation results of air combat are given, and the decision-making ability of the proposed method is discussed according to the simulation results. The method in this article is summarized in Section Conclusion.

## Aircraft model and missile model

The aircraft model adopts normal overload, tangential overload, and roll angle as control parameters. To simplify the complexity of the problem, the angle of attack and the angle of side slip are regarded as zero and the ground coordinate system is treated as the inertial system, meanwhile, the effects of the rotation of the earth are overlooked. The kinematic and dynamic model is shown as follows (Williams, 1990):

$$\begin{cases} \dot{x} = v \cos \gamma \cos \psi \\ \dot{y} = v \cos \gamma \sin \psi \\ \dot{z} = v \sin \gamma \\ \dot{v} = g(n_x - \sin \gamma) \\ \dot{\gamma} = \frac{g}{v}(n_z \cos \mu - \cos \gamma) \\ \dot{\psi} = \frac{g}{v \cos \gamma} n_z \sin \mu \end{cases} \quad (1)$$

where  $x$ ,  $y$ , and  $z$  indicate the positions of the aircraft in the inertial coordinate system;  $\gamma$  is the pitch angle,  $\psi$  is the yaw angle,  $v$  is the velocity, and  $g$  is the acceleration of gravity. Roll angle  $\mu$ , tangential overload  $n_x$ , and normal overload  $n_z$  are control parameters. The kinematic model of the missile is Wang et al. (2019):

$$\begin{cases} \dot{x}_m = v_m \cos \gamma_m \cos \psi_m \\ \dot{y}_m = v_m \cos \gamma_m \sin \psi_m \\ \dot{z}_m = v_m \sin \gamma_m \end{cases} \quad (2)$$

where  $x_m$ ,  $y_m$ , and  $z_m$  indicate the positions of the missile in the inertial coordinate system;  $v_m$  is the velocity,  $\gamma_m$  is the pitch angle, and  $\psi_m$  is the yaw angle. The dynamic model of the missile is:

$$\begin{cases} \dot{v}_m = \frac{(P_m - Q_m)g}{G_m} - g \sin \gamma_m \\ \dot{\psi}_m = \frac{n_{mh}g}{v_m \cos \gamma_m} \\ \dot{\gamma}_m = \frac{n_{mh}g}{v_m} - \frac{g \cos \gamma_m}{v_m} \end{cases} \quad (3)$$

where  $P_m$  and  $Q_m$  are thrust and air resistance,  $G_m$  is the mass of the missile, and  $n_{mc}$  and  $n_{mh}$  are control overload in the yaw direction and pitch direction.  $P_m$ ,  $Q_m$ , and  $G_m$  can be calculated by the following formula (Fang et al., 2019):

$$P_m = \begin{cases} 12000 & t \leq t_w \\ 0 & t > t_w \end{cases} \quad (4)$$

$$Q_m = \frac{1}{2} \rho v_m^2 S_m C_{Dm} \quad (5)$$

$$G_m = \begin{cases} 173.6 - 8.2t & t \leq t_w \\ 108 & t > t_w \end{cases} \quad (6)$$

where  $t_w = 8.0s$ ,  $\rho = 0.607$ ,  $S_m = 0.0324$ , and  $C_{Dm} = 0.9$ . It is assumed that the guidance coefficient of proportional guidance law is  $K$  in control planes. The two overloads in yaw and pitch directions are defined as:

$$\begin{cases} n_{mc} = K \cdot \frac{v_m \cos \gamma_t}{g} [\dot{\beta} + \tan \varepsilon \tan(\varepsilon + \beta) \dot{\varepsilon}] \\ n_{mh} = \frac{v_m}{g} \frac{\dot{\beta}}{\cos(\varepsilon + \beta)} \dot{\varepsilon} \end{cases} \quad (7)$$

$$\begin{cases} \beta = \arctan(r_y/r_x) \\ \varepsilon = \arctan(r_z/\sqrt{r_x^2 + r_y^2}) \end{cases} \quad (8)$$

$$\begin{cases} \dot{\beta} = (\dot{r}_y r_x - r_y \dot{r}_x)/(r_x^2 + r_y^2) \\ \dot{\varepsilon} = \frac{(r_x^2 + r_y^2) \dot{r}_z - r_z(\dot{r}_x r_x + \dot{r}_y r_y)}{R^2 \sqrt{r_x^2 + r_y^2}} \end{cases} \quad (9)$$

where  $\beta$  and  $\varepsilon$  are yaw angle and pitch angle of the line of sight, and  $\dot{\beta}$  and  $\dot{\varepsilon}$  are the corresponding derivatives. The line of sight vector is the distance vector  $\vec{r}$ , where  $r_x = x_t - x_m$ ,  $r_y = y_t - y_m$ ,  $r_z = z_t - z_m$  and  $R = \|\vec{r}\| = \sqrt{r_x^2 + r_y^2 + r_z^2}$ .

The maximum overload of the missile is 40. When the minimum distance between the missile and the target is  $< 12m$ , the target is regarded as a hit; when missile flight time exceeds 120s and it still fails to hit the target, the target is regarded as missed; during the midcourse guidance stage, the target is regarded as missed when its azimuth relative to the aircraft exceeds  $85^\circ$ ; during the final guidance stage, the target is regarded as missed when its azimuth relative to missile axis exceeds  $70^\circ$ .

## Maneuver decision-making method based on deep reinforcement learning and MCTS

He et al. (2017) uses MCTS to find the maneuver that makes the most air combat advantage among the seven basic maneuvers, in which human knowledge is used to define the air combat advantage. However, its action space is discrete and only contains seven basic maneuvers. In this paper, the search scope of maneuver is extended from seven basic maneuvers to continuous action space, which contains countless maneuvers theoretically, and human knowledge is not used to assist maneuver decision-making, but only the outcome of air combat simulations. The main idea of the proposed reinforcement learning algorithm is to use neural networks to generate the maneuver and value in each state and then use the neural network-guided MCTS to search the maneuver in the continuous action space. The maneuver selected by MCTS is more effective than the maneuver directly generated by the neural network. Then, repeat the above steps in the self-play to generate training samples and update the neural network with these training samples to make the neural network more closely match the improved maneuver and self-play winner. The repetition steps are stopped and the training is regarded as good

enough usually when the rating of the agent (Silver et al., 2016, 2017; Schrittwieser et al., 2020) or the scores obtained by the agent (Mnih et al., 2015; Hado et al., 2016) does not increase visibly. The new network is used in the next iteration to make MCTS more powerful.

AlphaGo with human knowledge is defeated by AlphaGo Zero without human knowledge, and AlphaZero can defeat the world champion without human knowledge and has found several joseki that human players have never found before. Meanwhile, AlphaGo with human knowledge was once defeated by the world champion Lee Sedol, but AlphaGo Zero without human knowledge has not been defeated by any human players ever since. Therefore, we write “While human knowledge or dense reward can make the algorithm achieve the goal quickly, it also limit the diversity and potential of the algorithm to the scope of human experience” in the introduction, which mainly refers to the game of GO but not the autonomous weapon deployment. However, it is a reasonable conjecture that human knowledge is not good enough for the training purposes for autonomous weapon deployment, thus we propose this method for air combat to investigate whether it is feasible for maneuver decision-making without human knowledge.

Our method is inspired by and built upon AlphaGo Zero. However, AlphaGo Zero is not suitable for air combat because of continuous action space, so we modified it to make it able to handle continuous action space. Since AlphaGo with human knowledge is defeated by AlphaGo Zero without human knowledge, we want to know if the method without human knowledge is feasible in air combat or even better than the method with human knowledge; therefore, we investigate the problem in this paper. It is true that human knowledge is indeed useful, and we will study maneuver methods with human knowledge in future. On the other hand, considering the development of AlphaGo, although the AlphaStar approach used human knowledge, a new approach called AlphaStar Zero may appear just like AlphaGo Zero, which can defeat AlphaStar and the world champion in the game of StarCraft II without using any human knowledge.

## MCTS in continuous space

MCTS is usually used for searching in discrete action space (He et al., 2017; Silver et al., 2017; Hu et al., 2021). In this paper, we use neural networks to guide MCTS as in Silver et al. (2017). Since MCTS is typically used in discrete space and cannot be used in continuous space directly, we propose the method of MCTS in continuous space to address the problem of maneuver decision-making in air combat. The generation and selection of action in continuous space are shown in Figure 1.

The green rectangle in Figures 1, 2 is the continuous action space, which contains countless actions theoretically. Therefore, it cannot be searched by MCTS directly and we propose the

following method to make MCTS able to search in continuous action space. First, a state is sent to neural networks as input and the neural network outputs the action and value according to the state, in which the action is regarded as the mean of a Gaussian distribution, the action output by the neural network is represented by the red rectangle in Figure 1. After that, a Gaussian distribution is acquired as shown in the blue shadow part in Figure 1. Then,  $N-1$  actions are sampled from the Gaussian distribution, which are represented by the black rectangles in Figure 1, so  $N$  actions are acquired totally and MCTS is used to search for these  $N$  actions. Figure 2 illustrates the search process of MCTS in continuous space.

Each node  $s$  in the tree contains all actions of edges  $(s, a)$ , and each edge stores a set  $\{N(s, a), W(s, a), Q(s, a), P(s, a)\}$ , where  $N$  represents the number of visits,  $W$  represents the total action value,  $Q$  represents the average action value, and  $P$  is the a priori probability of selecting this action, which can be computed by the Gaussian probability density function.

MCTS repeats four operations to find the action: selection, play, expansion, and backpropagation. Selection: take the current state as the root node, start the simulation from the root node, and stop until the simulation reaches the leaf node at time-step  $L$ . Before time-step  $L$ , the action is selected according to the a priori probability and average action value in the tree,  $a_t = \operatorname{argmax} [Q(s_t, a) + U(s_t, a)]$  (Rosin, 2011),

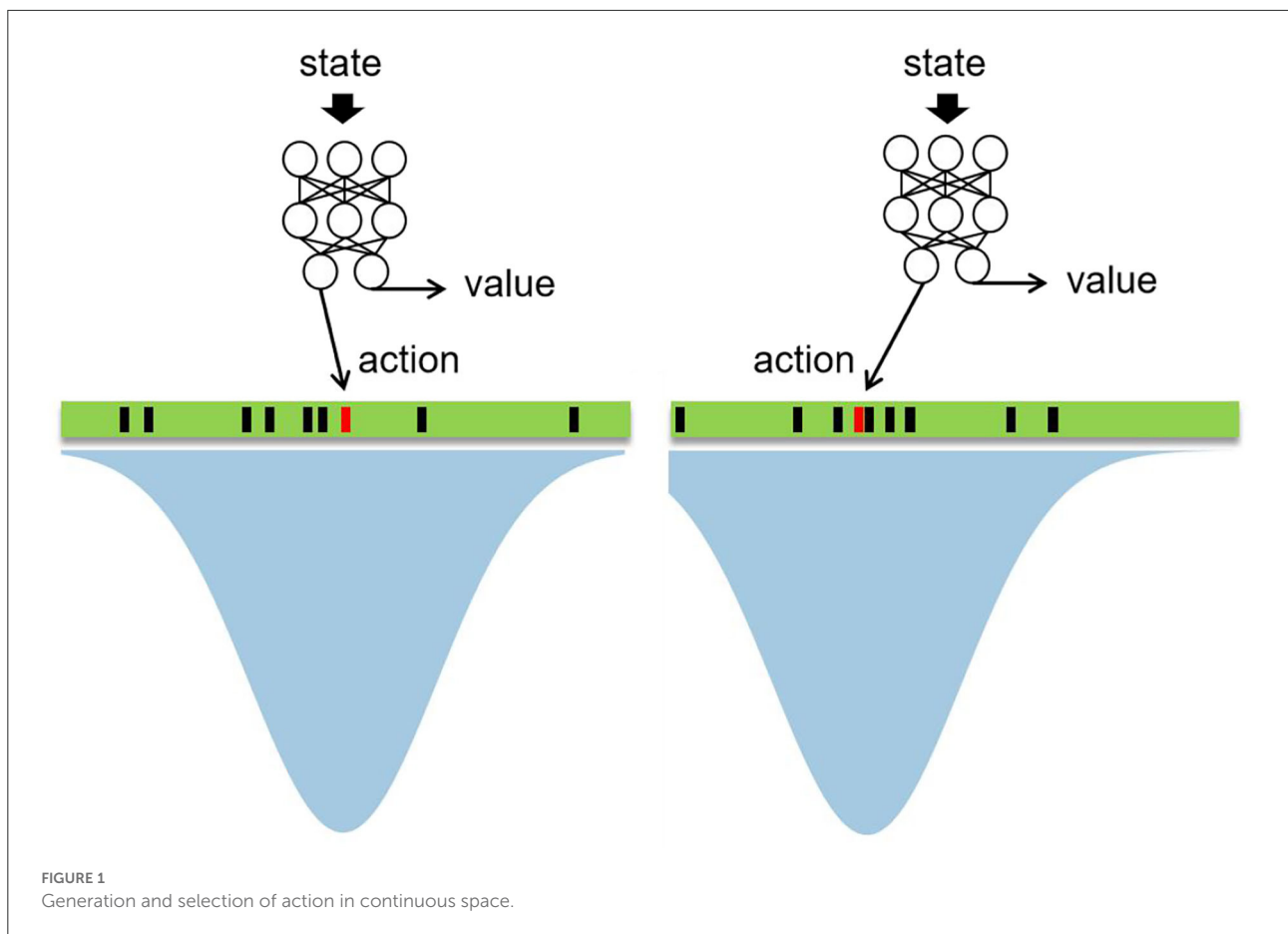
$$U(s, a) = P(s, a) \sqrt{\frac{\sum_b N(s, b)}{1 + N(s, a)}}$$

the probability of  $a_t$  is proportional to the maximum of  $Q(s_t, a) + U(s_t, a)$ , in which  $Q(s_t, a_t) = W(s_t, a_t) / N(s_t, a_t)$ . Here,  $W(s_t, a_t)$  is computed by the value head of neural networks, and actions are generated by the acting head of neural networks, which is different from the original MCTS used in He et al. (2017) and Hu et al. (2021) since the original MCTS chooses action randomly instead of using neural networks.

Play: when in the selection step an action is chosen, which has not been stored in the tree, the play starts. Actions are selected in self-play until the leaf node  $s_L$  is reached, and the leaf node  $s_L$  means it has not been expanded.

Expansion: the neural network is used to evaluate the leaf node  $s_L$  added to the queue, expand the leaf node  $s_L$ , and each edge  $(s_L, a)$  is initialized to  $\{N(s_L, a) = 0, W(s_L, a) = 0, Q(s_L, a) = 0, P(s_L, a) = p_a\}$  and  $p_a$  is the priori probability of the action. This part is another different part from the original MCTS used in He et al. (2017) and Hu et al. (2021), since the original MCTS evaluates the leaf node  $s_L$  by rollouts. However, the proposed method evaluates the leaf node  $s_L$  by neural networks, that is, the MCTS is guided by neural networks.

Backpropagation: update the number of visits and value of each step  $t$  in turn,  $N(s_t, a_t) = N(s_t, a_t) + 1$ ,  $W(s_t, a_t) = W(s_t, a_t) + v$ ,  $Q(s_t, a_t) = W(s_t, a_t) / N(s_t, a_t)$ .



After several iterations, MCTS outputs the action according to  $a_t = \operatorname{argmax}[Q(s_t, a) + U(s_t, a)]$  among  $N$  actions in continuous action space, as shown in the top right of Figure 2.

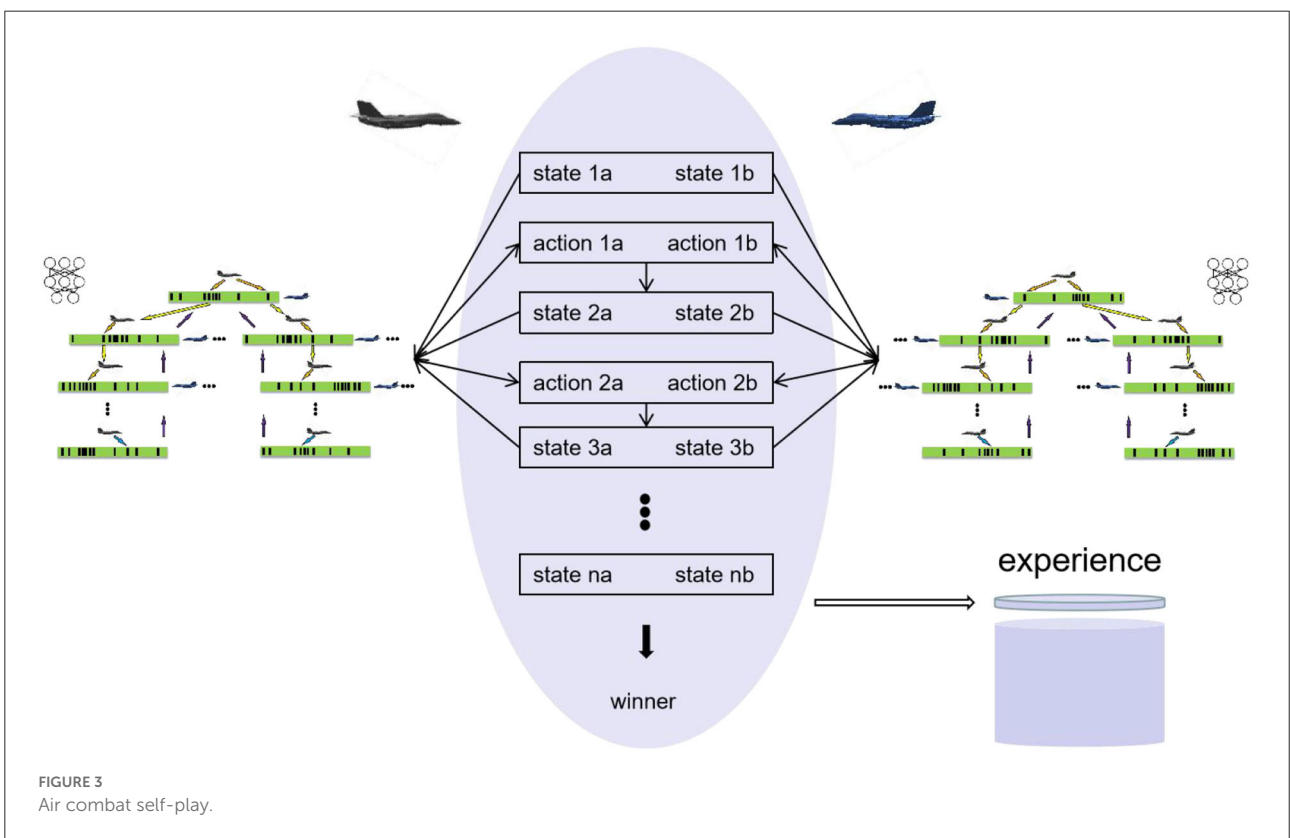
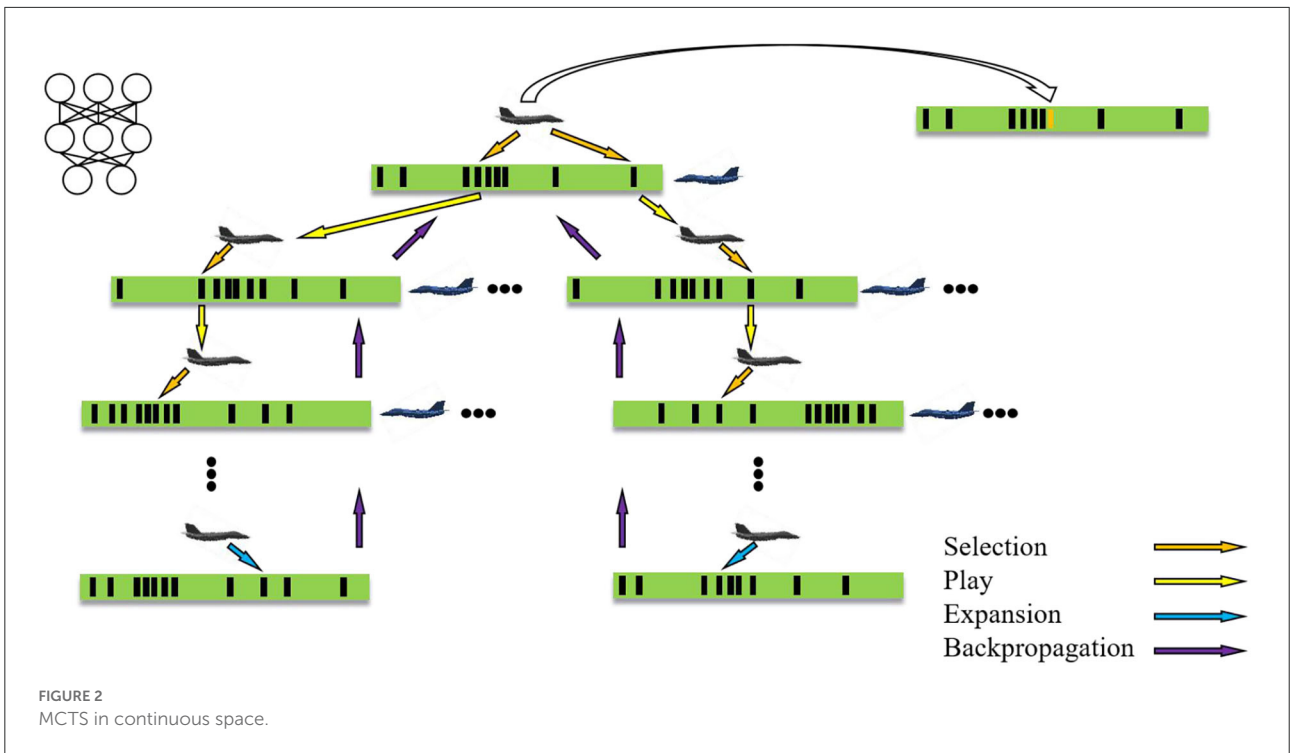
## Reinforcement learning from self-play

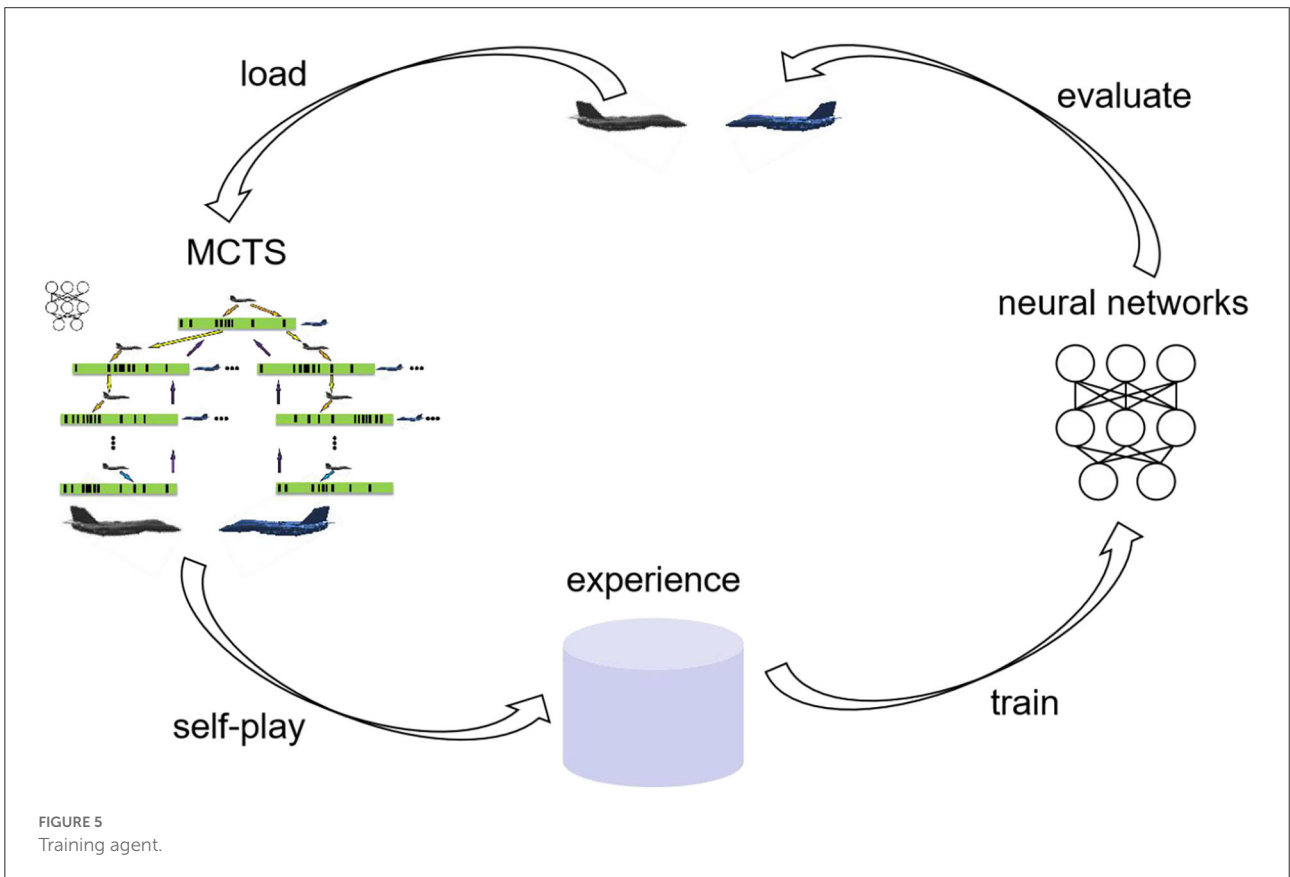
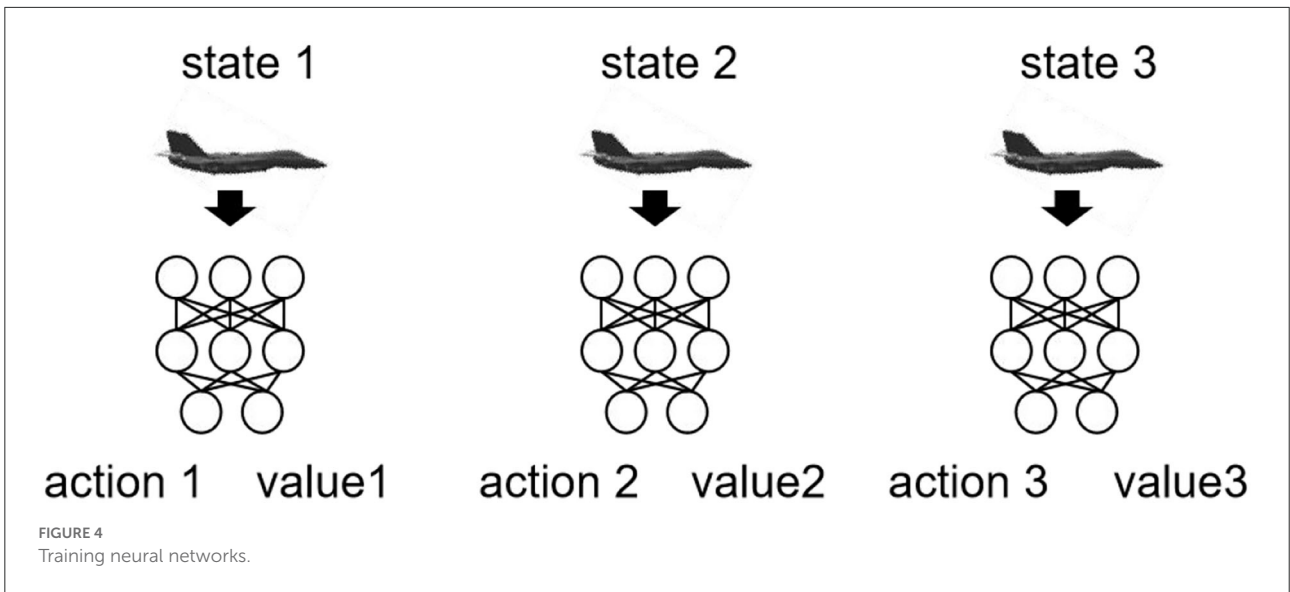
Self-play reinforcement learning method has achieved professional performance in such games: chess (Baxter et al., 2000), othello (Sheppard, 2002), and poker (Moravčík, 2017). Therefore, this paper adopts self-play reinforcement learning for maneuver decision-making, and does not use any human knowledge. Starting from a completely random maneuver strategy, the neural network is trained by the data generated by self-play, so that the neural network can gradually produce effective maneuver strategies during the training pipeline. Figure 3 illustrates the self-play procedure.

As shown in Figure 3, at each time-step, the two sides of air combat execute the maneuvers selected by MCTS and reach the next time-step and a new state. In this state, the two sides continue to execute the maneuver obtained by MCTS until the final result of the simulation is obtained. The final result at the

end  $T$  is  $r_T = \{-1, 0, 1\}$ , where  $-1$  represents lose,  $0$  represents draw, and  $1$  represents win. It can be seen that there is no reward function in of self-play process except the final result of air combat, that is, human knowledge is not added to self-play, which is another feature of the proposed method. Self-play uses MCTS to generate state-action pairs in each iteration and takes these state-action pairs as samples to train the neural network. As shown in Figure 3, the air combat data of each time-step  $t$  is saved as  $(s_t, a_t, z_t)$  in the experience pool,  $z_t = \pm r_T$  is the winner from the perspective of the current aircraft at time  $t$ . Uniform sampling  $(s_t, a_t, z_t)$  from all time-steps of the last iteration of self-play is used to train the network to minimize the error of prediction value and winner and the error of neural network output and MCTS output as shown in Figure 4, and the loss function is the sum of mean square error and L2 weight regularization.

To ensure the generalization ability of the neural network, the initial state of each game is randomly selected from the following scope: azimuth scope ( $-45^\circ, 45^\circ$ ), speed scope (250, 400 m/s), and the distance between aircraft (40, 100 km). In self-play, MCTS is used to search 90 times for each decision. The first 10 maneuvers are sampled according to the visit count of each node and the subsequent maneuvers are those with the largest





visit count, so as to balance the exploration and exploitation of the algorithm.

Figure 5 indicates the whole procedure of agent training. First, the agent generates air combat state-action pairs by MCTS in self-play and stores these data in the experience pool. Then,

the neural network is trained with the data generated by 350 times of air combat self-play. During each training, 64 samples are uniformly sampled from the experience pool. The optimizer is stochastic gradient descent with a momentum of 0.9, and the L2 regularization coefficient is 0.0001. After 1,000 times of



```

Build neural networks with random weights
For iteration = 1,..., M do:
  Randomly initialize state  $s_0$ 
  For  $t = 0, \dots$ , max step do
    State  $s_t$ 
    Red side selects action by MCTS
    Blue side selects action by MCTS
    Simulate and reach the next state
    If find the winner:
      Beak
    Else:
      Continue
     $t = t+1$ 
  Store the state-action pairs and the winner
  If required amount of experience:
    Beak
  Else:
    Continue
  Sample data from experience pool and train the neural networks
  Save and evaluate the neural networks
  If wins > 5 + failures:
    Load it as the current best neural network

```

Algorithm 1. Training agent.

training, a new neural network is obtained and saved. To ensure the quality of the data generated from self-play, the latest neural network after each training is evaluated: use the latest neural network to simulate air combat against the current best neural network 100 times. If the number of wins of the latest network is five more than failures, the latest neural network is loaded as the current best neural network and it is used to generate data in subsequent self-play, otherwise, the latest neural network is only saved but not loaded as the current best neural network. Algorithm 1 describes one iteration of agent training in Figure 5.

## Air combat state and neural network architecture

The input of the neural network is a one-dimensional vector with 44 elements, which are composed of the state of the current time-step and the state of the first three time-steps. As shown in Table 1, each state contains 11 quantities:  $\psi, \gamma, v, z, d, f_1, \psi_1, \gamma_1, d_1, \beta, f_2$ , where  $\psi$  and  $\gamma$  are yaw angle and pitch angle of velocity vector relative to the line of sight,  $v$  is the velocity of the aircraft,  $z$  is the flight altitude,  $d$  is the distance between the two sides in air combat, and  $r_1$  and  $r_2$  are the coordinates of the two sides, respectively.  $f_1$  represents whether our side launched a missile. Where  $\psi_1$  and  $\gamma_1$  are yaw angle and pitch angle of the missile's velocity vector relative to line of sight.

TABLE 1 Air combat state.

State	Symbol	Formula
Yaw angle	$\psi$	$\psi = \psi + \int \frac{g}{v \cos \gamma} n_z \sin \mu \, dt$
Pitch angle	$\gamma$	$\gamma = \gamma_0 + \int \frac{g}{v} (n_z \cos \mu - \cos \gamma) \, dt$
Velocity	$v$	$v = v_0 + \int g (n_x - \sin \gamma) \, dt$
Altitude	$z$	$z = z_0 + \int v \sin \gamma \, dt$
Distance between the two sides	$d$	$d = \ r_1 - r_2\ $
Launch missile	$f_1$	0 or 1
Yaw angle of missile	$\psi_1$	$\psi_m = \psi_{m0} + \int \frac{n_{mz} g}{v_m \cos \gamma_m} \, dt$
Pitch angle of missile	$\gamma_1$	$\gamma_m = \gamma_{m0} + \int \frac{n_{mg}}{v_m} - \frac{g \cos \gamma_m}{v_m} \, dt$
Distance between the missile and the other side	$d_1$	$d = \ r_{m1} - r_2\ $
Heading crossing angle	$\beta$	$\beta = \arccos\left(\frac{v_1 \cdot v_2}{\ v_1\  \ v_2\ }\right)$
Launch missile of the other side	$f_2$	0 or 1

$d_1$  is the distance between the missile and the other side and  $r_{m1}$  is the coordinate of the missile of the side in air combat.  $\beta$  is heading crossing angle, that is the angle between two velocity vectors of the two sides, which is represented by  $v_1$  and  $v_2$  in Table 1.  $f_2$  represents whether the other side launched missile. The input layer is followed by three hidden layers. The number of neurons of the first two layers is 128 and the number of neurons of the third layer is 64. Finally, it output five quantities. The first three outputs are normal overload, tangential overload, and roll angle, respectively. The fourth output is whether to launch the missile and the fifth output is the value of the current state. The activation function is tanh.

## Experiments and results

### Parameter setting

The maximum flight speed is 420 m/s, and the minimum flight speed is 90 m/s; The maximum flight altitude is 20,000 m and the minimum flight altitude is 50 m; the initial roll angle is always zero; the decision interval is 1 s and the maximum simulation time is 200 s. The outcome of the air combat simulation is defined as follows: if the missile hit the target, record it as a win; either the aircraft or the missile misses the target, it is regarded as missing the target; when the flight altitude of one side is greater than the maximum altitude or less than the minimum altitude, if the other side has launched missile and does not miss the target, record it as lose, otherwise, record it as a draw; when both sides miss the target, record it as a draw. The decision interval is 1 s, because it is common in the field since previous work (Guo et al., 2017; Du et al., 2018; Huang et al., 2018) usually uses the decision interval of 1 s. Meanwhile, a shorter decision interval requires more computational sources and a longer time span is obviously irrational.

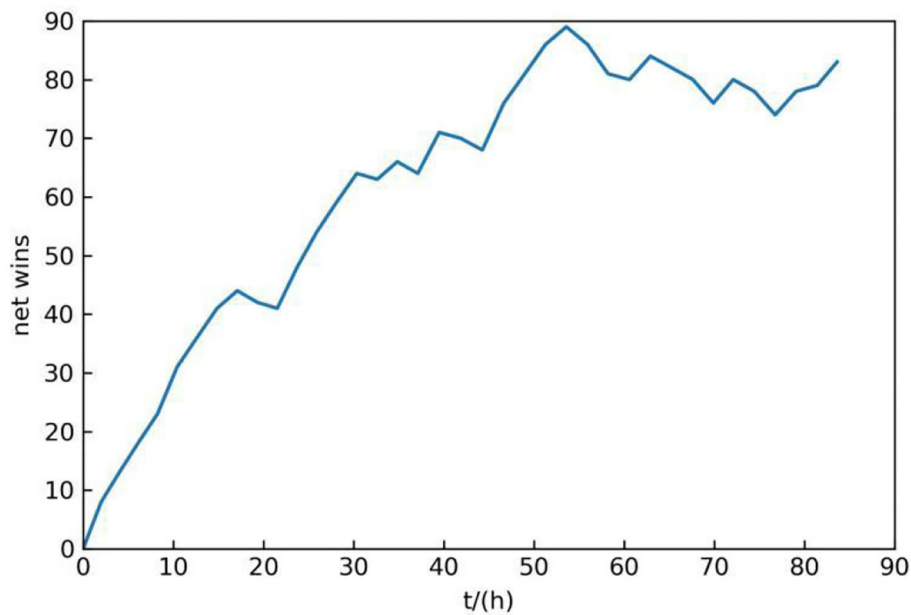


FIGURE 6  
Neural networks training result.

TABLE 2 Statistic results.

Initial state	Win	Lose	Draw	Average time(s)
Fixed	23	17	60	0.38
Random	22	21	57	0.37

It is true that maneuver decisions are not of any use if decisions cannot be done in a reasonable time span. Here, the rate for maneuver decision of 1 per 1 s does not mean that the maneuver is static within 1 s. For example, the aircraft takes the maneuver of changing the roll angle from 0 degrees to 45 degrees within 1 s (the case of 1 maneuver per second), thus, it gradually increases its roll angle from 0 to 45 degrees, which is a dynamic process. On the other hand, increasing the roll angle from 0 to 45 degrees may be interpreted as three maneuvers as well, for example, 0–15, 15–25, and 25–45 degrees. More importantly, even if we send several different maneuvers to the real aircraft within 1 s (such as changing the roll angle from 0 to 30 degrees, then changing it from 30 to –10 degrees, and then changing it from –10 to 50 degrees), it may not be able to realize it because of the limitations of the hardware (e.g., aircraft servomechanism). On the other hand, even if the real aircraft can realize it, it is unacceptable, because it is harmful to the aircraft to change its maneuver several times within 1 s (lack of aircraft strength). Minimum reaction time of the human brain is  $\sim 0.1$  s. Meanwhile, it takes much more than 0.1 s for a human

to decide what to do before the reaction, namely decision-making time. Therefore, the time span of 1 s is appropriate for a real-world application.

## Results and analysis

### Neural networks training result

In the process of self-play, record the net number of wins of each latest neural network in 100 times air combat, that is, subtract the number of failures from the number of wins. The reason why 100 times is selected is that 100 times is enough to distinguish the better one from both sides of the competition and does not cause too much time consumption. The total training time is about 84 h, and the change of net wins with time is shown in Figure 6.

As can be seen from Figure 6, the number of net wins is increasing along with the training. Although it sometimes decreases in the training process, it generally shows an upward trend, which indicates that the maneuvering decision-making ability of the proposed method gradually becomes more effective during self-play.

### Air combat simulation results

We verify the effectiveness of the method we proposed by a fight against the MCTS method (He et al., 2017): (1) 100 simulations with a fixed initial state of the following simulation

2, which is a fair initial state for both sides. (2) 100 simulations with a random initial state. Table 2 indicates the win, lose, and draw times and the average time consumed by each decision-making of the proposed method. As shown in Table 2, on the one hand, the proposed method won five more times than the MCTS method, and 60 simulations is drawn. These results indicate that the proposed method is feasible and effective, even though the proposed method is just slightly better. On the other hand, when simulations started from a random initial state, the proposed method is almost the same as the MCTS method, which indicates that the initial state has a significant influence on the decision-making method. As can be seen from Table 2, for the proposed method, the average time taken for each decision-making is  $\sim 0.38$  s. We also compute the average time of the original MCTS method (He et al., 2017), which is 0.11 s, this means that the proposed method is slower.

Next, we show the process of the MCTS maneuver decision-making method in continuous action space, then we show the process of the method we proposed by a fight against the MCTS method. The initial position of aircraft 1 is (40,000, 40,000, 10,000), the pitch angle is  $0^\circ$ , the yaw angle is  $180^\circ$ , and the initial velocity is 300 m/s. The initial position of aircraft 2 is (0, 0, 10,000), the pitch angle is  $0^\circ$ , the yaw angle is  $0^\circ$ , and the initial velocity is 300 m/s. Aircraft 1 moves at a constant speed in a straight line, and aircraft 2 maneuvers using the MCTS method with human knowledge. The simulation result is shown in Figure 7.

Figure 7A shows the trajectory of both sides, in which the blue solid line represents the flight trajectory of aircraft 1 and the orange solid line represents the flight trajectory of aircraft 2. In Figure 7B, the solid blue line indicates the velocity change of aircraft 1 and the orange solid line represents the velocity change of aircraft 2. Figure 7C indicates the overload change of missiles of aircraft 2. It can be seen that the MCTS method with human knowledge can react to the aircraft with simple maneuver and at the end of the simulation, the missile of aircraft 2 hit the target, which suggests the effectiveness of the MCTS method.

In simulation 2, aircraft 1 uses the proposed method and aircraft 2 uses the MCTS method (He et al., 2017), but the action space of the two methods is the same. As described in He et al. (2017), it combines the angle advantage function, distance advantage function, velocity advantage function, and height advantage function with MCTS, which means that it makes maneuver decisions with human knowledge. These advantage functions which stem from human knowledge can guide the aircraft to approach the target. However, our method uses only the final result  $r_T = \{-1, 0, 1\}$ , as described in Section Reinforcement learning from self-play, including no human knowledge.

The initial position of aircraft 1 is (70,000, 70,000, 10,000), the pitch angle is  $0^\circ$ , the yaw angle is  $180^\circ$ , and the initial velocity is 300 m/s. The initial position of aircraft 2 is (0, 0, 10,000), the pitch angle is  $0^\circ$ , the yaw angle is  $0^\circ$ , and the initial velocity is

300 m/s. As can be seen that the initial situation of both sides is equal. The simulation result is shown in Figure 8.

Figure 8A shows the trajectory of both sides, in which the blue solid line represents the flight trajectory of aircraft 1, the orange solid line represents the flight trajectory of aircraft 2, the green dotted line represents the flight trajectory of missile 1, and the red dotted line represents the flight trajectory of missile 2. In Figure 8B, the solid blue line indicates the velocity change of aircraft 1 and the orange solid line represents the velocity change of aircraft 2. Figure 8C indicates the overload change of missiles of the two sides, and it can be seen from Figure 8C that the missile overload is small when it is far from the target and reaches the maximum when it hit the target.

As can be seen from Figure 8A, when the simulation begins, both sides deflect toward each other and launch missiles, but their decision-making principles are different: aircraft 1 concludes that deflecting to aircraft 2 is of high value according to a large number of self-play data, while aircraft 2 chooses to deflect to aircraft 1 because it can increase the value of the air combat advantage function. In the end, the missile of aircraft 1 hit aircraft 2, and the distance between missile 2 and aircraft 1 is about 8 km. This suggests that the proposed method without human knowledge is stronger.

The initial position of aircraft 1 is (80 000, 80 000, 10 000), the pitch angle is  $0^\circ$ , the yaw angle is  $180^\circ$ , and the initial velocity is 300 m/s. The initial position of aircraft 2 is (0, 0, 10 000), the pitch angle is  $0^\circ$ , the yaw angle is  $45^\circ$ , and the initial velocity is 300 m/s. As can be seen that the initial situation of aircraft 2 is at an advantage. The simulation result is shown in Figure 9.

The simulation ended because the altitude of aircraft 2 exceeded the maximum altitude. The air combat advantage function of aircraft 2 includes the constraint on flight altitude to keep the altitude difference between it and the target within a certain range. However, although it used the advantage function based on human knowledge to guide maneuver decision-making, it failed to control the flight altitude properly because of the randomness of MCTS. On the contrary, the proposed method also based on MCTS can keep the flight altitude within a reasonable range without human knowledge, which indicates the effectiveness of the proposed method.

At the same time, according to Figures 7B, 8B, 9B, it can be seen that decision-making guided by human knowledge always increases the speed, while decision-making without human knowledge accelerates and decelerates, which shows that the method without human knowledge is more reasonable. Because the maximum speed is set as 420 m/s, it can be seen from the speed-increasing trend in Figures 7B, 8B, 9B that if the maximum speed is not set, the decision-making method guided by human knowledge will continue to increase the speed and always maintain the maximum speed in the subsequent air combat, which is not reasonable. Therefore, the method proposed in this paper without human knowledge is more reasonable and effective.

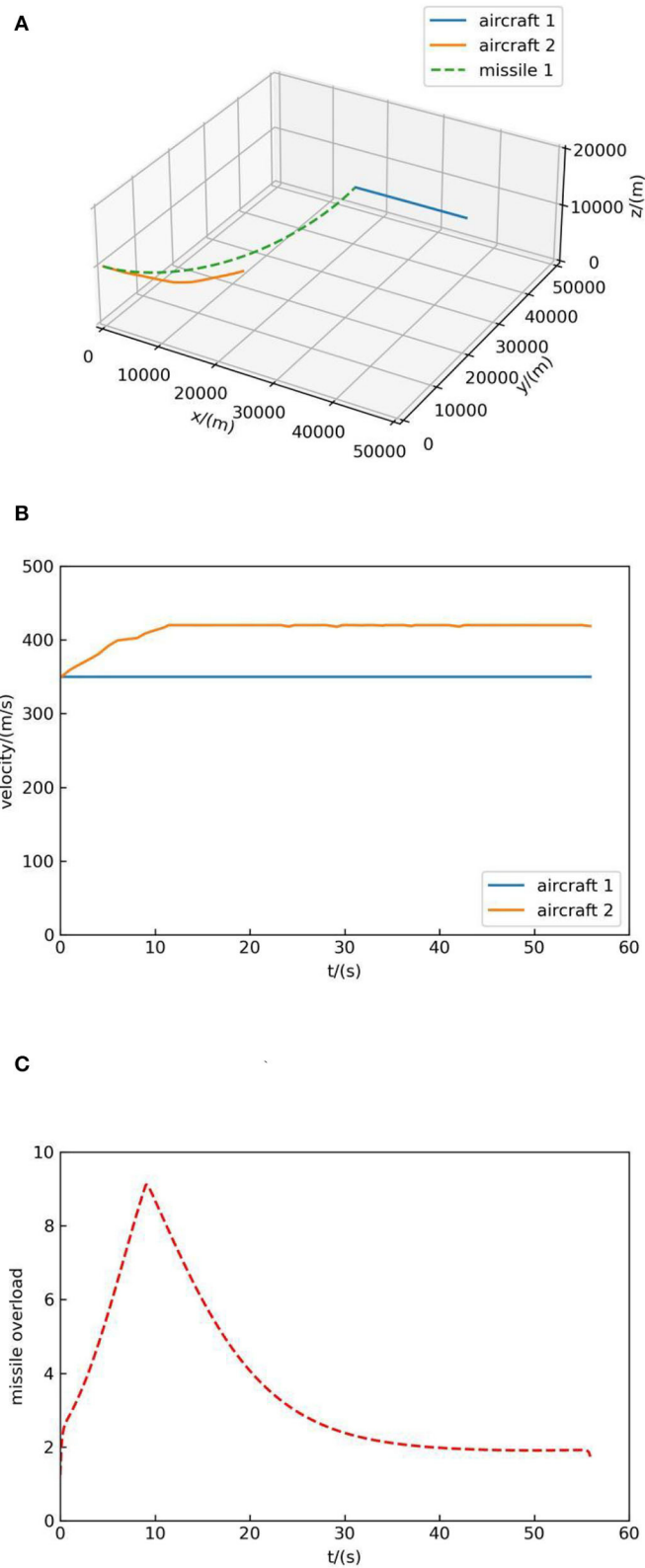


FIGURE 7  
Simulation result 1. (A) Air combat trajectory. (B) Velocity. (C) Missile overload.

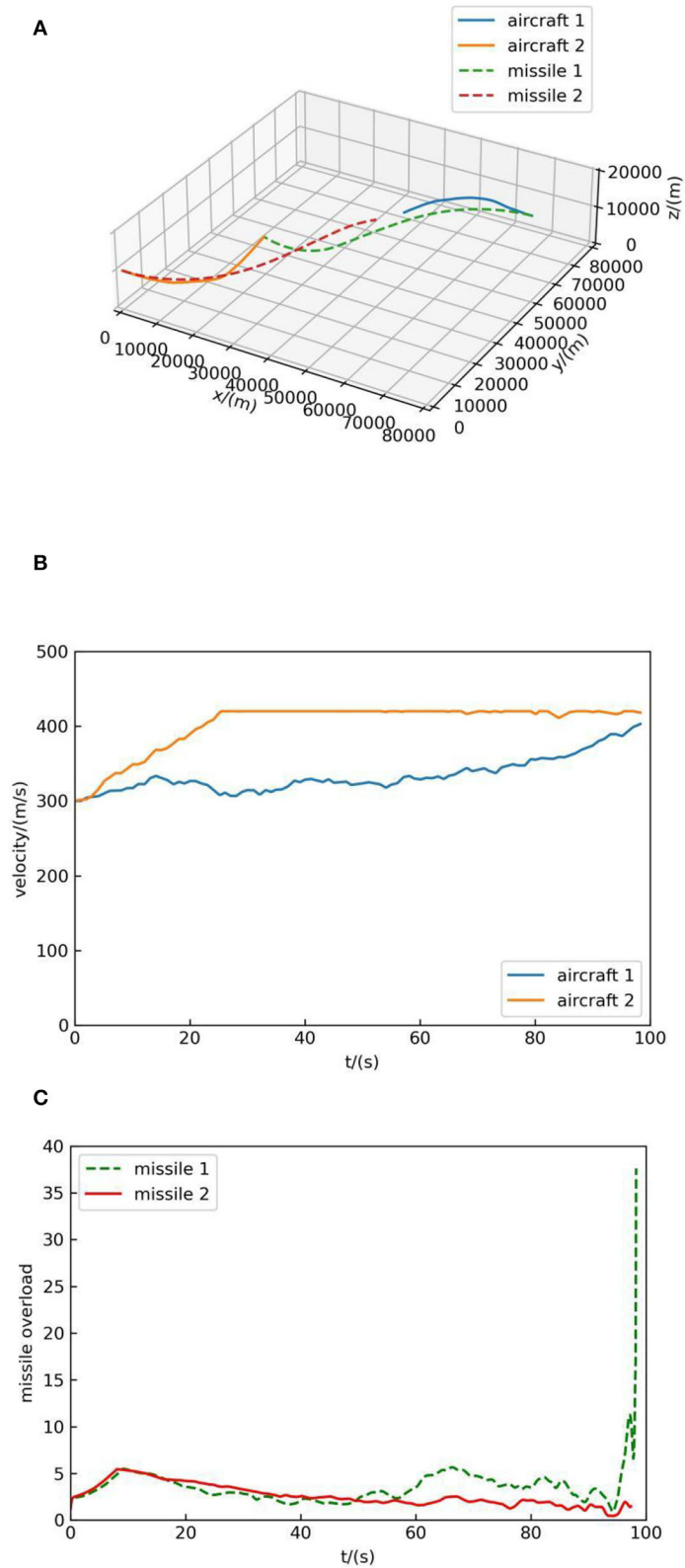


FIGURE 8  
Simulation result 2. (A) Air combat trajectory. (B) Velocity. (C) Missile overload.

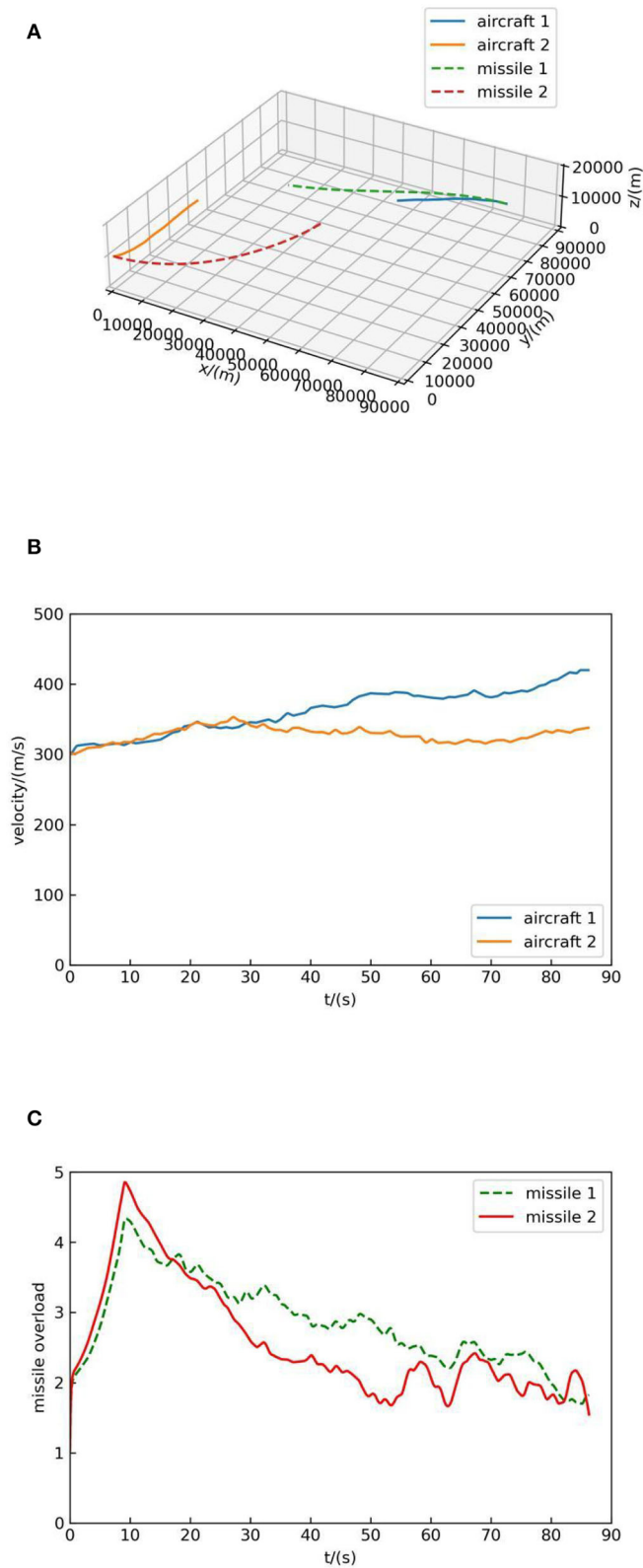


FIGURE 9  
Simulation result 3. (A) Air combat trajectory. (B) Velocity. (C) Missile overload.

## Conclusion

The maneuver decision-making method based on deep reinforcement learning and Monte Carlo tree search without human knowledge is proposed in this paper. According to the simulation results, it can be concluded that a pure reinforcement learning approach without human knowledge is feasible and efficient for autonomous air combat maneuver decision-making. On the one hand, the strengths of the proposed method are as follows: (1) The method can achieve similar performance as the method with human knowledge. (2) The method is simple to implement since elaborately designed reward based on human knowledge is not necessary. (3) The method can train neural networks from scratch without using any data from human pilots, which indicates that it can be used in the domains where human data are deficient or expensive to acquire. On the other hand, the weaknesses of the proposed method are as follows: (1) The performance of the method is not as good as its counterparts in board games, such as Go and chess. (2) The time consumption of the method is more than some traditional methods. (3) It takes plenty of time for training an agent using this method.

We aim to investigate whether it is feasible for maneuver decision-making without human knowledge by means of simulations and using the results for a recommendation system or pilots in manned aircraft is out of the scope of the article. In future work, considering AlphaGo Zero without human knowledge can defeat previous algorithms and human players in Go, and it is necessary to improve the performance of the method without human knowledge since the proposed method does not completely defeat the methods with human knowledge. Meanwhile, decreasing the time consumption of the method is also another future work because the time consumption of the proposed method is more than some traditional methods. And the training procedure needs to be improved since it takes plenty of time for training an agent.

## References

- Baxter, J., Tridgell, A., and Weaver, L. (2000). Learning to play chess using temporal differences. *Mach. Learn.* 40, 243–263. doi: 10.1023/A:1007634325138
- Du, H. W., Cui, M. L., Han, T., Wei, Z., Tang, C., and Tian, Y. (2018). Maneuvering decision in air combat based on multi-objective optimization and reinforcement learning. *J. Beij. Uni. Aero. Astronau.* 44, 2247–2256. doi: 10.13700/j.bh.1001-5965.2018.0132
- Eloy, G., David, W. C., Dzung, T., and Meir, P. (2020). “A differential game approach for beyond visual range tactics,” in *2021 American Control Conference* (New Orleans, LA).
- Fang, X., Liu, J., and Zhou, D. (2019). Background interpolation for on-line situation of capture zone of air-to-air missiles. *J. Syst. Eng. Electron.* 41, 1286–1293. doi: 10.3969/j.issn.1001-506X.2019.06.16
- Goodfellow, I., Bengio, Y., and Courville, A. (2017). *Deep Learning*. Beijing: Posts Telecom Press.
- Guo, H., Hou, M., Zhang, Q., and Tang, C. (2017). UCAV robust maneuver decision based on statistics principle. *Acta Arma.* 38, 160–167. doi: 10.3969/j.issn.1000-1093.2017.01.021
- Hado, H., Arthur, G., and David, S. (2016). “Deep reinforcement learning with double q-learning,” in *National Conference of the American Association for Artificial Intelligence* (Phoenix, AZ: PMLR), 1813–1825.
- He, X., Jing, X., and Feng, C. (2017). Air combat maneuver decision based on MCTS method. *J. Air For. Eng. Uni.* 18, 36–41. doi: 10.3969/j.issn.1002-0640.2018.03.008
- Hinton, G., and Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. *Science* 313, 504–507. doi: 10.1126/science.1127647
- Hu, D., Yang, R., Zuo, J., Zhang, Z., Wu, J., and Wang, Y. (2021). Application of deep reinforcement learning in maneuver planning of beyond-visual-range air combat. *IEEE Access.* 9, 32282–32297. doi: 10.1109/ACCESS.2021.3060426

## Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

## Author contributions

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

## Funding

This work was supported by the National Natural Science Foundation of China (Grant No. 62101590) and the Natural Science Foundation of Shaanxi Province (Grant No. 2020JQ-481).

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

- Huang, C., Dong, K., Huang, H., Tang, S., and Zhang, Z. (2018). Autonomous air combat maneuver decision using Bayesian inference and moving horizon optimization. *J. Syst. Eng. Electron.* 29, 86–97. doi: 10.21629/JSEE.2018.01.09
- Jonas, D., Federico, F., and Jonas, B. (2022). Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature* 602, 414–419. doi: 10.1038/s41586-021-04301-9
- Li, S., Ding, Y., and Gao, Z. (2019). UAV air combat maneuvering decision based on intuitionistic fuzzy game theory. *J. Syst. Eng. Electron.* 41, 1063–1070. doi: 10.3969/j.issn.1001-506X.2019.05.19
- Li, W., Shi, J., and Wu, Y. (2020). A multi-UCAV cooperative occupation method based on weapon engagement zones for beyond-visual-range air combat. *Def. Tech.* 4, 1–17. doi: 10.1016/j.dt.2021.04.009
- Li, X., Zhou, D., and Feng, Q. (2015). Air-to-air missile launch envelopes fitting based on genetic programming. *J. Project. Rockets Missile Guid.* 35, 16–18. doi: 10.15892/j.cnki.djzdx.2015.03.005
- Li, Z. (2010). China radar guided air-to-air missile. *Shipborne Weap.* 2, 22–35. doi: 10.15892/j.cnki.jzqw.2010.02.013
- Ma, X., Li, X., and Zhao, Q. (2018). “Air combat strategy using deep Q-learning,” in *Chinese Automation Congress*, 3952–3957. doi: 10.1109/CAC.2018.8623434
- Ma, Y., Wang, G., Hu, X., Luo, H., and Lei, X. (2020). Cooperative occupancy decision making of multi-UAV in beyond-visual-range air combat: a game theory approach. *IEEE Access*. 8, 11624–11634. doi: 10.1109/ACCESS.2019.2933022
- Matteo, H., Joseph, M., and Hado, H. (2017). *Rainbow: Combining Improvements in Deep Reinforcement Learning*. Available online at: <http://arxiv.org/abs/1710.02298v1>.
- Mcgregor, J. S., How, J. P., and Williams, B. (2010). Air-combat strategy using approximate dynamic programming. *J. Guid. Control Dynam.* 33, 1641–1654. doi: 10.2514/1.46815
- Mnih, V., Kavukcuoglu, K., and Silver, D. (2015). Human-level control through deep reinforcement learning. *Nature* 518, 529–533. doi: 10.1038/nature14236
- Moravčík, M. (2017). DeepStack: expert-level artificial intelligence in heads-up no-limit poker. *Science* 356, 508–513. doi: 10.1126/science.aam6960
- Oriol, V., Igor, B., and Silver, D. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 350–354. doi: 10.1038/s41586-019-1724-z
- Rosin, D. (2011). Multi-armed bandits with episode context. *Ann. Math. Artif. Intell.* 61, 203–230. doi: 10.1007/s10472-011-9258-6
- Schrittwieser, J., Antonoglou, I., and Silver, D. (2020). Mastering Atari, Go, Chess and Shogi by planning with a learned model. *Nature* 588, 604–609. doi: 10.1038/s41586-020-03051-4
- Sheppard, B. (2002). World-championship-caliber Scrabble. *Artif. Intell.* 134, 241–275. doi: 10.1016/S0004-3702(01)00166-7
- Silver, D., Huang, A., and Maddison, C. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature* 529, 484–489. doi: 10.1038/nature16961
- Silver, D., Schrittwieser, J., and Simonyan, K. (2017). Mastering the game of Go without human knowledge. *Nature* 550, 354–359. doi: 10.1038/nature24270
- Wang, J., Ding, D., Xu, M., Han, B., and Lei, L. (2019). Air-to-air missile launchable area based on target escape maneuver estimation. *J. Beij. Uni. Aero Astronau.* 45, 722–734. doi: 10.13700/j.bh.1001-5965.2018.0462
- Watkins, H., and Dayan, P. (1992). Q-learning. *Mach. Learn.* 8, 279–292. doi: 10.1007/BF00992698
- Wei, R. X., Zhou, K., Ru, C. J., Guan, X., and Che, J. (2015). Study on fuzzy cognitive decision-making method for multiple UAVs cooperative search. *Sci. Sin. Tech.* 45, 595–601. doi: 10.1360/N092015-00130
- Williams, P. (1990). Three-dimensional aircraft terrain-following via real-time optimal control. *J. Guid. Control Dynam.* 13, 1146–1149.
- Wu, S., and Nan, Y. (2013). The calculation of dynamical allowable lunch envelope of air-to-air missile after being launched. *J. Project. Rockets Missile Guid.* 33, 49–54. doi: 10.15892/j.cnki.djzdx.2013.05.012
- Wu, W., Zhou, S., Gao, L., and Liu, J. (2011). Improvements of situation assessment for beyond-visual-range air combat based on missile launching envelope analysis. *J. Syst. Eng. Electron.* 33, 2679–2685. doi: 10.3969/j.issn.1001-506X.2011.12.20
- Yang, Z., Zhou, D., Piao, H., Zhang, K., Kong, W., and Pan, Q. (2020). Evasive maneuver strategy for UCAV in beyond-visual-range air combat based on hierarchical multi-objective evolutionary algorithm. *IEEE Access*. 8, 46605–46623. doi: 10.1109/ACCESS.2020.2978883
- You, Y., Li, J., and Shen, L. (2019). An effective Bayesian network parameters learning algorithm for autonomous mission decision-making under scarce data. *Int. J. Mach. Learn. Cyber.* 10, 549–561. doi: 10.1007/s13042-017-0737-x
- Zhang, P., Song, C., and Zhang, J. (2015). Development analysis of radar air-to-air missile. *Aerodyn. Missile J.* 4, 30–33. doi: 10.15892/j.cnki.fhdd.2015.04.008
- Zhang, Q., Yang, R., Yu, L., Zhang, T., and Zuo, J. (2018). BVR air combat maneuvering decision by using Q-network reinforcement learning. *J. Air For. Eng. Uni.* 19, 8–14. doi: 10.3969/j.issn.1009-3516.2018.06.002
- Zhou, T., Chen, M., Wang, Y., He, J., and Yang, C. (2020a). Information entropy-based intention prediction of aerial targets under uncertain and incomplete information. *Entropy* 22, 1–19. doi: 10.3390/e22030279
- Zhou, T., Chen, M., Yang, C., and Nie, Z. (2020b). Data fusion using Bayesian theory and reinforcement learning method. *Sci. China Inform. Sci.* 63, 170209. doi: 10.1007/s11432-019-2751-4