applied sciences

MDPI

Article

Maneuver Decision-Making for Autonomous Air Combat Based on FRE-PPO

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

Aeronautics Engineering College, Air Force Engineering University, Xi'an 710038, China
* Correspondence: hphpzhang@126.com

Abstract: Maneuver decision-making is the core of autonomous air combat, and reinforcement learning is a potential and ideal approach for addressing decision-making problems. However, when reinforcement learning is used for maneuver decision-making for autonomous air combat, it often suffers from awful training efficiency and poor performance of maneuver decision-making. In this paper, an air combat maneuver decision-making method based on final reward estimation and proximal policy optimization is proposed to solve the above problems. First, an air combat environment based on aircraft and missile models is constructed, and an intermediate reward and final reward are designed. Second, the final reward estimation is proposed to replace the original advantage estimation function of the surrogate objective of proximal policy optimization to improve the training performance of reinforcement learning. Third, sampling according to the final reward estimation is proposed to improve the training efficiency. Finally, the proposed method is used in a self-play framework to train agents for maneuver decision-making. Simulations show that final reward estimation and sampling according to final reward estimation are effective and efficient.

Keywords: autonomous air combat; maneuver decision-making; reinforcement learning; final reward estimation; proximal policy optimization

check for updates

Citation: Zhang, H.; Wei, Y.; Zhou, H.; Huang, C. Maneuver Decision-Making for Autonomous Air Combat Based on FRE-PPO. *Appl. Sci.* **2022**, *12*, 10230. https://doi.org/10.3390/app122010230

Academic Editor: Alessandro Gasparetto

Received: 27 September 2022
Accepted: 9 October 2022
Published: 11 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the development of artificial intelligence, autonomous air combat has become more and more attractive. In August 2020, the US Defense Advanced Research Projects Agency held the Alpha DogFight competition. The air combat agent developed by Heron systems defeated the human pilot by 5:0 in close air combat [1]. This competition totally reflects the great advantages and potential of artificial intelligence in autonomous air combat, and the key to the victory of air combat is the maneuver decision-making method. Therefore, it is very important to study maneuver decision-making methods based on artificial intelligence. However, artificial intelligence-based methods require numerous samples for training agents, and sometimes, even much more samples cannot guarantee performance or efficiency, which is an urgent problem needed to be solved. Therefore, we mainly focus on improving the performance and efficiency of the maneuver decision-making method based on reinforcement learning (RL).

In recent studies, most of the research on maneuver decision-making for air combat focused on within-visual-ranges air combat [2–6], which excludes the kinematic model and the dynamic model of missiles and thus is insufficient and defective for addressing the problems of maneuver decision-making. Wei et al. [7] proposed a cognitive control model with a three-layered structure for multi-UAV cooperative searching according to the cognitive decision-making mode of humans performing searching behavior. The mission attack area is based on cognitive matching, reduction, and division based on this model and the fuzzy cluster. The simulation experiments indicate the great performance of the fuzzy cognitive decision-making method for cooperative search. On the other hand, this method mainly focuses on cooperative search, which may not be effective for maneuver decision-making because these problems are different from each other. Zhang et al. [8] proposed

a maneuver decision-making method based on the Q network and Nash equilibrium strategy, combining the missile attack area in the reward function to improve the efficiency of RL. However, the maneuver library of this method contains only five maneuvers, which cannot meet the requirements of air combat. Hu et al. [9] proposed to use the improved deep Q network [10] for maneuver decisions in unmanned air combat, constructed the relative motion model, missile attack model and 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 action space of the proposed method is discrete. However, the action space should be continuous to make the proposed method more reliable.

It is worth noting that deep RL has achieved professional performance in video games [11–13], board games such as GO [14–16], real-time strategy games such as StarCraft [17] and the magnetic control of Tokamak plasmas [18], which indicates the generality of deep RL for solving decision-making problems. Therefore, using deep RL to improve the level of air combat maneuver decision-making is a feasible potential direction. Ma et al. [19] described the cooperative occupation decision-making problem of multiple UAVs [20] as a zero-sum matrix game problem and proposed a solution of a double oracle algorithm combined with neighborhood search. However, the method did not incorporate the missile into cooperative occupation decision-making problems. Ma et al. [21] built an air combat game environment and trained the agent with deep Q-learning with a discrete action space. Eloy et al. [22] studied attacks against static high-value targets in air combat. They analyzed the confrontation process with game theory and put forward a differential game method of air combat combined with the missile attack area [23–25]. However, their flight model was two-dimensional rather than three-dimensional. He et al. [26] proposed a maneuver decision-making method based on Monte Carlo tree search (MCTS), which uses MCTS to find the action with the maximum air combat advantage function value among seven basic maneuvers. This method verifies the feasibility of MCTS for maneuver decision-making, however, its action space is discrete as well.

Although the above studies have made some progress, there are still some problems in current air combat maneuver decision-making. First, the action space of air combat maneuver decision-making is continuous, but existing methods usually adopt the method of continuous space discretization, which is a common drawback of existing methods. Second, existing decision-making methods determine whether it wins or not according to whether the target enters the missile attack area, namely, excluding the kinematic model and the dynamic model of the missile, which is another drawback. However, the target may not be shot down by the missile even if it enters the missile attack area. Therefore, it is necessary to determine whether it wins or not according to whether the missile hits the target. Third, RL algorithms often perform poorly in maneuver decision-making, so it is urgent to propose more effective and efficient RL algorithms address maneuver decision-making. These problems still require to be solved, and the goal of this article is to solve these problems.

To solve the above problems of air combat, we propose a method based on the final reward estimation and proximal policy optimization (FRE-PPO). Our main contributions are as follows: (1) we used a deep RL method, namely proximal policy optimization, to handle maneuver decision-making in continuous action space. (2) We propose to use final reward estimation to replace the original advantage estimation function of the surrogate objective of proximal policy optimization to improve training performance and efficiency. (3) The proposed method is conducted in a self-play framework to generate air combat agents. (4) Ablation studies are provided to support our claims. Section 2 introduces the basic knowledge of RL. In Section 3, an air combat simulation environment based on an aircraft model and missile model is constructed, and an intermediate reward and final reward are designed. And the air combat agent training framework based on FRE-PPO is proposed, which can improve the maneuver decision-making ability of air combat agents

by means of self-play and FRE-PPO. In Section 4, ablation studies and simulation experiments are performed, and FRE-PPO is compared with the other four methods to verify its performance. Section 5 summarizes the experimental results and draws conclusions.

2. Background

The advantage of the maneuver decision-making method based on RL (e.g., PPO) is that it can directly use the continuous action space without discretization of the action space. At the same time, it does not rely on expert experience and can adapt to changes in the air combat environment. Meanwhile, RL can create agents outperforming human beings [14–16]. Therefore, RL has some advantages and potential in air combat maneuver decision-making. However, there are still some problems when using RL in air combat maneuver decision-making: it is difficult to train effective agents using RL; RL method usually performs well in Atari [27] and robot control tasks [28], but air combat maneuver decision-making is different from these tasks. Therefore, how to guarantee better and more stable performance in air combat maneuver decision-making using RL needs to be addressed.

The purpose of RL is to learn policies to maximize rewards in sequential decision-making [29]. Sequential decision-making is that, at each discrete time step t, an agent starts from a certain state, s_t, chooses an action, a_t, according to its policy and obtains a reward, r_t. Then, it takes the discounted sum of rewards (DSR) as its return with a discount factor. The aim of the agent acting in a stochastic environment by sequentially choosing actions over a sequence of time steps is to obtain more rewards or win the game. Air combat maneuver decision-making can be regarded as a sequential decision-making problem in continuous action space. Therefore, the maneuver decision-making problem is modeled as the Markov decision process (MDP), a sequential decision-making process, and the agent interacts with the air combat environment by its policies and then obtains a tuple (s_t, a_t, r_t).

RL can be divided into value-function-based RL (VBRL) and policy-based RL (PBRL). VBRL uses the temporal difference (TD) error to optimize the value function of the action and selects the action corresponding to the value function. Its disadvantage is that if the action space is continuous, the action space must be discretized at first, which results in the exponential growth of actions. Value-based algorithms include Q-learning [11], deep Q-network (DQN) [10] and its variants: (1) prioritized experience replay [30] weights sampling of data based on TD error to improve learning efficiency; (2) dueling DQN [31] improves the network structure by decomposing the action value function Q into the state value function and the advantage function to improve the function approximation ability; (3) double DQN [32] uses two independent action value functions to solve the problem of over estimation; (4) noisy DQN [33] adds noise to network parameters to increase the exploration ability of agents; (5) distributed DQN [34] refines the state action value estimation into the estimation of the state action value distribution; (6) rainbow [13] combines the characteristics of the above DQN based methods; (7) R2D2 [35] studies the effect of adding recurrent neural networks to RL; (8) NGU [36] uses short-term novel incentives in an episode and long-term novel incentives across multiple episodes to increase the exploration ability of agents; (9) agent57 [37], based on NGU, solves the problem of long-term credit allocation by improving the training stability and dynamically adjusting discount factors.

PBRL methods directly parameterize policies to maximize DSR by updating the policy using a policy gradient. Its advantage is its fast convergence speed and can be directly used in the continuous and high-dimensional action space without discretization. Policy-based algorithms include policy gradient algorithm [38], natural policy gradient algorithm [39], deterministic policy gradient algorithm [40], deep deterministic policy gradient algorithm [41], dual delay deep deterministic policy gradient algorithm [42], trust region policy optimization (TRPO) [43] and proximal policy optimization (PPO) [44]. TRPO and PPO limit the update of the policy. TRPO can guarantee its cost after policy update is non-increasing. PPO clips the probability ratio of the policy, making the algorithm more

stable. Because VBRL cannot be directly used in continuous action space, this paper uses PBRL to solve the air combat maneuver decision-making problem.

3. Method

3.1. Aircraft Model and Missile Model

In maneuver decision-making, the aircraft model adopts normal overload, tangential overload and roll angle as control parameters. To simplify the complexity of the problem, the angles of attack and angle of side slip are regarded as 0, 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 [45]:

$$\begin{cases} \dot{x} = v\cos\gamma\cos\psi \\ \dot{y} = v\cos\gamma\sin\psi \\ \dot{z} = v\sin\gamma \\ \dot{v} = g(n_x - \sin\gamma) \\ \dot{\gamma} = \frac{g}{v}(n_z\cos\mu - \cos\gamma) \\ \dot{\psi} = \frac{g}{v\cos\gamma}n_z\sin\mu \end{cases} \quad (1)$$
$$n_x \in [-0.5, 1.5]$$
$$n_z \in [-3, 9]$$
$$\mu \in [-\pi, \pi]$$

where x, y, and z indicate the positions of the aircraft in the inertial coordinate system; γ is the pitch angle, ψ is the yaw angle, v is the velocity, and g is the acceleration of gravity. Roll angle μ, tangential overload n_x, and normal overload n_z are control parameters. The action space consists of μ, n_x, and n_z, which is continuous, as shown in (1). This means that the control parameters are selected from the three intervals instead of using space discretization as in previous methods. The kinematic model of the missile is [46]:

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

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

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

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

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

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

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

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

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

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

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

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

The maximum overload of the missile is 40. When the minimum distance between the missile and the target is less than 12 m, the target is regarded as being hit; when the missile flight time exceeds 120 s, 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°; during the final guidance stage, the target is regarded as missed when its azimuth relative to missile axis exceeds 70°.

3.2. Reward Design

In RL, the agent maximizes its return through trial and error experience [47]. Due to sparse rewards, it is difficult for agents to obtain rewards in an air combat environment and complete the given task. Therefore, in order to avoid invalid training caused by sparse rewards, this paper proposes intermediate rewards and final rewards. The intermediate reward is determined by the situation information of air combat, and the agent can get an intermediate reward at each time step, so the reward is not sparse, and the agent may get the final reward according to the assistance of the intermediate reward. The final reward is determined by the result of air combat, namely win, lose or draw. As the agent can only get the final reward at the end of air combat, the final reward is sparse. The combination of intermediate reward and final reward can effectively avoid bad training performance when only using the final reward. The formula of intermediate reward r_M and the explanation of the meaning behind the formula is shown as follows:

$$\begin{aligned} r_a &= \omega_{a1}(\psi_{\max 1} - \psi_{r1}) + \omega_{a2}(\gamma_{\max 1} - \gamma_{r1}) + \omega_{a3}(\psi_{r2} - \psi_{r1}) + \omega_{a4}(\gamma_{r2} - \gamma_{r1}) + \min(\tfrac{\mathrm{D}}{d}, 1) \\ r_b &= \omega_b \tfrac{|z_1 - \mathrm{H}|}{\mathrm{H}} \\ r_c &= \omega_c t_{missile} \\ r_d &= \begin{cases} 1, \text{ launched} \\ 0, \text{ not launched} \\ -1, \text{ miss the target} \end{cases} \\ r_M &= r_a + r_b + r_c + r_d \end{aligned} \tag{10}$$

where, $\gamma_{\max 1}$ and $\psi_{\max 1}$ are the maximum pitch and yaw angle of the missile, respectively, γ_{r1} and ψ_{r1} are, respectively, the pitch and yaw angles of the speed vector of one aircraft relative to the line of sight, γ_{r2} and ψ_{r2} are, respectively, the pitch and yaw angles of the speed vector of another aircraft relative to the line of sight. ω_{a1}, ω_{a1}, ω_{a3}, and ω_{a4} are corresponding weights. As can be seen in (10), smaller γ_{r1} and ψ_{r1} mean larger r_a, which motivates the aircraft to fly to the target (it can be regarded as an attack); larger γ_{r1} and ψ_{r1} mean smaller r_a, which indicates that the target is approaching the aircraft (it can be regarded as being attacked). D is the typical attack distance of the missile, and d is the distance between both sides, so the item min(D/d, 1) motivates the aircraft to approach the target. Therefore, r_a is set as its opposite during the final guidance stage to make the

aircraft fly away from the target since, during the final guidance stage, the missile no longer needs the information transited by the aircraft. z_1 is the altitude of the aircraft, H is the typical flight altitude, and ω_b is the weight. Therefore, r_b encourages the aircraft to stay at its typical flight altitude. $t_{missile}$ is the flight time after the missile is launched and ω_c is the weight. This means that earlier launching corresponds to a larger reward. When the missile is launched, the agent gets a reward of 1. When the missile is not launched, the agent gets a reward of 0. When the missile loses its target, the agent gets a reward of -1. This item means launching the missile is encouraged, and missing the target is punished. The sum of the above four rewards is the intermediate reward r_M. The formula of the final reward r_F is:

$$r_F = \begin{cases} 30, \text{ win} \\ -30, \text{ lose} \\ 0, \text{ draw} \end{cases} \tag{11}$$

The result of the air combat simulation is defined as follows: if one side's missile hits its target, the side wins air combat, and the other side hit by the missile loses air combat; either the aircraft or the missile misses the target, it is regarded as missing the target, if both sides lose targets or the simulation reaches the time limit, this air combat is regarded as a draw. At the end of air combat, the winner gets a final reward of 30, and the loser gets a final reward of -30, and in other cases, the final reward is 0. Therefore, the reward obtained by the agent at each time step is $r_t = r_M + r_F$. Finally, we set ω_{a1}, ω_{a1}, ω_{a3} and ω_{a4} to 1 because the radar may lose its target if the azimuth angle of the target is bigger than the maximum pitch angle or maximum yaw angle. We set $\gamma_{\max 1}$ and $\psi_{\max 1}$ to $\pi/3$ because radar cannot detect the target from all directions simultaneously, and these parameters can be changed according to different radars. We set ω_b to -4 in order to reduce the altitude difference between the aircraft and its target. We set ω_c to $1/60$ to encourage the maneuvers which make the target within radar detection range.

3.3. FRE-PPO Algorithm

TRPO repeatedly performs the following steps:

- Use the single path or vine procedures to collect a set of state-action pairs along with Monte Carlo estimates of their Q-values;
- By averaging over samples, construct the estimated objective and constraint in

$$\begin{aligned} &\underset{\theta}{\text{maximize}}\ \mathrm{E}_{s\sim\rho_{\theta_{old}},a\sim q}\left[\frac{\pi_\theta(a|s)}{q(a|s)}Q_{\theta_{old}}(s,a)\right] \\ &\text{subject to } \mathrm{E}_{s\sim\rho_{\theta_{old}}}\left[D_{KL}(\pi_{\theta_{old}}(\cdot|s)||\pi_\theta(\cdot|s))\right] \le \delta. \end{aligned} \tag{12}$$

- Approximately solve this constrained optimization problem to update the policy's parameter vector θ by means of the conjugate gradient algorithm followed by a line search, which is altogether only slightly more expensive than computing the gradient itself. Where, θ_{old} is the parameter of the current strategy, ρ is the probability distribution of state transition, q is the sampling distribution, and D_{KL} is the *KL* divergence.

By establishing the Fisher information matrix, TRPO avoids storing a dense Hessian matrix or policy gradients from a batch of trajectories and improves computational efficiency. Theoretically, the penalty coefficient of larger *KL* divergence can lead to a small step size, so the penalty coefficient should be decreased. However, it is difficult to select a penalty coefficient that can ensure the stability of the effect, so TRPO uses a constraint term instead of a penalty term. At the same time, since the maximum *KL* divergence is not easy to be optimized and estimated, TRPO uses the average *KL* divergence.

PPO is an improved version of TRPO. TRPO may cause excessive policy updates when maximizing the objective function, so PPO modifies the objective function to punish the policy changes that make the probability ratio far from 1. The objective function of PPO is:

$$\begin{aligned} L^{CLIP}(\theta) =& \mathrm{E}[\min(r_t(\theta)A_t, \mathrm{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)A_t)] \\ P_r(\theta) =& \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}. \end{aligned} \tag{13}$$

by clipping the probability ratio P_r and removing the strategy change that makes the probability ratio exceed the interval $[1-\varepsilon, 1+\varepsilon]$. The formula of the DSR advantage function used in PPO is: $A_t = \mathrm{r}_t + \gamma \mathrm{r}_{t+1} + \cdots \gamma^{T-t+1}\mathrm{r}_{T-1} + \gamma^{T-t}V(s_T) - V(s_t)$. The formula of generalized advantage estimation (GAE) [48] used in PPO is: $A_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \cdots + (\gamma\lambda)^{T-t+1}\delta_{T-1}$, where $\delta_t = \mathrm{r}_t + \gamma V(s_{t+1}) - V(s_t)$. Since GAE and DSR are not effective for solving the air combat maneuver decision-making problem, this paper proposes a new function, namely, final reward estimation (FRE). The formula of FRE is defined as follows:

$$A_t = \begin{cases} \mathrm{r}_t + \gamma^{T-t}\mathrm{r}_T, t < T \\ \mathrm{r}_T, t = T \end{cases} \tag{14}$$

where T is the last time step in air combat. This method uses self-play samples obtained from the air combat environment to train the maneuver policy, which is represented by neural networks. The input of the neural networks is air combat situation information, and the output is the three control quantities of aircraft and whether to launch a missile. At the same time, in order to increase the exploration of the policy, Gaussian noise is added to each action. After obtaining training samples, a certain number of samples need to be selected from these samples to train the neural network. A common sampling method is randomly uniform sampling. In order to improve the efficiency and effect of training, the method of sampling according to FRE is proposed in order to collect training samples, that is, the higher FRE of the sample, the greater the probability of being collected. The probability of each sample is calculated by the following formula:

$$P(i) = \frac{A_i^\alpha}{\sum_k A_k^\alpha} \tag{15}$$

where A_i is calculated by Formula (14), and the index α represents the role of FRE in probability. When $\alpha = 0$, it is identical to uniform choice; when $\alpha = 1$, samples are completely sampled according to FRE. Figure 1 shows the FRE-PPO self-play architecture. First, run the policy in an air combat environment and acquire transitions (s_t, a_t, and r_t). Second, store these transitions in the experience pool and compute probabilities of transitions according to FRE. Then, sample transitions according to probabilities and use these transitions to train neural networks.

3.4. Air Combat Agent Training Framework Based on FRE-PPO

Combined with the aircraft model, missile model, reward design, and FRE-PPO algorithm, an air combat agent training framework based on FRE-PPO is proposed. The agent training framework consists of six parts: actor, experience pool, learner, player pool, evaluator and recorder. The framework is shown in Figure 2.

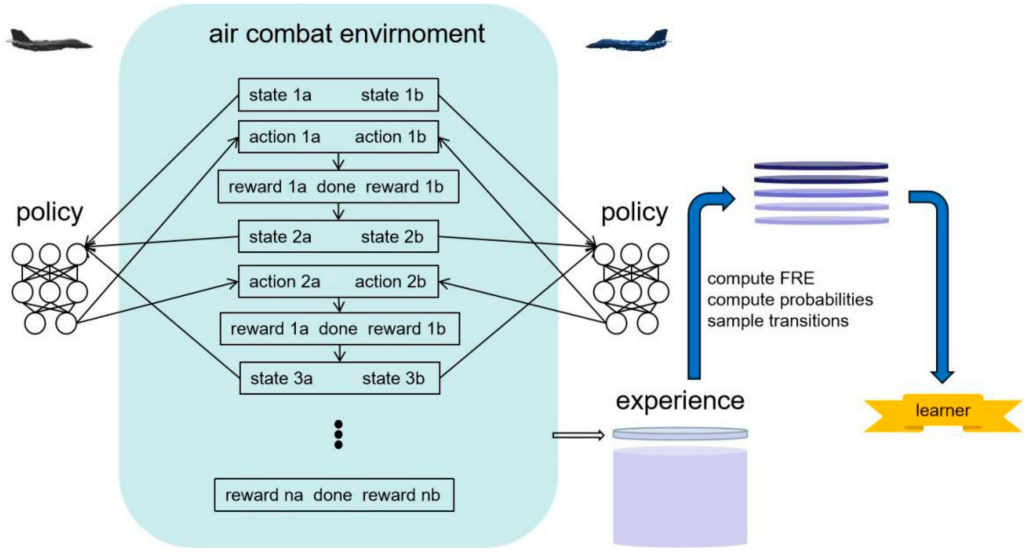


Figure 1. FRE-PPO Self-Play Architecture.

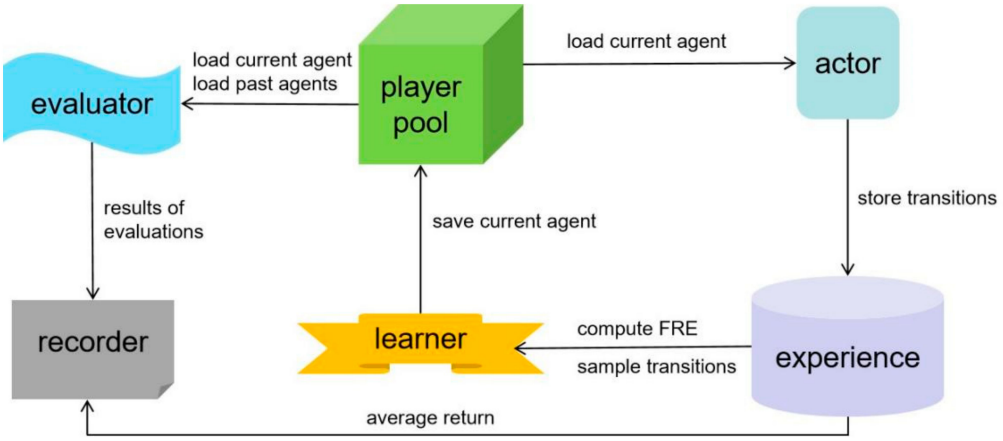


Figure 2. Air Combat Agent Training Framework.

In each iteration, the actor first generates self-play data in the air combat environment and stores these data in the experience pool. After that, the experience pool calculates the FRE of each sample and the average returns of all data and stores the average returns in the recorder. Then, a batch of samples with the size of 1024 is collected from the experience pool according to FRE and sent to the learner. The learner trains the current agent using these received samples and then stores the trained agent in the player pool. Then, the player pool randomly selects 36 past agents and sends these agents to the evaluator, where each past agent is compared against the current agent 3 times. The initial angle, speed and distance of air combat are randomly selected within a certain range. After the simulations, the number of wins, losses, and draws are stored in the recorder. According to the results of evaluations, it can be seen whether the decision-making ability of the agent has increased after training. Finally, the player pool sends the current agent to the actor to start the next iteration.

Obviously, the more wins, the stronger the decision-making ability of the agent. It should be pointed out that more losses also mean more effective agent training because the agent is compared against its own past version. Therefore, more losses mean a stronger past version of the current agent, which can indicate that the training of the agent is effective. On the contrary, a large number of draws means that the training of the agent may be invalid (a draw represents a missile missing the target or simulation timeout). At the beginning of training, there are usually a large number of draws; at this time, the agent often maneuvers randomly, so its missile easily misses the target, which results in a large number of draws. Therefore, if there are still a large number of draws after training for a period of time, the agent training may be invalid.

3.5. Air Combat State

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 ψ and γ are the yaw angle and pitch angle of the velocity vector relative to the line of sight, respectively, v is the velocity of the aircraft, z is the flight altitude, d is the distance between the two sides in air combat, r_1 and r_2 are the coordinates of the two sides, respectively, f_1 represents whether the agent should launch a missile, ψ_1 and γ_1 are the yaw angle and pitch angle of the missile's velocity vector relative to the line of sight, respectively, d_1 is the distance between the missile and the other agent, r_{m1} is the coordinate of the missile in air combat, β is the heading crossing angle, that is the angle between the 2 velocity vectors of the 2 sides, represented by v_1 and v_2 in Table 1, and f_2 represents whether the other side launched a missile. The input layer is followed by 3 hidden layers. Finally, it outputs 4 quantities. The first 3 outputs are normal overload, tangential overload and roll angle, respectively, and the fourth output is whether to launch the missile. The activation function is tanh.

Table 1. Air combat state.

State	Symbol	Formula
yaw angle	ψ	$\psi = \psi_0 + \int \frac{g}{v\cos\gamma} n_z \sin\mu \mathrm{d}t$
pitch angle	γ	$\gamma = \gamma_0 + \int \frac{g}{v}(n_z\cos\mu - \cos\gamma)\ \mathrm{d}t$
velocity	v	$v = v_0 + \int g(n_x - \sin\gamma)\ \mathrm{d}t$
altitude	z	$z = z_0 + \int v\sin\gamma\ \mathrm{d}t$
distance between the two sides	d	$d = \|r_1 - r_2\|$
launch missile	f_1	0 or 1
yaw angle of the missile	ψ_1	$\psi_m = \psi_{m0} + \int \frac{n_{mc}g}{v_m\cos\gamma_m}\ \mathrm{d}t$
pitch angle of the missile	γ_1	$\gamma_m = \gamma_{m0} + \int \frac{n_{mh}g}{v_m} - \frac{g\cos\gamma_m}{v_m}\ \mathrm{d}t$
distance between the missile and the other side	d_1	$d_1 = \|r_{m1} - r_2\|$
heading crossing angle	β	$\beta = \arccos(\frac{v_1 \cdot v_2}{\|v_1\|\|v_2\|})$
launch missile from the other side	f_2	0 or 1

4. Experiments and Results

In order to verify the performance of the proposed method, a large number of experiments are performed in this section. The proposed method is compared with the other four methods from two aspects of statistics and simulation results. In each simulation, the initial states of two sides of air combat (i.e., azimuth angles, velocities and distance between the two sides) are selected from corresponding intervals in Table 2. Then the two sides begin to maneuver by means of neural networks to generate training samples. Five independent experiments were performed for each method. Each experiment consists of 40 iterations in total, and each iteration consists of 20 cycles, as shown in Figure 2. The experimental hyperparameters are set as follows:

Table 2. Hyperparameters.

Hyperparameter	Value
azimuth angle	$(-\pi/4, \pi/4)$
distance	(40,000 m, 100,000 m)
velocity	(250 m/s, 400 m/s)
batch size	1024
optimizer	Adam
actor learning rate	0.0002
critic learning rate	0.001
actor architecture	(256, 256, 4)
critic architecture	(256, 256, 1)
activate function	tanh
epoch	8
γ	0.995
λ	0.98
ε	0.2

4.1. Ablation Studies

In this section, five methods are used for experiments: FRE, FRE-U, FRE-S, GAE and DSR. FRE stands for using FRE in the objective function and collecting samples according to FRE, FRE-U stands for using FRE in the objective function and collecting samples randomly, FRE-S stands for using FRE in the objective function without intermediate rewards and collecting samples according to FRE, GAE stands for using GAE in the objective function, DSR stands for using DSR in the objective function. In each iteration, the evaluator records the evaluation results of the method. Figure 3 shows the change in the average return of FRE, FRE-U and FRE-S during the agent training process. The solid line represents the mean of average returns of the five experiments, and one point is drawn on the solid line every five gradient steps. The shaded part represents the standard deviation of average returns. As can be seen from Figure 3, since agents have not yet acquired any effective maneuver policy, average returns in the early training are negative. After several times of training, the average returns of FRE and FRE-U have an obvious upward trend, while the average return of FRE-S is always around 0.

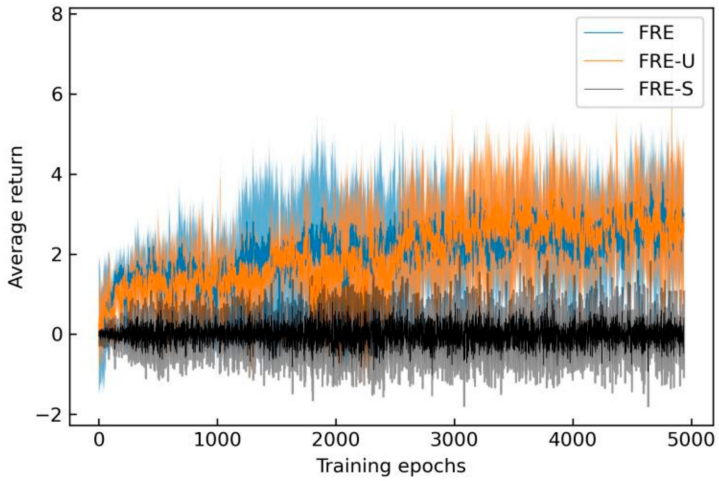


Figure 3. Average Return.

The blue line represents the average return of FRE, the orange line represents the average return of FRE-U and the black line represents the average return of FRE-S. Two points about the change of average return need to be declared: 1. In air combat maneuver decision-making, the return value obtained by the agent may be related to whether it can win air combat without causality, that is, the maneuver decision-making ability of air

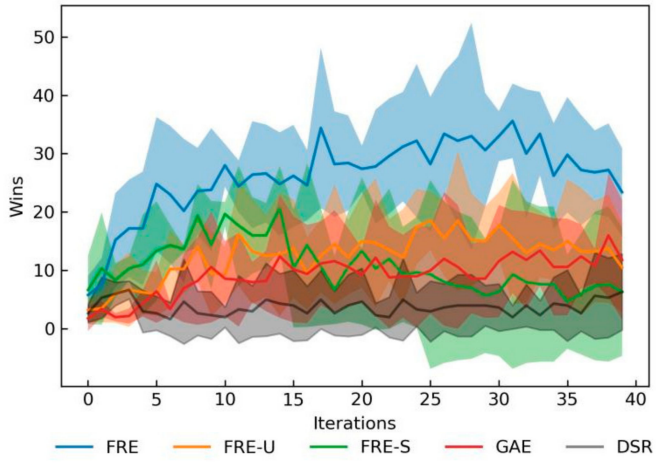
combat agent cannot be compared only according to returns it obtained. A high return value does not mean more probability of a win, and a low return value does not mean less probability of a win. The increase in the return value only means that the agent gradually learns the policy to increase the reward. 2. FRE-S only includes the final reward, while FRE and FRE-U include not only the final reward but also the intermediate reward. Therefore, the possible return value of FRE and FRE-U is more than that of FRE-S. According to the above two points, to accurately evaluate the performance of different methods, we need to use the results of air combat confrontation. Therefore, the results of different methods are compared in the evaluator, where each evaluation contains 108 rounds. The variations of the number of wins, losses, and draws in each method with the number of iterations are shown in Figure 4. The solid line represents the mean of the number of wins, loses or draws of this iteration over the five experiments, and the shaded part represents the standard deviation of the number of wins, loses or draws.

Obviously, the more wins, the stronger the decision-making ability of the agent. It should be pointed out that more losses also mean more effective agent training because the agent is playing against its own past versions. Therefore, more losses represent the stronger the decision-making ability of the past versions of the agent, which can indicate that the training of the agent is effective. On the contrary, a large number of draws means that the training of the agent may be invalid. At the beginning of training, there are usually a large number of draws because, at this time, the agent often maneuvers randomly, so its missile is likely to miss the target, which results in a large number of draws. Therefore, if there are still a large number of draws after training for a period of time, it indicates that the training may be invalid.

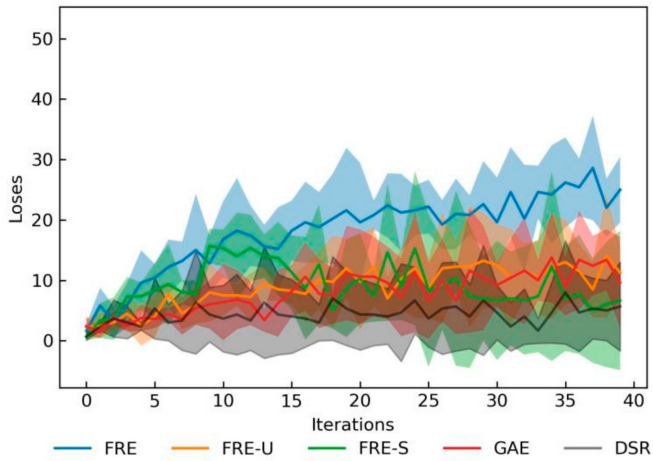
As shown in Figure 4a, the number of wins of FRE is the most among the five methods, and it increases faster than the other four methods. The number of wins of FRE-U and GAE also gradually increase, but their increasing rates are significantly slower than that of FRE, and the numbers of wins are only half of that of FRE. At the same time, the number of wins of FRE-U is slightly more than that of GAE, which indicates that final reward estimation is better than generalized advantage estimation in air combat maneuver decision-making. In the first 14 iterations, the number of wins of FRE-S increases rapidly, but it gradually decreases from the 15th iteration, which indicates that the early training of FRE-S is effective, while the later training is invalid. In addition, the winning rate of the first 14 iterations of FRE-S is faster than that of FRE-U, which indicates that sampling according to FRE is an effective method to improve training efficiency. However, the sparse rewards can make training more difficult, so the number of wins of FRE-S in the later period gradually decreases. The number of wins of DSR does not increase significantly, which indicates that DSR is invalid in air combat maneuver decision-making. It should be noted that some shaded parts of FRE-S and DSR are less than zero, it is because the mean values of the winning games of FRE-S and DSR are small, and their variances are large, which results in the shaded parts being less than zero. However, this does not mean that the wins are less than zero.

As shown in Figure 4b, The number of losses of FRE is also the most among the five methods, and it gradually increases during training. At the same time, its increased rate of losses is also the fastest. Because the agent fights against its own past versions, a large number of losses of the current agent indicates that the number of wins by its past versions is large, which means that the training method based on FRE is effective. The number of losses of FRE-U and GAE also increases, which shows that the two methods are also effective. The number of losses of FRE-S first increases and then decreases, with the same trend as its number of wins, which shows that the early training of FRE-S is effective, while the late training is invalid. At the same time, the decrease in the number of losses in the later period of the FRE-S is smaller than that of the number of wins. This is because the opponents are randomly selected from the player pool during evaluations, so it is possible to select the opponents who have been effectively trained in the early training stage. Therefore, its number of losses increases. As for FRE-S, the decrease in the number of losses is smaller than that of the number of wins. Similar to the number of wins, the

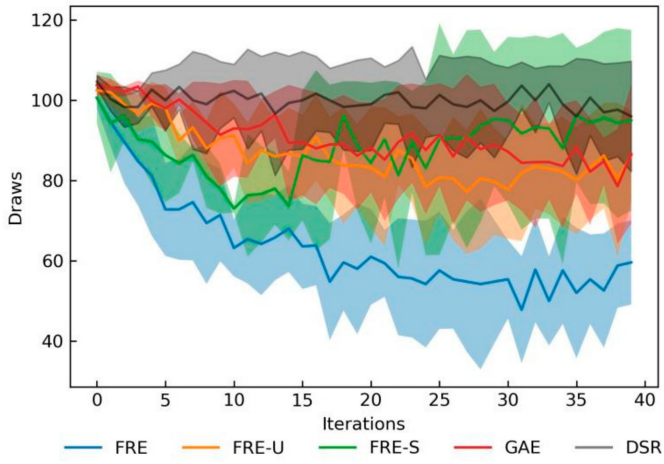
number of losses of DSR does not increase, which indicates that DSR is invalid in air combat maneuver decision-making.



(**a**)



(**b**)



(**c**)

Figure 4. (**a**) Wins, (**b**) Losses, and (**c**) Draws.

As shown in Figure 4c, the number of draws of FRE decreases obviously during training, and its amplitude is also large. The decreasing rate of the number of draws of FRE-U is faster than that of GAE, which indicates that the performance of FRE-U is better than GAE. The number of draws of FRE-S first decreased and then increased, and the number of draws of DSR is almost not decreased, which is consistent with the results in Figure 4a,b.

Table 3 shows the maximum number of wins, average number of wins, average number of losses and average number of draws from each method's overall evaluations. As can be seen from Table 3, the maximum number of wins for FRE over all evaluations was 55, the maximum number of wins for FRE-U and FRE-S were 32 and 28, respectively, and the maximum number of wins for GAE and DSR were 32 and 14, respectively. Therefore, the maximum number of wins for FRE was significantly more than that of the other four methods. The average number of wins for FRE was 26.22, 13.77 more than that of FRE-U, 15.44 more than that of FRE-S, 17.02 more than that of GAE and 22.47 more than that of DSR. The average number of losses for FRE was 18.08, 9.12 more than that of FRE-U, 9.05 more than that of FRE-S, 10.12 more than that of GAE, and 13.71 more than that of DSR. The average number of wins for DSR was less than the average number of losses, which shows again that DSR is invalid in air combat maneuver decision-making. In addition, the average wins and losses of FRE-S were more than that of GAE because FRE-S is more efficient in the early stage than GAE, and thus FRE-S accumulated more wins and losses. Table 3 also shows the time consumption of different methods. Because the average time of each decision-making is determined by the scale of neural networks, and the network scales of different methods are the same, so the average time of each decision-making is the same. Meanwhile, it can be seen that the average time of the proposed method is approximately 0.001 s, which meets the real-time requirements.

Table 3. Statistical results of evaluations.

	GAE	DSR	FRE-U	FRE-S	FRE
maximum wins	32	14	32	28	55
average wins	9.20	3.75	12.45	10.68	26.22
average loses	7.96	4.37	8.96	9.03	18.08
average draws	90.84	99.88	86.59	88.29	63.70
average time of each decision-making	0.001 s	0.001 s	0.001 s	0.001 s	0.001 s

Above all, comparing the experimental results of FRE-U with GAE and DSR, it can be seen that the effect of the final reward estimation proposed in this paper is better than GAE and DSR. Comparing the experimental results of FRE-S with FRE-U, GAE and DSR, it can be seen that the method of sampling according to FRE can significantly improve the training efficiency. Comparing the experimental results of FRE with FRE-S, FRE-U, GAE and DSR, it can be seen that the intermediate reward proposed in this paper can avoid invalid training caused by sparse reward in the later training stage. Meanwhile, combining FRE with the method of sampling according to FRE can improve training efficiency and performance.

4.2. Simulation Experiments

The above experiments show the statistical results of agent training, and the following experiments show the simulation results of agent training. Figure 5 shows the air combat simulation results of the final agent against the first, eighth, 16th, 24th and 32nd agents starting from the equal initial situation. The initial position of the final agent is (0, 0, 10,000), the initial yaw angle is 0°, and the initial pitch angle is 0°. The initial positions of other agents are (70,000, 70,000, and 10,000), the initial yaw angle is 180°, and the initial pitch angle is 0°. Figure 5a,c,e,g,i are three-dimensional trajectories of the final agent against the first, eighth, 16th, 24th and 32nd agents, respectively. Figure 5b,d,f,h,j are top views corresponding to the three-dimensional trajectories. In Figure 5, the solid line represents the trajectory of the aircraft, the dotted line represents the trajectory of the corresponding missile, and the two dots represent the initial positions of both sides of air combat.

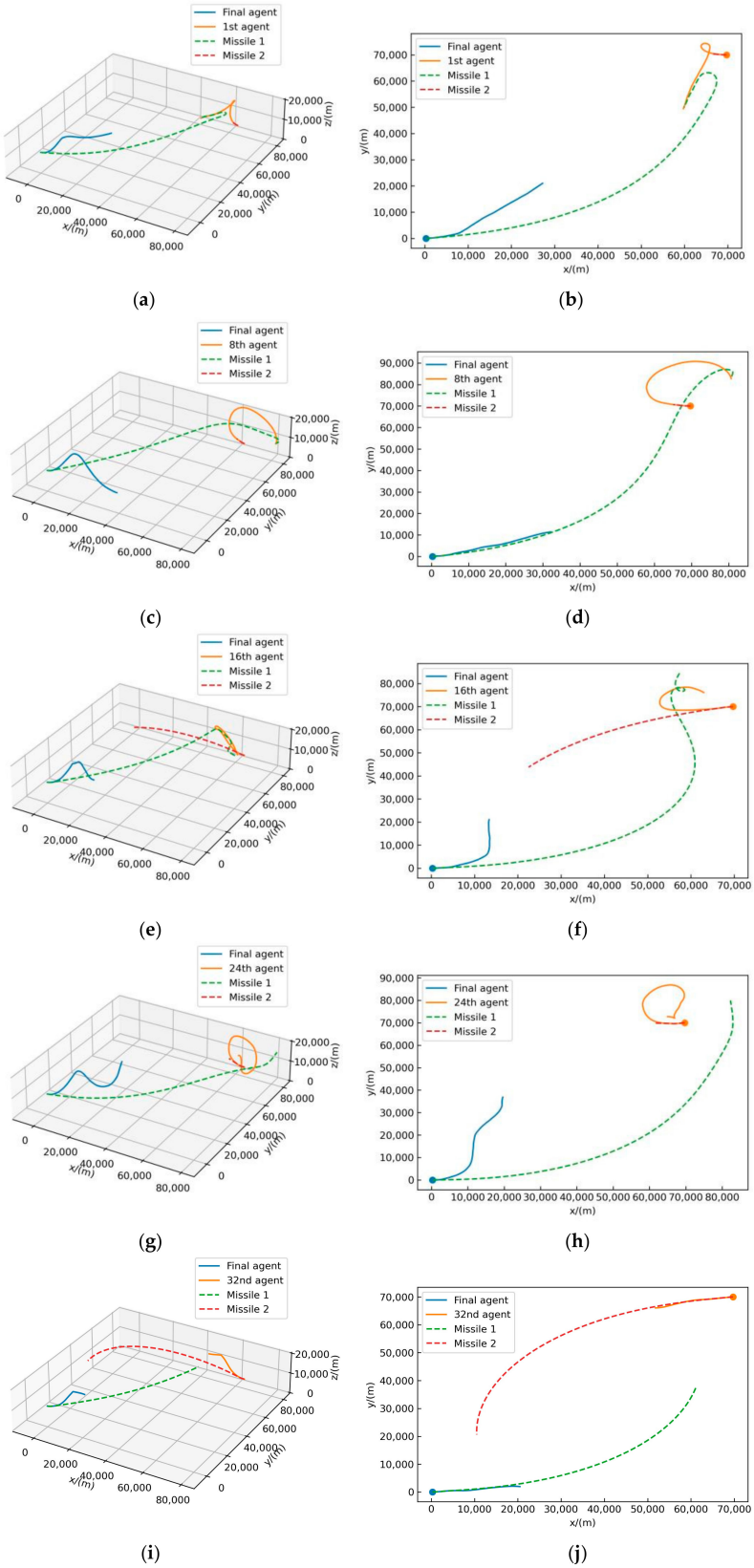


Figure 5. Comparison of Training Results of Different Iterations. (**a**) The 3D trajectory of the first agent. (**b**) Top view of the trajectory of the first agent. (**c**) The 3D trajectory of the eighth agent. (**d**) Top view of the trajectory of the eighth agent. (**e**) The 3D trajectory of the 16th agent. (**f**) Top view of the trajectory of 16th agent. (**g**) The 3D trajectory of the 24th agent. (**h**) Top view of the trajectory of the 24th agent. (**i**) The 3D trajectory of the 32nd agent. (**j**) Top view of the trajectory of the 32nd agent.

As can be seen from Figure 5a,b, because the first agent has not been trained, it maneuvers randomly, and it can be considered that the first agent cannot make effective maneuver decision-making. It can be seen from Figure 5a that the flight altitude of the first agent gradually rises with random maneuvers, and the final agent also gradually rises in order to defeat it. From the top view of the flight trajectory of the final agent, it can be seen that the final agent flies in the direction of the first agent and launches a missile, and the first agent is hit by the missile of the final agent at last, which indicates that the agent acquired how to attack its target after training of the proposed method.

As can be seen from Figure 5c,d, after launching its missile, the eighth agent soon loses its target, so it flies away from the final agent to avoid the missile of the final agent. At this time, the final agent does not lose its target by means of its maneuver decision-making. Finally, the eighth agent is hit by the missile of the final agent, and the simulation ends. According to the above analysis, the eighth agent does not acquire effective attack policies, but it acquires some evasive policies to get rid of missile tracking, although these policies are not effective enough.

5. Discussion

Trained from scratch, the agent can gradually improve its maneuver decision-making ability by means of the proposed method. Comparing Figure 5e,f with the above four figures, it can be seen that the flight time of the 16th agent's missile increases, which indicates that the 16th agent has learned some attack policies. However, during the confrontation with the final agent, the missile of the 16th agent loses its target, so the attack capability of the 16th agent is not as good as that of the final agent. 16th agent performs evasive maneuvers after its missile lost the target, and successfully avoids the attack of the final agent's missile, while the 8th generation agent does not avoid the attack of the final agent's missile, which indicates that the maneuver decision-making capability of 16th agent is better than that of the eighth agent.

Comparing the trajectory of the 24th agent with that of the 16th agent, it can be seen that the 24th agent is farther away from the missile of the final agent during flight, while the 16th agent is closer to the missile of the final agent during flight, which indicates that the evasive ability of the 24th agent is better than that of the 16th agent. As can be seen from Figure 5i,j, the maneuvers of the 32nd are similar to that of the final agent. After launching a missile, it gradually approaches its target to maintain the detection of the target. At the same time, neither side flies directly to the target because direct flying to the target would make them more likely to be hit by the opponent's missile. Finally, both sides lose their targets, and the simulation ends. According to the above analysis, the proposed FRE-PPO-based maneuver decision-making method is effective and efficient, which enables agents to learn evasive and attack policies. On the other hand, experiment results indicate that the final reward is more important than the intermediate reward in maneuver decision-making. A possible direction for future research is to improve the proposed method in order to increase the winning rate; for example, it can be combined with an evolution algorithm, which can enhance the exploration of reinforcement learning and generate more high-value samples.

6. Conclusions

Aiming at the problem of autonomous maneuver decision-making, an air combat agent training framework based on FRE-PPO is proposed. Based on the RL algorithm PPO, this method uses self-play to generate training data and evaluates the air combat ability of the trained agents during the training process. To address the problem of poor performance of PPO in maneuver decision-making, this paper proposes to use FRE to replace the advantage function in the PPO optimization objective and proposes to select training samples according to FRE.

Comparing the experimental results of FRE-U with GAE and DSR, it can be concluded that the final reward estimation proposed in this paper is more effective. Comparing the

experimental results of FRE-S with that of FRE-U, GAE and DSR, it can be concluded that the proposed method of sampling by FRE can significantly improve the training efficiency. Comparing the experimental results of FRE with those of FRE-S, FRE-U, GAE and DSR, it can be concluded that the proposed intermediate reward can avoid invalid training in the later stage caused by sparse rewards and combining FRE with the method of sampling according to FRE can improve the training efficiency and make the training performance better. According to the air combat simulation results of the final agent against the 1st, 8th, 16th, 24th and 32nd agents, respectively, the agent gradually acquires evasive and attack maneuvers during the training process and meets the real-time requirements.

To sum up, the maneuver decision-making method proposed in this paper has certain effectiveness and superiority and meets the real-time requirements. In future work, enhancing the winning rate is the main research direction since the average wins of the proposed method are not enough, and plenty of draws still exist (fewer draws mean better performance), which indicates that the proposed method should be improved further. Meanwhile, RL algorithms can be combined with evolution algorithms, such as genetic algorithms, to improve the exploration ability, which may lead to better performance and more wins.

Author Contributions: Conceptualization, C.H. and H.Z. (Huan Zhou); methodology, H.Z. (Huan Zhou); software, H.Z. (Hongpeng Zhang); validation, C.H., H.Z. (Huan Zhou) and Y.W.; formal analysis, H.Z. (Hongpeng Zhang); investigation, H.Z. (Hongpeng Zhang) and Y.W.; resources, C.H.; data curation, Y.W.; writing—original draft preparation, H.Z. (Hongpeng Zhang); writing—review and editing, Y.W.; visualization, H.Z. (Huan Zhou); supervision, C.H.; project administration, C.H.; funding acquisition, H.Z. (Huan Zhou). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant No. 62101590) and the Natural Science Foundation of Shaanxi Province (Grant No. 2020JQ-481).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. DARPA's AlphaDogfight Tests AI Pilot's Combat Chops, Breaking Defense. Available online: https://breakingdefense.com/2020/08/darpas-alphadogfight-tests-AI-pilot\T1\textquoterights-combat-chops/ (accessed on 12 August 2020).
2. Huang, C.; Dong, K.; Huang, H.; Tang, S.; Zhang, Z. Autonomous air combat maneuver decision using Bayesian inference and moving horizon optimization. *J. Syst. Eng. Electron.* **2018**, *29*, 86–97. [CrossRef]
3. Guo, H.; Hou, M.; Zhang, Q.; Tang, C. UCAV robust maneuver decision based on statistics principle. *Acta Armamentaria* **2017**, *38*, 160–167.
4. Du, H.W.; Cui, M.L.; Han, T.; Wei, Z.; Tang, C.; Tian, Y. Maneuvering decision in air combat based on multi-objective optimization and reinforcement learning. *J. Beijing Univ. Aeronaut. Astronaut.* **2018**, *44*, 2247–2256.
5. Mcgrew, J.S.; How, J.P.; Williams, B. Air-combat strategy using approximate dynamic programming. *J. Guid. Control Dyn.* **2010**, *33*, 1641–1654. [CrossRef]
6. Li, S.; Ding, Y.; Gao, Z. UAV air combat maneuvering decision based on intuitionistic fuzzy game theory. *J. Syst. Eng. Electron.* **2019**, *41*, 1063–1070.
7. Wei, R.X.; Zhou, K.; Ru, C.J.; Guan, X.; Che, J. Study on fuzzy cognitive decision-making method for multiple UAVs cooperative search. *Sci. Sin. Technol.* **2015**, *45*, 595–601.
8. Zhang, Q.; Yang, R.; Yu, L.; Zhang, T.; Zuo, J. BVR air combat maneuvering decision by using Q-network reinforcement learning. *J. Air For. Eng. Univ.* **2018**, *19*, 8–14.
9. Hu, D.; Yang, R.; Zuo, J.; Zhang, Z.; Wu, J.; Wang, Y. Application of deep reinforcement learning in maneuver planning of beyond-visual-range air combat. *IEEE Access* **2021**, *9*, 32282–32297. [CrossRef]
10. Mnih, V.; Kavukcuoglu, K.; Silver, D. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
11. Watkins, H.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292. [CrossRef]

12. Hado, H.; Arthur, G.; David, S. Deep reinforcement learning with double q-learning. In Proceedings of the National Conference of the American Association for Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 1813–1825.
13. Matteo, H.; Joseph, M.; Hado, H. Rainbow: Combining Improvements in Deep Reinforcement Learning. *arXiv* **2017**, arXiv:1710.02298v1.
14. Silver, D.; Huang, A.; Maddison, C. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489. [CrossRef]
15. Silver, D.; Schrittwieser, J.; Simonyan, K. Mastering the game of Go without human knowledge. *Nature* **2017**, *550*, 354–359. [CrossRef] [PubMed]
16. Schrittwieser, J.; Antonoglou, I.; Silver, D. Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model. *Nature* **2020**, *588*, 604–609. [CrossRef] [PubMed]
17. Oriol, V.; Igor, B.; Silver, D. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* **2019**, *575*, 350–354.
18. Jonas, D.; Federico, F.; Jonas, B. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature* **2022**, *602*, 414–419.
19. Ma, Y.; Wang, G.; Hu, X.; Luo, H.; Lei, X. Cooperative occupancy decision making of multi-UAV in beyond-visual-range air combat: A game theory approach. *IEEE Access* **2020**, *8*, 11624–11634. [CrossRef]
20. Kose, O.; Oktay, T. Simultaneous quadrotor autopilot system and collective morphing system design. *Aircr. Eng. Aerosp. Technol.* **2020**, *92*, 1093–1100. [CrossRef]
21. Ma, X.; Li, X.; Zhao, Q. Air combat strategy using deep Q-learning. In Proceedings of the Chinese Automation Congress, Xi'an, China, 30 November–2 December 2018; pp. 3952–3957.
22. Eloy, G.; David, W.C.; Dzung, T.; Meir, P. A differential game approach for beyond visual range tactics. *arXiv* **2020**, arXiv:2009.10640v1.
23. Wu, S.; Nan, Y. The calculation of dynamical allowable lunch envelope of air-to-air missile after being launched. *J. Proj. Rocket. Missile Guid.* **2013**, *33*, 49–54.
24. Li, X.; Zhou, D.; Feng, Q. Air-to-air missile launch envelops fitting based on genetic programming. *J. Proj. Rocket. Missile Guid.* **2015**, *35*, 16–18.
25. Wang, J.; Ding, D.; Xu, M.; Han, B.; Lei, L. Air-to-air missile launchable area based on target escape maneuver estimation. *J. Beijing Univ. Aeronaut. Astronaut.* **2019**, *45*, 722–734.
26. He, X.; Jing, X.; Feng, C. Air combat maneuver decision based on MCTS method. *J. Air For. Eng. Univ.* **2017**, *18*, 36–41.
27. Bellemare, M.G.; Naddaf, Y.; Veness, J.; Bowling, M. The arcade learning environment: An evaluation platform for general agents. *J. Artif. Intell. Res.* **2013**, *47*, 253–279. [CrossRef]
28. Mnih, V.; Badia, A.P.; Mirza, M. Asynchronous methods for deep reinforcement learning. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1928–1937.
29. Sutton, R.S.; Barto, A.G. Reinforcement learning: An introduction. *AI MAG* **2000**, *21*, 103. [CrossRef]
30. Schaul, T.; Quan, J.; Antonoglou, I. Prioritized experience replay. In Proceedings of the International Conference on Learning Representations, San Juan, WA, USA, 7–9 May 2015; pp. 1559–1566.
31. Wang, Z.; Schaul, T.; Hessel, M. Dueling network architectures for deep reinforcement learning. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1995–2003.
32. Van, H.; Guez, A.; Silver, D. Deep reinforcement learning with double Q-learning. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 1421–1432.
33. Fortunato, M.; Azar, M.G.; Piot, B. Noisy networks for exploration. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 23–26 April 2017; pp. 1177–1182.
34. Bellemare, M.G.; Dabney, W.; Munos, R. A distributional perspective on reinforcement learning. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 449–458.
35. Steven, K.; Georg, O.; John, Q. Recurrent experience replay indistributed reinforcement learning. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019; pp. 392–407.
36. Puigdomènech, B.A.; Sprechmann, P.; Vitvitskyi, A. Never give up: Learning directed exploration strategies. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020; PMLR: Addis Ababa, Ethiopia, 2020; pp. 1–28.
37. Adrià, P.B.; Bilal, P.; Steven, K. Agent57: Outperforming the Atari human benchmark. In Proceedings of the International Conference on Machine Learning, Virtual Event, 26–30 April 2020; pp. 507–517.
38. Sutton, R.S.; Mcallester, D.A.; Singh, S.P. Policy gradient methods for reinforcement learning with function approximation. In Proceedings of the Advances in Neural Information Processing Systems, Breckenridge, CO, USA, 1–2 December 2000; pp. 1057–1063.
39. Sham, K. A natural policy gradient. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 3–8 December 2001; pp. 1531–1538.
40. Silver, D.; Lever, G.; Heess, N.; Thomas, W.D.; Riedmiller, M. Deterministic policy gradient algorithms. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 387–395.
41. Timothy, P.L.; Jonathan, J.H.; Alexander, P. Continuous control with deep reinforcement learning. In Proceedings of the International Conference on Learning Representations, Puerto Rico, FL, USA, 2–4 May 2016; pp. 1692–1707.

42. Scott, F.; Herke, V.H.; David, M. Addressing function approximation error in actor-critic methods. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 1582–1591.
43. Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; Moritz, P. Trust region policy optimization. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 1889–1897.
44. Schulman, J.; Wolski, F.; Dhariwal, P. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
45. Williams, P. Three-dimensional aircraft terrain-following via real-time optimal control. *J. Guid. Control Dyn.* **1990**, *13*, 1146–1149. [CrossRef]
46. Fang, X.; Liu, J.; Zhou, D. Background interpolation for on-line situation of capture zone of air-to-air missiles. *J. Syst. Eng. Electron.* **2019**, *41*, 1286–1293.
47. David, S.; Satinder, S.; Doina, P.; Richard, S.S. Reward is enough. *Artif. Intell.* **2021**, *299*, 1–13.
48. John, S.; Philipp, M.; Sergey, L.; Michael, I.J.; Pieter, A. High-dimensional continuous control using generalized advantage estimation. In Proceedings of the International Conference on Learning Representations, Puerto Rico, FL, USA, 2–4 May 2016; pp. 1181–1192.