drones

MDPI

Article

PPO-Exp: Keeping Fixed-Wing UAV Formation with Deep Reinforcement Learning

Dan Xu [1], Yunxiao Guo [2], Zhongyi Yu [3], Zhenfeng Wang [2], Rongze Lan [2], Runhao Zhao [1], Xinjia Xie [4,*] and Han Long [2,*]

[1] College of System Engineering, National University of Defense Technology, Changsha 410073, China
[2] College of Sciences, National University of Defense Technology, Changsha 410073, China
[3] College of Advanced Interdisciplinary Studies, National University of Defense Technology, Changsha 410073, China
[4] College of Computer Science, National University of Defense Technology, Changsha 410073, China
* Correspondence: xiexinjia97@nudt.edu.cn (X.X.); longhan@nudt.edu.cn (H.L.)

Abstract: Flocking for fixed-Wing Unmanned Aerial Vehicles (UAVs) is an extremely complex challenge due to fixed-wing UAV's control problem and the system's coordinate difficulty. Recently, flocking approaches based on reinforcement learning have attracted attention. However, current methods also require that each UAV makes the decision decentralized, which increases the cost and computation of the whole UAV system. This paper researches a low-cost UAV formation system consisting of one leader (equipped with the intelligence chip) with five followers (without the intelligence chip), and proposes a centralized collision-free formation-keeping method. The communication in the whole process is considered and the protocol is designed by minimizing the communication cost. In addition, an analysis of the Proximal Policy Optimization (PPO) algorithm is provided; the paper derives the estimation error bound, and reveals the relationship between the bound and exploration. To encourage the agent to balance their exploration and estimation error bound, a version of PPO named PPO-Exploration (PPO-Exp) is proposed. It can adjust the clip constraint parameter and make the exploration mechanism more flexible. The results of the experiments show that PPO-Exp performs better than the current algorithms in these tasks.

Keywords: fixed-wing UAV; formation keeping; reinforcement learning

check for updates

Citation: Xu, D.; Guo, Y.; Yu, Z.; Wang, Z.; Lan, R.; Zhao, R.; Xie, X.; Long, H. PPO-Exp: Keeping Fixed-Wing UAV Formation with Deep Reinforcement Learning. *Drones* **2023**, *7*, 28. https://doi.org/10.3390/drones7010028

Academic Editor: Oleg Yakimenko

Received: 27 November 2022
Revised: 24 December 2022
Accepted: 28 December 2022
Published: 31 December 2022

1. Introduction

In recent years, unmanned aerial vehicles (UAVs) have been widely used in military and civil fields, such as in tracking [1], surveillance [2], delivery [3], and communication [4]. Due to the inherent defects, such as fewer platform functions and a light payload, it is difficult for a single UAV to perform diversified tasks in complex environments [5]. The cooperative formation composed of multiple UAVs can effectively compensate for the lack of performance and has many advantages in performing combat tasks. Thus, the formation control of UAVs has become a hot topic and attracted much attention [6,7].

Traditional solutions are usually based on accurate models of the platform and disturbance, such as model predictive control [8] and consistency theory [9]. This paper [10] proposed a group-based hierarchical flocking control approach, which did not need the global information of the UAV swarms. The study in [11] researched the mission-oriented miniature fixed-wing UAV flocking problem and proposed an architecture that decomposes the complex problem; it was the first work that successfully integrated the formation flight, target recognition, and tracking missions into simply an architecture. However, due to the influence of environmental disruption, these methods are difficult to accurately model [12]. This seriously limits the application scope of traditional analysis methods. Therefore, with the emergence of machine learning (ML), the reinforcement learning (RL) [13,14] method to solve the above problem has received increasing attention [15]. RL applies to decision-making control problems in unknown environments and has achieved successful applications in the robotics field [16–18].

At present, some works have integrated RL into the formation coordination control problem solution and preliminarily verified the feasibility and effectiveness in the simulation environment. Most existing schemes use the particle agent model for the rotary-wing UAV. The researchers [19] first researched RL in coordinated control, and applied the Q-learning algorithm and potential field force method to learn the aggregation strategy. After that, ref. [20] proposed a multi-agent self-organizing system based on a Q-learning algorithm. Ref. [21] investigates second-order multi-agent flocking systems and proposed a single critic reinforcement learning approach. The study in [22] proposes a UAV formation coordination control method based on the Deep Deterministic Policy Gradient algorithm, which enabled UAVs to perform navigation tasks in a completely decentralized manner in a large-scale complex environment.

Different from rotary-wing UAVs, the formation coordination control of fixed-wing UAVs is more complex and more vulnerable to environmental disturbance; therefore, different control strategies are required [23]. The Dyna-Q(λ) and Q-flocking algorithm are proposed [24,25] for solving the discrete state & action space fixed-wing UAV flocking problem under complex noise environments with deep reinforcement learning. To deal with the continuous space, ref. [26,27] proposed a fixed-wing UAV flocking method in continuous spaces based on deep RL with the actor–critic model. The learned policy can be directly transferred to the semi-physical simulation. Ref. [28] focused on the nonlinear attitude control problem and devised a proof-of-concept controller using proximal policy optimization.

However, the above methods also assume that UAVs fly with different attitudes, so the interaction (collision) between the followers can be ignored, and the followers in the above methods are seen as independent. Under the independent condition, these single-agent reinforcement learning algorithms can be effective due to the stationary environment [29]. However, in real tasks, even when the attitude is different, the collision still may happen when the attitude difference is not significant, and the UAVs adjust their roll angles.

In real tasks, the followers can interact with each other, and it is also common for them to collide in some scenarios, such as the identical attitude flocking task. However, this scenario is rarely studied. Ref. [30] proposed a collision-free multi-agent flocking approach MA2D3QN by using the local situation map to generate the collision risk map. The experimental results demonstrate that it can reduce the collision rate. The followers' reward function in MA2D3QN is only related to the leader and itself; however, other followers can also provide some information. This indicates that the method did not fully consider the interaction between the followers.

However, MA2D3QN did not demonstrate the ability to manage the non-stationary multi-agent environment [29], and the experiments also show collision judgments with high computation. With the number of UAVs rising, the computation time also increases. Furthermore, some problems in the above methods on fixed-wing UAVs have not been adequately solved, such as the generalization aspect and communication protocol; the most concerning problem is the minimum cost of the formation.

To consider the communication protocol of the formation, this paper takes the maximum communication distance between the UAVs into consideration, with a minimum cost communication protocol to guide the UAVs to send the message in the formation-keeping process. Under this protocol, the centralized training method for the UAVs is designed; only the leader needs to equip the intelligence chip. The main contributions of this work are as follows:

1. Research the formation keeping task with continuous space through reinforcement learning, and building the RL formation-keeping environment with OpenAI gym, and constructing the reward function for the task.
2. Design the communication protocol for the UAVs' formation with one leader who can make decisions intelligently and five followers who receive the decisions from

the leader. The protocol is feasible even when the UAVs are far away from each other. Under this protocol, the followers and leader can communicate at a low cost.

3. Analyze the PPO-Clip algorithm, give the estimation error bound of its surrogate, and elaborate on the relationship between the bound and hyperparameter ε: the higher ε, the more exploration, the larger the bound.
4. Propose a variation of PPO-clip: PPO-Exp. The PPO-Exp separates the exploration reward and regular reward in the task of formation keeping, and estimates the advantage function from them, respectively. The adaptive mechanism is used to adjust ε to balance the estimation error bound and exploration. The experiments demonstrate this mechanism with effectiveness for improving performance.

This paper is organized as follows. The first section introduced the current research on UAV flocking. Section 3 describes the background of the formation-keeping task and introduces reinforcement learning briefly. In Section 4, the formation-keeping environment is constructed, and the reward of the formation process is designed. Section 5 discusses the dilemma between the estimation error bound and exploration ability of PPO-Clip, and proposes PPO-Exp to balance the dilemma. Section 6 shows the experimental setup and results. Section 7 provides the conclusions of the paper.

2. Related Work

This section reviews current research about fixed-wing UAV flocking and formation-keeping approaches with deep reinforcement learning. According to the training architecture, this paper divides the current methods into the following two categories: centralized and decentralized. The difference between the two categories is as follows:

The centralized methods utilize the leader and all the followers' states in the training model, and the obtained optimal policy can control all of the followers so that they flock to the leader. The decentralized methods only use one follower and the leader's state to train the policy, and the obtained optimal policy could only control one follower. If there are several followers in the task, the policy and intelligence chip should be deployed on all of the followers.

2.1. Decentralized Approach

The paper [24] proposed a reinforcement learning flocking approach Dyna-Q(λ) to flock the fixed-wing UAV under the stochastic environment. To learn a model in the complex environment, the authors used Q(λ) [31] and Dyna architecture to train each fixed-wing follower to follow the leader, and combined internal models to deal with the influence of the stochastic environment. In [25], the authors further proposed Q-Flocking, which is a model-free and variable learning parameter algorithm based on Q-learning. Compared to Dyna-Q(λ), Q-Flocking removed the internal models and proved it could also converge to the solutions. For simplification, Q-Flocking and Dyna-Q(λ) also require that the state and action spaces are discrete, which is inappropriate. In [26], the authors first developed a DRL-based approach for the continuous state and action spaces fixed-wing UAV flocking. The proposed method is based on the Continuous Actor-Critic Learning Automation(CACLA) algorithm [32], with the experience replay technique embedded to improve the training efficiency. Ref. [33] considered a more complex flocking scenario, where the enemy threat is considered in the dynamic environment. To learn the optimal control policies, the authors use the situation assessment module to transfer the state of UAVs to the situation map stack. Then, the stack is input into the proposed Dueling Double Deep Q-network(D3QN) algorithm to update the policies until convergence. Ref. [34] proposed the Multi-Agent PPO algorithm to decentralize learning in the two–group fixed-wing UAV swarms dog fight control. To accelerate the learning speed, the classical rewarding scheme is added to the resource baseline, which could reduce the state and action spaces.

The advantage of decentralized methods is that these methods could be deployed on the distribution UAV systems, which could extend to the large-scale UAV formation. The disadvantage of the centralized methods is as follows:

- These methods also require all of the followers to be equipped with intelligence chips, which increase the costs.
- These methods do not consider the collision and communication problem, due to the use of only local information.

The decentralized approaches also assume that UAVs fly at different heights, and then the collision problem could be ignored. However, in real-world applications, the collision problem must be considered [30].

2.2. Centralized Approach

Ref. [35] studied the collision avoidance fixed-wing UAV flocking problem. To manage collision among the UAVs, the authors proposed the PS-CACER algorithm, which receives the global information of UAV swarms through the plug-n-play embedding module. Ref. [30] proposed a collision-free approach by transferring the global state information to the local situation map and constructing the collision risk function for training. To improve the training efficiency, the reference-point-based action selection technique is proposed to assist the UAVs' decisions.

The advantages of the centralized methods are as follows:

- These methods could reduce the cost of the formation. Under the centralized architecture, the formation system only requires the leader to equip the intelligence chip. The followers only need to send their state information to the leader and receive the feedback commands.
- These methods could consider collision avoidance and communication in the formation due to their use of global information.

The disadvantage of the centralized method is the dependence on the leader. Ref. [36] pointed out that the defect or jamming of the leader causes failure in the whole formation system.

When the number of UAVs increases or the tasks are complex, the centralized methods face the dimension curse and lack of learning ability problems. A popular approach is learning the complex tasks with a hierarchical method [37,38], which divides the complex tasks into several sub-tasks and uses the centralized method to optimize the hierarchies. The hierarchical reinforcement learning approaches are applied in the quadrotors swarm system [37,38], but are rarely used in fixed-wing UAV systems.

Even when using global information in training, the current centralized approaches fail to consider communication in the formation. Compared to current centralized approaches, the approach proposed in this paper considers the communication in formation, and provides the communication protocol. Through the communication protocol, the formation system could be considered as one leader with an intelligence chip and five followers without intelligence chips; the leader collects the followers' information, with a centralized train on the intelligence chip. The followers receive the command from the leader through this protocol and execute.

3. Background

This section will introduce the kinematic model of the fixed-wing UAV, restate the formation keeping problem, and briefly introduce reinforcement learning.

3.1. Problem Description

The formation task can be described as follows: At the beginning, the formation is orderly (shown in Figure 1), which is a common formation designed in [39]). The goal of the task is to reach the target area (the green circle area) with the formation in as orderly a way as possible; when the leader enters the target area, the mission is complete.

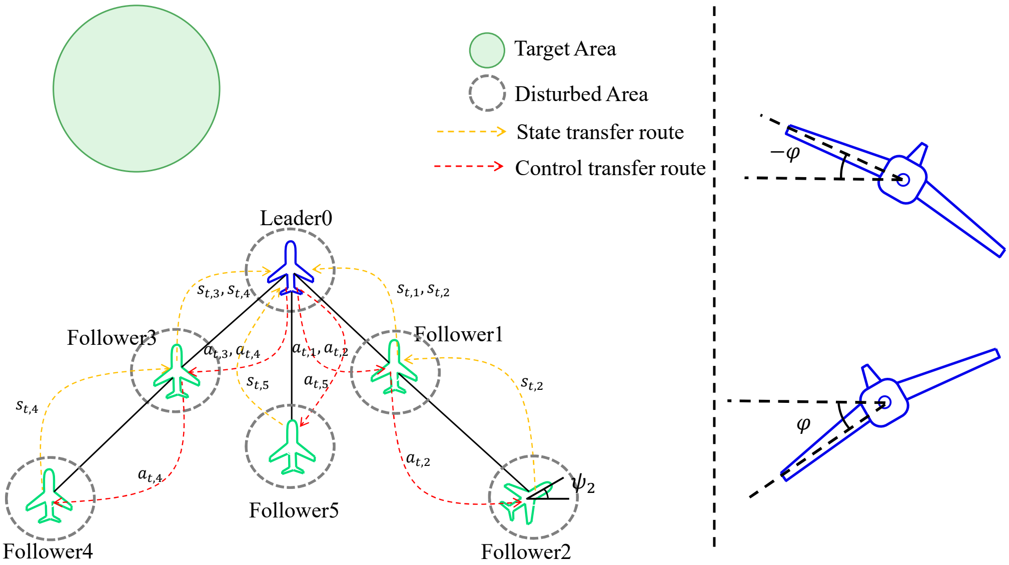


Figure 1. Left: The Leader–Follower formation topology structure and the task schematic diagram. **Right**: The action of UAV.

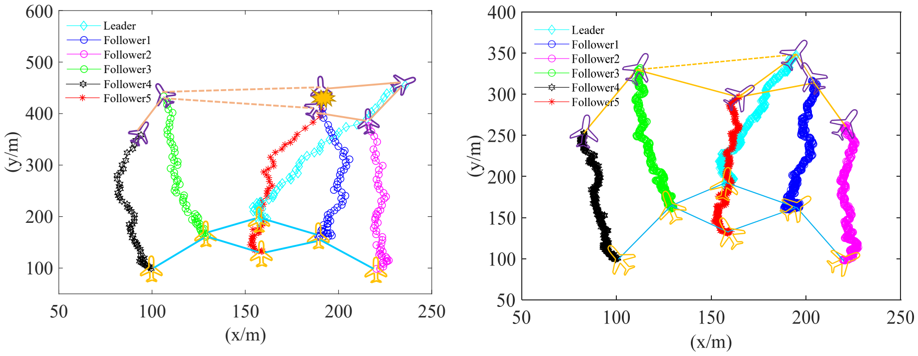
During the task, assume the UAVs are flying at a fixed attitude; then, each UAV in the formation can also be described as a six-degree of freedom (6DoF) dynamic model. However, analyzing the six-degree model directly is very complex; it will increase the space scale and make control more difficult. The 6DoF model can be simplified to the 4DoF model; to compensate for the loss incurred during this simplification, random noise is introduced into the model [27], and the dynamic equations of ith UAV in the formation can be written as follows:

$$\dot{\xi}_i = \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_i \\ y_i \\ \psi_i \\ \varphi_i \end{bmatrix} = \begin{bmatrix} v_i \cos\psi_i + \eta_{x_i} \\ v_i \sin\psi_i + \eta_{y_i} \\ -(\alpha_g/v_i)\tan\varphi_i + \eta_{\psi_i} \\ f(\varphi_i, \varphi_{i,d}) \end{bmatrix} \tag{1}$$

where $(x_i, y_i) \in \mathbb{R}^2$ is the planar position, and $\psi_i \in \mathbb{R}^1, \varphi_i \in \mathbb{R}^1$ represent the heading and roll angle, respectively, (see Figure 1). The v_i is the velocity, and α_g is the gravity acceleration. The random noise values $\eta_{x_i}, \eta_{y_i}, \eta_{\varphi_i}, \eta_{\psi_i}$ are the normal distributions, its means are $\mu_{x_i}, \mu_{y_i}, \mu_{\varphi_i}, \mu_{\psi_i}$, and its variances are $\sigma^2_{x_i}, \sigma^2_{y_i}, \sigma^2_{\varphi_i}, \sigma^2_{\psi_i}$, respectively, (the gray dotted circles in Figure 1 show the area of influence, of random factors); they represent the random factors introduced by simplification and environment noise.

A simple control strategy can make the formation satisfactory when the environment's noise is low. However, under a strong inference environment, such as one with strong turbulence, the random factors will be apparent, leading the formation to maintain the complexity of the task. If no effective control is provided, the formation will break up quickly, (this is demonstrated in Figure 2), and a crash may happen.

Furthermore, even though there is an effective control policy for the formation, the coupling between the control and communication protocol can also be an unsolved challenge. Because the communication range of UAVs is limited, if the UAV wants to know others' states, it has to wait for other UAVs out of range to send state information to UAVs it can communicate with, which in turn send state information to it. If no harmonic protocol is applied in the formation control, the asynchronous and nonstationary elements will be introduced into the formation control, making the control strategy more complex.



(**a**) (**b**)

Figure 2. (**a**): The ablation experiment result of environment noise: track of formation with no control; (**b**): The ablation experiment result of exploration balance point: PPO-Clip with $\varepsilon = 0.05$.

3.2. Reinforcement Learning

In the last part, the solution of differential Equation (1) can be represented as the current dynamic parameters adding the integral items by difference equation methods such as the Runge–Kutta method. So, the UAV formation control can be modeled as a Markov Decision Process(MDP), which refers to the decision process that satisfies the Markov property.

The MDP also can be described as the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$. $\mathcal{S}$ represents the state space, $\mathcal{A}$ represents the action space, and $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathcal{R}$ is the transition probability. The reward function is $r : \mathcal{S} \times \mathcal{A} \to \mathcal{R}$, and $\gamma \in (0, 1)$ is the discount factor, which leads the agent to pay more attention to the current reward.

Reinforcement learning can solve the MDP well to maximize the discounted return, as follows: $R_t = \sum_{t=0}^{\infty} \gamma^t r(s_t)$. The main approaches of RL are divided into the following three categories: value-based, model-based, and policy-based. The policy-based methods have been developed and widely used in various tasks in recent years. These methods directly optimize the value function by the policy gradient:

$$\nabla \mathcal{J}_\pi(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \sum_{t=0}^{T} \log \pi_\theta(s_t, a_t) A_\pi\right] \tag{2}$$

where A_π is the advantage function that is equal to the state-action value function, and the the state value function is subtracted, as follows:

$$A_\pi(S_t, a_t) = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | S_t = s, a_t = a\right] - E_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | S_t = s\right] \tag{3}$$

PPO (Proximal Policy Optimization) is one of the most famous policy gradient methods in continuous state and action space [40]. In policy gradient descent, PPO updates the following equation at each update epoch :

$$\mathcal{L}^{\text{Clip},\theta} = \mathbb{E}_{\pi_{\theta_{old}}}\left[\min\left(r_t(\theta) A_{\pi_{\theta_{old}}}, clip(r_t(\theta), 1 - \varepsilon, 1 + \varepsilon) A_{\pi_{\theta_{old}}}\right)\right] \tag{4}$$

However, using the constant clip coefficient ε, the PPO also proved its lack of exploration ability and difficulty in convergence. Therefore, designing an efficient dynamic mechanism to adjust ε and ensure greater exploration and faster convergence is also challenging.

4. Formation Environment

This section constructs the fixed-wing UAV formation-keeping environment, the formation topology, communication and control protocols, and collision. Communication loss is also considered in the environment through the reward design.

4.1. State and Action Spaces

In the course of the formation task, based on the 4DoF Equation (1), it is modified to a more realistic control environment. For the ith UAV, assume the thrust of the UAV is controllable, and it will generate a linear acceleration $\alpha_{v_i} = \dot{v}_i$. Moreover, assume the torque of the roll angle is controllable too, and add the roll angle acceleration $\alpha_{\varphi_i} = \dot{w}_i = \ddot{\varphi}_i$ into the dynamic equations. Finally, the dynamic equations of ith UAV can be modified as follows:

$$\dot{\boldsymbol{\xi}}_i = \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_i \\ y_i \\ \psi_i \\ \varphi_i \\ v_i \\ w_i \end{bmatrix} = \begin{bmatrix} v_i \cos\psi_i + \alpha_{v_i}\cos(\psi_i)t + \eta_{x_i} \\ v_i \sin\psi_i + \alpha_{v_i}\sin(\psi_i)t + \eta_{y_i} \\ -\left(\alpha_{\mathrm{g}}/v_i\right)\tan\varphi_i + \eta_{\psi_i} \\ \omega_i + \eta_{\omega_i} \\ \alpha_{v_i} \\ \alpha_{\varphi_i} \end{bmatrix} \tag{5}$$

To control the UAVs, linear acceleration and roll angle acceleration are input. For control, we have the dynamic model of ith UAV:

$$\ddot{\boldsymbol{\xi}}_i = \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\psi}_i \\ \dot{\varphi}_i \end{bmatrix} = \begin{bmatrix} \alpha_{v_i}\cos\psi_i \\ \alpha_{v_i}\sin\psi_i \\ \frac{-\alpha_g f(\varphi_i,\varphi_{i,d})}{v_i\cos^2\varphi_i} + \frac{\alpha_{v_i}\alpha_g\tan\varphi_i}{v_i^2} \\ \alpha_{\varphi_i} \end{bmatrix} \tag{6}$$

The state and action spaces for existing methods in UAVs controlled by reinforcement learning are often discrete, but in the real world, the state space is continuous and changes continuously as time goes on. Therefore, combining the analysis of the previous dynamics, we define the state tuple of the ith UAV as $\xi_i := (x_i, y_i, \psi_i, \varphi_i, v_i, w_i)$. The planar position $(x_i, y_i) \in \mathbb{R}^2$, heading $\psi_i \in S^1$, roll angle $\varphi_i \in S^1$, line and angle velocity $v, w \in \mathbb{R}$ are determined by solving the differential Equation (5).

In the action space, although the engine can produce fixed thrust, the real thrust acting on the UAVs in the nonuniform atmospheric environment is not of the same value as the engine product. So, we define the action space by $a_i := (\alpha_{v_i}, \alpha_{\varphi_i})$. Assume the UAVs can also produce the same acceleration in positive and negative directions, where we have $\alpha_{v_i} \in [-\alpha_{v_i max}, \alpha_{v_i max}]$, and $\alpha_{\varphi_i} \in [-\alpha_{\varphi_i max}, \alpha_{\varphi_i max}]$. The action will influence $\ddot{\xi}_i$ through Equation (6), and then influence the $\dot{\xi}_i$ indirectly.

After defining the individual state and action of the UAV, we define the formation system state and action by sticking to the individual state (action) as a vector. Define the state of system $\boldsymbol{\xi} := [\xi_1, \cdots, \xi_6]$, and the action of system $\boldsymbol{a} := [a_1, \cdots, a_6]$.

4.2. Communication and Control Protocol

To ensure the UAV formation consumes less energy in the information send and receive process, and ensure the reinforcement learning method can be helpful in the task, the communication and control protocol for the UAV formation will be provided in this part.

As is shown in Figure 1, the formation is of a Leader–Follower structure; in terms of hardware, all the UAVs are equipped with gyroscopes and accelerometers to monitor their action and state parameters. Only the leader has the "brain" chip that can make decisions intelligently; the followers only have the chips that can receive the control command signals, take the command action and send the state signals.

To describe this relationship, the graph model is introduced. Use the communication graph $\mathcal{G}_t$ to describe the communication ability of the formation at time t [39]:

$$\mathcal{G}_t = (6, \mathcal{V}_t, \mathcal{E}_t) \tag{7}$$

where $\mathcal{V}_t = \{v_1, \cdots, v_6\}$ is the set of nodes that represent UAVs, the $\mathcal{E}_t$ represent the arc set at time t, e.g., $e_{i,j} \in \mathcal{E}_t$ denote an arc from node i to node j, which means the UAV i can communicate with UAV j directly at time t. The adjacent matrix $\mathcal{A}_t = \{a_{i,j}\}$ of graph $\mathcal{G}_t$ is used to describe the communicated situation of formation in real-time, e.g., at the initial time, the adjacent matrix is as follows:

$$\mathcal{A}_0 = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{8}$$

The adjacent matrix is symmetric, and its element a_{ij} indicates the communication situation of UAV i and UAV j. If $a_{ij} = 1$, then $a_{ji} = 1$, and the ith and jth UAVs can share their state, and the control command can be sent from i to j or j to i. The adjacent matrix is updated in real-time. If the distance between two UAVs is greater than the communication limited distance d_{com}, the corresponding elements of the adjacent matrix will be 0.

Additionally, at the initial time, the formation is connected, and the connected component $\mathcal{W}$ is 1. l If the UAVs want to keep in communication with all the others, the graph $\mathcal{G}$ should only have one connected component. In the graph model, this condition could be transferred to $\mathcal{G}$. The methods that judge whether an undirected graph is connected include union-find disjoint sets, DFS, and BFS [41]. So, after DFS or BFS, the task fails when the connected component number $\mathcal{W}$ of graph $\mathcal{G}$ is more than 1. When the formation works, $\mathcal{W}$ should be 1.

When the formation works, the protocol should be active to support the UAVs communicating with each other. The communication protocol's primary purpose is to send all the UAVs' states to the leader for the decision; the control protocol sends the action command to all the UAVs. When the formation is as orderly as it was at first, the information only needs to obey the transfer route (shown in Figure 1), so the whole formation can be controlled well. However, when the noise disturbs the position of UAVs, it makes the connection between the UAVs that are not connected at the initial time. It breaks the connection between the UAVs that are connected at the initial time. To handle the chaos brought about by the noise, a communication and control protocol is shown in Figure 3.

In Figure 3a the communication protocol is shown, where the block in ith row represents the communication priority of the corresponding UAV. For the priority, the bigger the number, the higher the priority. Priority 1, 2 determines the order of communication. If the priority is 0, both parties have no communication probability. i.e., when the leader0 and follower3 are within the communication range of follower5, the follower5 will send the information to leader0 instead of follower3.

The protocol is designed based on the communication object: to send all the followers' state information to the leader to support the decision. So, the principle of the protocol is to give the followers closer to the leader higher priority, such as followers 1, 3 and 5.

Figure 3b has a similar meaning to the control protocol. The target of the control protocol sends the control information to all the UAVs. The control protocol motivates the leader to send the control information to the followers that connects as much as the followers. Therefore, leader0, and follower1, 2, and 5 have priority 2 because they can connect with up to 2 other followers.

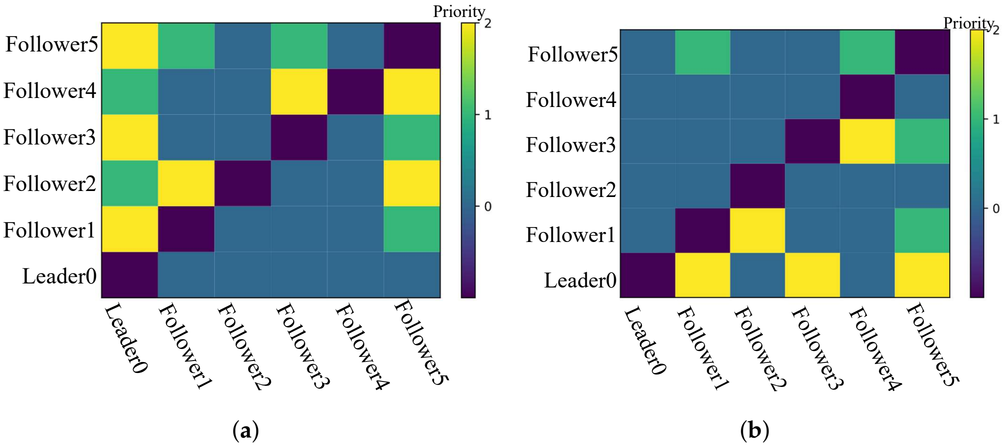


Figure 3. (**a**): The communication protocol of the UAVs formation; (**b**): The control protocol of the UAVs formation.

4.3. Reward Scheme

The goal of the formation-keeping task is to reach the target area and ensure the formation is as orderly as possible. At first, the orderliness of the formation is of primary concern. So, some geometric parameters are defined to describe the formation. The followers in the formation can be divided into two categories, one is on an oblique line with the leader, like followers 3 and 4, and another is on a straight line with the leader. Only follower 5 belongs to this category. The linear between the leader and the position where the follower should be located is called the baseline (see the back lines in Figure 4). Then, it is easy to know the first category followers have a baseline with a slope, and the second follower's baseline does not. For the follower i, the length of the initial baseline is l_i, and the initial slope is $k_i = \tan\theta$ (the first category).

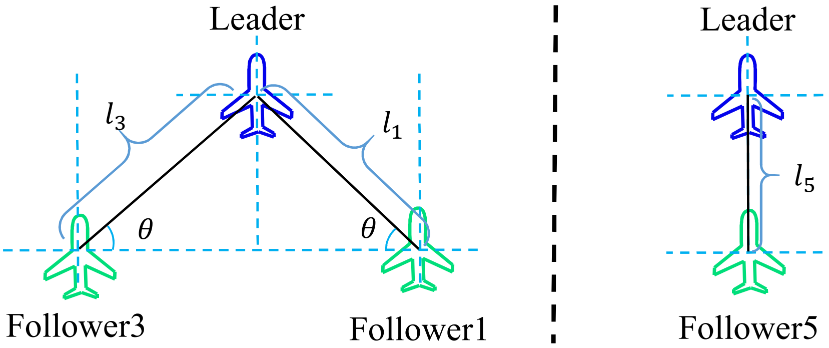


Figure 4. The communication and control protocol under the topology of the formation.

To make sure the UAV agent can return to the position that makes formation more orderly, for ith UAV, the formation reward is designed as follows:

$$R_{f,i} = -\max\{dis_{a,i}, |dis_{b,i} - l_i|\} \tag{9}$$

where $dis_{a,i}$ represents the distance between the follower i and the baseline along the vertical line of the baseline, and $dis_{b,i}$ represents the distance between the leader and follower i along the baseline. The formation reward is $R_{f,i}$.

When the UAVs belong to the first category follower (e.g., follower 3), the distance $dis_{a,3}$ can be calculated by the following formula:

$$dis_{a,3} = \frac{|x_3 \tan\theta - y_3 + (y_0 - x_0 \tan\theta)|}{\sqrt{1 + \tan\theta}} \tag{10}$$

Followers 2 and 4 have the same dis_a as in the above equation. The distance dis_b also can be obtained with the following formula:

$$dis_{b,3} = (x_3 - x_0)\cos\theta + (y_3 - y_0)\sin\theta + l_3 \tag{11}$$

For the second category follower (follower 5), it is easy to know that the reward can be represented as the following simple formation:

$$R_{f,5} = -\max\{|x_0 - x_5|, |y_0 - y_5 - l_5|\} \tag{12}$$

Furthermore, the main target of UAVs formation is to reach the target area, which is a circle with center coordinates (x_{tar}, y_{tar}) and radius r_{tar}. To encourage the formation to reach the target area, a sparse reward is designed as the destination reward:

$$R_d = \begin{cases} 0, \sqrt{(x_0 - x_{tar})^2 + (y_0 - y_{tar})^2} \le r_{tar} \\ 10{,}000, otherwise \end{cases} \tag{13}$$

We only calculate the distance of the leader. Only when the formation reaches the target area do the UAVs receive this sparse reward, and the learning process will halt. It leads to the UAVs not only needing to take minor actions to ensure that the orderly formation is not disorganized by the disturbance, but also needing to adjust direction to reach the target area. From the reward design view, UAV agents need to try different actions to discover and obtain a sparse signal. To accelerate the learning, the exploration rewards, as described in the literature [42], are designed as the incentive reward:

$$R_{e,i} = -max\{|x_i - x_{tar}|, |y_i - y_{tar}|\} \tag{14}$$

When the formation is closer to the target area, it will receive a higher exploration reward, leading the UAV agent to learn to reach the target area.

Meanwhile, some UAVs are too close and crash together, or they are too far and cease communicating with each other. In that case, the formation will suffer permanent destruction, and the task will halt.

Setting the minimum distance for crashes makes it easy to obtain the halt condition of UAV crashes. Then, the penalty should be added to avoid the above situation. This penalty is designed as a formal sparse reward as follows:

$$R_p = \begin{cases} -10{,}000, d_{i,j} \le d_{cra}, \forall i,j = 0,1,\cdots 5 \\ -10{,}000, \mathcal{W} > 1 \\ 0, \text{others} \end{cases} \tag{15}$$

where the $d_{\cdot,j}$ represents the minimum distance between the jth UAV and another five UAVs: $d_{\cdot,j} = \min_i\{d_{i,j}\}, \forall i = 1,\cdots,6, i \neq j$. The lowest communication distance is d_{com}, once the minimum distance $d_{\cdot,j}$ less than d_{com}, the jth UAV will lose the communication ability with other UAVs. In addition, d_{cra} is the crash distance; as long as the distance between two UAVs is less than this, the two UAVs might crash.

Finally, the reward of the formation system at time T can be represented as the sum of the following reward function:

$$R(T) = \sum_{i=1}^{6}\Big[R_{f,i}(T) + R_{e,i}(T)\Big] + R_d(T) + R_p(T) \tag{16}$$

5. PPO-Exp

PPO is one of the most popular deep reinforcement learning algorithms in continuous tasks that achieved outstanding performance. The PPO embedded the Actor–Critic algorithm, which uses a deep neural network as an Actor for policy generation, and another deep neural network as a Critic for policy estimating. The structure of PPO can be seen in

Figure 5; the Actor interacts with the environment, collects the trajectories: $\{s_t, a_t, r_t, s_{t+1}\}$ and stores them in the buffer, then it uses the buffer and the value function estimated by the Critic to optimize the Actor network's hyperparameter according to following surrogate:

$$\mathcal{L}_t^{\text{Clip},\theta} = \begin{cases} (1+\varepsilon)A_{\pi_{\theta_{t-1}}}; A_{\pi_{\theta_{t-1}}} > 0, r_t > 1+\varepsilon \\ (1-\varepsilon)A_{\pi_{\theta_{t-1}}}; A_{\pi_{\theta_{t-1}}} < 0, r_t < 1-\varepsilon \\ r_t \cdot A_{\pi_{\theta_{t-1}}}; otherwise \end{cases} \tag{17}$$

where the $A_{\pi_{\theta_{t-1}}}$ is the advantage function defined in Equation (3). The Critic network's hyperparameter ϕ is updated by minimizing the following MSE error:

$$\mathcal{L}_t^{\text{Clip},\phi} = \sum_t \left(y_t - Q_\phi(s_t, a_t)\right)^2 \tag{18}$$

$$y_t = r_t + \gamma \cdot Q_\phi(s_{t+1}, \pi_{\theta_{old}}(s_{t+1})) \tag{19}$$

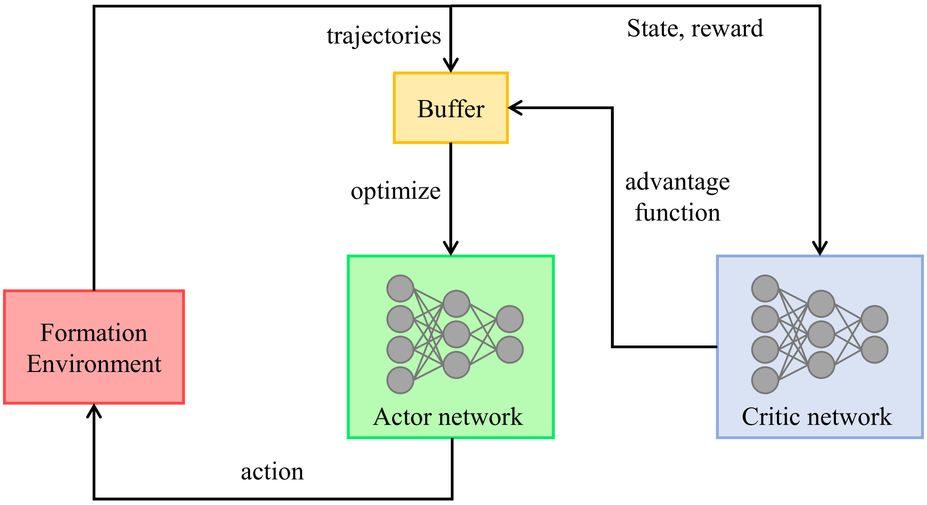


Figure 5. The structure of PPO with experience replay.

The gradient of Equations (17) and (18) is computed and used to update the hyperparameters θ and ϕ until they converge or reach maximum steps. In surrogate (17), the PPO restricted the difference between new and old policy by using the clip trick to restrain the ratio $r_t = \frac{\pi_\theta(s_t,a_t)}{\pi_{\theta_{old}}(s_t,a_t)}$. It could be considered a constraint on updated policy; under it, the ratio should satisfy the following constraint: $1-\varepsilon \le r_t \le 1+\varepsilon$. Then, the updated policy is restricted as follows:

$$\frac{|\pi_\theta(s_t, a_t) - \pi_{\theta_{old}}(s_t, a_t)|}{\pi_{\theta_{old}}(s_t, a_t)} \le \varepsilon \tag{20}$$

The coefficient ε is also a constant in the range $(0, 1)$ in PPO-Clip; from the inequality (20), it can be seen that the relative deviation is bound between $\pi_{\theta_{old}}$ and π_θ. When this deviation is under ε, as the increase in r_t is observed, the $\mathcal{L}_t^{\text{Clip},\theta}$ increase as well, but when the deviation exceeds ε, even if the r_t is increases, the $\mathcal{L}_t^{\text{Clip},\theta}$ maintains its value. It shows the exploration within the constraint ε; however, when the relative difference is beyond ε, the exploration is not encouraged by clipping the result to $(1+\varepsilon)A_{\pi_{\theta_{old}}}$. Figure 6 shows the surrogate of PPO-Clip in different ε. The large ε could encourage the agent to explore more and accept more policies. However, enlarging ε will lead to the estimated error of the surrogate. The PPO-Clip is the off-policy algorithm. The data generated by the old policy will be used as new policy updates. When $\|r_t - 1\| \le \varepsilon$, the estimated error bound of $\mathcal{L}^{\text{Clip},\theta}$ will increase as ε increases. For convenience, denote the following assumption:

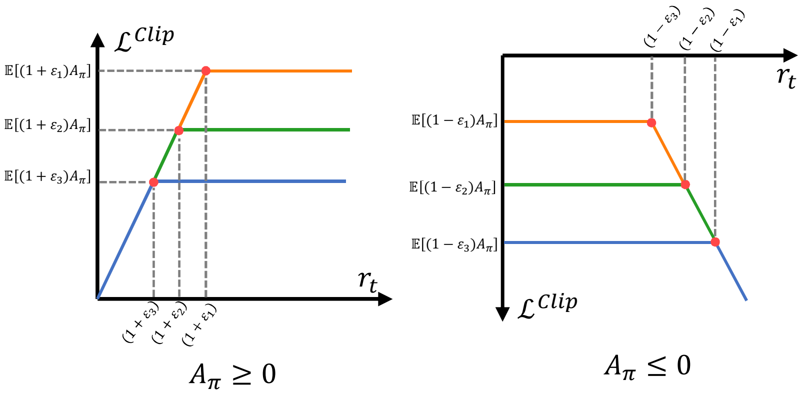


Figure 6. The surrogate of PPO-Clip in different ε. The relationship of different ε: $\varepsilon_3 > \varepsilon_2 > \varepsilon_1$.

Assumption 1. *In the previous t timestep of policy update, the ratio r_k satisfies $\|r_k - 1\| \le \varepsilon$, $\forall k = 1, \cdots, t$.*

Under Assumption 1, the following Lemma is given for auxiliary proof of the error bound:

Lemma 1. *Under Assumption 1, the difference of state distribution resulting from the policy satisfies the following inequality:*

$$\|\rho^{\pi_{\theta_t}} - \rho^{\pi_{\theta_{t-1}}}\| \le \frac{\varepsilon \cdot \gamma}{1-\gamma} \tag{21}$$

Proof. The distribution ρ^{π_θ} can be rewritten as [43]:

$$\rho^{\pi_\theta} = (1-\gamma)\sum_{k=0}^{\infty} \gamma^k \cdot d^k_{\pi_\theta} \tag{22}$$

where $d^k_{\pi_\theta}$ is the distribution resulting from π_θ at k timestep. Using the Markov property, $\forall s' \in \mathcal{S}$, the $d^k_{\pi_\theta}(s')$ could be decompose as follows:

$$d^k_{\pi_\theta}(s') = \sum_{s,a} d^{k-1}_{\pi_\theta}(s) \cdot \pi_\theta(a|s) \cdot P(s'|s,a) \tag{23}$$

Using the decomposition, the following equation holds:

$$\begin{aligned}
&d^k_{\pi_{\theta_t}}(s') - d^k_{\pi_{\theta_{t-1}}}(s') = \sum_{s,a}\Big[d^{k-1}_{\pi_{\theta_t}}(s)\cdot\pi_{\theta_t}(a|s) - d^{k-1}_{\pi_{\theta_{t-1}}}(s)\pi_{\theta_{t-1}}(a|s)\Big]P(s'|s,a) \\
&= \sum_{s,a}\Big[d^{k-1}_{\pi_{\theta_t}}(s)\cdot\pi_{\theta_t}(a|s) - d^{k-1}_{\pi_{\theta_{t-1}}}(s)\pi_{\theta_{t-1}}(a|s) + d^{k-1}_{\pi_{\theta_{t-1}}}(s)\pi_{\theta_{t-1}}(a|s) - d^{k-1}_{\pi_{\theta_{t-1}}}(s)\pi_{\theta_{t-1}}(a|s)\Big] \\
&\cdot P(s'|s,a) \\
&= \sum_{s,a}\big[\pi_{\theta_t}(a|s) - \pi_{\theta_{t-1}}(a|s)\big]\cdot d^{k-1}_{\pi_{\theta_{t-1}}}(s)\cdot P(s'|s,a) \\
&+ \sum_{s,a}\Big[d^{k-1}_{\pi_{\theta_{t-1}}}(s) - d^{k-1}_{\pi_{\theta_{t-1}}}(s)\Big]\cdot\pi_{\theta_{t-1}}(a|s)\cdot P(s'|s,a)
\end{aligned} \tag{24}$$

Using the triangle inequality, the following equation hold:

$$\begin{aligned}
&\sum_{s,a}\|\pi_{\theta_t}(a|s)-\pi_{\theta_{t-1}}(a|s)\|\cdot d^{k-1}_{\pi_{\theta_t}}(s)P(s'|s,a)+\sum_{s,a}\|d^{k-1}_{\pi_{\theta_t}}(s)-d^{k-1}_{\pi_{\theta_{t-1}}}(s)\|\cdot\pi_{\theta_{t-1}}(a|s)\cdot P(s'|s,a)\\
&\geq\|\sum_{s,a}\big[\pi_{\theta_t}(a|s)-\pi_{\theta_{t-1}}(a|s)\big]\cdot d^{k-1}_{\pi_{\theta_{t-1}}}(s)\cdot P(s'|s,a)\\
&+\sum_{s,a}\Big[d^{k-1}_{\pi_{\theta_{t-1}}}(s)-d^{k-1}_{\pi_{\theta_{t-1}}}(s)\Big]\cdot\pi_{\theta_{t-1}}(a|s)\cdot P(s'|s,a)\|=\|d^{k}_{\pi_{\theta_t}}(s')-d^{k}_{\pi_{\theta_{t-1}}}(s')\|
\end{aligned}\tag{25}$$

Sum up the inequality (26) to calculate the expectation on s':

$$\begin{aligned}
&\|d^{k}_{\pi_{\theta_t}}-d^{k}_{\pi_{\theta_{t-1}}}\|\\
&=\sum_{s'}\|d^{k}_{\pi_{\theta_t}}(s')-d^{k}_{\pi_{\theta_{t-1}}}(s')\|\leq\sum_{s,a}\|\pi_{\theta_t}(a|s)-\pi_{\theta_{t-1}}(a|s)\|\cdot d^{k-1}_{\pi_{\theta_t}}(s)\sum_{s'}P(s'|s,a)\\
&+\sum_{s,a}\|d^{k-1}_{\pi_{\theta_t}}(s)-d^{k-1}_{\pi_{\theta_{t-1}}}(s)\|\cdot\pi_{\theta_{t-1}}(a|s)\cdot\sum_{s'}P(s'|s,a)\\
&=\|\pi_{\theta_t}(a|s)-\pi_{\theta_{t-1}}(a|s)\|+\|d^{k-1}_{\pi_{\theta_t}}-d^{k-1}_{\pi_{\theta_{t-1}}}\|\\
&\leq\|\frac{\pi_{\theta_t}(a|s)-\pi_{\theta_{t-1}}(a|s)}{\pi_{\theta_{t-1}}(a|s)}\|\cdot\|\pi_{\theta_{t-1}}(a|s)\|+\|d^{k-1}_{\pi_{\theta_t}}-d^{k-1}_{\pi_{\theta_{t-1}}}\|\\
&\leq\|\frac{\pi_{\theta_t}(a|s)-\pi_{\theta_{t-1}}(a|s)}{\pi_{\theta_{t-1}}(a|s)}\|+\|d^{k-1}_{\pi_{\theta_t}}-d^{k-1}_{\pi_{\theta_{t-1}}}\|\leq\varepsilon+\|d^{k-1}_{\pi_{\theta_t}}-d^{k-1}_{\pi_{\theta_{t-1}}}\|\\
&\leq 2\varepsilon+\|d^{k-2}_{\pi_{\theta_t}}-d^{k-2}_{\pi_{\theta_{t-1}}}\|\leq k\varepsilon
\end{aligned}\tag{26}$$

Using Equation (22), the following equation holds:

$$\begin{aligned}
&\|\rho^{\pi_{\theta_t}}-\rho^{\pi_{\theta_{t-1}}}\|\leq\frac{1}{\gamma-1}\sum_{k=0}^{\infty}\gamma^{k}\|d^{k}_{\pi_{\theta_t}}-d^{k}_{\pi_{\theta_{t-1}}}\|\\
&\leq\frac{1}{\gamma-1}\sum_{k=0}^{\infty}\gamma^{k}\cdot k\cdot\varepsilon=\frac{\varepsilon\cdot\gamma}{1-\gamma}
\end{aligned}\tag{27}$$

□

Using this Lemma, the estimation error of the PPO-Clip could be obtained:

Theorem 1. *Under the Assumption 1, the estimation error of PPO-Clip is satisfied:*

$$Err\Big[\mathcal{L}^{Clip,\theta}\Big]=Err\Big[\mathbb{E}_{\pi_{\theta_{old}}}\Big[\frac{\pi_\theta}{\pi_{\theta_{old}}}A_{\pi_{\theta_{old}}}\Big]\Big]\leq\frac{\varepsilon\cdot\gamma}{1-\gamma}\mathbb{E}_{s\sim Unif_{\mathcal{S}},a\sim\pi_\theta}[A_\pi(s,a)]\tag{28}$$

Proof. When $\|r_t-1\|\leq\varepsilon$, the surrogate of the PPO-Clip will be degraded [40]:

$$\mathcal{L}^{\text{Clip},\theta}=\mathbb{E}_{\pi_{\theta_{old}}}\left[\frac{\pi_\theta}{\pi_{\theta_{old}}}A_{\pi_{\theta_{old}}}\right],\|\frac{\pi_\theta-\pi_{\theta_{old}}}{\pi_{\theta_{old}}}\|\leq\varepsilon\tag{29}$$

The above surrogate is the importance sampling estimator of the objective of the new policy [44]:

$$\mathbb{E}_{\pi_{\theta_{old}}}\left[\frac{\pi_\theta}{\pi_{\theta_{old}}}A_\pi(s,\pi_{\theta_{old}}(s))\right]\approx\mathbb{E}_{\pi_\theta}[A_\pi(s,\pi_\theta(s))]\tag{30}$$

However, the estimator uses the data generated by $\pi_{\theta_{old}}$, and the state distribution of $\mathcal{L}^{\text{Clip},\theta}$ is derived from $\rho^{\pi_{\theta_{old}}}$. Therefore, the estimation error is satisfied:

$$
\begin{aligned}
&\mathrm{Err}\left[\mathbb{E}_{\pi_{\theta_{old}}}\left[\frac{\pi_\theta}{\pi_{\theta_{old}}}A_\pi(s,\pi_{\theta_{old}}(s))\right]\right] = \left\|\mathbb{E}_{\pi_{\theta_{old}}}\left[\frac{\pi_\theta}{\pi_{\theta_{old}}}A_\pi(s,\pi_{\theta_{old}}(s))\right] - \mathbb{E}_{\pi_\theta}[A_\pi(s,\pi_\theta(s))]\right\| \\
&= \left\|\int_s \rho^{\pi_{\theta_{old}}}(s)\int_{a\sim\pi_{\theta_{old}}}\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}A_\pi(s,a)dads - \int_s \rho^{\pi_\theta}(s)\int_{a\sim\pi_\theta}A_\pi(s,a)dads\right\| \\
&\leq \int_s \left\|\rho^{\pi_{\theta_{old}}}(s) - \rho^{\pi_\theta}(s)\right\|\int_{a\sim\pi_\theta}\|A_\pi(s,a)\|dads \qquad (31)
\end{aligned}
$$

Consider the positive advantage situation and expand the integral of a; the following equation will hold:

$$
\mathrm{Err}\left[\mathbb{E}_{\pi_{\theta_{old}}}\left[\frac{\pi_\theta}{\pi_{\theta_{old}}}A_\pi(s,\pi_{\theta_{old}}(s))\right]\right] \leq \int_s \left\|\rho^{\pi_{\theta_{old}}}(s) - \rho^{\pi_\theta}(s)\right\|\int_a \pi_\theta(a|s)A_\pi(s,a)dads \qquad (32)
$$

Using the conclusion of Lemma 1, the following error bound could be obtained:

$$
\begin{aligned}
&\mathrm{Err}\left[\mathbb{E}_{\pi_{\theta_{old}}}\left[\frac{\pi_\theta}{\pi_{\theta_{old}}}A_\pi(s,\pi_{\theta_{old}}(s))\right]\right] \leq \int_s \frac{\varepsilon\cdot\gamma}{1-\gamma}\int_a \pi_\theta(a|s)A_\pi(s,a)dads \\
&= \int_s \frac{\varepsilon\cdot\gamma}{1-\gamma}\cdot|\mathcal{S}|\cdot\frac{1}{|\mathcal{S}|}\int_a \pi_\theta(a|s)A_\pi(s,a)dads \\
&= \frac{\varepsilon\cdot\gamma}{1-\gamma}\cdot|\mathcal{S}|\cdot\int_s \frac{1}{|\mathcal{S}|}\int_a \pi_\theta(a|s)A_\pi(s,a)dads \\
&= \frac{\varepsilon\cdot\gamma}{1-\gamma}\cdot|\mathcal{S}|\cdot\mathbb{E}_{s\sim\mathrm{Unif}_{\mathcal{S}},a\sim\pi_\theta}[A_\pi(s,a)] \qquad (33)
\end{aligned}
$$

where the $\mathrm{Unif}_{\mathcal{S}}$ represents the uniform distribution of the state. □

Theorem 1 confirms the positive relationship between the estimation error and ε. By using it, a more clear conclusion could be obtained:

Remark 1. *In PPO-Clip, the high ε could enhance the exploration but will result in a high estimation error bound of the surrogate; the low ε could decrease the error bound but will restrict the exploration.*

Therefore, to deal with the exploration and estimation error problems mentioned in Remark 1, this paper considers making the ε adaptive in different situations. The last part designed the sparse reward R_d, and the exploration reward R_e is designed as the incentive reward. The agent should explore more in the task to receive a high-level R_d and R_e. So, when these rewards are too low, the agent should release the restriction on r_t to encourage the exploration. When these rewards are high and stable, the restriction on r_t increases to ensure the estimation of the surrogate is accurate.

So, the exploration advantage function $A_\pi^{exp}(s_t,a_t)$ can be used to represent the advantage function that is estimated by R_d and R_e, which can reflect the exploration ability of the agent:

$$
\begin{aligned}
&A_\pi^{exp}(S_t,a_t) = E_\pi\left[\sum_{k=0}^{\infty}\gamma^k(R_d(t+k)+\sum_{i=1}^{6}R_{e,i}(t+k))|S_t=s,a_t=a\right] - \\
&E_\pi\left[\sum_{k=0}^{\infty}\gamma^k(R_d(t+k)+\sum_{i=1}^{6}R_{e,i}(t+k))|S_t=s\right] \qquad (34)
\end{aligned}
$$

According to the exploration function, an exploration PPO algorithm is proposed with an adaptive clip parameter ε. When the exploration advantage function is lower than last time, to improve the exploration ability, ε will be enlarged. Otherwise, the ε will be reduced, restraining the updated policy in a trust region. To sum up, the adaptive mechanism is designed as follows:

$$\varepsilon(t) = \begin{cases} \varepsilon(t-1) - clip(\frac{A^{exp}_{\pi_{\theta_t}} - A^{exp}_{\pi_{\theta_{t-1}}}}{A^{exp}_{\pi_{\theta_{t-1}}}}, 0, \frac{\varepsilon(t-1)}{2}); A^{exp}_{\pi_{\theta_t}} - A^{exp}_{\pi_{\theta_{t-1}}} > 0 \\ \varepsilon(t-1) + clip(\frac{A^{exp}_{\pi_{\theta_t}} - A^{exp}_{\pi_{\theta_{t-1}}}}{A^{exp}_{\pi_{\theta_{t-1}}}}, 0, \frac{\varepsilon(t-1)}{2}); A^{exp}_{\pi_{\theta_t}} - A^{exp}_{\pi_{\theta_{t-1}}} < 0 \\ \varepsilon(t-1); otherwise \end{cases} \tag{35}$$

The clip function in the above equations is to restrict the adaptive mechanism and avoid the ε being abnormal. Through the variation of the exploration advantage function, the exploration-based adaptive ε mechanism is proposed. When simply replacing the constant ε with the adaptive ε, the PPO will be PPO-Explorationε(PPO-Exp). With the restriction of old policies, new policies will be adjusted automatically. The surrogate of the PPO-Exp is as follows:

$$\mathcal{L}^{Exp,\theta} = \mathbb{E}_{\pi_{\theta_{old}}}\left[\min\left(r_t(\theta)A_{\pi_{\theta_{old}}}, clip(r_t(\theta), 1-\varepsilon(t), 1+\varepsilon(t))A_{\pi_{\theta_{old}}}\right)\right] \tag{36}$$

The Algorithm of PPO-Exp in the formation environment could be seen in Algorithm 1. The exploration and estimation error problem in PPO-Exp could be adapted without delay, and the following Proposition will give the exploration range and the estimation error decrease rate in different situations:

Algorithm 1 PPO-Exploration ε with formation keeping task.

```
Initialize $\pi_0$,$\phi_0$.
for $i = 0, 1, 2, \ldots$ N do
    for $t = 1, \cdots, T$ do
        The leader0 collects state information $\{s_{t,i} | i = 1, \cdots, 5\}$ through the communication protocol (Figure 3a)
        Run policy $\pi_\theta$, obtain the action $\{a_{t,i} | i = 0, 1, \cdots, 5\}$, and send them using the control protocol (Figure 3b).
        The leader and followers execute the action commands and receive a reward as follows: $(R_f(t), R_e(t), R_d(t), R_p(t))$
        Store $(s_t, a_t, s_{t+1}, R_t)$ at the buffer.
    end for
Transitions data from buffer, and estimate $\hat{A}_{\pi_{\theta_t}}$ , $\hat{A}^{exp}_{\pi_{\theta_t}}$, respectively.
    if $\hat{A}^{exp}_{\pi_{\theta_t}} - \hat{A}^{exp}_{\pi_{\theta_{t-1}}} > 0$ then
        $\varepsilon(t) = \varepsilon(t-1) - clip(\frac{\hat{A}^{exp}_{\pi_{\theta_t}} - \hat{A}^{exp}_{\pi_{\theta_{t-1}}}}{\hat{A}^{exp}_{\pi_{\theta_{t-1}}}}, 0, \frac{\varepsilon(t-1)}{2})$
    end if
    if $\hat{A}^{exp}_{\pi_{\theta_t}} - \hat{A}^{exp}_{\pi_{\theta_{t-1}}} < 0$ then
        $\varepsilon(t) = \varepsilon(t-1) + clip(\frac{\hat{A}^{exp}_{\pi_{\theta_t}} - \hat{A}^{exp}_{\pi_{\theta_{t-1}}}}{\hat{A}^{exp}_{\pi_{\theta_{t-1}}}}, 0, \frac{\varepsilon(t-1)}{2})$
    end if
    for $j = 1, \cdots, M$ do
        $\hat{\mathcal{L}}_\theta = \sum_{t=1}^{T} \min(r_t \cdot \hat{A}_{\pi_{\theta_t}}, clip(1-\varepsilon, 1+\varepsilon, r)\hat{A}_{\pi_{\theta_t}})$
        Update $\theta$ by SGD or Adam.
    end for
    Update critic network parameter $\phi_t$ by minimizing:
    $\sum_{k=1}^{T}(\sum_{t'>k} \gamma^{t'-t} R_t - V_\phi(s_t))^2$
end for
```

Proposition 1. *In PPO-Exp, when* $\hat{A}^{exp}_{\pi_{\theta_t}} - \hat{A}^{exp}_{\pi_{\theta_{t-1}}} < 0$*, the exploration range of next policy will be expanded to* $\frac{\|\pi_{\theta_t}-\pi_{\theta_{t-1}}\|}{\|\pi_{\theta_{t-1}}\|} \leq \varepsilon + \frac{\hat{A}^{exp}_t - \hat{A}^{exp}_{t-1}}{\hat{A}^{exp}_{t-1}} \leq \frac{3\varepsilon(t-1)}{2}$*; when* $\hat{A}^{exp}_{\pi_{\theta_t}} - \hat{A}^{exp}_{\pi_{\theta_{t-1}}} > 0$*, in next update, the error bound of the surrogate will decrease to* $O(\frac{\varepsilon(t-1)}{2})$*.*

Proof. When $\hat{A}^{exp}_{\pi_{\theta_t}} - \hat{A}^{exp}_{\pi_{\theta_{t-1}}} < 0$, according to Equation (35), it is easy to see the next policy will be expanded to $\frac{\|\pi_{\theta_t}-\pi_{\theta_{t-1}}\|}{\|\pi_{\theta_{t-1}}\|} \leq \varepsilon + clip(\frac{\hat{A}^{exp}_t - \hat{A}^{exp}_{t-1}}{\hat{A}^{exp}_{t-1}}, 0, \frac{\varepsilon(t-1)}{2})$. Then, the following inequality will hold:

$$0 \leq clip(\frac{\hat{A}^{exp}_t - \hat{A}^{exp}_{t-1}}{\hat{A}^{exp}_{t-1}}, 0, \frac{\varepsilon(t-1)}{2}) \leq \frac{\varepsilon(t-1)}{2} \tag{37}$$

So, the following inequality is held:

$$\frac{\|\pi_{\theta_t} - \pi_{\theta_{t-1}}\|}{\|\pi_{\theta_{t-1}}\|} \leq \varepsilon + \frac{\varepsilon(t-1)}{2} = \frac{3\varepsilon(t-1)}{2} \tag{38}$$

When $\hat{A}^{exp}_{\pi_{\theta_t}} - \hat{A}^{exp}_{\pi_{\theta_{t-1}}} > 0$, and Assumption 1 is satisfied, it is obvious that the conclusion of Theorem 1 could be used in PPO-Exp. So, using Equation (35) and Theorem 1, the PPO-Exp's decrease rate of the bound is as follows:

$$\begin{aligned}
&\Delta\mathrm{Err}\left[\mathcal{L}^{\mathrm{Exp}}_{\theta}\right] = \mathrm{Err}\left[\mathbb{E}_{\pi_{\theta_{t-1}}}\left[\frac{\pi_{\theta_t}}{\pi_{\theta_{t-1}}} A_{\pi_{\theta_{t-1}}}\right]\right] - \mathrm{Err}\left[\mathbb{E}_{\pi_{\theta_{t-2}}}\left[\frac{\pi_{\theta_{t-1}}}{\pi_{\theta_{t-2}}} A_{\pi_{\theta_{t-2}}}\right]\right] \\
&\leq \gamma \frac{\varepsilon(t) - \varepsilon(t-1)}{1-\gamma} \cdot |\mathcal{S}| \cdot \left\| \mathbb{E}_{s\sim \mathrm{Unif}_{\mathcal{S}}, a\sim\pi_{\theta_t}}[A_\pi(s,a)] - \mathbb{E}_{s\sim \mathrm{Unif}_{\mathcal{S}}, a\sim\pi_{\theta_{t-1}}}[A_\pi(s,a)] \right\| \\
&\leq \gamma \frac{\varepsilon(t) - \varepsilon(t-1)}{1-\gamma} \cdot |\mathcal{S}| \cdot \Gamma \\
&\leq \gamma \frac{(\varepsilon(t-1) + clip(\frac{\hat{A}^{exp}_t - \hat{A}^{exp}_{t-1}}{\hat{A}^{exp}_{t-1}}, 0, \frac{\varepsilon(t-1)}{2})) - \varepsilon(t-1)}{1-\gamma} \cdot |\mathcal{S}| \cdot \Gamma \\
&\leq \gamma \frac{\frac{3\varepsilon(t-1)}{2}}{1-\gamma} \cdot |\mathcal{S}| \cdot \Gamma = O(\frac{\varepsilon(t-1)}{2})
\end{aligned} \tag{39}$$

where the Γ is the upper bound of advantage:

$$\Gamma = \max_{\forall t} \left\| \mathbb{E}_{s\sim \mathrm{Unif}_{\mathcal{S}}, a\sim\pi_{\theta_t}}[A_\pi(s,a)] - \mathbb{E}_{s\sim \mathrm{Unif}_{\mathcal{S}}, a\sim\pi_{\theta_{t-1}}}[A_\pi(s,a)] \right\| \tag{40}$$

□

Proposition 1 indicates that the PPO-Exp could encourage the agents to adjust the exploration in different situations. The next section will validate it through numerical experiments.

6. Numerical Experiments

This section compares the PPO-Exp with four common reinforcement learning algorithms (PPO-Clip, PPO-KL, TD3, DDPG) in the formation-keeping task, and compared the performace of PPO-Exp and PPO-Clip in the formation changing task and obstacle avoidance task.

6.1. Experimental Setup

In terms of hardware, all the experiments are completed on the Windows 10 (64-bit) operating system, Intel(R) Core i7 processor, 16 GB memory, and 4 GB video memory. As for software, OpenAI-gym [45] is used to design the reinforcement learning environment and the physics rulers of the UAVs' formation.

The formation task is modeled on the OpenAI gym environment. See Figure 1; the position of the leader and followers can be seen in Table 1. The formation is updated by the dynamic equations solved by the difference method per 0.5 s per time mesh grid. The environment noises are set as $N(0,1)$ default. The target area is designed as a circle at $(200,400)$ with a radius of 40.

Table 1. The initial position of UAVs' formation.

	Leader0	**Follower1**	**Follower2**	**Follower3**	**Follower4**	**Follower5**
Position X	160	190	220	130	100	160
Position Y	190	160	100	160	100	130

6.2. Experiments on PPO-Exploration ε

The following famous continuous space RL algorithms are explored in this section: TD3, DDPG, PPO-KL, and PPO-clip; they are compared to the proposed method under the formation-keeping task.

- PPO-Clip [40]: Proximal Policy Optimization with Clip(PPO-Clip) function.
- PPO-KL [40]: Proximal Policy Optimization with KL-divergence(PPO-KL) constrain.
- DDPG [46]: Deep Deterministic Policy Gradient(DDPG) algorithm, which is a continuous action deep reinforcement learning algorithm that uses Actor–Critic architecture. In DDPG, the deterministic policy gradient is used to update the Actor parameter.
- TD3 [47]: Twin Delayed Deep Deterministic (TD3) policy gradient algorithm, which is a variant of DDPG. The TD3 introduced the delaying policy updates mechanism and the double network architecture to manage the per-update error and overestimation bias in DDPG.

The main hyperparameters of the contrast experiment are shown in Table 2. The blank area in the above table means the algorithm does not include this parameter.

Table 2. The main hyperparameters of the algorithm used in the experiment.

Parameter Name	**TD3**	**DDPG**	**PPO-KL**	**PPO-Clip**	**PPO-Exp**
γ	0.9	0.9	0.9	0.9	0.9
A_{LR}	0.00005	0.00005	0.00005	0.00005	0.00005
C_{LR}	0.0002	0.0002	0.0002	0.0002	0.0002
Batch	32	32	32	32	32
A_{US}			10	10	10
C_{US}			10	10	10
EPS			10^{-8}	10^{-8}	10^{-8}
$D_{KL}(target)$			0.01		
λ			0.5		
ε_{clip}			0.1	0.1	0.1
τ_{DDPG}		0.01			
VAR_{DDPG}		3			
Explore Step	500				
dim_{HIDDEN}	32				

Set the episode length be 200; the results of PPO-Exploration ε and other comparing algorithms are shown in Figure 7a. As the learning curves indicated, the PPO series methods achieved better performance; in all variations of PPO, the PPO-Exp has the best performance. It is validated that the adaptive mechanism based on exploration makes sense during policy updating. Figure 7b shows the change of ε; the series $\varepsilon(t)$ is stationary, and varies around 0.05, although the initial value is 0.1, which means 0.05 is the balance point between exploration and exploitation found by PPO-Exp. Meanwhile, the episode reward curve of PPO-Exp is higher than PPO-Clip's, validating the idea that exploration from PPO-Exp is efficient.

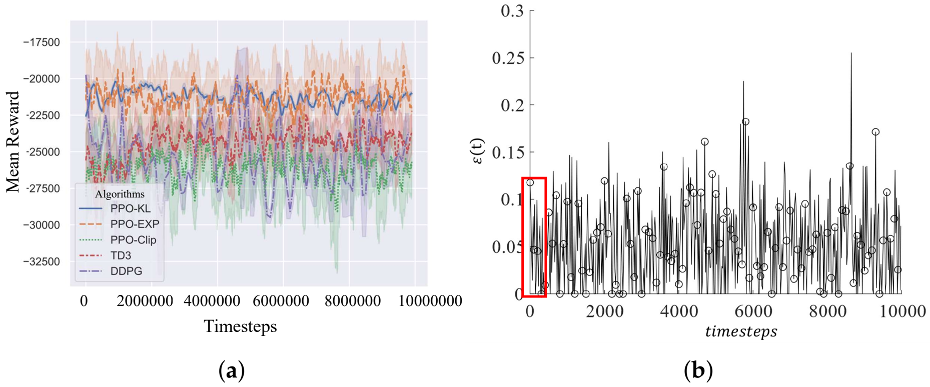


Figure 7. (**a**): Learning curves of TD3, DDPG, PPO-KL, PPO-Clip, and PPO-Exp; (**b**): The on of ε of PPO-Exploration ε during the training process.

6.3. Experiments on Formation Keeping

Only the learning curve was unable to declare whether the algorithm works well, so the trained PPO-Exp is used to perform 200s; the formation track can be seen in Figure 8. In this way, there is only a slight distortion in the formation, indicating that PPO-Exp can perform better in real tasks than PPO-Clip.

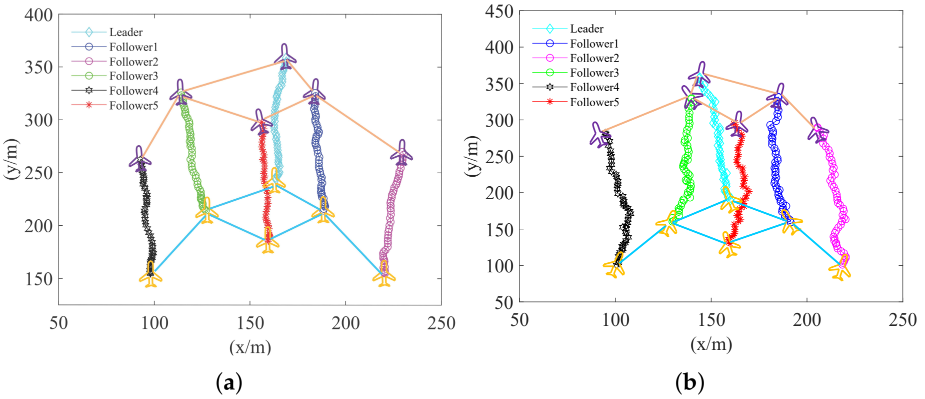
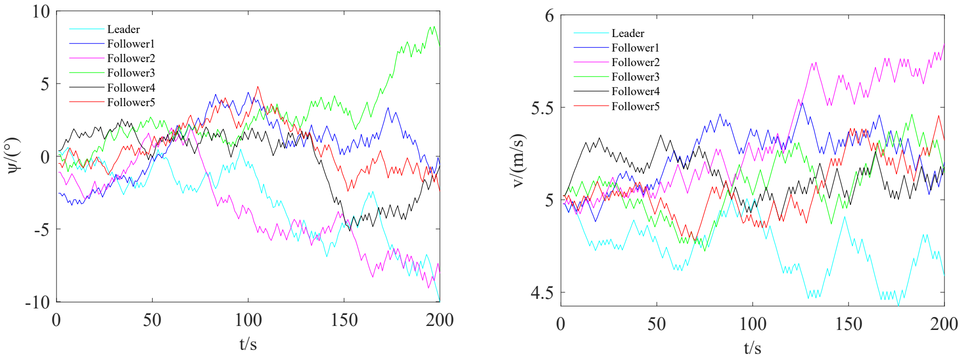


Figure 8. (**a**): The flight track of formation that is controlled by trained PPO-Exp ; (**b**): The flight track of formation that is controlled by trained PPO-Clip.

Furthermore, to evaluate the results, we plotted the heading ψ and the velocity v during 200 s in Figure 9. Figure 9a shows that followers 1, 4, and 5 are approaching gradually as time goes on. Followers 2, 3 and the leader, have no such trend to converge gradually; however, all the heading deviations are no more than 10°. In Figure 9b, the velocity of each UAV is shown. The velocities of followers 1, 3, 4, and 5 diverge a little and then converge. Corresponding to Figure 9a, followers 1, 4, and 5 are closer in terms of the value of velocity and heading; the leader and follower 2 are far away from these followers, but the velocity difference is not more than 1.5 m/s as well. This inspired us to design the reward based on the velocity and heading.

To illustrate the influence of environmental noise on formation keeping, the results show the formation track with no control in Figure 2a. To verify that the proposed centralized method saves time, this section further compares the decentralized version of PPO-Exp: PPO-Exp-Dec, which, similar to MAPPO, needs all six UAV agents to learn the control policy at the same time.



(**a**) (**b**)

Figure 9. (**a**) The test results in the heading angle of PPO-Exp; (**b**) The test results in the velocity of PPO-Exp.

To validate that the protocol can reduce the communication cost and avoid placing the UAVs out of the communication range, this section also compares the protocol-free version: PPO-exp-pro. The results can be seen in Table 3. Γ represents the episode reward, T represents time per episode, r_{col} and r_{fai} represents the collision rate and failure to communicate rate, respectively.

Table 3. The experimental results in different algorithms.

Algorithm	**Γ**	**T**	r_{coll}(%)	r_{fail}(%)
PPO-exp	**−19,197.2 ± 1307.4**	2.19 ± 0.04	**0.93 ± 0.01**	**0.32 ± 0.02**
PPO-exp-dec	−20,374.7 ± 1926.4	10.06 ± 0.08	1.01 ± 0.02	0.35 ± 0.01
PPO-exp-pro	−23,001.3 ± 2507.2	2.43 ± 0.03	0.98 ± 0.03	12.48 ± 1.76
PPO-clip	−20,305.7 ± 1588.6	2.14 ± 0.06	0.97 ± 0.02	0.94 ± 0.03
Greedy	−39,074.5 ± 3806.5	**1.15 ± 0.04**	12.32 ± 1.32	10.56 ± 0.65

To further verify the effectiveness of the proposed method, ablation experiments are performed (see Figures 2a,b and 8b). Figure 8b shows the trained PPO-clip without the exploration mechanism. Although there is no UAV crash, the leader and follower3 are very close, and the formation is not as orderly as the PPO-Exp. Figure 2a shows the result of no action taken, where the UAVs will crash, and the formation will break up. Figure 2b shows the trained PPO-clip with $\varepsilon = 0.05$, which is the balance point in the PPO-Exp. However, the experimental result shows it performs worse; there is one follower that loses communication with leader, and one follower almost crashes with the leader. The result illustrates that the PPO-Exp with adaptive ε is better than the PPO-Clip with a good ε. In summary, the ablation experiments also indicated that PPO-Exp performs better than other algorithms in terms of learning curves and the real-task.

6.4. Experiment on More Complex Tasks

To further show the efficiency of PPO-Exp in fixed-wing UAV formation keeping, this part design two more complex scenarios: formation changing and obstacle avoidance task, the UAV formation perform 120 s on each task. This part mainly compared the performance of PPO-Exp and PPO-Clip on these tasks.

The goal for the formation changing task is changing the formation shown in Figure 1 to the vertical formation. The vertical formation also expects the differences between leader and followers are as small as possible in coordinates on the x-axis. For guiding the followers to change the formation, this paper utilizes the absolute difference value of x

coordinates to modify the flocking reward. The modified flocking reward (9) and (12) could be represented as follows:

$$R_{f,i} = \|x_0 - x_i\|, \forall i = 1, \cdots, 5 \tag{41}$$

Then the total reward (47) can be rewritten as follows:

$$R(T) = \sum_{i=0}^{5} [\|x_0(T) - x_i(T)\| + R_{e,i}(T)] + R_d(T) + R_p(T) \tag{42}$$

where the $x_0(T), x_i(T)$ represent the x coordinates of leader and ith follower at time T, respectively. To encourage the UAV system to take more exploration on forming new formation, the flocking reward is added to the exploration advantage function:

$$A_\pi^{exp}(S_t, a_t) = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k (R_d(t+k) + \sum_{i=0}^{5} [\|x_0(T) - x_i(T)\| + R_{e,i}(t+k))] | S_t = s, a_t = a\right] - E_\pi\left[\sum_{k=0}^{\infty} \gamma^k (R_d(t+k) + \sum_{i=0}^{5} [\|x_0(T) - x_i(T)\| + R_{e,i}(t+k))] | S_t = s\right] \tag{43}$$

Training the task with PPO-Exp and PPO-Clip, the training parameters are kept as same as in the previous part except episode length. After training, the test result of PPO-Exp is shown in Figure 10a, and the PPO-Clip is shown in Figure 10b. To evaluate the performance, this paper draws the plots of the x coordinates and timesteps of the leader and followers in Figure 10c,d. The closer the x coordinates of followers to that of the leader, the better the performance will be. The x coordinates of followers in (c) converge to the leader faster than (d), representing that PPO-Exp can change vertical formation faster than PPO-Clip.

To further evaluate the formed vertical formation. Denote the terminal time as t_{ter}, calculate the average difference between the followers and leader in x coordinates in the last ten timesteps, and denote the result as δ_x, which can be represented as follows:

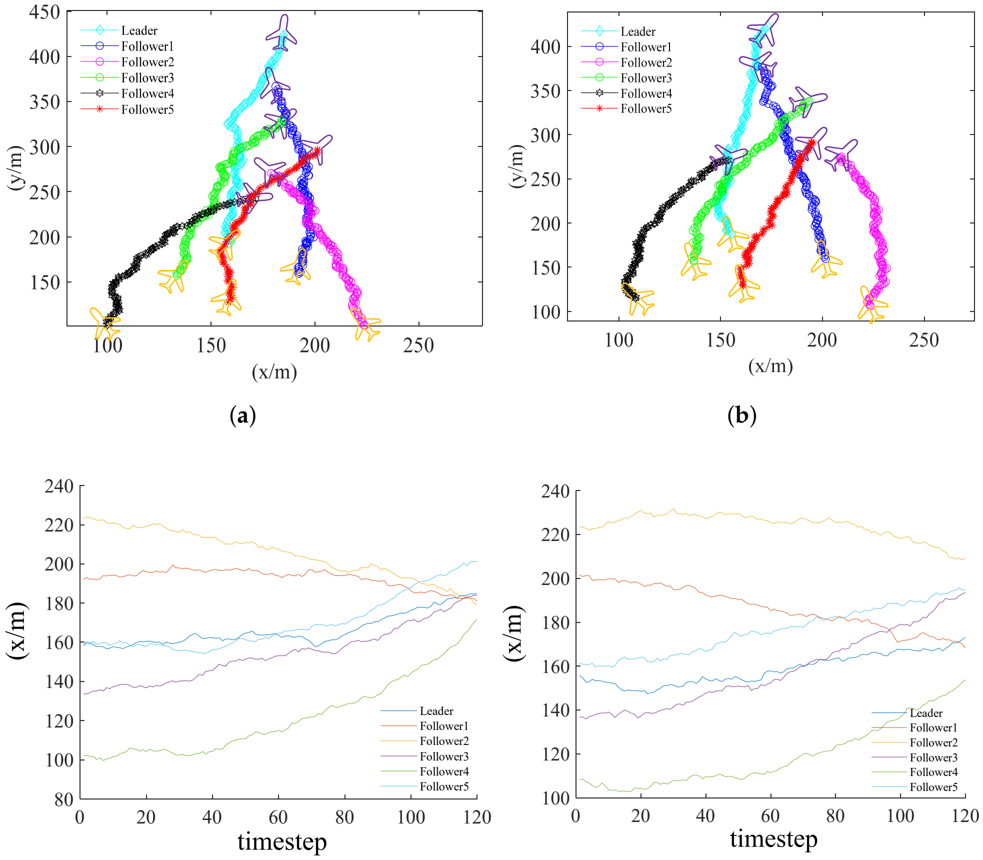
$$\delta_x = \frac{1}{5} \sum_{i=1}^{5} \sum_{t > t_{ter} - 10} \|x_0(t) - x_i(t)\| \tag{44}$$

The low δ_x indicates the follower is close to the leader in x coordinates. In PPO-Clip, the calculated $\delta_x \approx 95.383$, but in PPO-Exp, the calculated $\delta_x \approx 43.816$, which is nearly half of the PPO-Clip.

Compared to the control strategy in formation keeping, the followers in formation changing tasks perform good cooperation. All followers maneuver orderly to the position where the leader's x-coordinate is located. To avoid the UAVs collide each other, the followers decided to move to different positions on the y-axis. The followers take different maneuvers depending on their initial position to reach the position. e.g., follower 4, in the initial time, is far away from the leader in x-coordinates. For follower 4, a collision avoidance path is moving to the tail of the newly formed formation. Therefore, the follower4 achieves a large angle arc maneuver and moves to the tail of the formed vertical formation.

The target of the obstacle avoidance task is to reach the target area and avoid crashing into the obstacle. This paper considers a circle area on the plane as an obstacle. Denote the coordinates of the obstacle center is (x_{obs}, y_{obs}), and the radius is r_{obs}. A simple approach to consider this situation is to add a penalty on the formation system reward when the UAVs crash on the obstacle, the penalty effect. The penalty for crashing into the obstacle is denoted as follows:

$$R_{o,i} = \begin{cases} 0, \sqrt{(x_i - x_{obs})^2 + (y_i - y_{obs})^2} \le r_{obs} \\ -10{,}000, otherwise \end{cases} \tag{45}$$



(**a**) (**b**)

(**c**) (**d**)

Figure 10. (**a**): The performance of vertical formation changing task by PPO-Exp; (**b**): The performance of vertical formation changing task by PPO-Clip (**c**): The x coordinate of formation system in PPO-Exp; (**d**): The x coordinate of formation system in PPO-Clip

Similar to the exploration reward $R_{e,i}$, to

$$R^{obs}_{e,i} = min\{|x_i - x_{obs}|, |y_i - y_{obs}|\} \tag{46}$$

Then the total reward (47) can be rewritten as follows:

$$R(T) = \sum_{i=0}^{5}\Big[R_{f,i}(T) + R_{e,i}(T) + R^{obs}_{e,i}(T) + R_{o,i}(T)\Big] + R_d(T) + R_p(T) \tag{47}$$

To encourage the UAV system to take more exploration on avoid obstacle, the exploration reward in avoid obstacle $R^{obs}_{r,i}$ is added to the exploration advantage function:

$$A^{exp}_{\pi}(S_t, a_t) = E_{\pi}\left[\sum_{k=0}^{\infty}\gamma^k(R_d(t+k) + \sum_{i=0}^{5}\Big[R_{f,i}(T) + R_{e,i}(t+k) + R^{obs}_{e,i}(T))\Big]|S_t = s, a_t = a\right] -$$

$$E_{\pi}\left[\sum_{k=0}^{\infty}\gamma^k(R_d(t+k) + \sum_{i=0}^{5}[R_{f,i}(T) + R_{e,i}(t+k) + R^{obs}_{e,i}(T))]|S_t = s\right] \tag{48}$$

Training the obstacle to avoid task with PPO-Exp and PPO-Clip, the training parameters are kept as same as the previous part except episode length. After training with PPO-Exp and PPO-Clip, the test results of obstacle avoid task are shown in Figure 11a, and the results of PPO-Clip can be seen in Figure 11b. A follower in the formation trained by PPO-Clip crashed on the obstacle at 94 timesteps. The formation trained by PPO-Exp performed the arc maneuvers and avoided the obstacle. PPO-Exp performs better than PPO-Clip because it can explore more policies to reach the target area and discover a good path to avoid obstacles. However, the PPO-Clip still tries to reach the target area straight.

Compared to the formation keeping task without obstacles, the obstacle scenario requires the formation system to explore more to avoid the obstacle. Therefore, in this scenario, compared to the fixed ε PPO-Clip, the PPO-Exp shows better performance because it could adjust their ε to balance exploration and estimation error. Then the PPO-Exp explored the large-angle arc maneuvers and performed them to avoid the obstacle.

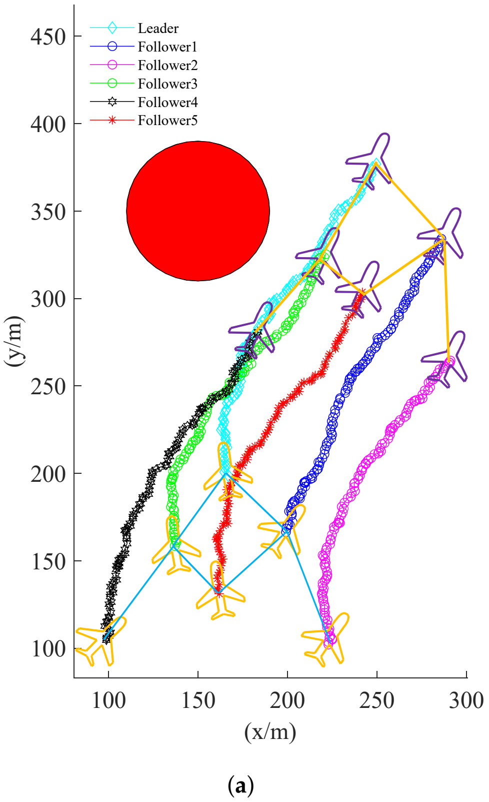


Figure 11. (**a**): The performance of formation keeping with obstacle avoid task by PPO-Exp; (**b**): The performance of formation keeping with obstacle avoid task by PPO-Clip.

7. Conclusions

This paper studies a flocking scenario consistent with one leader (with an intelligence chip) and several followers(without an intelligence chip). The reinforcement learning environment is constructed (continuous action and state space) with an OpenAI gym, and the reward is designed as a regular part and an exploration part. A low-communication cost protocol is provided to ensure the UAVs can communicate the state and action information between leader and followers. In addition, a variation of Proximal Policy Optimization is proposed to balance the dilemma between the estimation error bound and the exploration ability of PPO. The proposed method can help UAVs adjust the explore strategy, and the experiments demonstrate it has better performance than the current algorithms such as PPO-KL, PPO-clip, and DDPG.

Author Contributions: Conceptualization, D.X. and H.L.; Methodology, D.X., Y.G.; Supervision, D.X. and H.L.; Software, Y.G., Z.Y., Z.W., R.L., R.Z.; Formal analysis, Y.G. and H.L.; Writing—original draft, Y.G., R.Z.; Validation, Z.Y., Z.W., R.L., R.Z.; Visualization, Z.Y., Z.W.; Funding acquisition, D.X.; Resources, R.L.; Investigation, X.X., Data curation, X.X., Writing—review and editing, X.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

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

References

1. Zhou, W.; Li, J.; Zhang, Q. Joint Communication and Action Learning in Multi-Target Tracking of UAV Swarms with Deep Reinforcement Learning. *Drones* **2022**, *6*, 339. [CrossRef]
2. Tian, S.; Wen, X.; Wei, B.; Wu, G. Cooperatively Routing a Truck and Multiple Drones for Target Surveillance. *Sensors* **2022**, *22*, 2909. [CrossRef] [PubMed]
3. Wu, G.; Fan, M.; Shi, J.; Feng, Y. Reinforcement Learning based Truck-and-Drone Coordinated Delivery. *IEEE Trans. Artif. Intell.* **2021**. [CrossRef]
4. Gupta, L.; Jain, R.; Vaszkun, G. Survey of important issues in uav communication networks. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 1123–1152. [CrossRef]
5. Wu, Q.; Zeng, Y.; Zhang R. Joint trajectory and communication design for multi-uav enabled wireless networks. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 2109–2121. [CrossRef]
6. Eisenbeiss, H. A mini unmanned aerial vehicle (uav): System overview and image acquisition. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2004**, *36*, 1–7. Available online: https://www.isprs.org/proceedings/XXXVI/5-W1/papers/11.pdf (accessed on 29 November 2022).
7. Wang, Y.; Xing, L.; Chen, Y.; Zhao, X.; Huang, K. Self-organized UAV swarm path planning based on multi-objective optimization. *J. Command. Control* **2021**, *7*, 257–268. [CrossRef]
8. Kuriki, Y.; Namerikawa, T. Formation control with collision avoidance for a multi-uav system using decentralized mpc and consensus-based control. *SICE J. Control Meas. Syst. Integr.* **2015**, *8*, 285–294. [CrossRef]
9. Saif, O.; Fantoni, I.; Zavala-Río A. Distributed integral control of multiple uavs: Precise flocking and navigation. *IET Contr. Theory Appl.* **2019**, *13*, 2008–2017. [CrossRef]
10. Chen, H.; Wang, X. Formation flight of fixed-wing UAV swarms: A group-based hierarchical approach. *Chin. J. Aeronaut.* **2021**, *34*, 504–515. [CrossRef]
11. Liu, Z.; Wang, X.; Shen, L.; Zhao, S.; Cong, Y.; Li, J.; Yin, D.; Jia, S.; Xiang, X. Mission-Oriented Miniature Fixed-Wing UAV Swarms: A Multilayered and Distributed Architecture. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *1*, 2168–2216. [CrossRef]
12. Koch, W.; Mancuso, R.; West, R.; Bestavros, A. Reinforcement learning for uav attitude control. *ACM Trans. Cyber-Phys. Syst.* **2019**, *3*, 1–21. [CrossRef]
13. Kaelbling, L.; Littman, M.; Moore, A. Reinforcement learning: A survey. *J. Artif. Intell. Res.* **1996**, *4*, 237–285. 10.1613/jair.301. [CrossRef]
14. Li, Y. Deep reinforcement learning: An overview. *arXiv* **2017**, arXiv:1701.07274. Available online: https://arxiv.org/pdf/1701.07274.pdf (accessed on 29 November 2022).
15. Huy, P.; Hung, L.; David, S. Autonomous uav navigation using reinforcement learning. *arXiv* **2018**, arXiv:1801.05086. Available online: https://arxiv.org/pdf/1801.05086.pdf (accessed on 29 November 2022).
16. Gullapalli, V.; Franklin, J.; Benbrahim, H. Acquiring robot skills via reinforcement learning. *IEEE Control Syst. Mag.* **1994**, *14*, 13–24. [CrossRef]
17. Huang, J.; Mo, Z.; Zhang, Z.; Chen, Y. Behavioral control task supervisor with memory based on reinforcement learning for human—Multi-robot coordination systems. *Front. Inf. Technol. Electron. Eng.* **2022**, *23*, 1174–1188. FITEE.2100280. [CrossRef]
18. Zhang, F.; Leitner, J.; Milford, M.; Upcroft, B.; Corke, P. Towards vision-based deep reinforcement learning for robotic motion control. *arXiv* **2017**, arXiv:1511.03791. Available online: https://arxiv.org/pdf/1511.03791.pdf (accessed on 29 November 2022).
19. Tomimasu, M.; Morihiro, K.; Nishimura, H. A reinforcement learning scheme of adaptive flocking behavior. In Proceedings of the 10th International Symposium on Artificial Life and Robotics (AROB), Oita, Japan, 4–6 February 2005.
20. Morihiro, K.; Isokawa, T.; Nishimura, H.; Matsui, N. Characteristics of flocking behavior model by reinforcement learning scheme. In Proceedings of the 2006 SICE-ICASE International Joint Conference, Busan, Republic of Korea, 18–21 October 2006. [CrossRef]
21. Shao, W.; Chen, Y.; Huang, J. Optimized Formation Control for a Class of Second-order Multi-agent Systems based on Single Critic Reinforcement Learning Method. In Proceedings of the 2021 IEEE International Conference on Networking, Sensing and Control (ICNSC), Xiamen, China, 3–5 December 2021; pp. 1–6. [CrossRef]
22. Wang, C.; Wang, J.; Zhang, X. A deep reinforcement learning approach to flocking and navigation of uavs in large-scale complex environments. In Proceedings of the 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Anaheim, CA, USA, 26–28 November 2018. [CrossRef]
23. Beard, R.; Kingston, D.; Quigley, M.; Snyder, D.; Christiansen, R.; Johnson, W.; McLain, T.; Goodrich, M. Autonomous vehicle technologies for small fixed-wing uavs. *J. Aerosp. Comput. Inf. Commun.* **2005**, *2*, 92–108. [CrossRef]
24. Hung, S.; Givigi, S.; Noureldin, A. A dyna-q (lambda) approach to flocking with fixed-wing uavs in a stochastic environment. In Proceedings of the 2015 IEEE International Conference on Systems, Man, and Cybernetics(SMC), Hong Kong, China, 9–12 October 2015. [CrossRef]

25. Hung, S.; Givigi, S. A Q-learning approach to flocking with UAVs in a stochastic environment. *IEEE Trans. Cybern.* **2016**, *47*, 186–197. [CrossRef]
26. Yan, C.; Xiang, X.; Wang, C. Fixed-wing uavs flocking in continuous spaces: A deep reinforcement learning approach. *Robot. Auton. Syst.* **2020**, *131*, 103594. [CrossRef]
27. Wang, C.; Yan, C.; Xiang, X.; Zhou, H. A continuous actor-critic reinforcement learning approach to flocking with fixed-wing UAVs. In Proceedings of the 2019 Asian Conference on Machine Learning(ACML), Nagoya, Japan, 17–19 November 2019. Available online: http://proceedings.mlr.press/v101/wang19a/wang19a.pdf (accessed on 29 November 2022).
28. Bøhn, E.; Coates, E.; Moe, E.; Johansen, T.A. Deep reinforcement learning attitude control of fixed-wing uavs using proximal policy optimization. In Proceedings of the 2019 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 11–14 June 2019. [CrossRef]
29. Hernandez, P.; Kaisers, M.; Baarslag, T.; de Cote, E.M. A survey of learning in multiagent environments: Dealing with non-stationarity. *arXiv* **2017**, arXiv:1707.09183. Available online: https://arxiv.org/pdf/1707.09183.pdf (accessed on 29 November 2022).
30. Yan, C.; Wang, C.; Xiang, X.; Lan, Z.; Jiang, Y. Deep reinforcement learning of collision-free flocking policies for multiple fixed-wing uavs using local situation maps. *IEEE Trans. Ind. Inform.* **2021**, *18*, 1260–1270. [CrossRef]
31. Peng, J.; Williams, R. Incremental multi-step Q-learning. *Mach. Learn.* **1996**, *22*, 283–290.:1018076709321. [CrossRef]
32. Hasselt, H.; Marco, W. Reinforcement Learning in Continuous Action Spaces. In Proceedings of the 2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning, Honolulu, HI, USA, 1–5 April 2007; pp. 272–279. [CrossRef]
33. Wang, C.; Wu, L.; Yan, C.; Wang, Z.; Long, H.; Yu, C. Coactive design of explainable agent-based task planning and deep reinforcement learning for human-UAVs teamwork. *Chin. J. Aeronaut.* **2020**, *33*, 2930–2945. [CrossRef]
34. Zhao, Z.; Rao, Y.; Long, H.; Sun, X.; Liu, Z. Resource Baseline MAPPO for Multi-UAV Dog Fighting. In Proceedings of the 2021 International Conference on Autonomous Unmanned Systems (ICAUS), Changsha, China, 24–26 September 2021._327. [CrossRef]
35. Yan, C.; Xiang, X.; Wang, C.; Lan, Z. Flocking and Collision Avoidance for a Dynamic Squad of Fixed-Wing UAVs Using Deep Reinforcement Learning. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 4738-4744. [CrossRef]
36. Song, Y.; Choi, J.; Oh, H.; Lee, M.; Lim, S.; Lee, J. Improvement of Decentralized Flocking Flight Efficiency of Fixed-wing UAVs Using Inactive Agents. In Proceedings of the AIAA Scitech 2019 Forum, San Diego, CA, USA, 7–11 January 2019.
37. Yan, Y.; Wang, H.; Chen, X. Collaborative Path Planning based on MAXQ Hierarchical Reinforcement Learning for Manned/Unmanned Aerial Vehicles. In Proceedings of the 39th Chinese Control Conference (CCC), Shenyang, China, 27–29 July 2020; pp. 4837–4842. [CrossRef]
38. Ren, T.; Niu, J.; Liu, X.; Hu, Z.; Xu, M.; Guizani, M. Enabling Efficient Scheduling in Large-Scale UAV-Assisted Mobile-Edge Computing via Hierarchical Reinforcement Learning. *IEEE Internet Things J.* **2021**, *9*, 7095–7109. [CrossRef]
39. Yang, H.; Jiang, B.; Zhang, Y. Fault-tolerant shortest connection topology design for formation control. *Int. J. Control Autom. Syst.* **2014**, *12*, 29–36. [CrossRef]
40. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347. Available online: https://arxiv.org/pdf/1707.06347.pdf (accessed on 29 November 2022).
41. Banerjee, N.; Chakraborty, S.; Raman, V.; Satti, S.R. Space efficient linear time algorithms for bfs, dfs and applications. *Theory Comput. Syst.* **2018**, *62*, 1736–1762. [CrossRef]
42. Bansal, T.; Pachocki, J.; Sidor, S.; Sutskever, I.; Mordatch, I. Emergent Complexity via Multi-Agent Competition. *arXiv* **2017**, arXiv:1710.03748.
43. Sutton, R.; Barto, A. *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge MA, USA, 2018.
44. Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; Moritz, P. Trust Region Policy Optimization. In Proceeding of the 2015 International Conference on Machine Learning(ICML), Lille, France, 6–11 July 2015; pp. 1889–1897.
45. Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; Zaremba, W. Openai gym. *arXiv* **2016**, arXiv:1606.01540. Available online: https://arxiv.org/pdf/1606.01540.pdf (accessed on 29 November 2022).
46. Lillicrap, T.; Hunt, J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Wierstra, D. Continuous control with deep reinforcement learning. In Proceeding of the 4th International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2–4 May 2016; pp. 1582–1591.
47. Fujimoto, S.; Herke, H.; David, M. Addressing Function Approximation Error in Actor-Critic Methods. In Proceeding of the 2018 International Conference on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018; pp. 1582–1591. Available online: http://proceedings.mlr.press/v80/fujimoto18a/fujimoto18a.pdf (accessed on 29 November 2022).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.