




Article

Deep Reinforcement Learning-Driven Collaborative Rounding-Up for Multiple Unmanned Aerial Vehicles in Obstacle Environments

Zipeng Zhao ^{1,†} , Yu Wan ^{1,†}  and Yong Chen ^{2,3,*} 

¹ Laboratory for Big Data and Decision, National University of Defense Technology, Changsha 410003, China; zhaozipeng22@nudt.edu.cn (Z.Z.); wanyu13@nudt.edu.cn (Y.W.)

² Chengdu Fluid Dynamics Innovation Center No.75, West 2nd Section, 2nd Ring Road, Qingyang District, Chengdu 610071, China

³ School of Systems Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China

* Correspondence: literature_chen@nudt.edu.cn

† These authors contributed equally to this work.

Abstract: With the rapid advancement of UAV technology, the utilization of multi-UAV cooperative operations has become increasingly prevalent in various domains, including military and civilian applications. However, achieving efficient coordinated rounding-up of multiple UAVs remains a challenging problem. This paper addresses the issue of collaborative drone hunting by proposing a decision-making control model based on deep reinforcement learning. Additionally, a shared experience data pool is established to facilitate communication between drones. Each drone possesses independent decision-making and control capabilities while also considering the presence of other drones in the environment to collaboratively accomplish obstacle avoidance and rounding-up tasks. Furthermore, we redefine and design the reward function of reinforcement learning to achieve precise control of drone swarms in diverse environments. Simulation experiments demonstrate the feasibility of the proposed method, showcasing its successful completion of obstacle avoidance, tracking, and rounding-up tasks in an obstacle environment.

Keywords: multi-UAV; obstacle avoidance; rounding-up; multi-agent deep reinforcement learning; soft actor-critic algorithm



Citation: Zhao, Z.; Wan, Y.; Chen, Y. Deep Reinforcement Learning-Driven Collaborative Rounding-Up for Multi-UAV in Obstacle Environments. *Drones* **2024**, *8*, 464. <https://doi.org/10.3390/drones8090464>

Academic Editor: Pablo Rodríguez-González

Received: 26 July 2024

Revised: 25 August 2024

Accepted: 4 September 2024

Published: 6 September 2024



Copyright: © 2024 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

The round up of UAV swarms is a complex and challenging research area that intersects multiple disciplines, including computer science, control theory, communication technology, and aerospace engineering. With the rapid advancement of UAV technology, multi-UAV systems have become a hot topic in both research and application. Compared to single UAVs, multi-UAV systems exhibit numerous advantages in task execution. Single UAVs are often limited by endurance, payload capacity, and coverage area when faced with complex tasks and vast regions. However, multi-UAV systems, through collaborative operations, can effectively overcome these limitations, enhancing the efficiency and reliability of task execution. By employing cooperative strategies such as task allocation, path planning, and formation control, multi-UAV systems can cover larger areas and accomplish more complex tasks in a shorter time while offering higher redundancy and fault tolerance. These advantages make multi-UAV systems highly promising in various fields, including military operations, disaster relief [1], communication services [2], mapping and exploration [3], resource delivery [4], and agriculture [5]. However, the challenge lies in achieving efficient control and coordination of UAV swarms in complex environments.

The problem of UAV swarm pursuit involves controlling one or more groups of UAVs to track and capture targets swiftly and accurately under specific conditions. This problem encompasses challenges in target tracking, path planning, communication coordination,

and obstacle avoidance. To address these challenges, researchers need to conduct comprehensive investigations from both theoretical and practical perspectives. From a theoretical standpoint, the pursuit problem of UAV swarms requires knowledge from multiple fields, including optimization algorithms and control theory. Research in these domains can provide theoretical support and technical guidance for the pursuit problem of UAV swarms. Additionally, the study of communication protocols and collaboration strategies among drones is crucial for achieving efficient collaboration within drone swarms. From a practical perspective, the pursuit problem of drone swarms needs to be researched and validated in conjunction with real-world application scenarios. This includes testing and analyzing the performance parameters of drones, as well as simulating and replicating actual environments. Through these experiments and simulations, the effectiveness and feasibility of proposed control strategies and algorithms can be evaluated.

In the field of UAV cooperative control, traditional optimization algorithms such as ant colony optimization and wolf pack algorithms face limitations in computation time, flexibility, and intelligence, which hinder their ability to fully achieve intelligent control and decision-making in UAV swarms. On the other hand, the recently emerging deep reinforcement learning (DRL) algorithms, with their powerful high-dimensional perception capabilities and ability to handle nonlinear problems, have proven effective in a wide range of tasks, including path planning [6], collision avoidance [7], and autonomous driving [8]. DRL has also seen widespread applications in UAV swarm control. For instance, in [9], DRL algorithms were used to control multiple UAVs to track a person in an obstacle-laden environment, and in [10], multiple UAVs were utilized as communication relays, with DRL algorithms optimizing to maximize the communicable area. Given the extensive applications of deep reinforcement learning in the multi-UAV domain, DRL can similarly be effectively employed to accomplish complex tasks involving UAV swarm coordination, pursuit, and strike missions in challenging environments.

In this paper, our main contributions are as follows:

- We present a collaborative control strategy for unmanned aerial vehicle (UAV) formations utilizing a multi-agent deep reinforcement learning algorithm. This approach is designed to address the challenge of coordinating UAV groups for cooperative rounding-up tasks, with the overarching goal of enhancing strategic control capabilities in battlefield environments and optimizing the efficiency of collaborative operations within UAV groups.
- We have developed a shared experience data pool that allows each UAV to learn from the collective experiences of the entire swarm. This shared mechanism enables each UAV to leverage the operational experiences of others, continuously improving its own behavioral strategies. This approach significantly enhances the adaptability of the UAV system in dynamic environments, enabling it to respond more intelligently to complex scenarios.
- An innovative reward function has been developed to address challenges encountered by unmanned aerial vehicle (UAV) swarms in diverse scenarios, facilitating the successful accomplishment of target missions within a formation. This unique approach aims to optimize the performance of UAV swarms by tailoring the reward structure to the intricacies of various situations.

2. Related Work

In this section, we summarize existing works in multi-UAV control. The emerging paradigm of multi-UAV systems has given rise to a plethora of research avenues, with the primary focus on three pivotal components: perception, coordination, and navigation. Within this complex tapestry, the incorporation of neural network controls offers a promising dimension, enhancing the efficiency and adaptability of multi-UAV operations.

2.1. Optimization-Based Approach

Researchers have put forward the idea of utilizing traditional optimization methods to regulate the obstacle avoidance and pursuit of multiple unmanned aerial vehicles (UAVs). Optimization-based approaches have emerged as a robust framework for controlling multiple drones, employing mathematical techniques to identify the most optimal solution among a range of viable alternatives. These methods approach the planning dilemma as a constrained optimization problem with the objective of minimizing a designated cost function while adhering to constraints such as collision avoidance and formation maintenance [11–14].

For the obstacle avoidance navigation problem, this can be ensured by hard constraints that limit the trajectory to a specified convex space [13,15], or by soft constraints that directly embed the penalty function into the objective function [11]. For optimization-based controls, it is common to separate mapping, collaboration, and planning tasks from the broader navigation process. This approach allows for parallel execution across components, enhancing the transparency and explainability of the overall system. X. Zhou et al. [16] formulated the cluster control problem of UAV swarms as a multi-objective optimization problem and improved the multi-objective pigeon-inspired optimization (MPIO) based on the hierarchical learning behavior of the pigeon swarm to solve it in a distributed manner. In multi-UAV systems, model predictive control (MPC) is gradually used in various environments, including obstacle-free environments, space environments with obstacles [17], and multi-UAV collision-free trajectory generation [18,19]. Kuriki and Namerikawa [20] proposed a multi-UAV collaborative formation control strategy based on decentralized MPC and consensus control with collision avoidance capabilities. Their method ensures that each UAV can make decisions independently while ensuring the consistency of collision avoidance coupling constraints.

Regarding the tracking and rounding-up problem, Fei Yu et al. [21] analyzed the pursuit process based on differential games, gave a boundary setting process, and determined the game strategy based on the kinematic characteristics of the aircraft to achieve pursuit in an obstacle environment. Mikhail Khachumov et al. [22] combined intelligent control theory and jointly applied precise and precise collective control with flexible intelligent control to construct a mathematical model of UAV motion and implement pursuit research in a simulation system. Bingda Tong et al. [23] proposed a method for Harris hawks (*Parabuteo unicinctus*) to cooperatively hunt and intercept enemy drones, constructed a simplified drone guidance model, implemented a lead–follow strategy, and successfully intercepted enemy drones. Jiaxin Li et al. [24] designed a pursuit avoidance strategy, combined with the UAV motion control method, and achieved good results in improving interception time.

The optimization-based approach has undoubtedly elevated the capabilities of multi-UAV control, encompassing obstacle avoidance, tracking, and rounding-up tasks. However, this method encounters certain challenges that warrant attention. Particularly, in dynamic or unpredictable environments, traditional approaches struggle to guarantee the formulation of suitable control strategies at every stage of UAV swarm movement. Consequently, this limitation can result in errors, delays, and ultimately, catastrophic crashes. Addressing these challenges is crucial to ensure the safe and efficient operation of UAV swarms in complex and dynamic scenarios.

2.2. Learning-Based Approach

In recent years, the field of drone control has witnessed significant advancements with the integration of deep learning and reinforcement learning technologies. Researchers have increasingly turned to these learning-based approaches to enhance the adaptability and efficiency of unmanned aerial vehicle (UAV) systems. By leveraging learning methods, it becomes possible to design an end-to-end controller that directly translates input data into control commands, thereby greatly enhancing the capabilities and efficiency of the UAV system. This paper explores the potential of learning-based technologies in improving the

performance of UAV systems, offering insights into the design and implementation of an efficient and adaptable end-to-end controller.

Inspired by biological swarm intelligence, Longting Jiang et al. [25] constructed a communication multi-agent deterministic policy gradient (COM-MADDPG) framework to achieve dynamic rounding-up of target points. Chaoxu Mu et al. [26] also proposed a leader–follower strategy model combined with the actor–critic framework to realize the formation control problem, which effectively reduces the time complexity compared to traditional methods. Bo Li et al. [27] considered the relationship between the enemy and ourselves, designed an alternative maneuver strategy, and trained it using the soft actor–critic algorithm, which improved the flexibility during the pursuit and achieved good rounding-up effects. Xiaowei Fu et al. [28] designed a quasi-proportional guidance control law to generate effective learning samples as an empirical data pool for the DDPG algorithm. Experimental results show that the trained UAV can quickly track other target points, improving efficiency. Qianxin Xia et al. [29] proposed a method based on multi-agent reinforcement learning for formation target capture using an adaptive allocation strategy to complete the allocation of capture points. The trained strategy model can successfully capture escape targets through formation. Ruilong Zhang et al. [30] studied the target pursuit–avoidance problem of multiple quadcopters in an obstacle environment, expanded the MADDPG algorithm, and constructed a two-way coordinated target prediction network to ensure the effectiveness in the pursuit and escape process. Yihao Sun et al. [31] proposed a multi-agent deep deterministic policy gradient algorithm based on attention, which solved the problem of collaborative tracking control of autonomous driving and showed scalability in dynamic environments with obstacles. Yuanda Wang et al. [32] proposed a distributed cooperative pursuit strategy with communication based on reinforcement learning. Based on the deep deterministic policy gradient (DDPG) algorithm, a novel training algorithm for ring and leader–follower network topologies was developed. Verified through extensive simulations, the pursuer can capture agile escapees with a high success rate.

From this perspective, the challenge of cooperative obstacle avoidance and pursuit among multiple UAVs in a three-dimensional environment have predominantly been addressed through the utilization of traditional optimization methods, albeit lacking intelligent elements. However, the advent of reinforcement learning techniques presents an opportunity to train more efficient and intelligent control strategies. In light of this, we propose a novel approach based on deep reinforcement learning to tackle the issue of multi-UAV collaborative pursuit. By leveraging the power of reinforcement learning, our method aims to enhance the cooperative capabilities of UAVs in pursuit scenarios, enabling them to navigate complex environments with greater efficiency and intelligence.

In recent years, reinforcement learning algorithms have been widely applied in decision-making scenarios, particularly in multi-agent deep reinforcement learning (MARL) [33]. Knowledge transfer mechanisms, including shared buffers, action advising, and experience sharing, have been extensively explored in the MARL domain. Ming Tan [34] introduced the concept of shared buffers, where agents improve learning efficiency by sharing experiences. This approach demonstrated the potential of cooperative agents to outperform independent agents by leveraging shared experiences in multi-agent environments. Building on this foundation, recent work has focused on optimizing the conditions under which knowledge transfer occurs. For example, Tonghao Wang et al. [35] proposed an automated design of action-advising trigger conditions using a genetic programming-based method, highlighting how intelligent advising can further enhance agent performance by dynamically adjusting advising strategies according to the environment and agent needs. In research by Tonghao Wang [36] on experience sharing-based memetic transfer learning in MARL (MeTL-ES), a scalable approach was demonstrated where agents share implicit knowledge without the computational burden of traditional action-advising methods. Furthermore, in the work of Songyang Han [37], information sharing in MARL was utilized to enhance the safety and efficiency of connected autonomous vehicles (CAVs) through innovative techniques such as truncated Q-functions and safe action mapping. These studies all highlight

the importance of knowledge transfer mechanisms in MARL, providing the basis for our approach to advance more advanced techniques.

Our contributions are delineated as follows:

- Collaborative control strategy: Unlike existing approaches, our work introduces a novel multi-agent deep reinforcement learning algorithm specifically designed for UAV formations. This strategy focuses on the coordination of UAV groups in cooperative rounding-up tasks, aiming to enhance strategic control capabilities in complex battlefield environments. The innovative aspect of our approach lies in its ability to optimize the efficiency of collaborative operations within UAV groups, which has not been fully addressed in the literature.
- Shared experience data pool: We have developed a shared experience data pool that enables each UAV to not only make independent decisions and analyses but also learn from the experiences of the entire UAV swarm. This shared mechanism allows each UAV to leverage the operational experiences of others, continuously refining its own behavioral strategies. This approach significantly enhances the UAV system's adaptability in dynamic environments, enabling it to respond more intelligently to complex scenarios, marking a substantial advancement over traditional methods.
- Innovative reward function: Our work also introduces an innovative reward function tailored to the challenges faced by UAV swarms in diverse operational scenarios. This reward function is specifically designed to optimize UAV swarm performance, ensuring successful mission completion within a formation. The novelty of our approach lies in its customization of the reward structure to account for the complexities of various situations, which we believe sets our work apart from other reinforcement learning-based UAV decision-making approaches.

3. Preliminaries

3.1. Problem Scenario

In this paper, we intend to solve the problem of cooperative obstacle avoidance and round up of UAV swarms. This task can be divided into UAV formation coordination, UAV swarm obstacle avoidance, and UAV swarm rounding-up.

UAV swarm coordination is an effective approach for multiple drones to collaboratively and efficiently accomplish tasks. This relies on utilizing cameras and sensors within the drone cluster to gather crucial information, such as the position and velocity of nearby aircrafts. By maintaining a safe distance from each other, the swarm can navigate seamlessly during mission execution. Collaborative drone swarms improve task efficiency, enhance system reliability, and meet mission requirements in complex environments. In addition to collaborative tasks, drone swarms also need to consider obstacle avoidance to ensure safe navigation in challenging environments. This strategy enables drones to detect and evade obstacles while maintaining a safe distance, thereby enhancing the safety and efficiency of the drone swarm. Furthermore, drone swarms effectively encircle and capture target objects during pursuit missions. This is particularly useful for tracking and capturing dynamic and static moving targets, such as escaping vehicles, pedestrians, or other drones. Applying these capabilities to various domains, including security surveillance, border patrol, and crime prevention, can significantly improve the success rate and efficiency of these operations.

Overall, cooperative obstacle avoidance and rounding-up of UAV formations enables UAV swarms to operate effectively in diverse and challenging environments. We expect to combine the UAV swarm control decision-making model to complete obstacle avoidance, tracking, and rounding-up tasks. To this end, we designed a schematic diagram of UAV cooperative obstacle avoidance and rounding-up in an urban scene, as shown in Figure 1. Among them, the overall task can be divided into two sub-tasks, as shown in Figure 2, where (a) is the formation reconstruction process of the UAV group after avoiding obstacles, and (b) is the process of the UAV group chasing and surrounding the target point.

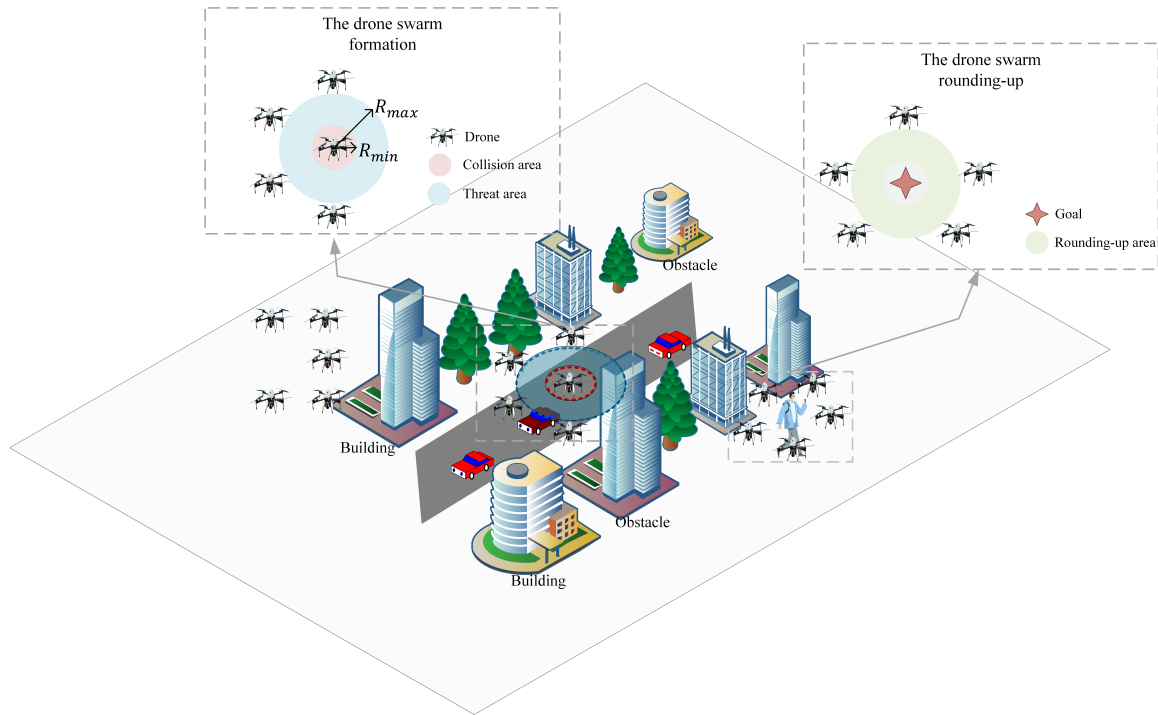


Figure 1. Problem scenario considered in this article.

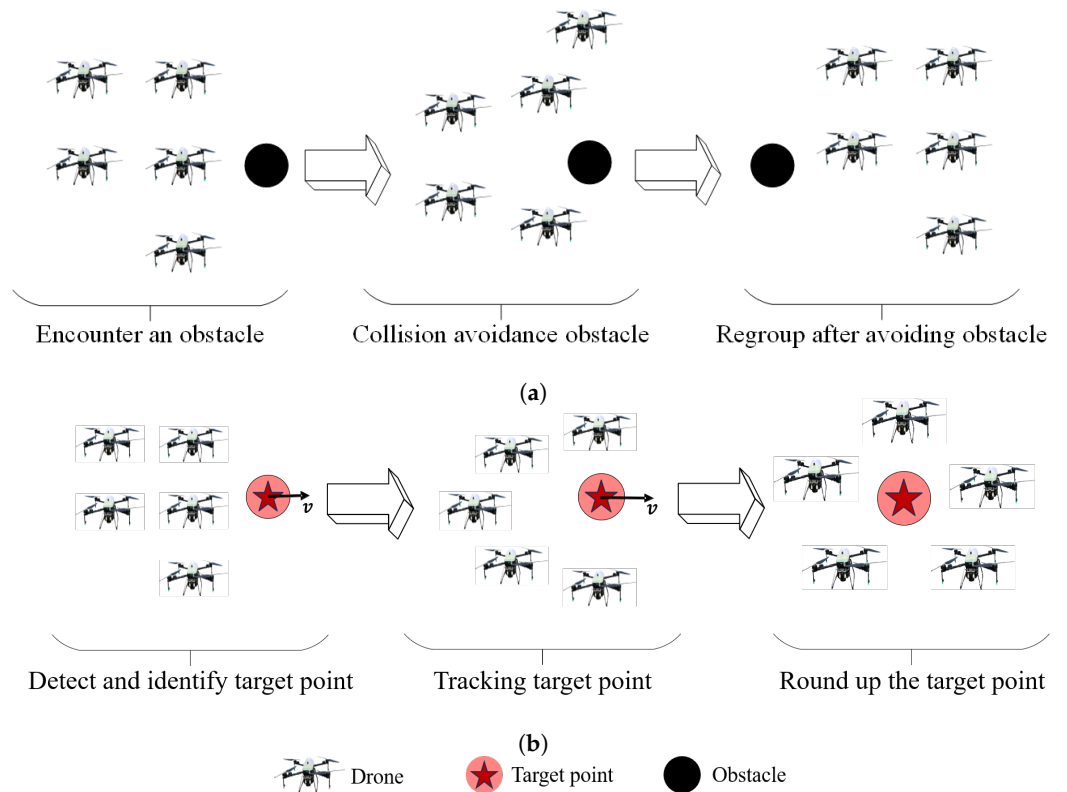


Figure 2. UAV swarm obstacle avoidance and rounding-up. (a) The UAV swarm in formation to avoid obstacles. (b) The UAV swarm chasing and rounding-up at the target point.

3.2. Problem Formulation

In this section, we present a comprehensive analysis of the problem of UAV swarm cooperative rounding-up, taking into account necessary assumptions and mitigating influential factors. Additionally, we provide a concise yet effective design of the experimental

scene. Furthermore, we establish a robust flight control model for the UAV, serving as a fundamental framework for subsequent experiments. By addressing these crucial aspects, we aim to lay a solid foundation for our research and facilitate a deeper understanding of the cooperative pursuit dynamics within UAV swarms. In this study, we address the challenge of coordinating a multi-UAV swarm to pursue a mission target point within an obstacle environment. In real-world mission scenarios, the drone swarm and the target point are subject to various external factors, such as wind and gravity. However, to ensure the research aligns with our objectives, we focus solely on the kinematic model of UAV speed and acceleration during the simulation process. By eliminating extraneous factors, we aim to achieve precise motion control of the UAVs. The speed of the UAV can be expressed as $v = (v_x, v_y, v_z)$, and the acceleration of the UAV can be expressed as $a = (a_x, a_y, a_z)$.

In this experiment, during the movement of the UAV swarm, we regard each UAV as a particle. The kinematic equation of UAVs in a three-dimensional space environment can be expressed as follows:

$$\begin{cases} X_{t+dt} = X_t + v_x * dt + 1/2 * a_x * dt^2 \\ Y_{t+dt} = Y_t + v_y * dt + 1/2 * a_y * dt^2 \\ Z_{t+dt} = Z_t + v_z * dt + 1/2 * a_z * dt^2 \end{cases} \quad (1)$$

In Equation (1), X_t, Y_t, Z_t represent the spatial coordinates of the UAV at time t , where X_t is the coordinate along the x -axis, Y_t is the coordinate along the y -axis, and Z_t is the coordinate along the z -axis. $X_{t+dt}, Y_{t+dt}, Z_{t+dt}$ represent the spatial coordinates of the UAV at time $t + dt$, indicating the new position after a time increment dt . v_x, v_y, v_z represent the velocity components of the UAV along the x -axis, y -axis, and z -axis at time t , respectively. a_x, a_y, a_z represent the acceleration components of the UAV along the x -axis, y -axis, and z -axis at time t , respectively. dt denotes the time increment, which is the elapsed time from t to $t + dt$. $1/2 * a_x * dt^2, 1/2 * a_y * dt^2, 1/2 * a_z * dt^2$ represent the displacement components caused by the accelerations a_x, a_y, a_z over the time increment dt . These are derived from the equations of motion for uniformly accelerated linear motion. The kinematic model of UAV is shown in Figure 3.

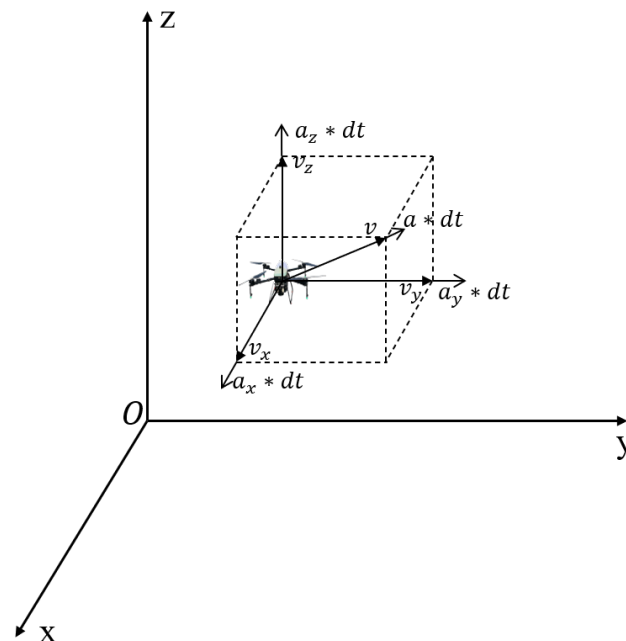


Figure 3. The kinematic model of the UAV.

To enhance the realism of our UAV model, we introduce a threat area concept, which establishes a radius around each UAV. This area serves as a boundary within which any approaching obstacle or aircraft is deemed a potential threat to the UAV. The threat area is defined by two parameters: the maximum threat distance, denoted as R_{max} , and the minimum threat distance, denoted as R_{min} . Figure 4 provides a visual representation of the threat area, illustrating its schematic diagram. By incorporating this concept, we aim to simulate real-world scenarios where UAVs must navigate and respond to potential threats in their surroundings.

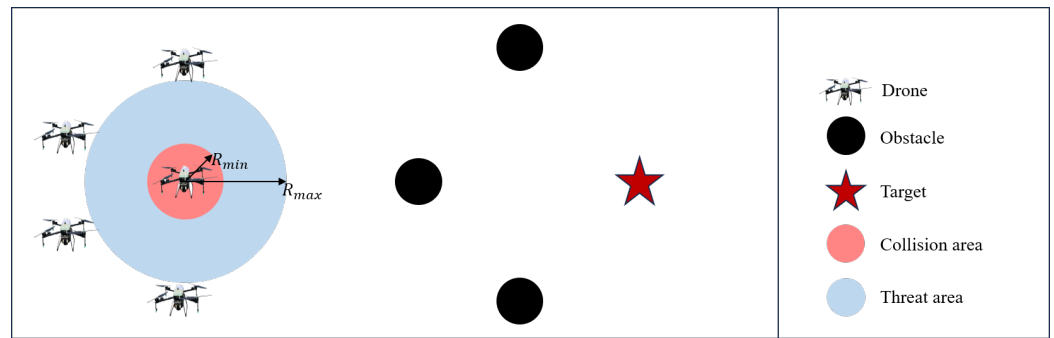


Figure 4. Visual representation of the threat area.

In our research, we introduce a critical distance parameter, denoted as R_{min} , which determines the collision threshold between the drone and any obstacle or approaching aircraft. If the distance between the drone and an object falls below R_{min} , it is considered a collision, resulting in the drone crashing. Additionally, we define a capture criterion based on the target point's position within the internal area of R_{max} . If the target point remains within this region for a significant duration, we conclude that the UAV swarm has successfully captured it. These criteria play a crucial role in evaluating the performance and effectiveness of the UAV swarm in pursuit missions.

In response to the above problem of cooperative obstacle avoidance and rounding-up of UAV groups, we define the following constraints:

$$\begin{cases} \|P_i - P_g\| \leq d_{goal}, i = 1, 2, \dots, N \\ R_{min} \leq \|P_i - P_j\| \leq R_{max}, i, j = 1, 2, \dots, N; i \neq j \\ \|P_i - P_{obs}\| \geq d_{min}^{obs}, i = 1, 2, \dots, N \\ |V_{uav}| \leq V_{max} \end{cases} \quad (2)$$

where $P_i = (x_i, y_i, z_i)$, $P_j = (x_j, y_j, z_j)$ represent the position of the drone; $P_g = (x_g, y_g, z_g)$ represents the position of the target point; $P_{obs} = (x_o, y_o, z_o)$ represents the position of the obstacle; we consider d_{goal} to be the target area of the drone; R_{min} and R_{max} define the safe area between drones; and d_{min}^{obs} means that the drone will not collide with obstacles. Finally, to consider drone safety issues, we also set an upper speed threshold V_{max} for the drones.

4. Approach

In this section, we provide a comprehensive analysis of approaches to solving the rounding-up problem in obstacle spaces by leveraging UAV formations. We delve into the details of deep reinforcement learning-based methods, providing a comprehensive description of each aspect. These include an introduction to the basic theory of reinforcement learning, the design of the swarm control decision model, the architecture of the training network, the configuration of the UAV swarm reward function, and the pseudocode for the implementation of our algorithm. By illuminating these key components, we aim to provide a comprehensive understanding of the proposed approach and its potential implications in solving the rounding-up problem.

4.1. Multi-Agent Deep Reinforcement Learning

Multi-agent deep reinforcement learning is a powerful approach to address the challenge of control decision-making for multiple agents in collaborative or competitive environments. Unlike traditional reinforcement learning, where agents operate independently and solely consider their own states and actions, multi-agent deep reinforcement learning enables agents to collaborate or compete with each other in order to achieve a shared objective or maximize their individual returns. This methodology is particularly relevant in real-world scenarios where agents must work together or engage in competition to accomplish common goals.

In the realm of multi-agent decision-making, the effective sharing of data among agents and the enhancement of overall decision-making performance are crucial. To achieve this, we propose the establishment of a comprehensive shared experience and information buffer, which undergoes continuous optimization strategies to improve its control capabilities. In complex environments, where multiple agents are involved, the state input can be represented as $s = [s_1, s_2, \dots, s_n]$, while the action output can be expressed as $a = [a_1, a_2, \dots, a_n]$, with n denoting the number of agents. Each agent is driven by a common task goal and aims to maximize its individual reward return. By executing the strategy $\pi_\theta(s_t|a_t)$, the state transitions from s_t to s_{t+1} , and each agent receives its own reward r . Consequently, the overall reward value $\sum r$ can be calculated. Our primary objective is to maximize the task reward income, expressed as $\max \sum (r|s, a)$.

4.2. Soft Actor–Critic Algorithm

The soft actor–critic algorithm (SAC) is a deep reinforcement learning algorithm that integrates the principles of policy gradient and Q-learning. By incorporating entropy regularization, SAC enhances the exploration and learning capabilities of the agent while ensuring policy stability. This algorithm serves as a powerful tool for addressing complex decision-making problems in various domains. Through the combination of policy gradient and Q-learning, SAC offers a comprehensive framework for optimizing the agent's behavior and achieving efficient policy convergence. The introduction of entropy regularization further enriches the algorithm's ability to explore the environment and adapt to dynamic scenarios. As a result, SAC stands as a significant advancement in the field of deep reinforcement learning, with promising applications in diverse domains.

The soft actor–critic (SAC) algorithm is a prominent approach in the field of reinforcement learning, aimed at learning the optimal action strategy by optimizing both the strategy and value function. The primary objective of the policy function optimization is to maximize the weighted sum of the expected return and entropy. This optimization goal is encapsulated in the objective function, which can be mathematically expressed as follows [38]:

$$J(\pi) = E_{\pi_\theta} \left[\sum_{t=0}^{T-1} r^t R(s_t, a_t) + \alpha H(\pi(\cdot|s_t)) \right] \quad (3)$$

The policy function is optimized to maximize the weighted sum of the expected return and entropy, striking a balance between exploration and exploitation. The expected return is computed using a value function estimator, while the entropy value serves to encourage exploration and prevent the strategy from becoming overly deterministic. By adjusting the weight of entropy, we can effectively control the trade-off between exploration and exploitation.

The optimization goal of the value function is to minimize the mean square error of the value function so that it can accurately estimate the value of the state–action pair. We define the value function to be parameterized by ψ , and we learn the training value function by minimizing the squared residual, which can be expressed as follows [38]:

$$J_V(\psi) = E_{s_t \sim D} \left[\frac{1}{2} \left(V_\psi(s_t) - E_{a_t \sim \pi_\phi} [Q_\theta(s_t, a_t) - \log \pi_\phi(a_t|s_t)] \right)^2 \right] \quad (4)$$

The soft actor–critic (SAC) algorithm incorporates two value function estimators to enhance learning ability and stability. The primary value function estimator is responsible for estimating the value function, while the secondary estimator is utilized for calculating the target value. This dual-Q network structure effectively reduces the estimation error of the value function, leading to improved performance. By employing two value function estimators, the SAC algorithm mitigates the overestimation bias commonly observed in single-estimator approaches. The main value function estimator is updated based on the temporal difference error, while the target value is computed using the secondary estimator. This separation of estimation and target calculation helps to stabilize the learning process and improve the accuracy of value function estimation. Furthermore, the SAC algorithm incorporates a target entropy regularization term to encourage exploration and prevent premature convergence. By maximizing the entropy of the policy distribution, the algorithm promotes exploration in the early stages of learning, facilitating the discovery of optimal action strategies.

Based on an understanding of the above formula, training the value function network needs to minimize the squared difference between the prediction of the value network and the expected prediction of the Q function. By deriving the above formula, we can obtain the following:

$$\hat{\nabla}_{\psi} J_V(\psi) = \nabla_{\psi} V_{\psi}(s_t)(V_{\psi}(s_t) - Q_{\theta}(s_t, a_t) + \log \pi_{\phi}(a_t | s_t)) \quad (5)$$

The training Q function is based on the following equation:

$$J_Q(\theta) = E_{(s_t, a_t) \sim D} \left[\frac{1}{2} (Q_{\theta}(s_t, a_t) - \hat{Q}(s_t, a_t))^2 \right] \quad (6)$$

where $\hat{Q}(s_t, a_t) = r(s_t, a_t) + \gamma E_{s_{t+1} \sim p} [V_{\bar{\psi}}(s_{t+1})]$. Bringing this into the above formula results in the following:

$$J_Q(\theta) = E_{(s_t, a_t) \sim D} \left[\frac{1}{2} Q(s_t, a_t) - r(s_t, a_t) + \gamma E_{s_{t+1} \sim p} [V_{\bar{\psi}}(s_{t+1})] \right] \quad (7)$$

In order to use the gradient strategy for training, we can derive the above formula as follows:

$$\hat{\nabla}_{\theta} J_Q(\theta) = \nabla_{\theta} Q_{\theta}(s_t, a_t)(Q_{\theta}(s_t, a_t) - r(s_t, a_t) - \gamma V_{\bar{\psi}}(s_{t+1})) \quad (8)$$

The training policy network is based on the following formula:

$$J_{\pi}(\phi) = E_{s_t \sim D, \epsilon_t \sim N} [\log \pi_{\phi}(f_{\phi}(\epsilon_t, s_t) | s_t) - Q_{\theta}(s_t, f_{\phi}(\epsilon_t, s_t))] \quad (9)$$

where $a_t = f_{\phi}(\epsilon_t, s_t)$. To train it using the gradient strategy, we differentiate it as follows:

$$\hat{\nabla}_{\phi} J_{\pi}(\phi) = \nabla_{\phi} \log \pi_{\phi}(a_t | s_t) + (\nabla_{a_t} \log \pi_{\phi}(a_t | s_t) - \nabla_{a_t} Q(s_t, a_t)) \nabla_{\phi} f_{\phi}(\epsilon_t, s_t) \quad (10)$$

During the training process, the stochastic gradient descent method is commonly employed to update both the policy network and the value function network. As the multi-agent interacts with the environment, the collected experience data are stored in a buffer. These data play a crucial role in calculating the loss values of the policy function and the value function. Subsequently, the network parameters are updated using the backpropagation algorithm, aiming to minimize the loss function. This iterative process facilitates the refinement and optimization of the policy and value function networks, enhancing the overall training performance.

4.3. Network Architecture

This paper presents a novel approach utilizing two deep neural networks (DNNs) to train actor networks and critic networks. The architecture of the networks is illustrated in Figure 5. In the actor network, the input data are passed through a four-layer neural

network with dimensions Linear [18, 128], Linear [128, 128], Linear [128, 64], and Linear [64, 2], along with the LeakyReLU activation function, to derive the corresponding strategic actions. On the other hand, the critic network processes the input data through a four-layer neural network with dimensions Linear [128, 128], Linear [128, 64], Linear [64, 32], and Linear [32, 1], along with the LeakyReLU activation function, to estimate the evaluation value of the corresponding state–action pair.

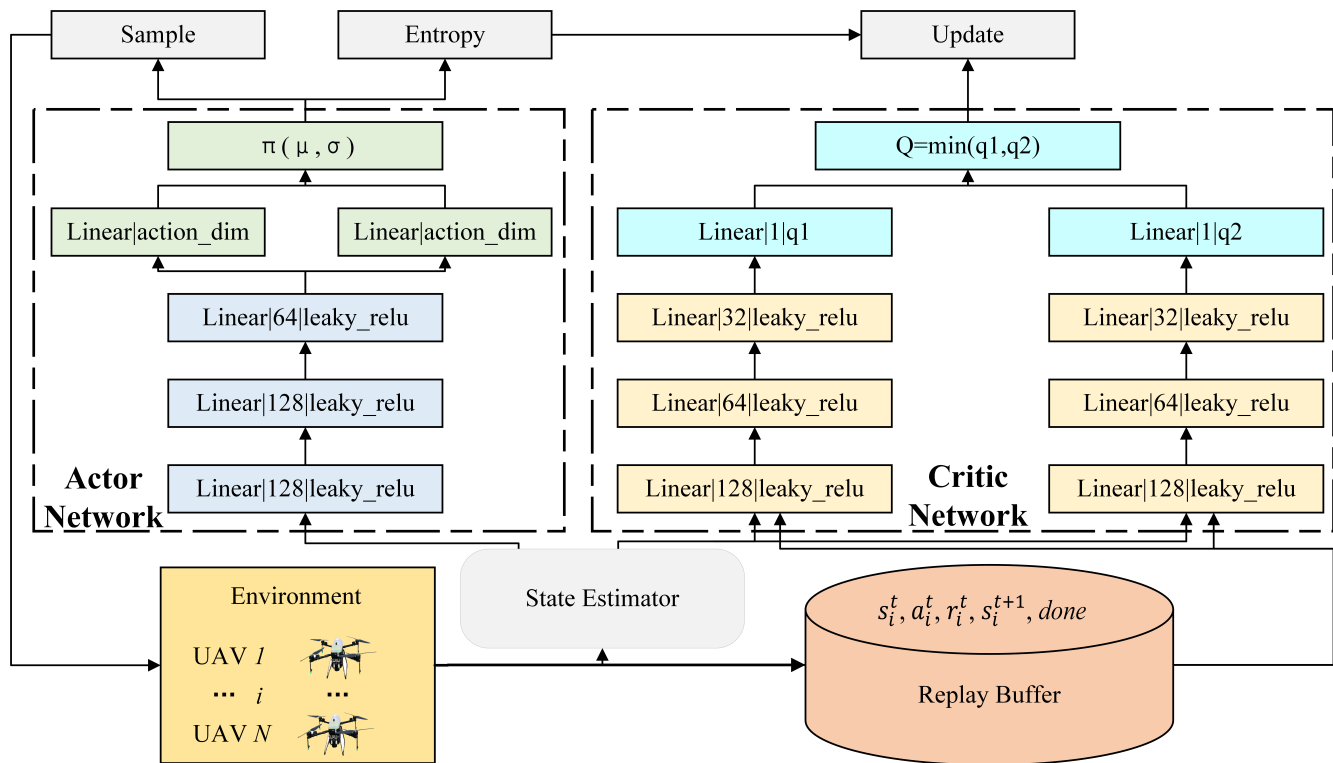


Figure 5. The architecture of the network.

In the actor network, empirical information observations obtained from the interaction between the UAV group and the environment are taken as input. The output of the actor network consists of a mean vector and a standard deviation vector. These vectors are used to determine a Gaussian distribution, representing the strategy $\pi(\mu, \sigma)$. Subsequently, an action is sampled from this distribution.

On the other hand, the critic network takes the status and actions of the drone swarm as input and outputs a value function that evaluates the current status and actions. The training objective of the critic network is to minimize the mean square error of the value function and reduce the discrepancy between the predicted value and the true value of the value function.

By simultaneously training the actor and critic networks using the SAC algorithm, policy learning and value function learning are implemented. At each time step, the actor network selects an action based on the Gaussian distribution generated by the mean and standard deviation vectors. The resulting interaction with the environment provides the next state and reward. The critic network evaluates the value of the next state, and the TD (temporal difference) error is computed. The parameters of the critic network are updated based on this error, improving the accuracy of the value function prediction. Additionally, the output of the critic network serves as the advantage function for the actor network. The actor network parameters are updated using the gradient of the advantage function, enhancing the strategy.

Overall, this research paper presents a comprehensive framework that combines actor and critic networks to optimize the decision-making process of a UAV group in

an interactive environment. The proposed approach demonstrates promising results in improving the performance and efficiency of the UAV group.

4.4. Reward Function

In order to assess the learning performance of drone swarms in the context of swarm rounding-up, it is crucial to establish a reliable measure of success. To address this, this research paper introduces a novel reward function that serves as an evaluation metric for the drones' learning capabilities.

The defined reward function takes into account various factors that contribute to the overall success of the drone swarm. These factors include the efficiency of the swarm in completing the rounding-up task, the accuracy of the drones' actions, and coordination and collaboration among the individual drones. By considering these key aspects, the reward function provides a comprehensive assessment of the learning performance of the drone swarm.

In the experimental setup, we defined the operating space as a 30×30 area, with each UAV having a radius of 1, a safety distance of 1.2, and a maximum speed of 2. Obstacles in the environment were also assigned a radius of 1. In the context of a rounding-up mission, drones play a crucial role in starting from a designated point, reaching the target location for rounding-up, and executing precise directional strikes. Throughout the itinerary of the drone group, several factors require careful consideration, including maintaining a proper formation, avoiding obstacles, and ensuring the successful completion of the rounding-up task. To effectively evaluate the performance of the drone group based on these aspects, a mixed reward function is proposed in this research paper:

$$r = r_g + r_a + r_f + r_s \quad (11)$$

where r_g represents the reward obtained by the UAV group reaching the rounding-up target position, r_a represents the reward obtained by the UAV group avoiding collision during the march, r_f represents the reward obtained by the UAV group maintaining the formation during the march, and r_s represents the reward obtained by controlling the speed of the drone swarm approaching the target position.

Based on the environmental parameter settings, we consider a UAV to have reached the target when it is within a distance of 2 from the target point. Additionally, we describe the UAV's progress toward the target using a negative scoring function. Furthermore, we define the UAV's heading as moving toward the target when the heading angle is less than 70° . The UAV's performance is considered optimal when the heading angle is less than 40° —this parameter was refined through continuous adjustments based on training outcomes. The reward r_g for the UAV swarm reaching the rounding-up target position consists of two parts. One part is the reward obtained when the position is closed, and the other part is the reward obtained when the angle towards the target position is correct, which can be expressed as follows:

$$r_g = r_{goal} + r_{angle} \quad (12)$$

where

$$r_{goal} = \begin{cases} -d_g/14 + 1 & d_g > 2 \\ +100 & d_g \leq 2 \end{cases} \quad (13)$$

$$r_{angle} = \begin{cases} 0 & angle \leq 40^\circ \\ -1.5 & 40^\circ < angle \leq 70^\circ \\ -3 & else \end{cases} \quad (14)$$

Considering the parameters of the UAVs and obstacles, we determined that if the edge-to-edge distance between UAVs, or between a UAV and an obstacle, is greater than 0.2—meaning the center-to-center distance between UAVs is greater than 2.2—then the UAVs are within a safe distance. Additionally, we consider that when the distance between

UAVs is less than 0.5, or the distance between a UAV and an obstacle is less than 1.2, the UAV is deemed to be at serious risk of damage, and a significant negative reward is applied.

The reward r_a obtained by the UAV swarm to avoid collision during its travel is set based on the relative distance d_o between the UAV swarm and the obstacle, which can be expressed as follows:

$$r_a = \begin{cases} -100 & d_o \leq 1.2 \\ -30 & 1.2 < d_o \leq 1.5 \\ -20 & 1.5 < d_o \leq 2.2 \\ 0 & \text{else} \end{cases} \quad (15)$$

The reward r_f obtained by the UAV swarm maintaining the formation during the march is based on the relative distance d_u within the UAV swarm, which can be expressed as follows:

$$r_f = \begin{cases} -50 & d_u \leq 0.5 \\ -8 & 0.5 < d_u \leq 1.3 \\ -4 & 1.3 < d_u \leq 2.2 \\ 0 & \text{else} \end{cases} \quad (16)$$

Regarding speed, our goal is to keep the drone close to its maximum speed while maintaining a certain margin, so we assign a positive reward when the speed is below 1.8 and a large negative reward when the speed exceeds 2.2.

The reward r_s obtained by the speed control of the UAV group approaching the target position is based on the speed v_u setting of the UAV when the UAV group approaches the target position, which can be expressed as follows:

$$r_s = \begin{cases} 0 & 0 < v_u \leq 1.8 \\ -2.0 & 1.8 < v_u \leq 2.2 \\ -6.5 & v_u > 2.2 \end{cases} \quad (17)$$

In this section, we propose the implementation of a mixed reward function to evaluate the performance of drone clusters in rounding-up missions. The reward function takes into account various factors such as obstacle collisions, aircraft collisions, and yaw flight, assigning a negative reward value in these instances, which ultimately leads to mission failure. However, this approach also serves the purpose of optimizing the overall strategy employed.

4.5. Swarm Control Algorithm

In this study, we employed the soft actor–critic (SAC) algorithm as the foundation for our control strategy. The pseudocode for the training algorithm is outlined as follows. Initially, the network undergoes a random initialization process, setting the starting and ending mission coordinates for the UAV formation. Subsequently, the process enters a loop where the drone utilizes visual information to make decisions and execute actions based on factors such as the proximity between adjacent drones, obstacles, and target points. These actions lead to transitions to the next state, accompanied by corresponding rewards. Furthermore, we store the previous state value, action value, reward value, current state value, and completion signal value in a cache space, which is then utilized for network training through value sampling. Notably, information sharing is facilitated among the drones within this system. The overall algorithm is described in Algorithm 1, providing a comprehensive overview of the methodology employed in this research.

Algorithm 1: Soft Actor–Critic (SAC) Algorithm for Multi-UAV Flocking Rounding-Up.

Input: Iterations MAX_EPISODE, MAX_STEP_SIZE;
 Initialize actor network π_θ and critic network q_1 and q_2 ;
for $episode=1$ to MAX_EPISODE **do**
 Initialize the initial and final position coordinates of each UAV;
 for $step = 1$ to MAX_STEP_SIZE **do**
 for each UAV ($e = 1$ to 5) **do**
 Calculate current observation s_t ;
 Determine if collision has occurred and if the destination has been reached;
 end
 Determine action a_t by executing π_θ ;
 Execute a in the environment;
 if the size of memory > 10000 **then**
 Randomly sample a batch of transitions $(s_t, a_t, r, s_{t+1}, d)$ from buffer;
 Compute targets for the Q functions;
 Update Q-functions (q_1 network and q_2 network) parameters;
 Update policy (π_θ network) weights;
 Adjust temperature α ;
 Update target network weights;
 for each UAV ($e = 1$ to 5) **do**
 Observe the next state s_{t+1} , the reward r , and the done signal d to indicate whether s_{t+1} is terminal;
 Store data $(s_t, a_t, r, s_{t+1}, d)$ in replay buffer;
 end
 if All reached **then**
 Initialize the initial and final position coordinates of each UAV;
 end
end
Output: Trained actor network π_θ

5. Experiment and Result Analysis

In this section, we present a comprehensive evaluation of our novel multi-UAV formation control, cooperative rounding-up, and obstacle avoidance methods. To assess the effectiveness of our approach, we conducted experiments using the Gazebo9 simulation environment. Gazebo is a highly advanced platform that offers a realistic simulation environment closely resembling the physical characteristics of real-world robots. In this evaluation, we focus on the performance of the controller trained using the soft actor-critic (SAC) algorithm and compare it with other state-of-the-art reinforcement learning algorithms. Through rigorous comparison and verification, we demonstrate that our proposed system outperforms existing methods. To further validate the functionality of our designed system, we devised two complex mission scenarios for evaluation: (1) fixed-point rounding-up, and (2) dynamic tracking and rounding-up. These scenarios provide a comprehensive assessment of the system's capabilities and showcase its potential in real-world applications.

Fixed-point rounding-up and strike can be explained as knowing that the target point needs to be hit and sending a drone group to round up and strike the target point. Dynamic tracking and rounding-up can be explained as knowing the movement trajectory of the target point to be hit and sending a group of drones to track and round up the moving target point.

5.1. Experimental Parameters and Hardware Settings

In our simulation environment, we configured a drone swarm consisting of five drones, each equipped with two simulated cameras: a grayscale camera and a stereo camera. The grayscale camera is capable of recording grayscale images at a resolution of 720×540 pixels, with a frame rate of 20 frames per second. On the other hand, the stereo cameras capture depth images at a resolution of 224×224 pixels, also at a frame rate of 20 Hz. These cameras work in conjunction with an onboard PyTorch-based end-to-end controller, which is responsible for controlling the drones. The drone swarm relies on visual perception to gather information about nearby drones and obstacles and utilizes obstacle avoidance and rounding-up actions to earn rewards. In the experimental setup, we trained the system within a 30×30 area ($\{(x,y) | -20 \leq x \leq 10, -20 \leq y \leq 10\}$). We deployed five UAVs, each with a radius of 1 m. To ensure collision avoidance, a safety distance of 1.2 m was maintained, and the maximum speed was capped at 2 m. Additionally, four cylindrical obstacles with a radius of 1 m each were placed within the environment. During the training process, the start and end points were randomly generated to enhance the system's adaptability and intelligence. For a detailed overview of the environmental parameters used in our experiments, please refer to Table 1.

Table 1. Environment parameter settings.

| Parameters | Value |
|-------------------------|--|
| Regional space | $\{(x,y) -20 \leq x \leq 10, -20 \leq y \leq 10\}$ |
| Number of obstacles | 4 |
| Radius of obstacle | 1 |
| Number of target points | 1 |
| Number of drones | 5 |
| Radius of the drone | 1 |
| Drone maximum speed | 2 |

In the experiment, the PPO algorithm, DDPG algorithm, and SAC algorithm were used to train the strategy. During the training process, batch size is the number of samples extracted from the experience pool during each training of the UAV, and discount rate represents the number of samples taken when calculating the loss value, or the discount on future rewards. The optimization algorithm is the Adam algorithm, the initial value of the entropy regularization coefficient is 1, the target entropy value is -2 , etc. The specific parameter settings are shown in Tables 2–4.

When training the policy model, we used the PyTorch framework to design the neural network model and optimizer. The CPU was Intel@Xeon Silver 4210R 2.4G, 10C/20T, and the GPU was NVIDIA RTX 3090-24G, which was used to accelerate the training of the policy network. The computer memory was 32G.

Table 2. SAC algorithm parameter settings.

| Parameters | Value |
|---------------------------|---------|
| Number of training rounds | 2000 |
| Batch size | 500 |
| Maximum number of steps | 600 |
| Experience pool size | 800,000 |
| Actor learning rate | 0.0001 |
| Evaluator learning rate | 0.0002 |
| α learning rate | 0.0001 |
| Init alpha | 1 |
| Target entropy | -2 |
| Discount rate | 0.98 |

Table 3. DDPG algorithm parameter settings.

| Parameters | Value |
|---------------------------|---------|
| Number of training rounds | 2000 |
| Batch size | 500 |
| Maximum number of steps | 600 |
| Experience pool size | 800,000 |
| Actor learning rate | 0.0001 |
| Evaluator learning rate | 0.0002 |
| α learning rate | 0.0001 |
| Init alpha | 1 |
| Target entropy | −2 |
| Discount rate | 0.98 |

Table 4. PPO algorithm parameter settings.

| Parameters | Value |
|---------------------------|--------|
| Number of training rounds | 2000 |
| Batch size | 500 |
| Maximum number of steps | 600 |
| Actor learning rate | 0.0001 |
| Evaluator learning rate | 0.0002 |
| Update times | 10 |
| Clipping value | 0.2 |
| Attenuation coefficient | 0.99 |
| Discount rate | 0.98 |

5.2. Results and Analysis

In this section, we present the configuration of our experimental task environment, which is based on the description provided in the previous section. We proceed to display and analyze the experimental results obtained from the drone swarm. The primary objective of the swarm is to learn and explore, with the aim of achieving formation, obstacle avoidance, and tracking and rounding-up capabilities. To facilitate the training process of the strategy, we devised a hybrid reward function that assigns appropriate rewards and penalties to the UAV swarm. Additionally, we conducted tests to evaluate the performance of the resulting policy in completing tasks within an obstacle environment.

5.2.1. Learning Curve

In this article, we utilize three currently mainstream reinforcement learning algorithms (SAC algorithm, DDPG algorithm, and PPO algorithm) to solve the drone swarm rounding-up problem. It should be noted that we ensured that the three algorithm training strategies achieved convergence. Based on the reward function mentioned in the previous article, we measured the rewards obtained by the five drones in each episode and drew a learning curve to evaluate whether the five drones could perform better in the current policy network. During the training process, the fluctuations in the reward function curve were normal. These fluctuations typically arise from the algorithm's exploratory phase, where various action combinations are attempted; the complexity of the tasks being tackled; the inherent randomness of the environment; and the non-linear nature of the problems being addressed. As shown in Figure 6, the polyline represents the cumulative rewards obtained by the five drones in the current mission space at the end of each episode of training, where (a) is the learning curve of fixed-point rounding-up, and (b) is the learning curve of dynamic rounding-up. As can be seen from Figure 6, under the same important parameter settings, whether it is a static tail scene or a dynamic rounding-up scene, the SAC algorithm can obtain higher rewards than the DDPG algorithm and the PPO algorithm.

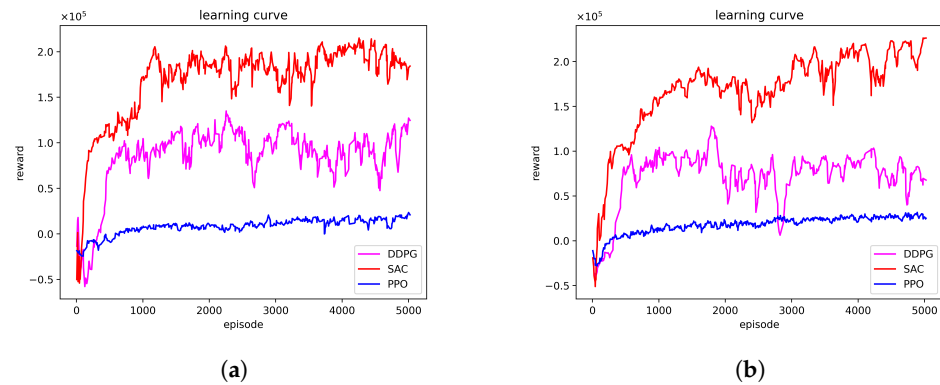


Figure 6. Learning curve of rounding-up. (a) The learning curve of fixed-point rounding-up. (b) The learning curve of dynamic rounding-up.

5.2.2. Simulation Results and Visual Analysis

In order to test the performance of each strategy, we saved the previously trained control strategy model, then carried out the UAV’s fixed-point rounding-up and dynamic tracking rounding-up and strike missions and analyzed its flight trajectory.

A. UAV Swarm Fixed-Point Rounding-up and Strike Mission Test

The UAV group-targeted rounding-up strike mission is a military action that uses a group composed of multiple UAVs to round up and strike specific targets. In this type of mission, using information from onboard sensors, the drone swarm will work together to round up the target in a specific area, prevent it from escaping or administering further damage, and then strike the target in order to destroy or neutralize the target.

The flight trajectory is shown in Figure 7. It can be seen from the figure that, in the beginning, there was a certain distance between the UAV group and the target point, and there were obstacles in the way. As our UAV group adjusted, it gradually approached the enemy and finally moved around the enemy’s strike point, maintaining a certain distance from the target point and rounding-up the target in a specific area.

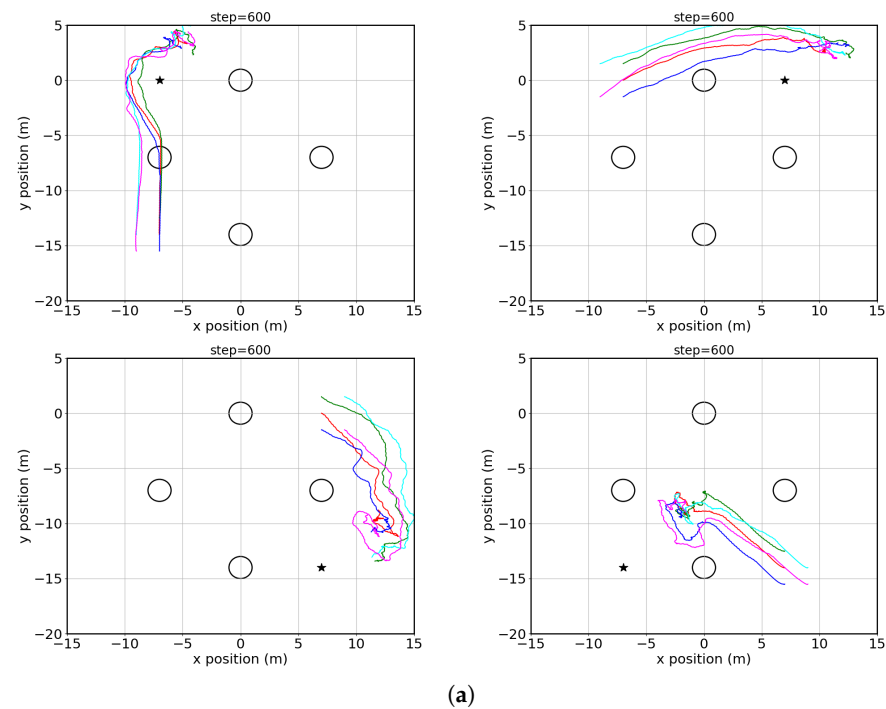


Figure 7. Cont.

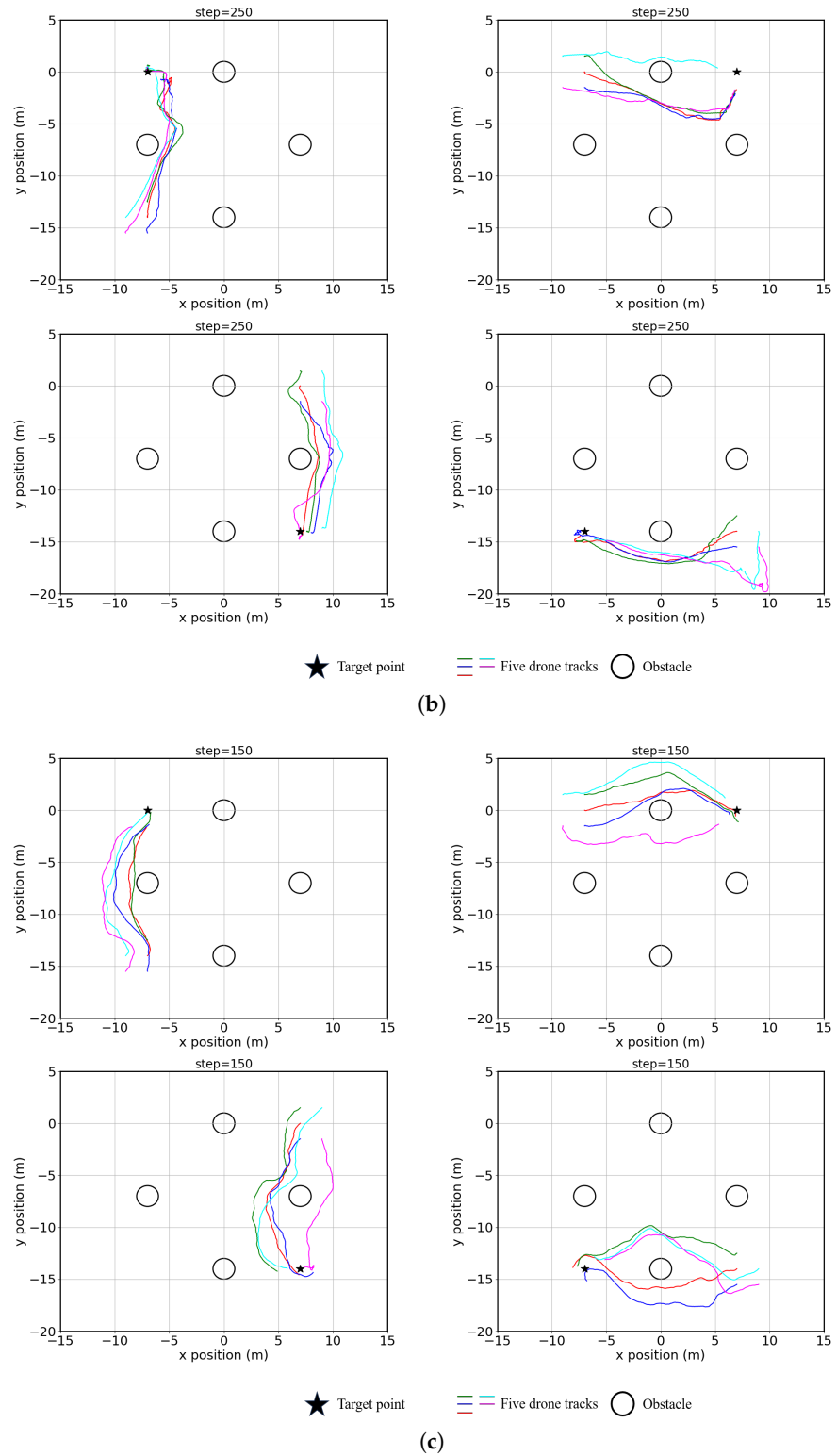


Figure 7. UAV swarm obstacle avoidance and fixed-point rounding-up test. (a) Test with PPO algorithm. (b) Test with DDPG algorithm. (c) Test with SAC algorithm.

Figure 7 illustrates the fixed-point search process of three algorithms across four directions. Panels (a), (b), and (c) show the motion trajectories of the PPO, DDPG, and SAC algorithms, respectively, in the upward, downward, leftward, and rightward directions. The PPO algorithm struggles with the fixed-point roundup task, demonstrating subpar performance. While the DDPG algorithm completes the task, it fails to produce optimal

movement trajectories. In contrast, the SAC algorithm excels in both obstacle avoidance and fixed-point rounding-up tasks, delivering consistently smooth and efficient trajectories.

B. UAV Swarm Dynamic Tracking, Rounding-up, and Strike Mission Test

UAV group dynamic tracking and rounding strike missions refer to military actions that use a group composed of multiple UAVs to track, round-up, and strike moving targets in real time. In this kind of mission, we have anticipated the enemy’s movement, and under the condition of avoiding obstacles, the drone swarm works together to round-up the dynamic target, force it to stop, or directly destroy it.

The flight trajectory is shown in Figure 8. It can be observed from the figure that with time, the UAV group constantly adjusts its direction and speed, gradually approaches the moving enemy target point, reduces the distance from the enemy target point, and finally successfully captures the enemy target.

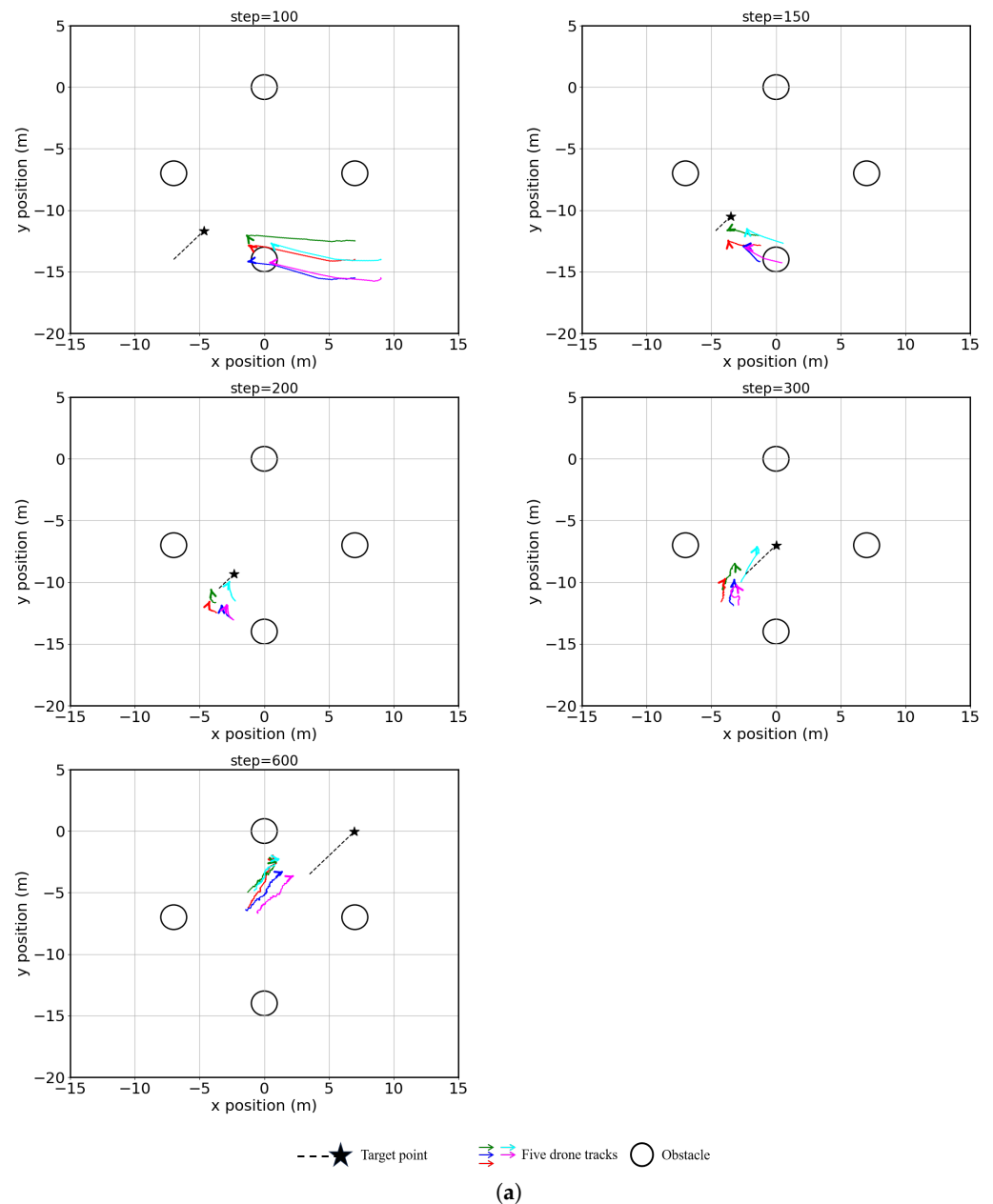


Figure 8. Cont.

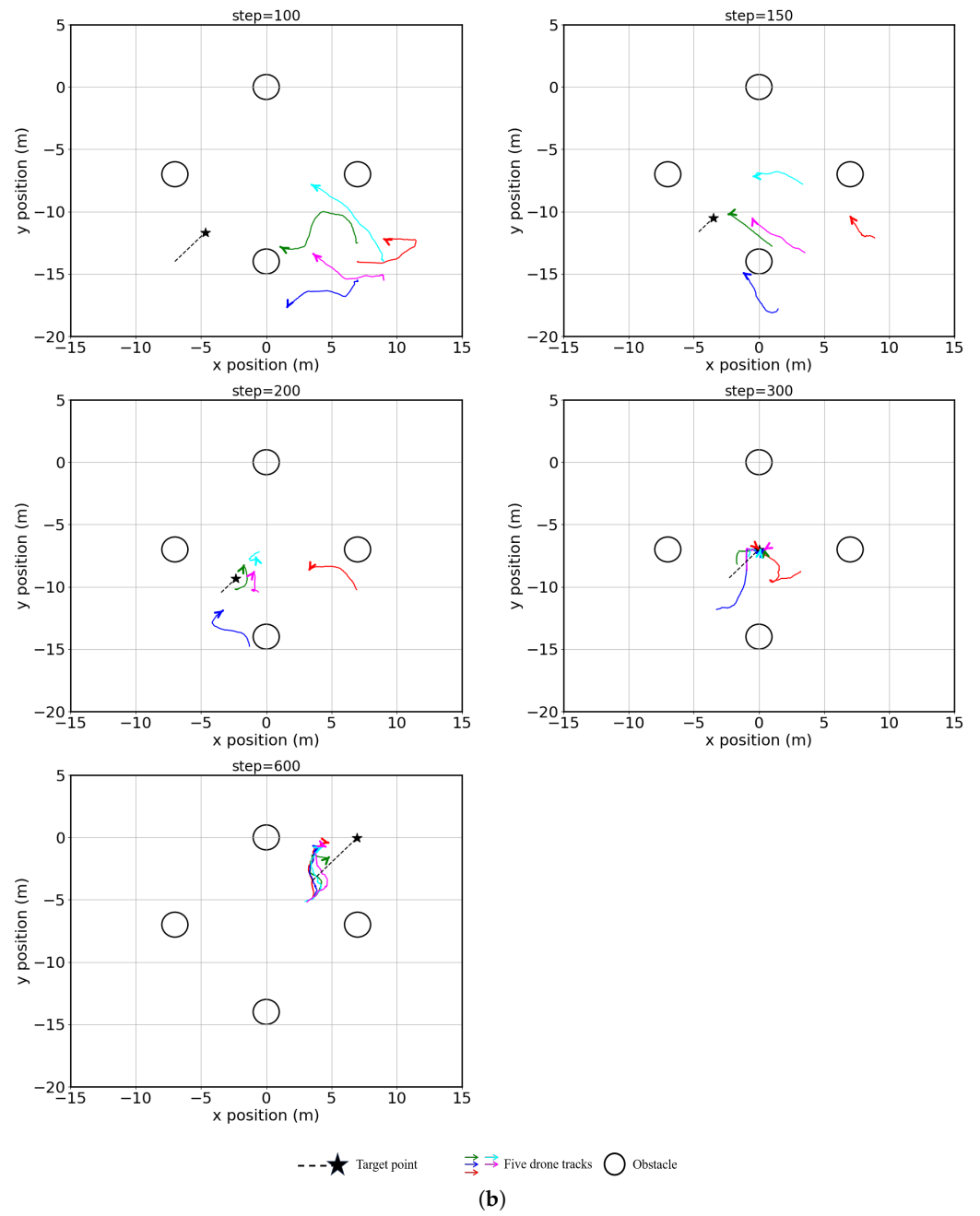


Figure 8. Cont.

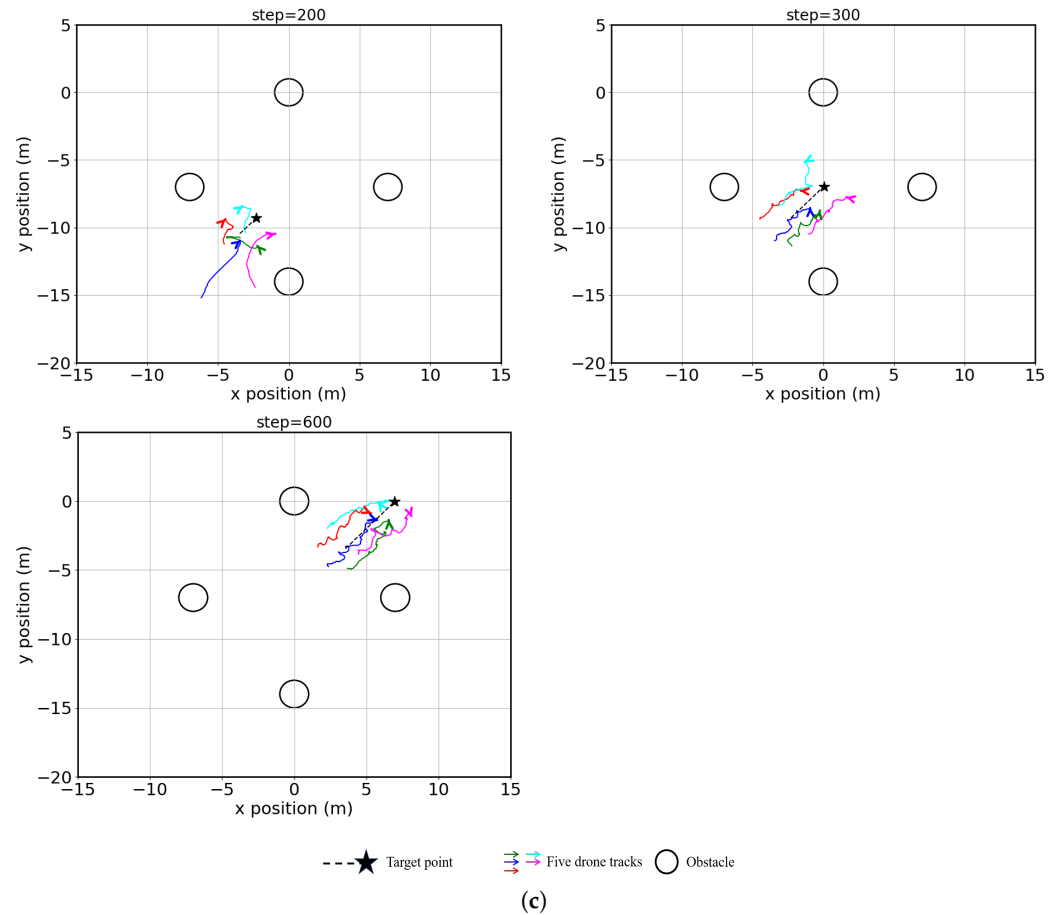


Figure 8. UAV swarm dynamic obstacle avoidance and rounding-up test with three algorithms (The scenes of the 100th, 150th, 200th, 300th, and 600th steps in the exercise process are shown, respectively). (a) Test with PPO algorithm. (b) Test with DDPG algorithm. (c) Test with SAC algorithm

Figure 8 shows the movement trajectories of five drones and three algorithms chasing the target point in the dynamic rounding-up scene, where (a) is the test trajectory diagram of the PPO algorithm, (b) is the test trajectory diagram of the DDPG algorithm, and (c) is the test trajectory diagram of the SAC algorithm. Three of the algorithms intercepted the trajectory state diagrams when step = 100, 150, 200, 300, and 600, respectively.

As can be seen in Figure 8, in the PPO algorithm, the UAV swarm cannot successfully bypass obstacles and collide with obstacles. Although there is a tendency to move closer to the target point, it cannot successfully complete the dynamic round-up task. In the DDPG algorithm, the UAV swarm can successfully bypass obstacles and complete the round-up task, but the internal distance of the UAV is not maintained very well, and there is a risk of collision. In the SAC algorithm, the UAV swarm can complete the obstacle avoidance and round-up tasks very well, and the obstacle avoidance and round-up trajectories are also more in line with the ideal state.

To summarize, by testing the effects of the three algorithms in two rounding-up scenarios, we found that compared to the PPO algorithm and DDPG algorithm, the SAC algorithm can complete these tasks better. The SAC algorithm can be completed faster and better in both fixed-point rounding-up tasks and dynamic rounding-up tasks, while PPO is not suitable for these tasks.

5.3. Evaluation Metrics

To assess the efficacy of the three algorithms in both fixed-point and dynamic round-up tasks, we devised three key performance indicators: efficiency, safety, and energy

consumption. By leveraging these indicators, we aim to discern which algorithm is better suited for the specific demands of this application scenario.

5.3.1. Efficiency Assessment

In terms of efficiency, we measured the number of steps taken by various algorithms to complete the two tasks, with each step set at 0.05 s. A shorter acquisition time indicates that the algorithm is capable of quickly identifying and rounding-up the target. Among them, we believe that a distance of 2 m from the target point represents a successful rounding-up.

Figure 9a–c represent the changes in the distance of the drone swarm to the target point over time in the fixed-point rounding-up mission of the three algorithms, respectively. It can be observed that the three algorithms are all approaching the mission target point, but the UAV under the PPO algorithm converges prematurely and has not yet fully reached the rounding-up range. Both the DDPG and SAC algorithms can complete this task, but judging from the number of running steps, DDPG requires about 156 steps to complete, while SAC only takes 130 steps to complete. From this point of view, the SAC algorithm is better than the DDPG algorithm. Figure 9d–f represent the changes in the distance of the drone swarm to the target point over time in the dynamic rounding-up task of the three algorithms, respectively. It can be observed that while all three algorithms tend to complete the task, the PPO algorithm fails to maintain the surrounding state after approaching the target point. This inability to closely follow the mission point ultimately leads to mission failure. Both the DDPG algorithm and the SAC algorithm can complete this task, but in terms of rounding-up stability and the number of steps completed, the DDPG algorithm takes about 248 steps to complete, and the distance remains unstable, while the SAC algorithm only takes about 130 steps to complete, and the distance remains stable. From this point of view, the SAC algorithm has a better performance for both fixed-point rounding-up tasks and dynamic rounding-up tasks.

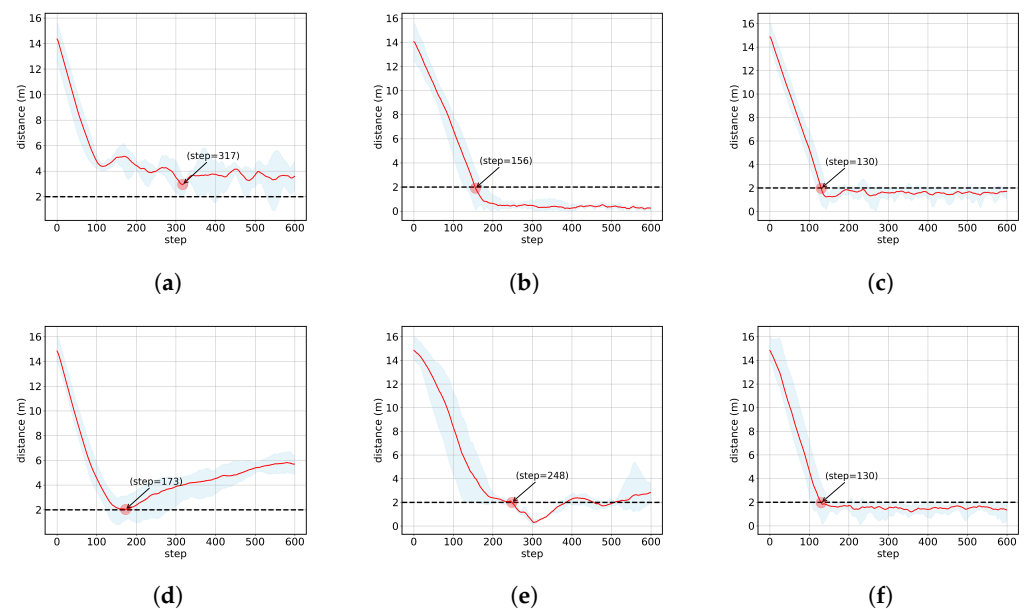


Figure 9. Efficiency assessment. (a) Test fixed-point rounding-up with PPO. (b) Test fixed-point rounding-up with DDPG. (c) Test fixed-point rounding-up with SAC. (d) Test dynamic rounding-up with PPO. (e) Test dynamic rounding-up with DDPG. (f) Test dynamic rounding-up with SAC.

5.3.2. Safety Assessment

In terms of safety, we measured and recorded the minimum distance between drones and obstacles, the minimum distance between drones, and the minimum average distance between drones throughout the mission. Reducing the risk of collision shows that the algorithm can effectively prevent drone collisions and ensure mission safety. In addition, a higher obstacle avoidance ability indicates that the algorithm can quickly identify and avoid obstacles, thereby

ensuring the safe flight of the drone. Among them, we consider a safe distance of 1.2 m between drones and obstacles and a safe distance of 0.5 m between drones.

Figure 10a,b represent the safety indicators of the three algorithms in fixed-point rounding-up tasks and dynamic rounding-up tasks: the minimum distance between drones relative to obstacles, the minimum distance between drones, and the average distance between drones, respectively, where (a) is a fixed-point roundup task. Both the DDPG algorithm and the SAC algorithm can achieve the purpose of safe obstacle avoidance and collision prevention, while the UAV collides with obstacles under the PPO algorithm. (b) represents the dynamic round-up task, in which only UAVs under the SAC algorithm can safely avoid obstacles. UAVs operating under both the DDPG algorithm and the PPO algorithm are at risk of colliding with obstacles or aircrafts.

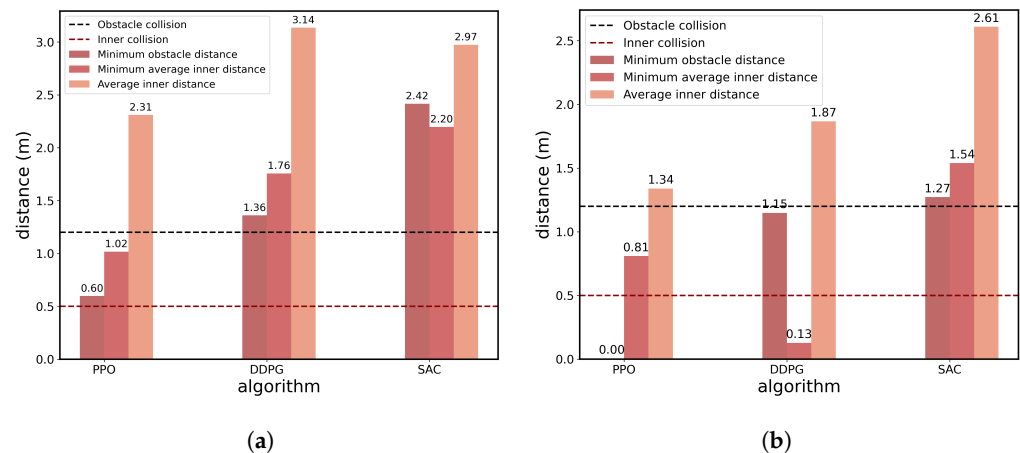


Figure 10. Safety assessment. (a) Test fixed-point rounding-up. (b) Test dynamic rounding-up.

5.3.3. Energy Consumption Assessment

In terms of energy consumption, we recorded the flight acceleration of the drone when using different methods to complete two tasks for evaluation. In terms of energy consumption, we recorded the average acceleration of the UAV swarm during flight when completing two tasks using different methods to evaluate the UAV energy consumption. We believe that the smaller the acceleration, the smoother the drone operates, and the smaller the energy consumption. A lower energy consumption means the algorithm can use the drone's energy more efficiently, extending the duration of the mission and the energy consumption of the drone. The smaller the acceleration, the smaller the energy consumption. A lower energy consumption means that the algorithm is able to use the drone's energy resources more efficiently, extending the duration of the mission.

Figure 11 represents the average acceleration of the UAV swarm for the two tasks controlled by the three algorithms obtained after running the strategy 20 times. Comparing the three algorithms, it is obvious that whenever there is a fixed-point rounding-up task or a dynamic rounding-up task, the average acceleration of the UAV group under the SAC algorithm is the smallest; that is, the energy consumption is lower than the other two algorithms.

Based on the analysis in Section 5.2 regarding learning rate and simulation visualization, as well as the various evaluation metrics discussed in Section 5.3, several conclusions can be drawn. First, the SAC algorithm demonstrates a significantly better learning rate compared to the DDPG and PPO algorithms. With the same critical parameter settings, SAC consistently achieves higher rewards in both static tail scenarios and dynamic rounding scenarios. Second, the trajectory visualization analysis reveals that the SAC algorithm produces smoother paths and more accurately reaches target positions, especially in dynamic scenarios, compared to DDPG and PPO. Third, the efficiency evaluation shows that SAC allows the UAVs to approach the target point more quickly and maintain a more stable formation around the target. In terms of safety, the SAC-based UAVs maintain safer

distances from neighboring UAVs and obstacles throughout the entire flight compared to DDPG and PPO. Lastly, the energy consumption assessment indicates that UAVs under the SAC algorithm exhibit the lowest average acceleration, implying lower energy consumption compared to the other two algorithms. The comparison results of the three result in two different outcomes after 20 repeated test run scenarios, as shown in Table 5.

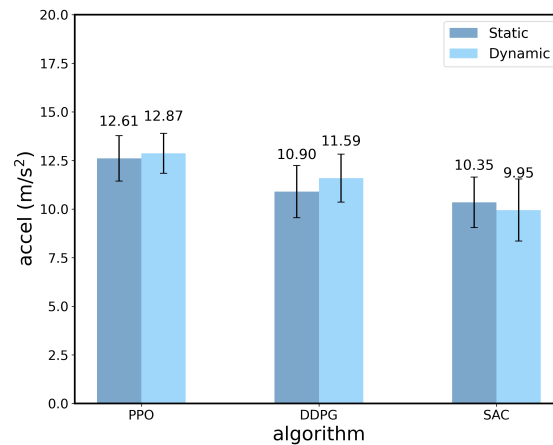


Figure 11. Energy consumption assessment.

Table 5. Comparison results of three results in two different scenarios.

| Senarios | Method | Efficiency | Safety—Obstacle | Safety—Inner | Consumption | Success Rate |
|----------|--------|------------|-----------------|--------------|--------------|--------------|
| Static | SAC | 153 ± 35 | 2.42 ± 0.82 | 2.20 ± 0.28 | 10.35 ± 1.30 | 100% |
| | DDPG | 221 ± 123 | 1.36 ± 1.41 | 1.76 ± 0.47 | 10.90 ± 1.34 | 75% |
| | PPO | / | 0.60 ± 0.16 | 1.02 ± 0.27 | 12.61 ± 1.17 | 0 |
| Dynamic | SAC | 147 ± 16 | 1.27 ± 0.39 | 1.54 ± 0.28 | 9.95 ± 1.60 | 100% |
| | DDPG | 238 ± 33 | 1.15 ± 0.58 | 0.13 ± 0.06 | 11.59 ± 1.23 | 100% |
| | PPO | 157 ± 10 | 0.00 ± 0.10 | 0.81 ± 0.36 | 12.87 ± 1.03 | 50% |

"/" indicates that the UAV did not reach the target point.

In the table above, we have summarized and compared the performance of the three algorithms across two different scenarios. The comparison includes the number of steps required to reach the target point (efficiency); the minimum distance between UAVs and obstacles, as well as between UAVs themselves (safety); the average acceleration during UAV operation (consumption); and the overall success rate.

In conclusion, UAV swarms utilizing the SAC algorithm can effectively accomplish obstacle avoidance and encirclement tasks, demonstrating superior performance in both static and dynamic scenarios. In contrast, the DDPG and PPO algorithms show relatively poor performances, making them less suitable for handling high-complexity tasks.

6. Conclusions

This research paper proposes a new method using multi-agent deep reinforcement learning to solve the challenge of collaborative obstacle avoidance and round-up of drone swarms, enabling multiple drones to exhibit enhanced collaborative round-up capabilities and higher intelligence. Through a series of experiments, we evaluate the effectiveness of the policy model trained using reinforcement learning by analyzing the average reward function curve and drone-tracking graph. Second, we propose three performance evaluation metrics to discern which algorithm is more suitable for the specific needs of this application scenario. The feasibility and effectiveness of the proposed multi-agent deep reinforcement learning method were verified based on a large number of simulations. Experimental and simulation results show that the method based on multi-agent deep reinforcement learning achieves good results in the rounding-up scenario and successfully

achieves the goal of using multiple drones to capture target points. However, it is important to note that the current scenario is relatively simple.

In our research, we utilized Gazebo to establish a robust and versatile simulation environment that closely mirrors real-world UAV dynamics, providing a strong foundation for validating our theoretical approaches and enabling potential real-world applications. While the direct transfer from simulation to reality, as demonstrated in previous studies, is promising, we recognize Gazebo's inherent limitations, particularly its idealization of environmental complexity and simplification of hardware constraints. These factors must be carefully addressed during the transition from simulation to real-world implementation to ensure the effectiveness and reliability of UAV systems. Moving forward, we plan to extend our research by conducting real-world UAV flight experiments.

Author Contributions: Conceptualization, Z.Z., Y.W., and Y.C.; methodology, Z.Z.; software, Z.Z. and Y.W.; validation, Z.Z., Y.W., and Y.C.; formal analysis, Z.Z., Y.W., and Y.C.; investigation, Z.Z., Y.W., and Y.C.; resources, Z.Z., Y.W., and Y.C.; data curation, Z.Z., Y.W., and Y.C.; writing—original draft preparation, Z.Z.; writing—review and editing, Z.Z.; visualization, Z.Z., Y.W., and Y.C.; supervision, Y.W. and Y.C.; project administration, Y.W. and Y.C.; funding acquisition, Y.W. and Y.C. 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 number 62073330, 52072408.

Data Availability Statement: The data are unavailable due to privacy or ethical restrictions.

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

References

- Oh, D.; Han, J. Smart search system of autonomous flight UAVs for disaster rescue. *Sensors* **2021**, *21*, 6810. [[CrossRef](#)]
- Bejaoui, A.; Park, K.H.; Alouini, M.S. A QoS-oriented trajectory optimization in swarming unmanned-aerial-vehicles communications. *IEEE Wirel. Commun. Lett.* **2020**, *9*, 791–794. [[CrossRef](#)]
- Liu, S.; Mohta, K.; Shen, S.; Kumar, V. Towards collaborative mapping and exploration using multiple micro aerial robots. In *Experimental Robotics: The 14th International Symposium on Experimental Robotics*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 865–878.
- Huang, Y.; Han, H.; Zhang, B.; Su, X.; Gong, Z. Supply distribution center planning in UAV-based logistics networks for post-disaster supply delivery. In Proceedings of the 2020 IEEE International Conference on E-health Networking, Application & Services (HEALTHCOM), Shenzhen, China, 1–2 March 2021; pp. 1–6.
- Ju, C.; Son, H.I. A distributed swarm control for an agricultural multiple unmanned aerial vehicle system. *Proc. Inst. Mech. Eng. Part I J. Syst. Control. Eng.* **2019**, *233*, 1298–1308. [[CrossRef](#)]
- Yan, C.; Xiang, X.; Wang, C. Towards real-time path planning through deep reinforcement learning for a UAV in dynamic environments. *J. Intell. Robot. Syst.* **2020**, *98*, 297–309. [[CrossRef](#)]
- Huang, Z.; Yang, Z.; Krupani, R.; Şenbaşlar, B.; Batra, S.; Sukhatme, G.S. Collision avoidance and navigation for a quadrotor swarm using end-to-end deep reinforcement learning. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 300–306.
- Wu, J.; Huang, Z.; Huang, W.; Lv, C. Prioritized experience-based reinforcement learning with human guidance for autonomous driving. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *35*, 855–869. [[CrossRef](#)] [[PubMed](#)]
- Moon, J.; Papaioannou, S.; Laoudias, C.; Kolios, P.; Kim, S. Deep reinforcement learning multi-UAV trajectory control for target tracking. *IEEE Internet Things J.* **2021**, *8*, 15441–15455. [[CrossRef](#)]
- Ho, T.M.; Nguyen, K.K.; Cheriet, M. UAV control for wireless service provisioning in critical demand areas: A deep reinforcement learning approach. *IEEE Trans. Veh. Technol.* **2021**, *70*, 7138–7152. [[CrossRef](#)]
- Lopez, B.T.; How, J.P. Aggressive 3-D collision avoidance for high-speed navigation. In Proceedings of the ICRA 2017, Singapore, 29 May–3 June 2017; pp. 5759–5765.
- Florence, P.R.; Carter, J.; Ware, J.; Tedrake, R. Nanomap: Fast, uncertainty-aware proximity queries with lazy search over local 3D data. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 7631–7638.
- Florence, P.; Carter, J.; Tedrake, R. Integrated perception and control at high speed: Evaluating collision avoidance maneuvers without maps. In *Algorithmic Foundations of Robotics XII: Proceedings of the Twelfth Workshop on the Algorithmic Foundations of Robotics*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 304–319.
- Bucki, N.; Lee, J.; Mueller, M.W. Rectangular pyramid partitioning using integrated depth sensors (rappids): A fast planner for multicopter navigation. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4626–4633. [[CrossRef](#)]

15. Zhou, B.; Pan, J.; Gao, F.; Shen, S. Raptor: Robust and perception-aware trajectory replanning for quadrotor fast flight. *IEEE Trans. Robot.* **2021**, *37*, 1992–2009. [[CrossRef](#)]
16. Zhou, X.; Zhu, J.; Zhou, H.; Xu, C.; Gao, F. Ego-swarm: A fully autonomous and decentralized quadrotor swarm system in cluttered environments. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 4101–4107.
17. Srinivasa, N.R.N. Towards a Swarm of Agile Micro Quadrotors. *Auton. Robot.* **2013**, *35*, 287–300.
18. Preiss, J.A.; Honig, W.; Sukhatme, G.S.; Ayanian, N. CrazySwarm: A large nano-quadcopter swarm. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3299–3304.
19. Soria, E.; Schiano, F.; Floreano, D. Predictive control of aerial swarms in cluttered environments. *Nat. Mach. Intell.* **2021**, *3*, 545–554. [[CrossRef](#)]
20. Tordesillas, J.; How, J.P. MADER: Trajectory planner in multiagent and dynamic environments. *IEEE Trans. Robot.* **2021**, *38*, 463–476. [[CrossRef](#)]
21. Yu, F.; Zhang, X.; Li, Q. Determination of The Barrier in The Qualitatively Pursuit-evasion Differential Game. In Proceedings of the 2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC), Xiamen, China, 10–12 August 2018; pp. 1–6.
22. Khachumov, M.; Khachumov, V. Notes on the pursuit-evasion games between unmanned aerial vehicles operating in uncertain environments. In Proceedings of the 2021 International Conference Engineering and Telecommunication (En&T), Dolgoprudny, Russia, 24–25 November 2021; pp. 1–5.
23. Tong, B.; Liu, J.; Duan, H. Multi-UAV Interception Inspired by Harris' Hawks Cooperative Hunting Behavior. In Proceedings of the 2021 IEEE International Conference on Robotics and Biomimetics (ROBIO), Sanya, China, 27–31 December 2021; pp. 1656–1661.
24. Li, J.; Zhu, J.; Liu, Y.; Fu, X. Dynamic Evasive Strategy of UAV Swarm Active Interception. In *Proceedings of the 2021 5th Chinese Conference on Swarm Intelligence and Cooperative Control*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 251–260.
25. Jiang, L.; Wei, R.; Wang, D. UAVs rounding up inspired by communication multi-agent depth deterministic policy gradient. *Appl. Intell.* **2023**, *53*, 11474–11489. [[CrossRef](#)]
26. Mu, C.; Peng, J.; Sun, C. Hierarchical multiagent formation control scheme via actor-critic learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 8764–8777. [[CrossRef](#)] [[PubMed](#)]
27. Li, B.; Zhang, H.; He, P.; Wang, G.; Yue, K.; Neretin, E. Hierarchical Maneuver Decision Method Based on PG-Option for UAV Pursuit-Evasion Game. *Drones* **2023**, *7*, 449. [[CrossRef](#)]
28. Fu, X.; Zhu, J.; Wei, Z.; Wang, H.; Li, S. A UAV pursuit-evasion strategy based on DDPG and imitation learning. *Int. J. Aerosp. Eng.* **2022**, *2022*, 3139610. [[CrossRef](#)]
29. Xia, Q.; Li, P.; Shi, X.; Li, Q.; Cai, W. Research on Target Capturing of UAV Circumnavigation Formation Based on Deep Reinforcement Learning. In Proceedings of the 2022 International Conference on Autonomous Unmanned Systems (ICAUS 2022), Xi'an, China, 23–25 September 2022.
30. Zhang, R.; Zong, Q.; Zhang, X.; Dou, L.; Tian, B. Game of drones: Multi-uav pursuit-evasion game with online motion planning by deep reinforcement learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 7900–7909. [[CrossRef](#)]
31. Sun, Y.; Yan, C.; Lan, Z.; Lin, B.; Zhou, H.; Xiang, X. A Scalable Deep Reinforcement Learning Algorithm for Partially Observable Pursuit-Evasion Game. In Proceedings of the 2022 International Conference on Machine Learning, Cloud Computing and Intelligent Mining (MLCCIM), Xiamen, China, 5–7 August 2022; pp. 370–376.
32. Wang, Y.; Dong, L.; Sun, C. Cooperative control for multi-player pursuit-evasion games with reinforcement learning. *Neurocomputing* **2020**, *412*, 101–114. [[CrossRef](#)]
33. Sutton, R.S.; Barto, A.G. Reinforcement learning: An introduction. *Robotica* **1999**, *17*, 229–235. [[CrossRef](#)]
34. Tan, M. Multi-agent reinforcement learning: Independent vs. cooperative agents. In Proceedings of the Tenth International Conference on Machine Learning, Amherst, MA, USA, 27–29 June 1993; pp. 330–337.
35. Wang, T.; Peng, X.; Wang, T.; Liu, T.; Xu, D. Automated design of action advising trigger conditions for multiagent reinforcement learning: A genetic programming-based approach. *Swarm Evol. Comput.* **2024**, *85*, 101475. [[CrossRef](#)]
36. Wang, T.; Peng, X.; Jin, Y.; Xu, D. Experience Sharing Based Memetic Transfer Learning for Multiagent Reinforcement Learning. *Memetic Comput.* **2021**, *14*, 3–17. [[CrossRef](#)]
37. Han, S.; Zhou, S.; Wang, J.; Pepin, L.; Ding, C.; Fu, J.; Miao, F. A Multi-Agent Reinforcement Learning Approach for Safe and Efficient Behavior Planning of Connected Autonomous Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 3654–3670. [[CrossRef](#)]
38. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Proceedings of the International Conference on Machine Learning, PMLR, 2018, Stockholm, Sweden, 10–15 July 2018; pp. 1861–1870.

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.