drones

MDPI

Article

Collaborative Target Tracking Algorithm for Multi-Agent Based on MAPPO and BCTD

Yuebin Zhou [1,†], Yunling Yue [2,†], Bolun Yan [2], Linkun Li [2], Jinsheng Xiao [2,3,*] and Yuan Yao [1,*]

[1] College of Physical Science and Technology, Central China Normal University, Wuhan 430072, China; yuebin1118@mails.ccnu.edu.cn
[2] School of Electronic Information, Wuhan University, Wuhan 430072, China; yanbl1@whu.edu.cn (B.Y.); lilinkun@whu.edu.cn (L.L.)
[3] Pingyang Institute of Science and Technology Innovation, Wenzhou 325400, China
* Correspondence: xiaojs@whu.edu.cn (J.X.); yaoyuan@ccnu.edu.cn (Y.Y.)
† These authors contributed equally to this work.

Abstract

Target tracking is a representative task in multi-agent reinforcement learning (MARL), where agents must collaborate effectively in environments with dense obstacles, evasive targets, and high-dimensional observations—conditions that often lead to local optima and training inefficiencies. To address these challenges, this paper proposes a collaborative tracking algorithm for UAVs that integrates behavior cloning with temporal difference (BCTD) and multi-agent proximal policy optimization (MAPPO). Expert trajectories are generated using the artificial potential field (APF), followed by policy pre-training via behavior cloning and TD-based value optimization. MAPPO is then employed for dynamic fine-tuning, enhancing robustness and coordination. Experiments in a simulated environment show that the proposed MAPPO+BCTD framework outperforms MAPPO, QMIX, and MADDPG in success rate, convergence speed, and tracking efficiency. The proposed method effectively alleviates the local optimization problem of APF and the training inefficiency problem of RL, offering a scalable and reliable solution for dynamic multi-agent coordination.

Keywords: multi-drone collaborative target tracking; behavior cloning with temporal difference; artificial potential field; multi-agent proximal policy optimization

check for updates

Academic Editor: Pablo Rodríguez-Gonzálvez

Received: 11 June 2025
Revised: 15 July 2025
Accepted: 21 July 2025
Published: 24 July 2025

Citation: Zhou, Y.; Yue, Y.; Yan, B.; Li, L.; Xiao, J.; Yao, Y. Collaborative Target Tracking Algorithm for Multi-Agent Based on MAPPO and BCTD. *Drones* **2025**, *9*, 521. https://doi.org/10.3390/drones9080521

Copyright: © 2025 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

Target tracking in dynamic environments requires drones to efficiently locate and track moving targets while avoiding obstacles. Traditional methods, such as the APF [1], are effective in path planning and obstacle avoidance [2] but are prone to local optima, limiting their effectiveness in dynamic settings [3]. MARL methods [4], on the other hand, can optimize collaborative strategies but suffer from slow convergence and high computational consumption [5], especially as the number of agents increases [6].

In response to these challenges, this paper introduces a novel framework that integrates BCTD and MAPPO, which enhances both learning efficiency and stability by leveraging pre-training and fine-tuning strategies [7]. Taking into account the above problems encountered in the multi-drone tracking task [8], the proposed multistage collaborative tracking framework integrates expert guidance, imitation learning (IL) [9] pre-training, and RL [10] fine-tuning. First, the policy network is pre-trained via behavior cloning on expert data of the artificial potential field, enabling rapid acquisition of fundamental path

planning and obstacle avoidance capabilities. Subsequently, the potential field weights are dynamically adjusted during reinforcement learning fine-tuning, using the MAPPO [11] algorithm. A global reward signal, such as successful collaborative encirclement, is introduced to overcome local force balance and drive UAVs to bypass obstacles and continuously approach the target.

The MARL training process is often plagued by significant time and resource consumption. Traditional RL methods rely on agents exploring strategies solely through environmental interactions, necessitating numerous trial-and-error iterations to generate interaction data. This approach results in extended training periods and low sample efficiency. To address these challenges, this study employs BCTD to enhance the value network. This method enables drones to acquire skills in path planning and obstacle avoidance within dynamic environments, thereby streamlining subsequent agent fine-tuning processes.

A novel method was proposed in this study to enhance the operational efficiency of collaborative target tracking and encirclement tasks [12]. The key contributions are as follows: 1. The integration of APF with IL. Drones are trained to acquire path planning and obstacle avoidance skills within the artificial potential field framework by initially imitating a pre-trained strategy based on APF principles, followed by fine-tuning through MARL to improve task performance. 2. A methodology employing BCTD is introduced to concurrently optimize the policy and value network. During the behavior cloning phase, the value network is refined using the temporal difference method to enable drones to learn path planning and obstacle avoidance in a dynamic environment, thereby facilitating subsequent agent fine-tuning. Generalized advantage estimation (GAE) is utilized to rectify the bias of the value function, prevent policy oscillation, reduce the duration of policy exploration, and enhance both evaluation accuracy and training efficiency.

2. Related Work

Traditional methods like APF frequently result in agents being trapped in local optima, unable to escape the potential field's low points. Meanwhile, MARL suffers from low training efficiency and slow convergence [13]. To address these issues, various solutions have been proposed. Rubio et al. modified the APF function to mitigate the local optimum problem [14]. Pateria et al. [15] improved training efficiency by enhancing the experience replay mechanism and introduced a hierarchical reinforcement learning framework to accelerate training. Xiao et al. [16] aimed to reduce evaluation bias and instability by employing multiple critic networks. Despite advancements, existing methods still face limitations, such as policy oscillation and low sample utilization, which complicate the balance between rapid learning and policy robustness.In contrast, the MAPPO+BCTD algorithm introduced here markedly surpasses traditional benchmarks like APF, MADDPG, and QMIX, achieving a 100% tracking success rate, an average of 19.6 steps, and a convergence speed of 65,000 episodes. To address the local optimum issue inherent in the APF method, we integrate BCTD. By using APF to generate expert trajectories, the policy network is pre-trained to avoid local minima. The TD-based value optimization in BCTD further refines the policy, ensuring that agents can overcome local optima by learning from high-quality expert data, thus improving stability during reinforcement learning fine-tuning and enhances training efficiency to 7.7 times that of conventional RL (65,000 vs. 500,000 episodes), offering an efficient and robust solution for multi-agent collaborative tracking in dynamic environments. The subsequent sections detail the relevant research and applications of each technology.

2.1. Artificial Potential Field

The conventional APF method, commonly employed for path planning and obstacle avoidance, operates by simulating attractive and repulsive fields to direct agent motion. The attractive force is given by

$$F_{\text{att}} = k_{\text{att}} \cdot \left(p_{\text{target}} - p_{\text{agent}}\right) \tag{1}$$

where $k_{\text{att}} = 2.0$ is a positive constant, p_{agent} is the current position of the agent, and p_{target} is the position of the target. The repulsive force is

$$F_{\text{rep}} = \begin{cases} k_{\text{rep}} \cdot \left(\frac{1}{d(p_{\text{agent}}, p_{\text{obs}})} - \frac{1}{d_0}\right) \cdot \frac{1}{d(p_{\text{agent}}, p_{\text{obs}})^2} \cdot \left\|p_{\text{target}} - p_{\text{agent}}\right\|, & \text{if } d \geq d_0 \\ 0, & \text{if } d < d_0 \end{cases} \tag{2}$$

where $k_{\text{rep}} = 1.5$ is a positive constant, d is the distance to the obstacle, and $d_0 = 0.5$ is the influence radius. Whenever the distance between the agent and an obstacle is below a predefined threshold $d < d_0$, a repulsive force is exerted, thus enabling the avoidance of the obstacle. Conversely, no force is exerted when the separation distance exceeds the threshold $d \geq d_0$. These forces guide the agents' motion and are combined to form the overall potential field. The approach involves constructing a cohesive potential field, guiding the agent's movement toward the resultant force, which is a combination of the attractive force from the target and the repulsive force from the obstacles. However, the traditional APF method is susceptible to getting stuck in local potential field equilibria in densely cluttered obstacle environments [17]. Fan et al. [18] proposed enhancements to the conventional APF framework by introducing a threshold limit for target actions, a correction factor for repulsion field distance, and an additional force mechanism. Despite these modifications, the method still exhibits limited dynamic adaptability. In a different approach, Zhang et al. [19] integrated APF with reinforcement learning by utilizing APF to generate expert trajectories and subsequently refining the strategy through RL [20]. This study builds on this concept by leveraging the potential field function of the APF as expert knowledge in imitation learning (IL) to address the local optimization challenges of traditional APF, offering high-quality initialization strategies for subsequent reinforcement learning processes.

2.2. Imitation Learning

IL is effective in reducing the exploration cost of RL by mimicking expert behavior, particularly beneficial in intricate tasks [21]. Behavior cloning (BC), a common approach in IL, aligns with expert state-action mapping through supervised learning. However, it is susceptible to the compounding error issue [22], particularly evident in multi-agent settings where policy deviations can disrupt cooperation [23]. Addressing this challenge, Arora et al. [24] introduced inverse reinforcement learning (IRL), a technique that deduces the intrinsic reward mechanism by studying the expert's actions, facilitating the agent's optimization based on this reward structure. In parallel, Bhattacharyya et al. [25] devised an approach that merges generative adversarial imitation learning (GAIL) with generative adversarial networks. By concurrently training the generative and discriminative models adversarially, the agent can replicate the expert's behavior. This paper leverages behavior cloning for network pretraining, utilizing expert trajectories from APF to expedite initial tracking proficiency and optimizing action distribution matching through cross-entropy loss. In contrast to previous research, this study builds upon Xiao et al.'s approach [26] of improving tiny target recognition by purifying features in target detection. It integrates the time difference method to enhance the optimization of the value network, reduce

oscillations in policy fine-tuning, and enhance the adaptability of imitation learning in dynamic settings, differing from prior research.

2.3. Behavior Cloning with Temporal Difference

BCTD merges TD methods with BC to optimize policy and value networks simultaneously in IL. This approach improves strategy stability and training efficiency in multi-agent systems. TD learning, a prevalent method in reinforcement learning, updates the state-value function by minimizing TD errors, thereby guiding policy adjustments. Its primary advantage is its efficient online evaluation. Nevertheless, TD methods generally require a considerable amount of environmental data and, when used in isolation, are inadequate for IL. To overcome these limitations, previous research has investigated the integration of reinforcement learning mechanisms into IL. Perlaza et al. [27] introduced the dataset aggregate method (DAgger) to mitigate policy drift by iteratively incorporating expert decision making. However, DAgger faces challenges such as high computational demands, complex deployment, and frequent expert interaction, limiting its applicability in high-dimensional multi-agent contexts. Addressing these issues, this paper presents the BCTD method, which integrates TD learning into the behavioral cloning phase via "pseudo-temporal difference paths." During training, the policy network employs cross-entropy loss to align with expert actions while utilizing state-reward information from expert trajectories to compute the TD error, thereby refining the value network parameters. The introduction of GAE [28] for weighted accumulation of TD error reduces valuation variance and enhances the stability of value learning. Unlike traditional BC, the BCTD method not only benefits from the rapid convergence due to expert behavior guidance but also offers value evaluation for current strategy behavior quality. This approach mitigates training oscillations caused by policy drift and ensures precise value initialization for subsequent reinforcement learning phases, such as MAPPO fine-tuning, thereby improving training efficiency and policy robustness.

3. Problem Formulation

3.1. Multi-Robot Pursuit Problem

In this work, we consider a cooperative target tracking problem involving multiple UAV agents operating in a 2D environment with obstacles. The set of pursuers is denoted by $\mathbf{V} = \{1, 2, \cdots, N\}$, and the target as E. Each agent $i \in \mathbf{V}$ can observe the relative positions of nearby agents, obstacles, and the evader, but not the global state. Unlike traditional first-order models, all pursuers in this work are modeled as second-order point-mass systems, where the control input is acceleration, and the state includes both position and velocity. The continuous-time kinematic equations are

$$\begin{cases} \dot{\mathbf{p}}_i = \mathbf{v}_i \\ \dot{\mathbf{v}}_i = \mathbf{a}_i \end{cases} \tag{3}$$

where $\mathbf{p}_i$, $\mathbf{v}_i$, and $\mathbf{a}_i$ represent the position, velocity, and acceleration of pursuer i, respectively. In discrete time, the state transition is

$$\begin{cases} \mathbf{p}_{i,t+1} = \mathbf{p}_{i,t} + \mathbf{v}_{i,t} \cdot \Delta t + \frac{1}{2}\mathbf{a}_{i,t} \cdot \Delta t^2 \\ \mathbf{v}_{i,t+1} = \mathbf{v}_{i,t} + \mathbf{a}_{i,t} \cdot \Delta t \end{cases} \tag{4}$$

In this paper, the movement of the tracking target selects the acceleration direction according to the APF rule, and thus also follows the second-order point-mass systems.

The pursuit task is deemed successful when all of the following conditions are met at a predefined terminal time $\mathbf{t}_{\max}$:

- Capture condition: Each pursuer is within a capture radius d_c from the target:

$$d_{E,i}(t_{\max}) < d_c, \quad \forall i \in V \tag{5}$$

- Collision avoidance: All inter-agent distances must exceed twice the safety radius d_s, and each agent must maintain a minimum distance from any obstacle:

$$d_{i,j}(t) > 2d_s, \quad d_{o,i}(t) > d_s, \quad \forall i \neq j, \forall t \in [0, t_{\max}] \tag{6}$$

where $d_{E,i}(t)$ is the Euclidean distance between pursuer i and the evader; $d_{i,j}(t)$ is the distance between pursuers i and j; $d_{o,i}(t)$ is the distance between pursuer i and the nearest obstacle.

Given that each agent relies exclusively on its local observation o_i to make decisions, the aim is to develop a set of decentralized policies $\pi_i(a_i|o_i)$ that together meet the specified constraints and complete the pursuit task in the most efficient manner. The primary objective is to maximize the expected cumulative discounted reward as follows:

$$\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{t_{\max}} \gamma^t R(s_t, \mathbf{a}_t)\right] \tag{7}$$

where $\gamma \in (0,1)$ is the discount factor, and $\mathbf{a}_t = (a_{1,t}, a_{2,t}, \ldots, a_{N,t})$ is the joint action vector. This issue showcases the amalgamation of dynamic constraints, partial observability, and decentralized coordination. The incorporation of acceleration-based control and second-order dynamics enhances the realism and intricacy of the scenario, thereby escalating the requirements for policy stability and generalization.

3.2. Decentralized Partially Observable Markov Decision Process (Dec-POMDP)

The multi-UAV pursuit problem can be formulated as a partially observable Markov Game (POMG) that is represented by the tuple $\langle S, A, O, P, R, \gamma \rangle$, where S, A, O, P, R, γ denote the global state space, joint action space, local observation space, state transition probability, reward function, and discount factor, respectively. Each agent chooses an action based on its local observations, which are sampled from the global state, receives a reward, and collectively impacts the environment's state. The objective for each agent is to learn the optimal policy $\pi^*(a_i|o_i)$ that maximizes the cumulative rewards.

(1) State space S: The state $s \in S$ represents the global configuration of the environment, which includes the position and velocity of all drones, the position and velocity of the target, and the positions of static obstacles. The total three-drone system state is

$$S = \left[\left\{S_i^{\text{agent}}\right\}_{i=1}^{N}, \left\{S_j^{\text{target}}\right\}_{j=1}^{K}, \left\{S_k^{\text{obs}}\right\}_{k=1}^{M}\right] \tag{8}$$

where N, K, M represents the number of tracking drones, targets, and obstacles, respectively.

For each drone $i \in I$, the state consists of its 2D position $p_i = (x_i, y_i)$, velocity $v_i = (v_{i_x}, v_{i_y})$, agent radius R_i, maximum speed $v_i^{\max}$.

$$S_i^{\text{agent}} = [p_i, v_i, R_i, v_i^{\max}] \tag{9}$$

Similarly, the state of target is

$$S_j^{\text{target}} = \left[p_j^{\text{target}}, v_j^{\text{target}}, R_j^{\text{target}}\right] \tag{10}$$

where p_j^{target}, v_j^{target}, and R_j^{target} are the location, velocity, and radius of the tracked entity, respectively. Additionally, the state of obstacle k is

$$S_k^{\text{obs}} = \left[p_k^{\text{obs}}, R_k^{\text{obs}}\right] \tag{11}$$

where R_k^{obs} is the radius of the obstacle k.

(2) Observation space O_i: Due to limited sensing and communication, each agent i receives only partial observations O_i. The local observation of agent i is as follows:

$$O_i = \left[O_{i,self}, O_{i,UAV}, O_{i,obs}, O_{i,Tar}\right] \tag{12}$$

In the one-target scenario, the observation of targets is $O_{i,Tar} = O_{i,Tar1}$, where the observation O_{i,obs_j} includes the relative position $\Delta p_{i,Tarj}$ between the agent and the obstacle j, relative velocity $\Delta p_{i,Tarj}$, and its radius of size.

$$O_{i,Tar_j} = \left[\Delta p_{i,Tar_j}, \Delta v_{i,Tar_j}, R_{\text{target}}\right] \tag{13}$$

where $\Delta p_{i,Tarj} = p_{j,target} - p_i$ represents the relative position between the target and the agent. In the three-drone system, to consider cooperation between agents, all drones are included in the observation range.

$$O_{i,UAV} = \left[O_{i,UAV_1}, O_{i,UAV_j}, \ldots, O_{i,UAV_{N_1}}\right], \quad j \neq i \tag{14}$$

where $O_{i,UAV_j} = \left[\Delta p_{i,UAV_j}, \Delta v_{i,UAV_j}, R_{UAV}\right]$, and uses zero padding when the agent is invisible. The observation of an obstacle contains the position relative to the agent and the obstacle radius, so the observation of each visible obstacle j is expressed as follows:

$$O_{i,\text{Obs}} = \left[O_{i,Obs_1}, O_{i,Obs_j}, \ldots, O_{i,Obs_{N_2}}, \right] \tag{15}$$

Similarly, compose all obstacle observations into $O_{i,UAV_j} = \left[\Delta p_{i,Obs_j}, R_{UAV}\right]$.

Song et al. [29] demonstrated that sparse feature selection enhances decision-making efficiency in high-dimensional settings. In line with this principle, the study's observation space design focuses solely on essential interaction information (such as relative poses of agents/targets) to prevent irrelevant features from impacting policy learning.

(3) Action space A^i: The action of agent i is denoted as $a_i \in A_i$. A_i represents a set of feasible velocity vectors subject to fundamental kinematic equations and restrictions imposed by map boundaries, maximum velocity, and maximum acceleration limits.

$$a \in \left\{a_{static}, a_{up}, a_{down}, a_{left}, a_{right}, a_{ur}, a_{lr}, a_{ll}, a_{ul}\right\} \tag{16}$$

The values represent acceleration in nine directions: static, upward, downward, left, right, upper right, lower right, lower left, and upper left. This design provides directional precision for path adjustments and obstacle avoidance. The discrete action space simplifies policy learning compared to continuous spaces.

(4) State transition model P: The transition model $P(s_{t+1}|s_t, \mathbf{a}_t)$ governs how the global state evolves based on the joint action $\mathbf{a}_t = [a_1, \ldots, a_N]$ from all agents. Each UAV agent updates its velocity and position using Newtonian dynamics:

$$v_{i,t+1} = v_{i,t} + a_{i,t} \cdot \Delta t, \quad p_{i,t+1} = p_{i,t} + v_{i,t} \cdot \Delta t + \frac{1}{2} a_{i,t} \cdot \Delta t^2 \tag{17}$$

In the context of algorithm research, it is reasonable to assume a constant speed of action by the drone over the extremely short time interval Δt.

(5) The reward function $R^i(s, a)$ guides agent i by providing rewards for actions executed. It is a composite function combining dense and sparse components to promote effective and cooperative target tracking while discouraging unsafe or inefficient behaviors.

$$R(s, \mathbf{a}) = \sum_{i=1}^{N} \left(r_i^{\text{track}} + r_i^{\text{collide}} + r_i^{\text{step}} \right) + r^{\text{success}} \tag{18}$$

where $r_i^{\text{track}} = d_{i,t-1} - d_{i,t}$ measures the change in distance between an agent and a target over time, promoting movement towards the target. r_i^{collide} represents a penalty for collisions between agents or obstacles. This term encourages safe navigation and cooperative spacing:

$$r_i^{\text{collide}} = \begin{cases} -1, & \text{if collision occurs} \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

r_i^{step} is a step penalty that per timestep to penalize unnecessary wandering and incentivize task efficiency. It is typically linear with respect to the elapsed step ratio $r_i^{\text{step}} = -\frac{\text{current step}}{\text{max steps}}$. Additionally, r^{success} is a high-value sparse reward awarded only when all agents successfully surround the target within a predefined radius threshold. This strongly incentivizes cooperation and convergence as follows:

$$r^{\text{success}} = \begin{cases} +10, & \text{if capture condition is met} \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

The reward structure combines short-term behavior shaping with long-term goal completion to encourage individual approach and effective coordination in completing a cooperative task, while minimizing collisions and delays. The weights of the reward components (such as r^{track}, $r^{collide}$, r^{step}, and $r^{success}$) were manually tuned using a grid search approach. We iterated over a set of values for each weight and selected the combination that yielded the most efficient training process and the best tracking performance. This manual tuning allowed us to strike a balance between encouraging rapid target approach, minimizing collisions, and penalizing unnecessary movements, ensuring robust collaboration among agents.

4. Approach

In response to the issue of the traditional methods of artificial potential field being susceptible to local optima in complex scenarios [30], as well as the bottleneck of low training efficiency and slow convergence in MARL, this paper considers an innovative collaborative tracking framework. The key innovation lies in the architecture driven by expert experience in pre-training and fine-tuning architecture [31], along with the combined optimization mechanism based on BCTD. Specifically, this framework effectively integrates expert trajectories produced through APF, BC in IL, value estimation mechanism optimized by TD, and reinforcement policy optimization capability of MAPPO. The goal is to attain effective initialization and consistent iteration of the agent's strategy. The core process is structured as follows:

(1) Pre-training Guided by APF Expert Experience: Leverage expert trajectories from APF to pre-train the policy network, enabling rapid acquisition of fundamental tracking capabilities.

(2) BCTD Combined Optimization Mechanism: Employ BCTD to synchronize the optimization of both the policy and evaluation networks.

(3) MAPPO Dynamic Fine-tuning: Incorporate the MAPPO clipping mechanism along with a globally designed reward system to facilitate efficient multi-agent collaboration.

Figure 1 illustrates the algorithmic framework, which is systematically divided into three stages: "pre-training—combined optimization—dynamic adjustment," ensuring a balance between path planning reliability and reinforcement learning efficiency.

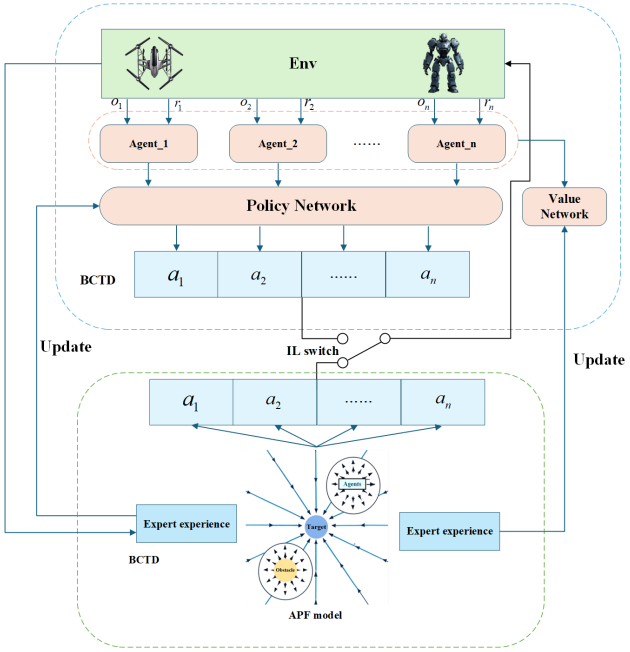


Figure 1. Algorithm framework.

In collaborative tracking and pursuit, the action space of the agent is usually represented as a set $\{a_1, a_2, \ldots, a_n\}$. The reinforcement learning policy is denoted as $\pi_\theta(a|s)$, where the input is the current state s and the output is a vector f with a dimension of $|A|$, which is the size of the action space. Each element of the vector f corresponds to the probability value of an action, indicating the probability of choosing that action.

Initially, the APF method generates expert demonstration trajectories, forming a high-quality dataset that captures interactions among agents, targets, and obstacles. At this stage, the policy network undergoes pre-training via BC based on rule-driven trajectories. This process allows the agent to swiftly acquire fundamental path planning and obstacle avoidance skills, thereby minimizing exploration costs and easing learning challenges in the subsequent pure reinforcement learning phase. The policy network, structured as a multi-classifier, outputs the probability distribution of each discrete action. Expert actions serve as the supervision signal, and the network is trained using the cross-entropy loss function.

$$\mathcal{L}_{\text{policy}}(\theta) = -\sum_t a_t \cdot \log \pi_\theta(a_t|s_t) \tag{21}$$

where a_t represents the expert actions in one-hot encoding, and π_θ is the policy network.

Then, to address the limitations of BC in strategy generalization and stability, we present the BCTD mechanism. Using the attention mechanism for cross-modal alignment as proposed by Xu et al. [32], the BCTD framework in this investigation accomplishes multiscale temporal feature integration by utilizing the GAE-weighted TD error, thus improving the robustness of value estimation. This approach simultaneously optimizes the policy and value networks during the imitation learning phase. Expert trajectories provide

state–action–reward sequences, which are used to form pseudo-environment interaction paths. The GAE is employed for estimating and propagating value returns. The temporal difference error is defined by the following equation:

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t) \tag{22}$$

where δ_t is the TD error at time step t, indicating the difference between the current valuation and the return after one step.The corresponding GAE estimate is as follows:

$$\hat{A}_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l} \tag{23}$$

where γ is the discount factor (set to 0.95 in our experiments). The GAE smooths the TD error across multiple timesteps to reduce variance. In our experiments, we used $\lambda = 0.9$, which allowed us to effectively balance bias and variance. For example, in one timestep, the TD error for an agent might be calculated as $\delta_t = 1.5$, and the GAE estimate $\hat{A}_t$ would sum over multiple δ_t values, weighted by γ and λ. Meanwhile, the value network aims to minimize the mean squared error. The primary advantage of BCTD is its ability to quantitatively evaluate and correct policy quality without adding to the environmental interaction burden. It mitigates training oscillations from early policy deviations, thereby enhancing convergence stability and the generalization capability of the policy during the imitation learning phase.

$$\mathcal{L}_{\text{value}}(\phi) = \frac{1}{2} \sum_t \left(V_\phi(s_t) - \hat{V}_t\right)^2 \tag{24}$$

Finally, building on pre-training, we utilize MAPPO for policy fine-tuning via RL. Employing a centralized training and distributed execution (CTDE) framework, MAPPO shares global value network information while maintaining policy independence, enabling efficient collaboration. Agents receive real-time feedback from the environment, including rewards for target approach, penalties for collisions and step counts, and rewards for successful encirclement. MAPPO mitigates policy update fluctuations through a clipping mechanism and, combined with the value baseline from BCTD pre-training, significantly improves the collaborative capacity and learning efficiency of the multi-agent system in dynamic settings.

The evaluation metrics for the experiments in this chapter include the tracking success rate at convergence, the number of steps required to complete the task, the average reward tested after model convergence, and the number of training episodes required for the model to converge.

In the three-UAV target tracking scenario, analysis of the agent observation space reveals that computational complexity grows linearly with the number of nearby drones and obstacles when each UAV independently executes this cooperative tracking algorithm. The MAPPO+BCTD algorithm achieves convergence at approximately 65 k episodes, significantly lower than alternative approaches such as standard MAPPO, which requires 500 k episodes to converge. On an NVIDIA GeForce GTX 3090 GPU, each episode requires approximately 35 ms of processing time, resulting in a total training duration of roughly 6.3 h. These results highlight the computational efficiency of the proposed method, which benefits from the pre-training phase. Meanwhile, we conducted experiments with parameters detailed in Table 1.

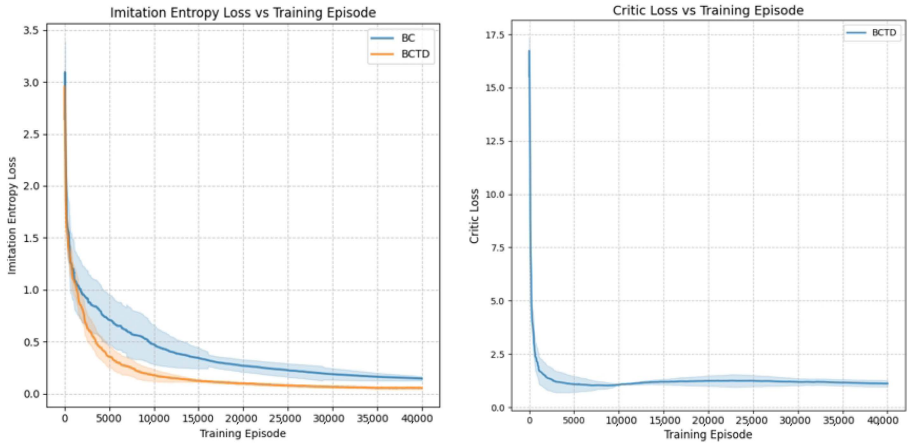
Table 1. The setting of simulation.

Objects	Properties	Descriptions
Computer	operating system	Linux Ubuntu 20.04
	CPU	i9-9900K
	memory	32GB
	GPU	NVIDIA GeForce GTX 3090
	video memory	24GB
Algorithm	learning rate	0.0001
	discount factor	0.95
	batch size	150
	hidden layer size	(100, 100, 100)
	loss function	Mean-Square Error (MSE)
	optimization method	Stochastic Gradient Descent (SGD)
Scenario Configuration	environment side Length	2.0
	agent radius	0.06
	target radius	0.04
	maximum speed (agent)	1.0
	maximum speed (target)	0.8

4.1. Three-UAV Target Tracking Test

The BCTD method is used to introduce the tracking ability of the APF into the tracking agents, allowing the agents to learn this ability. By using cross-entropy loss as the optimization function, the agent's policy model successfully learns the tracking ability of the APF model. Additionally, TD methods are used to achieve convergence of the value network.

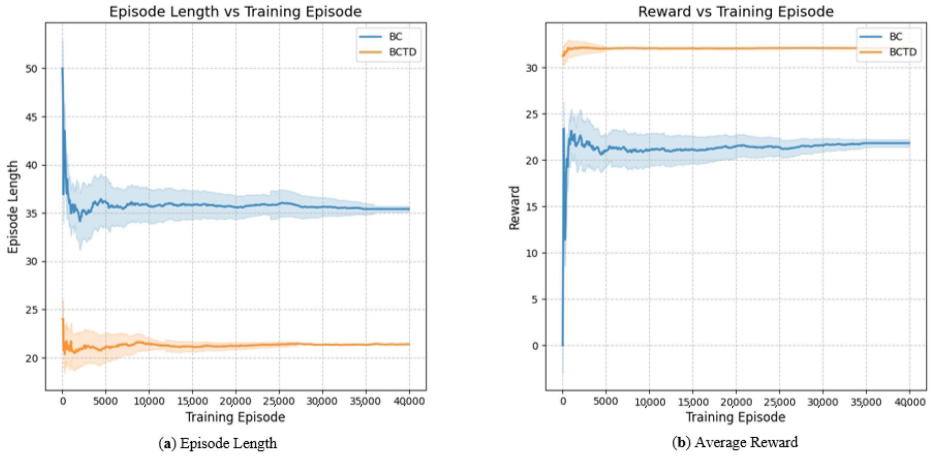
The convergence curve of the policy network's cross-entropy loss indicates that the cross-entropy loss may initially be large due to the complexity of the environment during the IL process. However, as the episodes progress, the cross-entropy loss gradually converges towards 0, showing that the policy network has effectively learned the tracking ability of the APF model. As shown in Figure 2, the convergence of the value network through TD Mean Squared Error (MSE) Loss curve.



(**a**) Cross-Entropy Loss (**b**) Value Network Loss

Figure 2. The policy network cross-entropy loss curve and the value network loss convergence curve.

The role of APF is merely to generate expert behaviors through imitation learning to guide the path planning of the agent, without relying on the update of the value function. Therefore, there is no critic network and no critic loss. And the loss curve of the value network via TD shows rapid convergence in the early stages. Value loss based on TD decreases quickly. However, during the convergence process, due to the local optimal limitations of the APF mean that there are cases where agents cannot quickly catch up with the tracking target. This leads to some oscillation in the evaluation of the current value even after the tracking ability of the APF model has been learned. The average reward curve and the average number of steps to complete the task for the APF model were then tested, as shown in Figure 3.



(**a**) Episode Length (**b**) Average Reward

Figure 3. Performance testing of the APF model.

Through the reward and step completion curves, even with some local optima, the APF model still effectively completes the corresponding tracking tasks. Additionally, due to its rule-based nature, the tracking performance remains relatively stable.

In the experiments in this chapter, the number of drones is 3, and the number of tracking targets is 1. MAPPO + BCTD and other classic centralized training and distributed execution algorithms were selected for comparison experiments, including MAPPO based on policy, the MADDPG algorithm [33], and the QMIX algorithm [34] based on value. First, we analyze the application of the BCTD algorithm in the MAPPO algorithm, and then compare the performance of different algorithms based on the evaluation metrics in the collaborative tracking scenario.

The average reward curve and the average number of steps for different algorithms during training were tested as shown in Figure 4, which presents the average reward curve of the episode for different algorithms in the collaborative tracking scenario.

The curve with higher opacity and fluctuations represents the actual reward curve used, and the curve with lower opacity represents the curve after smoothing the raw rewards. The analysis of the episode reward convergence for different algorithms is as follows: In this experiment, MAPPO + BCTD demonstrated significant advantages. Compared with other algorithms, MAPPO + BCTD rapidly converged early in the training and stabilized at higher reward values, indicating that the pre-training via behavior cloning effectively accelerated the agents' policy learning process. In contrast, MAPPO gradually improved rewards but had slower convergence and larger reward fluctuations during training, suggesting that without pre-training, the agent requires more time to adapt to the task. MADDPG showed unstable performance in the multi-agent environment with consistently low reward values, failing to effectively improve tracking ability. QMIX, though showing smoother reward changes, had lower final reward levels and could not overcome

the influence of obstacles in the environment, leading to incomplete optimization of the policy. In complex tracking tasks, MAPPO + BCTD accelerated convergence and improved stability through pre-training in behavior cloning, making it the most effective algorithm.

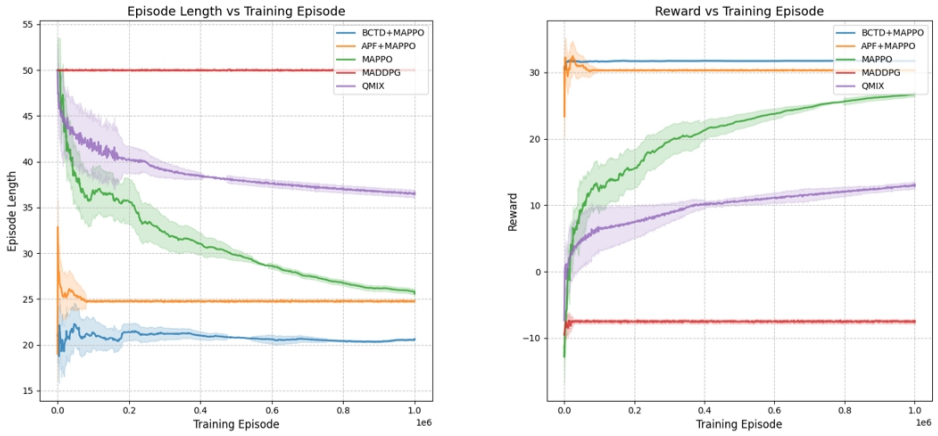
As shown in Figure 4, episode completion step curves for different algorithms in a collaborative tracking scenario, the curve with higher opacity and fluctuations represents the actual episode step curve, and the curve with lower opacity represents the episode step curve after the raw rewards have been smoothed.

In this experiment, MAPPO + BCTD performed the best overall, benefiting from the pre-training via behavior cloning, which effectively improved the agents' learning efficiency and strategy collaboration capability. This led to a rapid decrease in the number of steps per episode, stabilizing around 20 steps, demonstrating excellent convergence speed and decision-making efficiency. MAPPO followed, with the number of steps stabilizing around 26, indicating its good multi-agent collaboration and environmental adaptation. In contrast, MADDPG performed poorly, with steps remaining around 50 and showing significant fluctuations, exposing its instability in learning for multi-agent cooperation tasks. QMIX, although better than MADDPG, still had a higher number of steps than MAPPO-like algorithms in complex environments, indicating limited policy flexibility and environmental adaptability. In summary, MAPPO + BCTD, with the advantage of pre-training, significantly improved policy convergence efficiency and collaboration capabilities, exhibiting optimal performance in multi-agent target tracking tasks.

The evaluation indicators of the collaborative tracking scenarios in the evaluation indicators of the MAPPO+BCTD, MAPPO, QMIX, MADDPG and APF algorithms were quantified, and the comparison of the tracking performance after convergence of different algorithms is shown in Table 2:

Table 2. Comparison of tracking performance after convergence for different algorithms.

Metric	MADDPG	QMIX	MAPPO	APF + MAPPO	BCTD + MAPPO
Success Rate	0.00 ± 0.0	0.72 ± 0.15	1.00 ± 0.12	1.00 ± 0.08	1.00 ± 0.0
Episode Length	50 ± 0	36.2 ± 2.1	25.2 ± 1.6	20.2 ± 0.8	19.6 ± 0.2
Average Reward	−2.7 ± 1	14.6 ± 3.2	31.3 ± 3.8	31.3 ± 3.5	32.4 ± 1.2
Training Timestep	1000 k	300 k	500 k	150 k	65 k



(**a**) Episode Length (**b**) Average Reward

Figure 4. Comparison of different algorithms.

As shown in the table, MAPPO + BCTD demonstrated significant advantages in multi-agent target tracking tasks, particularly in convergence speed, task completion efficiency, and strategy learning ability. With pre-training via behavior cloning, MAPPO + BCTD can converge quickly, completing training in only 65 k episodes, significantly reducing training time compared to other algorithms. At the same time, this algorithm stands out in terms of tracking success rate and average reward, with a tracking success rate of 100% and the highest average reward, indicating its stability and efficiency in task execution. Moreover, MAPPO + BCTD has the fewest average steps, further demonstrating that its optimized policy model can efficiently complete target tracking tasks. In terms of model convergence, MAPPO + BCTD, leveraging the advantages of IL, required the fewest episodes to converge among reinforcement learning algorithms.

4.2. Experimental Visualization Comparison

After all models completed convergence, the effects of different algorithms were tested in the tracking task environment. As shown below, the episode ends when the tracking task is completed within the maximum number of steps; otherwise, the episode ends when the maximum action steps are reached. Figure 5 shows the visual demonstration of the experimental effects of different algorithms. The tracking drone's trajectory is represented by a red line, and the tracking target robot's trajectory is represented by a blue line. By comparing various algorithms, it is clear that MAPPO + BCTD demonstrates smoother drone paths and significantly fewer steps to complete the task. In contrast, the QMIX algorithm shows poor obstacle avoidance performance when faced with dense obstacles, causing the drone swarm to be unable to effectively avoid obstacles, which impacts tracking performance. The MADDPG algorithm, on the other hand, has a noticeable local optimum issue, where the drone swarm ultimately converges to a stationary state, failing to effectively track the target, performing much worse than other algorithms. These results indicate that MAPPO + BCTD can achieve more efficient and smoother tracking paths in complex environments, with stronger capabilities for handling dynamic obstacles.

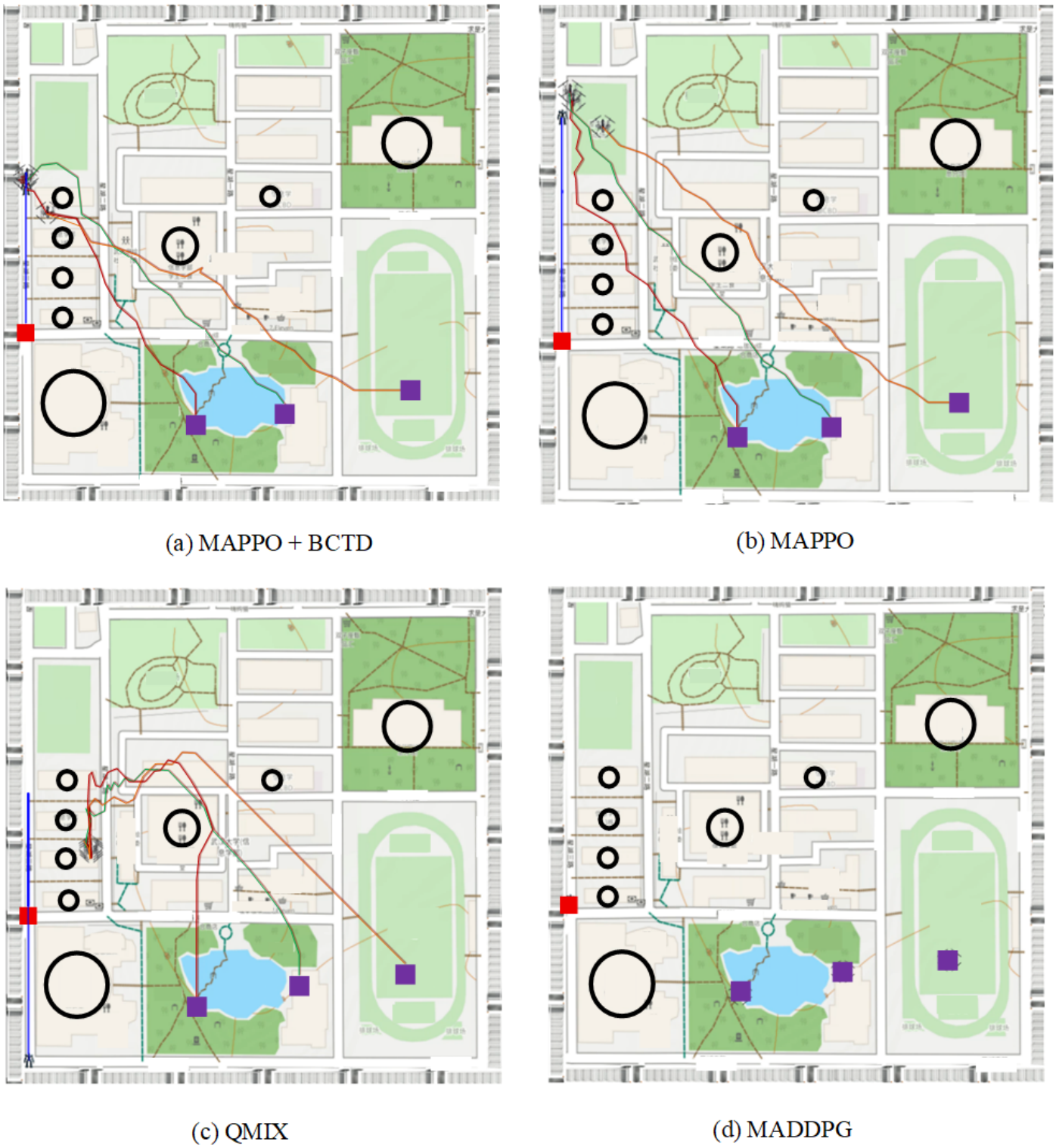


Figure 5. Demonstration diagrams of different algorithms. The red and purple rectangles indicate the initial positions of the target agent and pursuing agents, respectively. The movement trajectory of the target agent is represented by the blue line, while the movement trajectories of the pursuing agents are shown with red, green, and brown lines respectively. The black circles in the diagram represent generated obstacles.

5. Conclusions

An algorithmic model for multi-drone target tracking tasks is thoroughly discussed, with the combination of MAPPO and BCTD being proposed and its application in actual

tracking tasks. By introducing the APF model and combining the advantages of IL and RL, especially with optimizations in the MAPPO algorithm, we have significantly enhanced the model's convergence speed and task completion efficiency. Specifically, MAPPO + BCTD accelerates the policy learning process through pre-training, significantly reducing the number of episodes required for training, and demonstrating extremely high stability and efficiency during training. Furthermore, this method successfully completes multi-drone target tracking tasks with fewer steps, outperforming traditional algorithms such as MAPPO, MADDPG, and QMIX in both convergence speed and task efficiency. Although the proposed algorithm demonstrated strong performance in simulation, future work will focus on testing the method in real-world environments and extending the framework to handle larger multi-agent systems. Additionally, we plan to explore the scalability of the approach in 3D environments, which introduces more complex dynamics and obstacle interactions.

Author Contributions: Conceptualization, Y.Z. and L.L.; methodology, Y.Y. (Yunling Yue); validation, B.Y., J.X. and Y.Y. (Yuan Yao); investigation, J.X. and L.L.; writing—original draft preparation, Y.Z. and Y.Y. (Yuan Yao); writing—review and editing, Y.Y. (Yunling Yue)and L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Major Program (JD) of Hubei Province under Grant 2023BAA026.

Data Availability Statement: Data will be made available upon request.

Acknowledgments: The numerical calculations were conducted at the Supercomputing Center of Wuhan University. The authors wish to express their gratitude to the editors and reviewers for their invaluable feedback, which significantly contributed to enhancing the quality of this paper.

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

References

1. Li, Y.; Hao, Y.J.; Li, H.Y.; Li, M. Path Planning of Unmanned Ships Using Modified APF Algorithm. In Proceedings of the 2023 IEEE 11th International Conference on Computer Science and Network Technology (ICCSNT), Dalian, China, 21–22 October 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 112–119.
2. Fan, J.; Chen, X.; Liang, X. UAV trajectory planning based on bi-directional APF-RRT* algorithm with goal-biased. *Expert Syst. Appl.* **2023**, *213*, 119137. [CrossRef]
3. Hassan, I.A.; Abed, I.A.; Al-Hussaibi, W.A. Path planning and trajectory tracking control for two-wheel mobile robot. *J. Robot. Control (JRC)* **2024**, *5*, 1–15. [CrossRef]
4. Westheider, J.; Rückin, J.; Popović, M. Multi-uav adaptive path planning using deep reinforcement learning. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 649–656.
5. Ejaz, M.; Gui, J.; Asim, M.; El-Affendi, M.A.; Fung, C.; Abd El-Latif, A.A. RL-Planner: Reinforcement learning-enabled efficient path planning in multi-UAV MEC systems. *IEEE Trans. Netw. Serv. Manag.* **2024**, *21*, 3317–3329. [CrossRef]
6. Mahjoub, O.; de Kock, R.; Singh, S.; Khlifi, W.; Vall, A.; Tessera, K.a.; Pretorius, A. Efficiently quantifying individual agent importance in cooperative marl. *arXiv* **2023**, arXiv:2312.08466
7. Cheng, X.; Wang, Z. Multi-UAV Autonomous Collaborative Target Tracking in Dynamic Environment Based on Multi-agent Reinforcement Learning. In Proceedings of the 2023 3rd International Conference on Electronic Information Engineering and Computer (EIECT), Shenzhen, China, 17–19 November 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 123–128.
8. Wang, Y.; Dong, L.; Sun, C. Cooperative control for multi-player pursuit-evasion games with reinforcement learning. *Neurocomputing* **2020**, *412*, 101–114. [CrossRef]
9. Cao, Z.; He, Y. Imitation Learning Based Manipulator Skill Training for Home Environment. In Proceedings of the 2024 China Automation Congress (CAC), Qingdao, China, 1–3 November 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 6110–6115.
10. Ernst, D.; Louette, A. *Introduction to Reinforcement Learning*; Feuerriegel, S., Hartmann, J., Janiesch, C., Zschech, P., Eds.; Springer Nature: Berlin/Heidelberg, Germany, 2024; pp. 111–126.
11. Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; Wu, Y. The surprising effectiveness of ppo in cooperative multi-agent games. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24611–24624.

12. Cai, W.; Chen, H.; Zhang, M. A survey on collaborative hunting with robotic swarm: Key technologies and application scenarios. *Neurocomputing* **2024**, *598*, 128008. [CrossRef]
13. Shakya, A.K.; Pillai, G.; Chakrabarty, S. Reinforcement learning algorithms: A brief survey. *Expert Syst. Appl.* **2023**, *231*, 120495. [CrossRef]
14. Rubio, F.; Valero, F.; Llopis-Albert, C. A review of mobile robots: Concepts, methods, theoretical framework, and applications. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1729881419839596. [CrossRef]
15. Pateria, S.; Subagdja, B.; Tan, A.h.; Quek, C. Hierarchical reinforcement learning: A comprehensive survey. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–35. [CrossRef]
16. Xiao, J.; Yan, B.; Xie, H.; Yu, Q.; Li, L.; Wang, Y.F. Optimizing Multi-AAV Formation Cooperative Control Strategies with MCDDPG Approach. *IEEE Internet Things J.* **2025**, *12*, 9775–9791. [CrossRef]
17. Keyu, L.; Yonggen, L.; Yanchi, Z. Dynamic obstacle avoidance path planning of UAV Based on improved APF. In Proceedings of the 2020 5th International Conference on Communication, Image and Signal Processing (CCISP), Chengdu, China, 13–15 November 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 159–163.
18. Fan, X.; Guo, Y.; Liu, H.; Wei, B.; Lyu, W. Improved artificial potential field method applied for AUV path planning. *Math. Probl. Eng.* **2020**, *2020*, 6523158. [CrossRef]
19. Zhang, Z.; Zhang, D.; Zhang, Q.; Pan, W.; Hu, T. DACOOP-A: Decentralized Adaptive Cooperative Pursuit via Attention. *IEEE Robot. Autom. Lett.* **2023**, *9*, 5504–5511. [CrossRef]
20. Ding, Q.; Xu, X.; Gui, W. Path planning based on reinforcement learning with improved APF model for synergistic multi-UAVs. In Proceedings of the 2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Rio de Janeiro, Brazil, 24–26 May 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 432–437.
21. Zare, M.; Kebria, P.M.; Khosravi, A.; Nahavandi, S. A survey of imitation learning: Algorithms, recent developments, and challenges. *IEEE Trans. Cybern.* **2024**, *54*, 7173–7186. [CrossRef] [PubMed]
22. Mahmoudi, S.; Davar, A.; Sohrabipour, P.; Bist, R.B.; Tao, Y.; Wang, D. Leveraging imitation learning in agricultural robotics: A comprehensive survey and comparative analysis. *Front. Robot. AI* **2024**, *11*, 1441312. [CrossRef] [PubMed]
23. Hua, J.; Zeng, L.; Li, G.; Ju, Z. Learning for a robot: Deep reinforcement learning, imitation learning, transfer learning. *Sensors* **2021**, *21*, 1278. [CrossRef] [PubMed]
24. Arora, S.; Doshi, P. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artif. Intell.* **2021**, *297*, 103500. [CrossRef]
25. Bhattacharyya, R.; Wulfe, B.; Phillips, D.J.; Kuefler, A.; Morton, J.; Senanayake, R.; Kochenderfer, M.J. Modeling human driving behavior through generative adversarial imitation learning. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 2874–2887. [CrossRef]
26. Xiao, J.; Guo, H.; Zhou, J.; Zhao, T.; Yu, Q.; Chen, Y.; Wang, Z. Tiny object detection with context enhancement and feature purification. *Expert Syst. Appl.* **2023**, *211*, 118665. [CrossRef]
27. Perlaza, S.M.; Esnaola, I.; Bisson, G.; Poor, H.V. On the validation of Gibbs algorithms: Training datasets, test datasets and their aggregation. In Proceedings of the 2023 IEEE International Symposium on Information Theory (ISIT), Taipei, Taiwan, 25–30 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 328–333.
28. Jacinto, E.; Martinez, F.; Martinez, F. Navigation of autonomous vehicles using reinforcement learning with generalized advantage estimation. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*. [CrossRef]
29. Song, Q.; Sun, B.; Li, S. Multimodal sparse transformer network for audio-visual speech recognition. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 10028–10038. [CrossRef] [PubMed]
30. Aldao, E.; González-deSantos, L.M.; Michinel, H.; González-Jorge, H. Uav obstacle avoidance algorithm to navigate in dynamic building environments. *Drones* **2022**, *6*, 16. [CrossRef]
31. Liu, Z.; Xu, Y.; Xu, Y.; Qian, Q.; Li, H.; Ji, X.; Chan, A.; Jin, R. Improved fine-tuning by better leveraging pre-training data. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 32568–32581.
32. Xu, C.; Ye, Z.; Mei, L.; Yu, H.; Liu, J.; Yalikun, Y.; Jin, S.; Liu, S.; Yang, W.; Lei, C. Hybrid attention-aware transformer network collaborative multiscale feature alignment for building change detection. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 5012914. [CrossRef]
33. Lowe, R.; Wu, Y.I.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. . [CrossRef]
34. Rashid, T.; Samvelyan, M.; De Witt, C.S.; Farquhar, G.; Foerster, J.; Whiteson, S. Monotonic value function factorisation for deep multi-agent reinforcement learning. *J. Mach. Learn. Res.* **2020**, *21*, 1–51.

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.