sensors

MDPI

Article

Searching and Tracking an Unknown Number of Targets: A Learning-Based Method Enhanced with Maps Merging

Peng Yan [1], Tao Jia [2] and Chengchao Bai [1,*]

[1] School of Astronautics, Harbin Institute of Technology, Harbin 150001, China; yanpeng@hit.edu.cn
[2] Aerospace Technology Research Institute, China Aerodynamics Research and Development Center, Mianyang 621000, China; 13946034815@163.com
* Correspondence: baichengchao@hit.edu.cn

Abstract: Unmanned aerial vehicles (UAVs) have been widely used in search and rescue (SAR) missions due to their high flexibility. A key problem in SAR missions is to search and track moving targets in an area of interest. In this paper, we focus on the problem of Cooperative Multi-UAV Observation of Multiple Moving Targets (CMUOMMT). In contrast to the existing literature, we not only optimize the average observation rate of the discovered targets, but we also emphasize the fairness of the observation of the discovered targets and the continuous exploration of the undiscovered targets, under the assumption that the total number of targets is unknown. To achieve this objective, a deep reinforcement learning (DRL)-based method is proposed under the Partially Observable Markov Decision Process (POMDP) framework, where each UAV maintains four observation history maps, and maps from different UAVs within a communication range can be merged to enhance UAVs' awareness of the environment. A deep convolutional neural network (CNN) is used to process the merged maps and generate the control commands to UAVs. The simulation results show that our policy can enable UAVs to balance between giving the discovered targets a fair observation and exploring the search region compared with other methods.

Keywords: unmanned aerial vehicle (UAV); search and track; deep reinforcement learning (DRL); maps merging; convolutional neural network (CNN)

Citation: Yan, P.; Jia, T.; Bai, C. Searching and Tracking an Unknown Number of Targets: A Learning-Based Method Enhanced with Maps Merging. *Sensors* **2021**, *21*, 1076. https://doi.org/10.3390/s21041076

Academic Editor: Mario Luca Fravolini
Received: 4 January 2021
Accepted: 2 February 2021
Published: 4 February 2021

1. Introduction

In the past decade, unmanned aerial vehicles (UAVs) have been widely used in military and civilian applications due to their low cost and high flexibility. Especially in search and rescue (SAR) missions, multiple UAVs working together can reduce mission execution time and provide timely relief to targets [1–3]. In a SAR mission, UAVs need to search out targets in an unknown region and continuously track them to monitor their status. However, in general, the number of targets is unknown and available UAVs are limited, which requires multiple UAVs to work together to keep track of the discovered targets while finding more unknown targets [4]. The problem of using robot teams to cooperatively observe multiple moving targets has been formalized first by Parker and Emmons [5], who termed this problem as Cooperative Multi-Robot Observation of Multiple Moving Targets (CMOMMT) and showed it is NP-hard.

Since the CMOMMT problem was raised, there has been a great deal of work to address it. A classical approach is the local force vector proposed by Parker and Emmons [5], in which a robot is subject to the attractive forces of nearby targets and the repulsive forces of nearby robots, and the direction of the robot's motion is determined by the combined force of the two. However, this method will cause overlapping observations on the same target. Thus, Parker [6] proposed an improved method called A-CMOMMT to solve this phenomenon, where the robots are controlled by the weighted local force vectors for tracking targets. Additionally, in [7], the authors proposed B-CMOMMT, in which a help behavior is added to reduce the risk of losing a target. In [8], the authors proposed an

algorithm called P-CMOMMT, considering the uniformity of the observation of the targets through the information entropy of targets' observations. The methods based on local force vector lack the prediction of the targets' behaviors and do not make full use of the targets' historical position information, resulting in a low efficiency for searching and tracking the targets.

A large number of optimization-based methods have been proposed to solve CMOMMT-like problems [9–12]. In [13], the authors used group of vision-based UAVs to search for multiple ground targets. The objective is to optimize the collective coverage area and the detection performance based on a distributed probability map updating model, in which the probability of target existence is calculated through the measurement information and information sharing among neighboring agents. In [14], the authors proposed a multi-objective optimization approach based on genetic algorithm (GA) to minimize the mission completion time for a team of UAVs finding a target in a bounded area. In [15], the UAVs' task sequence for a reconnaissance task assignment problem is considered, where the problem is formulated as a multi-objective, multi-constraint, nonlinear optimization problem solved with a modified Multi-Objective Symbiotic Organisms Search algorithm (MOSOS). In [16], searching and tracking an unknown ground moving target by multiple UAVs in an urban environment was modeled as a multi-objective optimization problem with preemptive priority constraints. The authors proposed a fuzzy multi-objective path planning method to solve this problem with target behavior predicted by extended Kalman filter (EKF) and probability estimation. In [17], the authors proposed a real-time path-planning solution enabling multiple UAVs to cooperatively search a given area. The problem is modeled as a Model Predictive Control (MPC) problem solved with Particle Swarm Optimization (PSO) algorithm. In [18], the authors emphasize the fairness of observations among different targets compared with the initial CMOMMT problem. They proposed an integer linear programming model to solve this problem where the motion of the targets is estimated in a Bayesian framework.

The above-mentioned approaches fail to balance between target searching and target tracking, which will make it difficult for UAVs to keep searching for undiscovered targets when the number of UAVs is less than the number of targets. To solve this problem, Li et al. [19] proposed a profit-driven adaptive moving targets search algorithm, which considers the impact of moving targets and collaborating UAVs in a unified framework through a concept called observation profit of cells. However, this approach assumes that the total number of targets is known, which is impractical in some complex environments. In [20], Dames proposed a method to enable multiple robots to search for and track an unknown number of targets. The robots use the Probability Hypothesis Density (PHD) filter to estimate the number of targets and the positions of the targets, and a Voronoi-based control strategy to search and track targets. This method assumes that each robot has a unique ID for creating a globally consistent estimate, which will limit the scalability of the robot team.

Recently, the development of deep reinforcement learning (DRL) [21] provides an alternative way to deal with the CMOMMT problem. DRL learns control policies through interacting with the environment, and it has reached or exceeded human levels in some game tasks [22,23]. There have been some studies using DRL to solve the targets search and tracking problem. In [24], the authors proposed a framework for searching for multiple static targets through a group of UAVs. The framework consists of a global planner based on a modern online Partially Observable Markov Decision Process (POMDP) solver and a local continuous-environment exploration controller based on a DRL method. In [25], the authors proposed a target following method based on deep Q-networks, considering visibility obstruction from obstacles and uncertain target motion. In [26], the authors proposed a DRL-based method to enable a robot to explore unknown cluttered urban environments, in which a deep network with convolutional neural network (CNN) [27] was trained by asynchronous advantage actor-critic (A3C) approach to generate appropriate frontier locations. In [28], the authors constructed a framework for automatically exploring unknown environments. The exploration process is decomposed into the decision, planning, and

mapping modules, in which the decision module is implemented by a deep Q-network for learning exploration policy from the partial map.

In this paper, we focus on the problem of Cooperative Multi-UAV Observation of Multiple Moving Targets (CMUOMMT), where a UAV team needs to search and track an unknown number of targets in a search region. Our objective is to enable UAVs to give the discovered targets a fair observation and meanwhile maximize the exploration rate of the environment to discover more targets. To achieve this objective, the problem is formulated as a POMDP and solved with a DRL method. During the mission, each UAV maintains four observation history maps, which can reduce the partial observability of the environment. Furthermore, maps merging among UAVs can further improve awareness of the environment. To extract environmental features, a deep network with CNN is used to process each UAV observation map. A modern DRL method is used to train the shared policy with a centralized training, decentralized execution paradigm. The main contributions of this work are as follows:

- The average observation rate of the targets, the standard deviation of the observation rates of the targets, and the exploration rate of the search region are simultaneously optimized to enable multiple UAVs to cooperatively achieve fair observation of discovered targets and continuous search for undiscovered targets.
- Each UAV maintains four observation maps recording observation histories, and a map merging method among UAVs is proposed, which can reduce the partial observability of the environment and improve awareness of the environment.
- A DRL-based multi-UAV control policy is proposed, which allows UAVs to learn to balance tracking targets and exploring the environment by interacting with the environment.

The remainder of this paper is organized as follows. In Section 2, the problem is formulated and the optimization objectives are introduced. In Section 3, the details of our method are proposed, including the maps merging method and the key ingredients of the DRL method. In Section 4, simulation experiments are conducted and the results are discussed. Finally, we conclude this paper in Section 5.

2. Problem Formulation

In this paper, we consider the problem of CMUOMMT described in [6,19], which is shown in Figure 1 and defined as follows:

- A bounded two-dimensional rectangular search region S discretized into $C_L \times C_W$ equally sized cells, where C_L and C_W represent the number of cells in the length and width directions of the search region, respectively.
- The time step is discretized and denoted by t within a mission time duration T.
- A set of N moving targets $\mathcal{V}$ in S. For target $v_j (v_j \in \mathcal{V}, j = 1, 2, \cdots N)$, the cell that lies at time step t is denoted by $c_t(v_j) \in S$. The mission is to observe these targets using multiple UAVs. To simplify this mission, we assume that the maximal speed of the targets is smaller than that of the UAVs.
- A team of M homogeneous UAVs $\mathcal{U}$ deployed in S to observe the targets. For UAV $u_i (u_i \in \mathcal{U},\ i = 1, 2, \cdots M)$, the cell that lies at time step t is denoted by $c_t(u_i) \in S$. Each UAV can observe the targets through its onboard sensor. The sensing range of each UAV is denoted by d_s. We assume that the UAVs are flying at a fixed altitude, and the size of the field of view (FOV) of each UAV is the same and remains constant. The term $FOV_t(u_i)$ denotes the FOV of the UAV u_i at time step t. In addition, each UAV is equipped with a communication device to share information to coordinate with other UAVs. The communication range is denoted by d_c, which is assumed to be larger than the sensing range d_s. The UAVs can only share information with UAVs within a communication range. We further assume that all UAVs share a known global coordinate system.

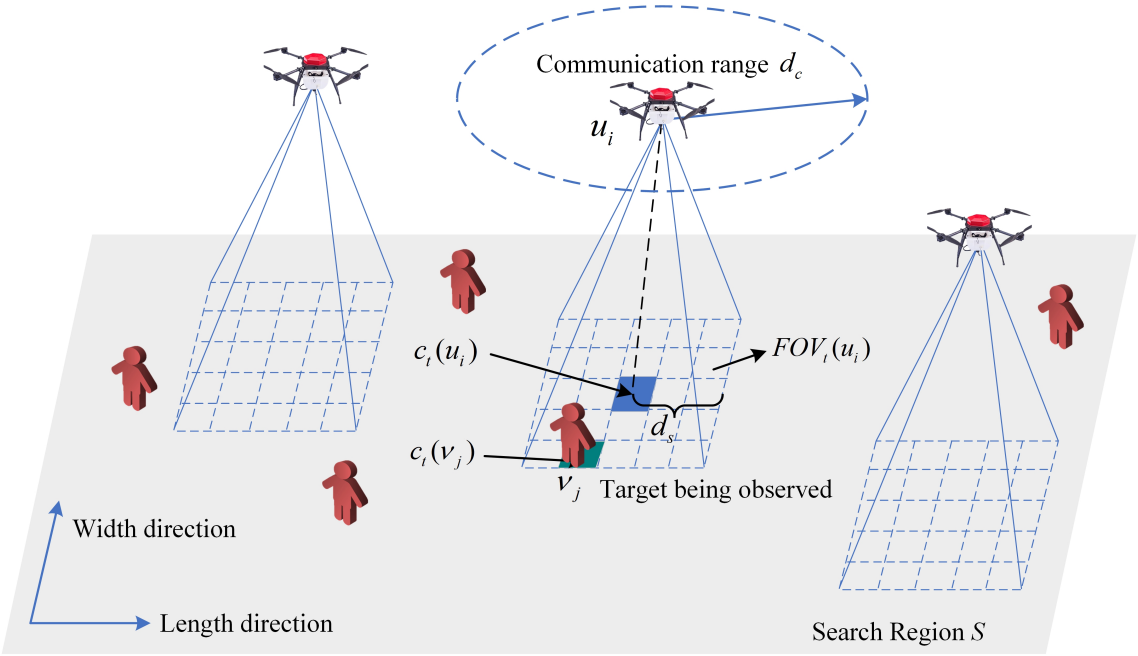

Figure 1. The environment considered in this paper. Unmanned aerial vehicles (UAVs) and targets are moving in a bounded two-dimensional rectangular search region S. $c_t(v_j)$ and $c_t(u_i)$ denote the cells in which target v_j and UAV u_i lie at time step t, respectively. The blue grids represent the FOV (field of view) of each UAV. The dashed ellipse indicates the communication range of the UAV u_i.

The target v_j is monitored when it is within the FOV of at least one UAV, which can be defined as

$$O_t(v_j) = \begin{cases} 1, \text{if } \exists u_i, c_t(v_j) \in FOV_t(u_i) \\ 0, \text{else} \end{cases} \tag{1}$$

where $O_t(v_j)$ indicates the observation state of target v_j.

During the mission, the observation rate of the target v_j can be defined as

$$\eta(v_j) = \frac{1}{T}\sum_{t=1}^{T} O_t(v_j) \tag{2}$$

where $\eta(v_j)$ represents the observation rate of the target v_j, which represents the proportion of time elapsed under the observation of at least one UAV during the mission.

The first objective for the UAV team is to maximize the average observation rate of N targets, which can be characterized by the metric $\bar{\eta}$:

$$\bar{\eta} = \frac{1}{N}\sum_{j=1}^{N} \eta(v_j) \tag{3}$$

Maximizing $\bar{\eta}$ alone is unfair, especially when the number of UAVs is less than the number of targets, which may result in some targets not being observed during the mission. To solve this problem, the second objective for the UAV team is to minimize the standard deviation σ_η of the observation rates of N targets:

$$\sigma_\eta = \sqrt{\frac{1}{N}\sum_{j=1}^{N} \left(\eta(v_j) - \bar{\eta}\right)} \tag{4}$$

A low value of σ_η means that all targets are observed relatively uniformly during the mission. In addition, since the UAV team does not know the total number of targets, it

needs to continuously explore the search region to discover new targets. Thus, the third objective for the UAV team is to maximize the exploration rate β of the search region, which is defined as

$$\beta = \frac{1}{T}\frac{1}{C_L C_W}\sum_{k=1}^{C_L}\sum_{l=1}^{C_W} t_{stamp}(c_{kl}) \tag{5}$$

where $t_{stamp}(c_{kl})$ represents the latest observed time for cell c_{kl}. k and l represent the indexes of the cell c_{kl} in the length and width directions of the search region, respectively. The maximum value of β is 1, which means that all cells in the search region are being observed by the UAV team at time step T. However, this is unrealistic since the maximum region observed by the UAV team is less than the total search region to be observed. That is,

$$\bigcup_{u_i \in \mathcal{U}} FOV(u_i) < S \tag{6}$$

The ultimate objective is a combination of $\bar{\eta}$, σ_η and β, which is different from [6,19], whose objectives only consider the average observation rate $\bar{\eta}$. In this study, the UAV team needs to balance between giving the known targets a fair observation and exploring the search region through an efficient method.

3. Methods

3.1. Overview

We formulate the CMUOMMT problem as a POMDP and solve it with a DRL method. In this method, all UAVs share a global control policy π to decide actions. The action is selected according to the observation from the environment, i.e., $\mathbf{a}_t \sim \pi(\mathbf{a}_t|\mathbf{o}_t)$, $\mathbf{o}_t \sim \mathcal{O}(\mathbf{s}_t)$, where $\mathbf{s}_t$ is the global state of the environment, $\mathbf{o}_t$ is the local observation of the environment state, $\mathcal{O}(\mathbf{s}_t)$ is the observation function determined by the UAVs' sensing range and communication range, and $\mathbf{a}_t$ is the selected action. The observation $\mathbf{o}_t$ includes four observation maps about the environment, which will be given in Section 3.2.

In a reinforcement learning (RL) framework, an RL agent learns an optimal policy $\mathbf{a}_t \sim \pi^*(\mathbf{a}_t|\mathbf{o}_t)$ through interacting with the environment. The goal of the RL agent is to maximize a long-term accumulated reward

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{7}$$

where r_{t+k+1} is the reward the RL agent received at time step $t+k+1$, $\gamma (0 < \gamma < 1)$ is the discount factor to make G_t a bounded value.

In the proposed DRL method, we use a deep neural network $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ parameterized by θ to approximate the UAVs' control policy. The objective is to use a DRL method to find the optimal parameters θ^*, which can make the UAV team balance between giving the known targets a fair observation and exploring the search region. The system architecture is shown in Figure 2.

As shown in Figure 2, each UAV first gets the observations from the environment through its onboard sensor to update its local observation maps. Then, each UAV receives the local maps of the other UAVs through its communication device, and the local maps are merged to provide the deep network π_θ an observation $\mathbf{o}_t$. Finally, the deep neural network π_θ outputs the action $\mathbf{a}_t$ to control the UAV and receives the reward r_{t+1} at the next time step. The maps merging method is introduced in the next subsection, and the ingredients of deep reinforcement learning are introduced in Section 3.3.

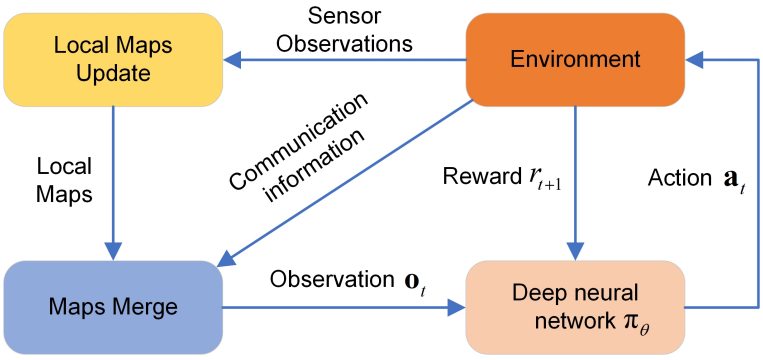

Figure 2. The system architecture.

3.2. Maps Merging

During the mission, each UAV maintains four observation maps:

(1) The observation map of the UAV's position in the search region, denoted by a $C_L \times C_W$ matrix $MS_t(u_i)$, $MS_t(u_i) \in \mathbb{R}^2$, defined as follows:

$$MS_t(u_i) = \left[ms_t^{kl}(u_i)\right]_{C_L \times C_W}, \quad ms_t^{kl}(u_i) = \begin{cases} 1, \text{ if } c_t(u_i) = c_{kl} \\ 0, \text{ else} \end{cases} \tag{8}$$

(2) The observation history map of the cells, which records the latest observed time for each cell. This map is denoted by a $C_L \times C_W$ matrix $MC_t(u_i)$, $MC_t(u_i) \in \mathbb{R}^2$, defined as follows:

$$MC_t(u_i) = \left[mc_t^{kl}(u_i)\right]_{C_L \times C_W} \tag{9}$$

The map $MC_t(u_i)$ is obtained in two steps. At each time step t, the map $MC_t(u_i)$ is first updated by the observation of the UAV u_i on the subset of cells within $FOV_t(u_i)$, that is,

$$mc_t^{kl}(u_i) = t, \quad \text{for } \ c_{kl} \in FOV_t(u_i) \tag{10}$$

In addition, the observation history maps from other UAVs within a communication range will also update the local maps. The values of corresponding cells in the observation history map will be updated with the latest observation time as follows:

$$mc_t^{kl}(u_i) = mc_t^{kl}(u_j), \text{ if } mc_t^{kl}(u_j) > mc_t^{kl}(u_i) \text{ and } d_t(u_i, u_j) < d_c, \ u_j \in \mathcal{U}, \ j \neq i \tag{11}$$

where $d_t(u_i, u_j)$ represents the distance between UAV u_i and UAV u_j.

(3) The position history map of the other UAVs, which records the history positions of the other UAVs. This map is denoted by a $C_L \times C_W$ matrix $MU_t(u_i)$, $MU_t(u_i) \in \mathbb{R}^2$, defined as follows:

$$MU_t(u_i) = \left[mu_t^{kl}(u_i)\right]_{C_L \times C_W}, \tag{12}$$

$$mu_t^{kl}(u_i) = \begin{cases} 1, \text{ if } c_t(u_j) = c_{kl} \text{ and } d_t(u_i, u_j) < d_c, \ u_j \in \mathcal{U}, \ i \neq j \\ 0, \text{ if } t = 0 \end{cases} \tag{13}$$

At each time step, the map $MU_t(u_i)$ is updated by the observation history maps from other UAVs within a communication range as follows:

$$mu_t^{kl}(u_i) = mu_t^{kl}(u_j), \text{ if } mu_t^{kl}(u_j) > mu_t^{kl}(u_i) \text{ and } d_t(u_i, u_j) < d_c, \ u_j \in \mathcal{U}, \ j \neq i \tag{14}$$

In addition, the knowledge about the positions of the other UAVs at the last observation time might be outdated. Thus, at the next time step, the values of cells in $MU_t(u_i)$ are decayed as follows:

$$mu_{t+1}^{kl}(u_i) = mu_t^{kl}(u_i) - \frac{1}{t_U}, \quad \text{if } \ mu_t^{kl}(u_i) \geq \frac{1}{t_U} \tag{15}$$

where t_U is a time constant, representing the decay period of the value of $mu_t^{kl}(u_i)$.

(4) The position history map of the targets, which records the historical positions of the targets. This map is denoted by a $C_L \times C_W$ matrix $MT_t(u_i)$, $MT_t(u_i) \in \mathbb{R}^2$, defined as follows:

$$MT_t(u_i) = \left[mt_t^{kl}(u_i) \right]_{C_L \times C_W}, \tag{16}$$

$$mt_t^{kl}(u_i) = \begin{cases} 1, \text{ if } c_t(v_j) = c_{kl} \text{ and } c_t(v_j) \in FOV_t(u_i),\ v_j \in \mathcal{V} \\ 0, \text{ if } t = 0 \end{cases} \tag{17}$$

The map $MT_t(u_i)$ is also updated by the observation history maps from other UAVs within a communication range, that is,

$$mt_t^{kl}(u_i) = mt_t^{kl}(u_j), \text{ if } mt_t^{kl}(u_j) > mt_t^{kl}(u_i) \text{ and } d_t(u_i,\ u_j) < d_c,\ u_j \in \mathcal{U},\ j \neq i \tag{18}$$

Same as the map $MU_t(u_i)$, the values of cells in $MT_t(u_i)$ are decayed as follows:

$$mt_{t+1}^{kl}(u_i) = mt_t^{kl}(u_i) - \frac{1}{t_T}, \text{ if } mt_t^{kl}(u_i) \geq \frac{1}{t_T} \tag{19}$$

where t_T is a time constant, representing the decay period of the value of $mt_t^{kl}(u_i)$.

One example of four observation maps for UAV u_0 is shown in Figure 3.

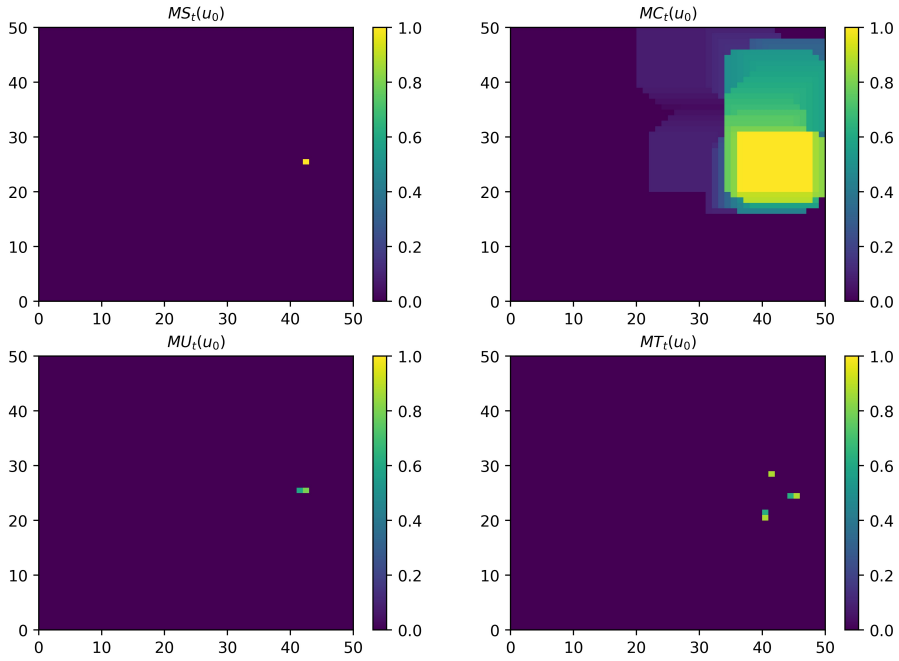

Figure 3. An illustration of four observation maps for UAV u_0. The parameters are set as follows: $N = 10$, $M = 5$, $C_L = 50$, $C_W = 50$, $d_s = 5$ cells, $d_c = 10$ cells, $t_U = 5$, $t_T = 8$. The current time step is $t = 200$. The map $MC_t(u_0)$ is normalized as follows: $mc_t^{kl}(u_0) = mc_t^{kl}(u_0) / \max(MC_t(u_0))$, where $\max(MC_t(u_0))$ represents the maximum value of elements in matrix $MC_t(u_0)$. From the map $MU_t(u_0)$, we can see that this map records one UAV's historical positions. From the map $MT_t(u_0)$, we can see that this map records three targets' historical positions.

3.3. Deep Reinforcement Learning

In this section, we introduce the key elements of the proposed DRL method, consisting of observation space, action space, network architecture, reward function, and training algorithm.

3.3.1. Observation Space

At time step t, the observation of UAV u_i consists of four parts, i.e., $\mathbf{o}_t(u_i) = [\mathbf{o}_t^1(u_i),\ \mathbf{o}_t^2(u_i),\ \mathbf{o}_t^3(u_i),\ \mathbf{o}_t^4(u_i)]$.

- The observation $\mathbf{o}_t^1(u_i)$ is a part of the map $MS_t(u_i)$ centered in the UAV's current cell $c_t(u_i)$, with length C_{input} and width C_{input}. That is, $\mathbf{o}_t^1(u_i)$ is a $C_{\text{input}} \times C_{\text{input}}$ matrix, representing the positional relationship of UAV u_i relative to the boundary of the search area S, which is defined as follows:

$$\mathbf{o}_t^1(u_i) = [{}^1o_t^{kl}(u_i)]_{C_{\text{input}} \times C_{\text{input}}}, \tag{20}$$

$${}^1o_t^{kl}(u_i) = \begin{cases} 1, \text{ if } c_t(u_i) = c_{kl} \text{ or } c_{kl} \notin S, k = 1, 2, \cdots C_{\text{input}}, \\ \quad l = 1, 2, \cdots C_{\text{input}} \\ 0, \text{ else} \end{cases} \tag{21}$$

- The observation $\mathbf{o}_t^2(u_i)$ is a part of the map $MC_t(u_i)$ centered in the UAV's current cell $c_t(u_i)$, with length C_{input} and width C_{input}. Similarly, $\mathbf{o}_t^2(u_i)$ is a $C_{\text{input}} \times C_{\text{input}}$ matrix, representing the observation state of the cells around UAV u_i, which is defined as follows:

$$\mathbf{o}_t^2(u_i) = [{}^2o_t^{kl}(u_i)]_{C_{\text{input}} \times C_{\text{input}}}, \tag{22}$$

$${}^2o_t^{kl}(u_i) = \begin{cases} mc_t^{mn}(u_i)/t, \text{ if } t > 0 \text{ and } c_{kl} \in S \text{ and } c_{kl} = c_{mn}, k = 1, 2, \cdots C_{\text{input}}, \\ \quad l = 1, 2, \cdots C_{\text{input}}, \ m = 1, 2, \cdots C_L, \ n = 1, 2, \cdots C_W \\ 1, \text{ elseif } c_{kl} \notin S \\ 0, \text{ else} \end{cases} \tag{23}$$

- The observation $\mathbf{o}_t^3(u_i)$ is a part of the map $MU_t(u_i)$ centered in the UAV's current cell $c_t(u_i)$, with length C_{input} and width C_{input}. Like $\mathbf{o}_t^2(u_i)$, $\mathbf{o}_t^3(u_i)$ is a $C_{\text{input}} \times C_{\text{input}}$ matrix, representing historical position information of other UAVs around UAV u_i, which is defined as follows:

$$\mathbf{o}_t^3(u_i) = [{}^3o_t^{kl}(u_i)]_{C_{\text{input}} \times C_{\text{input}}}, \tag{24}$$

$${}^3o_t^{kl}(u_i) = \begin{cases} mu_t^{mn}(u_i), \text{ if } c_{kl} \in S \text{ and } c_{kl} = c_{mn}, \ k = 1, 2, \cdots C_{\text{input}}, \\ \quad l = 1, 2, \cdots C_{\text{input}}, \ m = 1, 2, \cdots C_L, \ n = 1, 2, \cdots C_W \\ 0, \text{ else} \end{cases} \tag{25}$$

- The observation $\mathbf{o}_t^4(u_i)$ is a part of the map $MT_t(u_i)$ centered in the UAV's current cell $c_t(u_i)$, with length C_{input} and width C_{input}. Similarly, $\mathbf{o}_t^4(u_i)$ is a $C_{\text{input}} \times C_{\text{input}}$ matrix, representing historical position information of targets around UAV u_i, which is defined as follows:

$$\mathbf{o}_t^4(u_i) = [{}^4o_t^{kl}(u_i)]_{C_{\text{input}} \times C_{\text{input}}}, \tag{26}$$

$${}^4o_t^{kl}(u_i) = \begin{cases} mt_t^{mn}(u_i), \text{ if } c_{kl} \in S \text{ and } c_{kl} = c_{mn}, \ k = 1, 2, \cdots C_{\text{input}}, \\ \quad l = 1, 2, \cdots C_{\text{input}}, m = 1, 2, \cdots C_L, \ n = 1, 2, \cdots C_W \\ 0, \text{ else} \end{cases} \tag{27}$$

One example of observations for UAV u_0 is shown in Figure 4, which is consistent with the scenario shown in Figure 3.

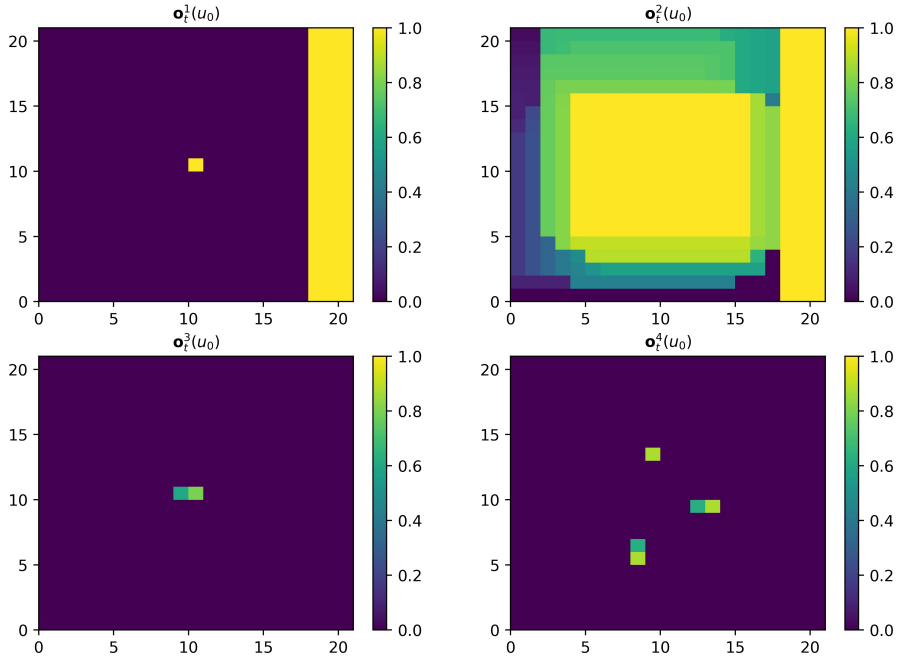

Figure 4. An illustration of observations for UAV u_0. The parameters are consistent with those in Figure 3. The value of C_{input} is set to 21 cells. From the observation $\mathbf{o}_t^1(u_0)$ and $\mathbf{o}_t^2(u_0)$, we can see that UAV u_0 is very close to the right boundary of the search region.

3.3.2. Action Space

The UAV's action space is a set of target cells around the UAV, that is, each UAV can move into one of its eight neighbor cells or stay at its current cell. Thus, the action space has a total of nine command actions. The actual command action is selected according to the selection probabilities calculated by the deep neural network.

3.3.3. Network Architecture

In this study, a deep neural network is used to process the observation $\mathbf{o}_t$, and its outputs are the selection probabilities of actions, denoted by $P(\mathbf{a}_t|\mathbf{o}_t)$. The deep neural network architecture is shown in Figure 5.

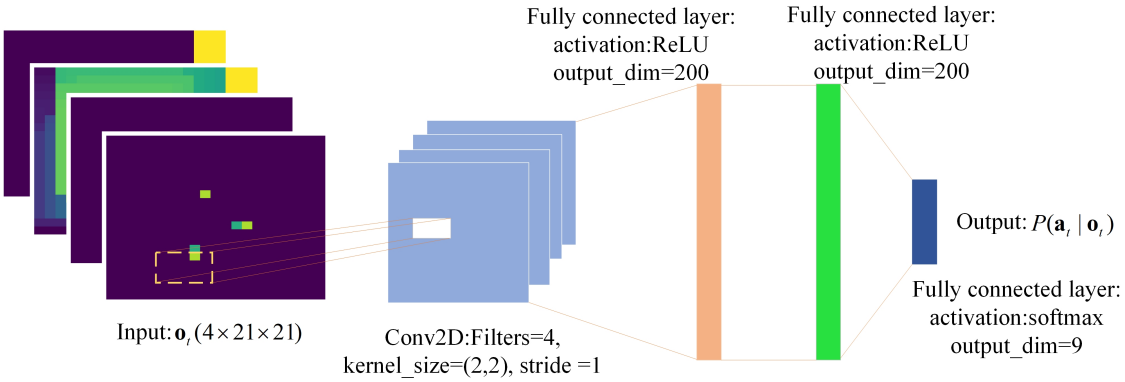

Figure 5. The deep neural network architecture. The value of C_{input} is set to 21.

As shown in Figure 5, we use four hidden layers to process the observation $\mathbf{o}_t$. The first hidden layer uses the CNN to process the input data, which has 4 two-dimensional filters with kernel size = (2, 2) and stride = 1, and its activation function is ReLU [29]. The second and third hidden layers are two fully connected layers with 200 rectifier units. The

last hidden layer contains nine nonlinear units with an activation function of Softmax, limiting the output to (0, 1), whose outputs are the selection probabilities of each action.

3.3.4. Reward Function

The design of the reward function is closely related to our objective, which is to enable the UAV team to balance between giving the known targets a fair observation and exploring the search region. Thus, a reward function is designed to achieve this objective:

$$r_t(u_i) = r_t^1(u_i) + r_t^2(u_i) + r_t^3(u_i) + r_t^4(u_i) \tag{28}$$

where $r_t(u_i)$ is the reward received by UAV u_i at time step t, which is a sum of four different rewards.

The reward $r_t^1(u_i)$ encourages UAV u_i to track targets that have been discovered, which consists of the following three terms:

$$r_t^1(u_i) = \lambda_1 {}^l r_t^1(u_i) + \lambda_2 {}^g r_t^1(u_i) + \lambda_3 {}^h r_t^1(u_i) \tag{29}$$

where ${}^l r_t^1(u_i)$ represents the local reward for tracking the discovered targets, ${}^g r_t^1(u_i)$ represents the global reward for tracking the discovered targets, ${}^h r_t^1(u_i)$ represents the reward for recording the historical positions of the targets, λ_1, and λ_2 and λ_3 are the positive coefficients. The rewards ${}^l r_t^1(u_i)$, ${}^g r_t^1(u_i)$ and ${}^h r_t^1(u_i)$ are designed as follows:

$$\begin{cases} {}^l r_t^1(u_i) = \min\left(\sum_{j=1}^{N} \frac{d_s}{20 * d_t(u_i, v_j)}, \ 1\right), \text{ for } d_t(u_i, v_j) < d_s \\ {}^g r_t^1(u_i) = \bar{\eta}_t - \bar{\eta}_{t-1} \\ {}^h r_t^1(u_i) = \min(\text{sum}(MT_t(u_i)), 1) \end{cases} \tag{30}$$

where $d_t(u_i, v_j)$ represents the distance between UAV u_i and target v_j at time step t, $\bar{\eta}_t$ represents the average observation rate of targets at time step t, $\text{sum}(MT_t(u_i))$ represents the sum of the values of the elements in matrix $MT_t(u_i)$, $\min(x, y)$ returns the minimum value of x and y.

The reward $r_t^2(u_i)$ encourages UAV u_i to explore the search region, which consists of the following two terms:

$$r_t^2(u_i) = \lambda_4 {}^l r_t^2(u_i) + \lambda_5 {}^g r_t^2(u_i) \tag{31}$$

where ${}^l r_t^2(u_i)$ is the local reward for exploring the search region, ${}^g r_t^2(u_i)$ is the global reward for exploring the search region, λ_4 and λ_5 are the positive coefficients. The rewards ${}^l r_t^2(u_i)$ and ${}^g r_t^2(u_i)$ are designed as follows:

$$\begin{cases} {}^l r_t^2(u_i) = \min\left(\frac{C_L C_W (\beta_t(u_i) - \beta_{t-1}(u_i))}{M * d_s}, \ 1\right) \\ {}^g r_t^2(u_i) = \min\left(\frac{C_L C_W (\beta_t - \beta_{t-1})}{M * d_s}, \ 1\right) \end{cases} \tag{32}$$

where $\beta_t(u_i)$ represents the local exploration rate of the search region known by UAV u_i at time step t, β_t represents the global exploration rate of the search region at time step t. $\beta_t(u_i)$ and β_t are calculated as follows:

$$\begin{cases} \beta_t(u_i) = \frac{1}{t} \frac{1}{C_L C_W} \sum_{k=1}^{C_L} \sum_{l=1}^{C_W} mc_t^{kl}(u_i) \\ \beta_t = \frac{1}{t} \frac{1}{C_L C_W} \sum_{k=1}^{C_L} \sum_{l=1}^{C_W} t_{stamp}(c_{kl}) \end{cases} \tag{33}$$

The reward $r_t^3(u_i)$ penalizes UAV u_i for approaching other UAVs too close, which is designed as follows:

$$r_t^3(u_i) = \sum_{j=1, j\neq i}^{M} r_t^3(u_i, u_j),\ r_t^3(u_i, u_j) = \begin{cases} -0.2, \text{ if } 0.8d_s \leq d_t(u_i, u_j) < d_s \\ -0.5, \text{ else if } 0.5d_s \leq d_t(u_i, u_j) < 0.8d_s \\ -1.0, \text{ else if } d_t(u_i, u_j) < 0.5d_s \\ 0.0, \text{ else} \end{cases} \tag{34}$$

The reward $r_t^4(u_i)$ penalizes UAV u_i for leaving the search region, which is designed as follows:

$$r_t^4(u_i) = \begin{cases} -5, \text{ if } c_t(u_i) \notin S \\ 0, \text{ else} \end{cases} \tag{35}$$

The reward function designed above can make UAVs receive dense rewards in the training process, which can reduce the difficulty of learning. In addition, we set $\lambda_1 = 0.6$, $\lambda_2 = 0.2$, $\lambda_3 = 0.2$, $\lambda_4 = 0.7$, and $\lambda_5 = 0.3$ in the training process.

3.3.5. Training Algorithm

In this study, we used a policy-based DRL algorithm, proximal policy optimization (PPO) [30], to train the deep neural network. The PPO has the benefits of optimizing control policies with guaranteed monotonic improvement and high sampling efficiency, and it has been widely used in the control of robots [31,32].

The algorithm flow is shown in Algorithm 1. In the training process, a centralized training, decentralized execution paradigm is used. Specifically, at each time step, each UAV independently obtains the observation and selects action through the shared policy, and the policy is trained with experiences collected by all UAVs during network training. The collected experience is used to construct the loss function $L^{CLIP}(\theta)$ for the policy network π_θ and the loss function $L^V(\phi)$ for the value network V_ϕ. The value network structure is the same as the policy network structure, except that it has only one linear unit in its last layer. In each episode, the policy network π_θ is optimized E_π times, and the value network V_ϕ is optimized E_V times on the same minibatch data sampled from the collected experience with Adam optimizer [33].

4. Results

In this section, simulation experiments are performed to evaluate the effectiveness of our proposed policy. We first describe the simulation setup and introduce the training process. Then, we compare our policy with other methods in various scenarios to validate its performance. Finally, we discuss the results.

4.1. Simulation Setup and Training Results

We conduct simulation experiments in a Python environment. The deep neural networks are implemented with Pytorch [34]. In the training process, we consider a search region of size 50×50 cells, i.e., $C_L = C_W = 50$ cells. The numbers of UAVs and targets in the search region are set to $M = 5$ and $N = 10$, respectively. The sensing range of each UAV is set to $d_s = 5$ cells and the communication range of each UAV is set to $d_c = 10$ cells. In addition, the maximum UAV speed is set to 1 cell per time step, and the maximum target speed is set to 0.5 cells per time step. The total mission time step is 200, i.e., $T = 200$. We set $t_U = 5$ and $t_T = 8$ for the decay period of the position history map of the UAVs and that of the position history map of the targets, respectively. The parameters in Algorithm 1 are listed in Table 1. In addition, the observation input size is set to $C_{\text{input}} = 21$ cells.

Algorithm 1: PPO with multiple UAVs for CMUOMMT

Initialize policy network π_θ, $\pi_{\theta'}$, and value function V_ϕ, let $\pi_{\theta'}=\pi_\theta$
for *episode = 1,2,...,* **do**
  for t = 1,2,...,T **do**
    for *UAV* i = 1, 2, ..., M **do**
      Run policy $\pi_{\theta'}$, collecting experience $\{\mathbf{o}_t(u_i),\ \mathbf{a}_t(u_i),\ r_{t+1}(u_i),\ \mathbf{o}_{t+1}(u_i)\}$.
      Estimate advantages using
      $\hat{A}_t(u_i) = -V_\phi(\mathbf{o}_t(u_i)) + r_{t+1}(u_i) + \cdots + \gamma^{T-t} r_{T+1}(u_i) + \gamma^{T-t+1} V_\phi(\mathbf{o}_{T+1}(u_i))$.
    end
  end
  for *UAV* i = 1, 2, ..., M **do**
    for j = 1,2,...,E_π **do**
      $L^{CLIP}(\theta) = -\widehat{\mathbb{E}}_t\left[\min\left(r_i^t(\theta)\hat{A}_t(u_i), \mathrm{clip}\left(r_i^t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t(u_i)\right)\right]$, $r_i^t(\theta)=\frac{\pi_\theta\left(\mathbf{a}_i^t|\mathbf{o}_i^t\right)}{\pi_{\theta'}\left(\mathbf{a}_i^t|\mathbf{o}_i^t\right)}$.
      Optimize surrogate $L^{CLIP}(\theta)$ wrt θ, with minibatch size B and the learning rate $l_{r\theta}$. (Note: $\mathrm{clip}(x, x_{min}, x_{max})$ limits the value of x between x_{min} and x_{max})
    end
    for k = 1,2,...,E_V **do**
      $L^V(\phi) = \sum_{t=1}^{T}\left(\sum_{t'>t}\gamma^{t'-t} r_{t'}(u_i) - V_\phi(\mathbf{o}_t(u_i))\right)$.
      Optimize surrogate $L^V(\phi)$ wrt ϕ, with minibatch size B and the learning rate $l_{r\phi}$.
    end
  end
  $\theta' \leftarrow \theta$
end

Table 1. Training parameters in Algorithm 1.

Parameters	Values
T	200
M	5
γ	0.99
E_π	10
ε	0.1
B	64
$l_{r\theta}$	0.00005
E_V	10
$l_{r\phi}$	0.001

The training process took 3000 episodes. At the beginning of each training episode, the positions of the UAVs and the targets are randomly reset. The speed of each target is randomly generated between [0, 0.5] cells per time step and remains unchanged during a training episode. We recorded the average and variance of each episode's cumulative reward every 100 episodes. The cumulative reward for each training episode is the average of the cumulative rewards received by all UAVs. The training results are shown in Figure 6. As training progresses, each UAV receives progressively larger rewards, which means that the control policy gradually converges, allowing each UAV to track discovered targets and explore unknown environments. It is worth noting that in the early stages of training, the UAVs receive negative rewards due to leaving the search region.

Figure 6. Training curve of the average and variance of each episode's cumulative reward every 100 episodes.

4.2. Comparison with Other Methods

In this subsection, we compare our policy with other methods, including A-CMOMMT [6], P-CMOMMT [8], PAMTS [19], and Random policy. A-CMOMMT is a traditional approach for solving the CMOMMT problem, which uses weighted local force vectors to control UAVs. P-CMOMMT considers the uniformity of the observation of the targets compared with A-CMOMMT. PAMTS is a novel distributed algorithm, considering tracking the targets and exploring the environment in a unified framework. Random policy serves as a baseline of the CMOMMT problem.

In each set of comparative simulation experiments, we ran 50 random test experiments for each method and calculated the average of the following three metrics:

- the average observation rate of the targets $\bar{\eta}$,
- the standard deviation σ_η of the observation rates of the N targets, and
- the exploration rate β of the search region.

We first compared our policy against other methods with different numbers of UAVs while the number of targets was fixed to 10. As shown in Figure 7a, the average observation rates of the targets continued to increase as the number of UAVs increased across all methods. Our policy had the best performance when the number of UAVs was 2, 10, or 15, and it was the second best method when the number of UAVs was 5 or 20. In addition, Figure 7b shows that our policy had the minimum standard deviation of the observation rates compared with A-CMOMMT and PAMTS in most cases, which shows that our policy can give the targets relatively fair observations. It is worth noting that the standard deviation of the observation rates gradually increased with the increase in the number of UAVs when using P-CMOMMT and Random policy. This is because the number of targets being observed increases when the number of UAVs increases, so that the standard deviation of the observation rates also increases with it. Figure 7c shows the exploration rate of the search region with the various number of UAVs. It can be seen that our policy had a high exploration rate in most cases relative to other methods except for the random policy. Overall, our policy can give targets a high and fair observation while maintaining a high exploration rate of the search region.

The impact of the total mission time on the three metrics was also studied. Figure 8a shows that the observation rates with A-CMOMMT, PAMTS, and our policy continued to improve as the total mission time increased. It is because that the increased mission time allows UAVs to search the environment sufficiently to find the targets. In addition, the observation rates of the A-CMOMMT, PAMTS, and our policy gradually approached as the total mission time increased, which means all three methods can find the targets in the search region with enough mission time. P-CMOMMT had a low observation rate because

it tries to give a uniform observation to the targets, which can also be seen from Figure 8b, where P-CMOMMT had a relatively low standard deviation of the observation rates. As shown in Figure 8b, our policy had a medium standard deviation of the observation rates. Similarly, as shown in Figure 8c, our policy had a medium exploration rate compared to the other methods. The results show that our policy can increase the observation rate of the targets when the mission time increases, while reducing the standard deviation of the observation rates and increasing the exploration rate of the search region.

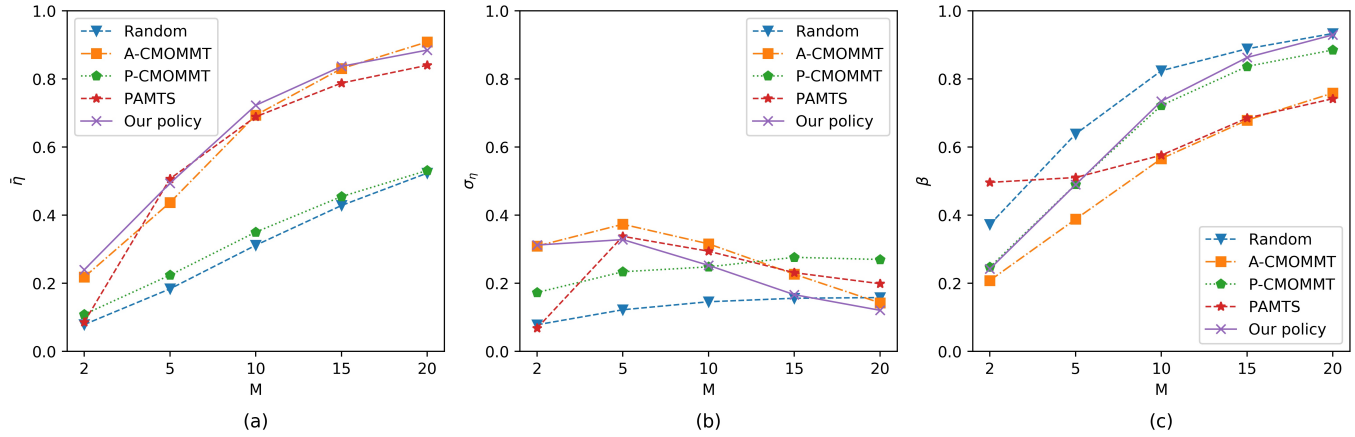

Figure 7. Comparison of results when the number of UAVs is increasing while the number of targets is fixed to 10. Other parameters are the same as those in the training process. (**a**) The results of the average observation rates change with the number of UAVs. (**b**) The results of the standard deviation of the average observation rates change with the number of UAVs. (**c**) The results of the exploration rates change with the number of UAVs.

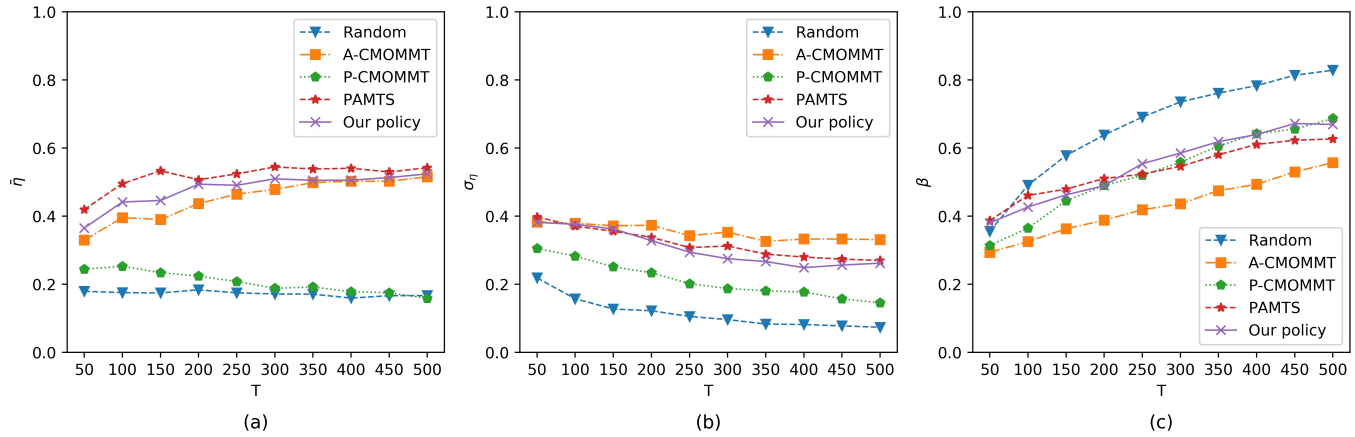

Figure 8. Comparison of results when the total mission time is increasing. Other parameters are the same as those in the training process. (**a**) The results of the average observation rates change with the total mission time. (**b**) The results of the standard deviation of the average observation rates change with the total mission time. (**c**) The results of the exploration rates change with the total mission time.

In addition, the impact of the size of the search region on the three metrics is studied. Figure 9a,c shows that the observation rate of the targets and the exploration rate of the search region decreased as the size of the search region increased. It is obvious that targets were more scattered in a larger search region, which makes it difficult for UAVs to find targets and explore the entire search region. As shown in Figure 9b, the increase in the standard deviation of the observation rates from $C_L = C_W = 25$ to $C_L = C_W = 50$ was due to the number of the discovered targets decreasing as the size of the search region increased. However, the decrease in the standard deviation of the observation rates from

$C_L = C_W = 50$ to $C_L = C_W = 125$ was due to the difficulty for UAVs to find the targets in a large search region.

Finally, we studied the impact of the communication range on the three metrics. As shown in Figure 10, for our policy, the observation rate and the exploration rate continued to improve, and the standard deviation of the observation rate continued to decrease as the communication range increased, until the communication range was greater than 10 cells, where all three metrics basically no longer changed. The impact of the communication range on the three metrics under PAMTS was consistent with our policy, except when there was no communication among UAVs, i.e., $d_c = 0$ cells. The results show that the information from the nearby UAVs can bring significant improvements, and the information from the remote UAVs has little impact on this mission. In addition, because A-CMOMMT and P-CMOMMT only consider the impact of UAVs within the sensing range, the variation in communication range has no effect on the three metrics. Like the above results, our policy has a high observation rate just below PAMTS and a low standard deviation of the observation rates and a high exploration rate of the search region compared with A-CMOMMT and PAMTS.

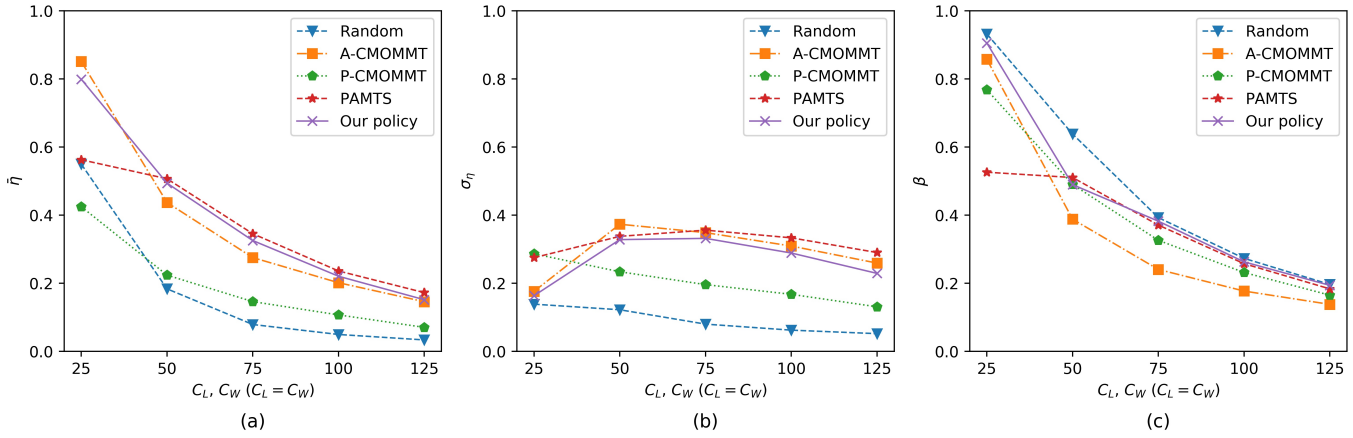

Figure 9. Comparison of results when the size of the search region is increasing. Other parameters are the same as those in the training process. (**a**) The results of the average observation rates change with the size of the search region. (**b**) The results of the standard deviation of the average observation rates change with the size of the search region. (**c**) The results of the exploration rates change with the size of the search region.

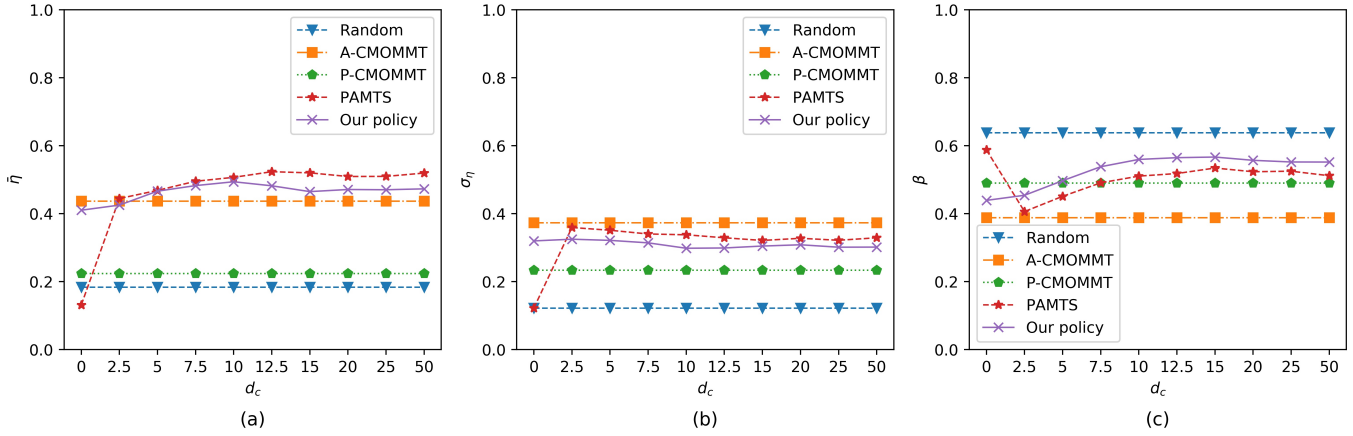

Figure 10. Comparison of results when the communication range is increasing. Other parameters are the same as those in the training process. (**a**) The results of the average observation rates change with the communication range. (**b**) The results of the standard deviation of the average observation rates change with the communication range. (**c**) The results of the exploration rates change with the communication range.

4.3. Discussion

The above comparison results show that our policy can find a balance between giving the known targets a fair observation and exploring the search region. Though our policy has a low observation rate compared with PAMTS in most cases, it can give a fair observation to the targets with a low standard deviation of the observation rates and continue a high exploration rate of the search region, which can enable UAVs to find more targets when the total number of the targets is unknown. It is worth noting that PAMTS assumes that the total number of targets is known, and we do not have this assumption.

5. Conclusions

In this paper, a DRL based approach is proposed to solve the CMUOMMT problem. Unlike traditional CMOMMT approaches, we considered the average observation rate of the targets, the standard deviation of the observation rates, and the exploration rate of the search region at the same time under the assumption that the total number of the targets is unknown. To achieve this objective, we used four observation maps to record the historical positions of targets and other UAVs, exploration status of the search region, and the UAV's position relative to the search region. In addition, UAVs' maps were merged from the maps of different UAVs within a communication range. The merged maps were then cropped and processed with a deep neural network to obtain the selection probabilities of actions. The reward function was designed carefully to provide UAVs with dense rewards in the training process. The results of the extensive comparison simulation experiments prove that our policy can give the targets a fair observation and meanwhile maintain a high exploration rate of the search region. Future work will study the CMUOMMT problem in a search region with obstacles and targets with evasive movements. This is a more challenging problem that requires smarter collaboration between UAVs.

Author Contributions: Conceptualization, P.Y., T.J., and C.B.; methodology, P.Y. and C.B.; software, P.Y. and T.J.; validation, C.B.; formal analysis, T.J.; data curation, P.Y.; writing—original draft preparation, P.Y. and T.J.; writing—review and editing, P.Y. and C.B.; visualization, P.Y. and T.J.; supervision, C.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

Abbreviations

The following abbreviations are used in this manuscript:

A3C	Asynchronous advantage actor-critic
CMOMMT	Cooperative Multi-Robot Observation of Multiple Moving Targets
CMUOMMT	Cooperative Multi-UAV Observation of Multiple Moving Targets
CNN	Convolutional neural network
DRL	Deep reinforcement learning
EKF	Extended Kalman filter
FOV	Field of view
GA	Genetic algorithm
MOSOS	Multi-Objective Symbiotic Organisms Search
MPC	Model Predictive Control
PHD	Probability Hypothesis Density

POMDP	Partially Observable Markov Decision Process
PPO	Proximal policy optimization
PSO	Particle Swarm Optimization
SAR	Search and rescue
UAV	Unmanned aerial vehicle

References

1. Queralta, J.P.; Taipalmaa, J.; Pullinen, B.C.; Sarker, V.K.; Gia, T.N.; Tenhunen, H.; Gabbouj, M.; Raitoharju, J.; Westerlund, T. Collaborative multi-robot systems for search and rescue: Coordination and perception. *arXiv* **2020**, arXiv:2008.12610.
2. Mendonça, R.; Marques, M.M.; Marques, F.; Lourenco, A.; Pinto, E.; Santana, P.; Coito, F.; Lobo, V.; Barata, J. A cooperative multi-robot team for the surveillance of shipwreck survivors at sea. In Proceedings of the OCEANS 2016 MTS/IEEE Monterey, Monterey, CA, USA, 19–23 September 2016; pp. 1–6.
3. Sampedro, C.; Rodriguez-Ramos, A.; Bavle, H.; Carrio, A.; de la Puente, P.; Campoy, P. A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based techniques. *J. Intell. Robot. Syst.* **2019**, *95*, 601–627. [CrossRef]
4. Khan, A.; Rinner, B.; Cavallaro, A. Cooperative robots to observe moving targets. *IEEE Trans. Cybern.* **2016**, *48*, 187–198. [CrossRef]
5. Parker, L.E.; Emmons, B.A. Cooperative multi-robot observation of multiple moving targets. In Proceedings of the International Conference on Robotics and Automation, Albuquerque, NM, USA, 20–25 April 1997; pp. 2082–2089.
6. Parker, L.E. Distributed algorithms for multi-robot observation of multiple moving targets. *Auton. Robot.* **2002**, *12*, 231–255. [CrossRef]
7. Kolling, A.; Carpin, S. Cooperative observation of multiple moving targets: an algorithm and its formalization. *Int. J. Robot. Res.* **2007**, *26*, 935–953. [CrossRef]
8. Ding, Y.; Zhu, M.; He, Y.; Jiang, J. P-CMOMMT algorithm for the cooperative multi-robot observation of multiple moving targets. In Proceedings of the 2006 6th World Congress on Intelligent Control and Automation, Dalian, China, 21–23 June 2006; Volume 2, pp. 9267–9271.
9. Peng, H.; Su, F.; Bu, Y.; Zhang, G.; Shen, L. Cooperative area search for multiple UAVs based on RRT and decentralized receding horizon optimization. In Proceedings of the 2009 7th Asian Control Conference, Hong Kong, China, 27–29 August 2009; pp. 298–303.
10. Yao, P.; Wang, H.; Ji, H. Multi-UAVs tracking target in urban environment by model predictive control and Improved Grey Wolf Optimizer. *Aerosp. Sci. Technol.* **2016**, *55*, 131–143. [CrossRef]
11. Rosalie, M.; Danoy, G.; Chaumette, S.; Bouvry, P. Chaos-enhanced mobility models for multilevel swarms of UAVs. *Swarm Evol. Comput.* **2018**, *41*, 36–48. [CrossRef]
12. Stolfi, D.H.; Brust, M.R.; Danoy, G.; Bouvry, P. A Cooperative Coevolutionary Approach to Maximise Surveillance Coverage of UAV Swarms. In Proceedings of the 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 10–13 January 2020; pp. 1–6.
13. Hu, J.; Xie, L.; Xu, J.; Xu, Z. Multi-agent cooperative target search. *Sensors* **2014**, *14*, 9408–9428. [CrossRef]
14. Hayat, S.; Yanmaz, E.; Brown, T.X.; Bettstetter, C. Multi-objective UAV path planning for search and rescue. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, Singapore, 29 May–3 June 2017; pp. 5569–5574.
15. Chen, H.X.; Nan, Y.; Yang, Y. Multi-UAV Reconnaissance task assignment for heterogeneous targets based on modified symbiotic organisms search algorithm. *Sensors* **2019**, *19*, 734. [CrossRef]
16. Hu, C.; Zhang, Z.; Yang, N.; Shin, H.S.; Tsourdos, A. Fuzzy multiobjective cooperative surveillance of multiple UAVs based on distributed predictive control for unknown ground moving target in urban environment. *Aerosp. Sci. Technol.* **2019**, *84*, 329–338. [CrossRef]
17. de Alcantara Andrade, F.A.; Reinier Hovenburg, A.; Netto de Lima, L.; Dahlin Rodin, C.; Johansen, T.A.; Storvold, R.; Moraes Correia, C.A.; Barreto Haddad, D. Autonomous unmanned aerial vehicles in search and rescue missions using real-time cooperative model predictive control. *Sensors* **2019**, *19*, 4067. [CrossRef]
18. Banfi, J.; Guzzi, J.; Amigoni, F.; Flushing, E.F.; Giusti, A.; Gambardella, L.; Di Caro, G.A. An integer linear programming model for fair multitarget tracking in cooperative multirobot systems. *Auton. Robot.* **2019**, *43*, 665–680. [CrossRef]
19. Li, X.; Chen, J.; Deng, F.; Li, H. Profit-driven adaptive moving targets search with UAV swarms. *Sensors* **2019**, *19*, 1545. [CrossRef]
20. Dames, P.M. Distributed multi-target search and tracking using the PHD filter. *Auton. Robot.* **2020**, *44*, 673–689. [CrossRef]
21. Nguyen, T.T.; Nguyen, N.D.; Nahavandi, S. Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications. *IEEE Trans. Cybern.* **2020**, *50*, 3826–3839. [CrossRef]
22. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; others. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
23. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; others. Mastering the game of go without human knowledge. *Nature* **2017**, *550*, 354–359. [CrossRef] [PubMed]

24. Walker, O.; Vanegas, F.; Gonzalez, F. A framework for multi-agent UAV exploration and target-finding in GPS-denied and partially observable environments. *Sensors* **2020**, *20*, 4739. [CrossRef]
25. Bhagat, S.; Sujit, P. UAV Target Tracking in Urban Environments Using Deep Reinforcement Learning. In Proceedings of the 2020 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 1–4 September 2020; pp. 694–701.
26. Niroui, F.; Zhang, K.; Kashino, Z.; Nejat, G. Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments. *IEEE Robot. Autom. Lett.* **2019**, *4*, 610–617. [CrossRef]
27. Sharif Razavian, A.; Azizpour, H.; Sullivan, J.; Carlsson, S. CNN features off-the-shelf: an astounding baseline for recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, Ohio, 23–28 June 2014; pp. 806–813.
28. Li, H.; Zhang, Q.; Zhao, D. Deep reinforcement learning-based automatic exploration for navigation in unknown environment. *IEEE Trans. Neural Networks Learn. Syst.* **2020**, *31*, 2064–2076. [CrossRef]
29. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; pp. 807–814.
30. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
31. Long, P.; Fanl, T.; Liao, X.; Liu, W.; Zhang, H.; Pan, J. Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 6252–6259.
32. Yan, P.; Bai, C.; Zheng, H.; Guo, J. Flocking Control of UAV Swarms with Deep Reinforcement Leaming Approach. In Proceedings of the 3rd International Conference on Unmanned Systems (ICUS), Harbin, China, 27–28 November 2020; pp. 592–599.
33. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
34. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; others. Pytorch: An imperative style, high-performance deep learning library. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 8026–8037.