Extrinsic-and-Intrinsic Reward-Based Multi-Agent Reinforcement Learning for Multi-UAV Cooperative Target Encirclement

Jinchao Chen, *Member, IEEE*, Yang Wang, Ying Zhang, *Member, IEEE*, Yantao Lu, *Member, IEEE*, Qiuhao Shu, and Yujiao Hu, *Member, IEEE*

***Abstract*— Due to their high flexibility and strong maneuverability, unmanned aerial vehicles (UAVs) have attracted lots of attention and are widely employed in many fields. Especially in target encirclement applications, UAVs have shown great advantages in adaptability and reliability, and can efficiently fly to and evenly surround the targets in complex and dynamic environments. In this paper, we concentrate on the cooperative target encirclement problem of heterogeneous UAVs and try to propose a multi-agent reinforcement learning approach to solve the problem. First, with the models of heterogeneous UAVs and obstacles, we analyze the collision avoidance, motion continuity, and energy consumption constraints of UAVs, and formulate the cooperative target encirclement problem as a multi-constraint combinatorial optimization one. Then, inspired by the humans' learning experience that curiosity provides a powerful motivator for humans to explore, discover, and acquire new knowledge, we propose an extrinsic-and-intrinsic reward-based multi-agent reinforcement learning approach to cooperatively control the behaviors of UAVs and achieve the target encirclement missions. Simulation experiments with randomly generated environments are conducted to evaluate the performance of our approach, and the results show that our approach has a significant advantage in terms of average reward, encirclement success rate, encirclement time, and encirclement energy consumption.**

***Index Terms*— Multi-agent reinforcement learning, heterogeneous unmanned aerial vehicle, extrinsic-and-intrinsic reward mechanism, cooperative target encirclement.**

I. Introduction

With the rapid advancements in technology, unmanned aerial vehicle (UAV) swarms have leveraged their high flexibility and strong maneuverability to achieve widespread adoption in both civilian and military domains. Particularly in the execution of intricate high-level missions, UAV swarms can significantly enhance the task execution efficiency and environmental adaptability in complex and dynamic environments through information exchange and behaviour collaboration. For example, in times of emergencies and natural disasters, UAV swarms can rapidly establish a temporary communication network to blanket the disaster-affected areas and provide smooth communication between rescuers and survivors by promoting mutual information sharing and task coordination [1], allowing the public to deliver massive amounts of information timely and boosting the overall efficiency of rescue operations. Similarly, in urban traffic monitoring, UAVs in a swarm can work together in a coordinated manner, communicate with each other, and share information to effectively monitor traffic congestion, accidents, road conditions, and other relevant factors in real-time [2], playing a significant role in expanding the monitoring range and enhancing the detection accuracy and the traffic management efficiency.

Cooperative target encirclement [3], which effectively mitigates potential threats and decreases the risk to personnel, is a typical application scenario of UAV swarm systems. In this scenario, autonomous UAVs are adopted to evenly surround a static or dynamic target to keep surveillance, prevent enemy targets from escaping, or protect friend targets [4]. As displayed in Fig. 1, during an encirclement mission, lots of radars continuously detect the enemy targets [5] and forward the detection information to the military base. Then, the military base releases an encirclement task to the UAV swarms. Once receiving the encirclement task, UAVs should accurately perceive the surrounding environment through on-board sensors [6] such as optical cameras, thermal imagers, and LiDAR, and devise an optimal encirclement strategy to encircle the targets while fully considering the environmental information and vehicle capabilities. Meanwhile, UAVs will dynamically adjust their flight and encirclement strategies according to real-time encirclement posture, efficiently achieve the cooperative encirclement of enemy targets, and greatly improve the security of the friend targets.

The cooperative target encirclement of multiple UAVs is complicated to solve, especially in dynamic environments where reasonable flight and encirclement strategies should

Received 19 May 2024; revised 6 September 2024, 19 October 2024, and 17 November 2024; accepted 29 December 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62106202, in part by the Key Research and Development Program of Shaanxi Province under Grant 2024GX-YBXM-118, in part by the Aeronautical Science Foundation of China under Grant 2023M073053003, and in part by the Fundamental Research Funds for the Central Universities. The Associate Editor for this article was A. Al-Dulaimi. *(Corresponding authors: Jinchao Chen; Yujiao Hu.)*

Jinchao Chen, Yang Wang, Ying Zhang, Yantao Lu, and Qiuhao Shu are with the School of Computer Science, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China (e-mail: cjc@nwpu.edu.cn; ywang0109@mail.nwpu.edu.cn; ying_zhang@nwpu.edu.cn; yantaolu@nwpu.edu.cn; qhsproanime@mail.nwpu.edu.cn).

Yujiao Hu is with the Future Network Research Center, Purple Mountain Laboratories, Nanjing, Jiangsu 211111, China (e-mail: huyujiao@pmlabs.com.cn).

Digital Object Identifier 10.1109/TITS.2024.3524562

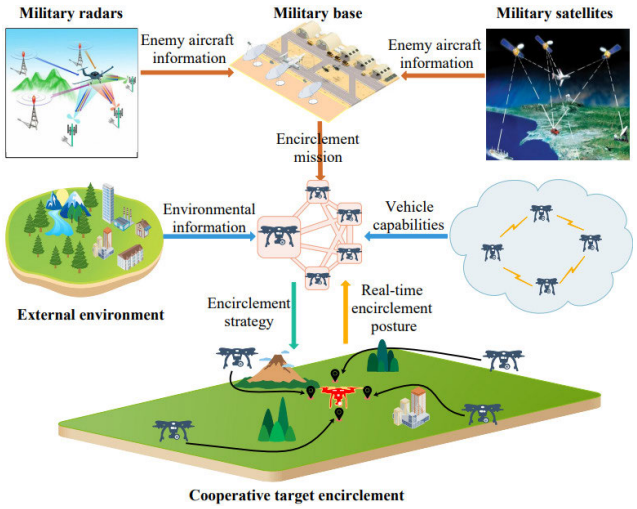

Fig. 1. Cooperative target encirclement with multiple UAVs.

be quickly provided to overcome the high uncertainties and successfully encircle the targets while satisfying many constraints, such as collision avoidance and energy consumption. Not only does the multi-constraint optimization property give it an NP-hard complexity, but also the dynamic environments add difficulties in obtaining the optimal strategies.

Although multi-UAV cooperative target encirclement has been widely studied in the literature, most existing research focuses on encircling targets by one or multiple homogeneous UAVs with uniform functionalities and performances to reduce system complexity. However, in many cases, those approaches cannot meet the complex requirements of practical encirclement tasks since homogeneous UAVs lack diversity and cannot be flexible and adaptable to different task requirements and environmental changes. In practice, there is a remarkable trend towards achieving encirclement missions with a heterogeneous UAV swarm composed of various types of drones, such as fixed-wing, multi-rotor, and vertical take-off and landing (VTOL) drones, and each type of drone has its specific abilities to handle particular tasks efficiently and provide a more comprehensive range of services. Meanwhile, current approaches pay more attention to encircling targets in static environments or reduced space without unexpected and anomalous changes, leading to unsatisfactory solutions that could not effectively adapt to dynamic and complex environments.

With the development of computer technology and artificial intelligence, reinforcement learning has rapidly received widespread attention from experts and scholars at both domestic and international levels in solving the cooperative target encirclement problem [7], [8], [9]. By interacting with environments through a distinctive trial-and-error mechanism, reinforcement learning helps UAVs intelligently chase and capture targets, continuously optimizing their strategies based on environmental feedback and improving the effectiveness and efficiency of encirclement operations. However, the training process of reinforcement learning-based approaches typically requires lots of samples and computational resources to search for optimal strategies, resulting in unsatisfactory solutions. Meanwhile, environments in the multi-UAV cooperative target encirclement problem are inherently complex, and the number of state-action combinations is extremely large [10], making it challenging to define well-behaved reward functions and provide clear reward signals. Due to the lack of sufficient positive rewards, the agents must carry out a large number of actions to find reasonable and reliable strategies, resulting in a slow or even non-converge learning solution, preventing the agents from achieving good performance.

In this work, we focus on the cooperative target encirclement problem of heterogeneous UAVs with different performances and try to develop an efficient multi-agent deep reinforcement learning framework to generate effective encirclement strategies for each UAV. The main contributions are summarized as follows:

1. Constraints and objectives of the multi-UAV cooperative target encirclement problem are analyzed, and the multi-UAV cooperative target encirclement problem is abstracted as a multi-constraint combinatorial optimization one to find the optimal and safe flight path for each UAV while satisfying the collision avoidance and energy consumption constraints.

2. Inspired by the human's learning process where curiosity plays a crucial role in exploring, learning, and understanding the unknowns, an extrinsic-and-intrinsic reward-based multi-agent reinforcement learning approach (EIR-MARL) is designed to train reasonable decision-making strategies and control the behaviors of UAVs such that the cooperative target encirclement mission would be achieved efficiently.

The remainder of this work is organized as follows. Section II briefly introduces the related work. In Section III, the models of heterogeneous UAVs and obstacles are built, and the multi-UAV cooperative target encirclement problem is formulated. Section IV establishes the Markov game model for the studied problem and introduces the proposed multi-agent reinforcement learning approach. Section V organizes the experiments and evaluates the performance of the proposed approach from several aspects. Finally, Section VI draws the conclusion of this work.

II. Related Work

Target encirclement, which aims to plan collision-free paths to surround a static or dynamic target, is a typical application scenario for UAVs and a research hotspot in the literature. In recent years, lots of efforts have been devoted to the target encirclement problem, and a number of feasible solutions have been presented with different constraints and objectives. Generally speaking, the existing approaches for the cooperative target encirclement problem can be categorized into three types: deterministic methods, heuristic methods, and reinforcement learning methods.

Deterministic methods, which rely on precise mathematical models and predefined control rules to coordinate the actions of UAVs [11], are frequently used in the target encirclement problem. By utilizing detailed environmental modeling and strategic planning, deterministic methods can optimize the behavior sequences of UAVs to effectively encircle the static or dynamic targets. Shen et al. [12] built a robot-target relative

dynamics model and designed a Lyapunov-based local controller to stabilize robots and achieve the cooperative hunting of multiple car-like robots. Hafez et al. [13] applied the model predictive control (MPC) scheme to describe the control ruler and implement the controllers and solved the multi-UAV encirclement problem by using different forms of MPC to control the positions of quad-rotors. Altan and Hacıoğlu [14] proposed a novel Hammerstein model-based MPC controller for real-time target tracking of a three-axis gimbal system in UAV flight scenarios to ensure robustness under external disturbances. Liu et al. [15] proposed an efficient path planner based on an improved rapidly exploring random tree (RTT) to deal with the cooperative target-tracking problem and generate feasible paths for robots in aquatic environments. Although deterministic methods typically exhibit high performance and are easy to implement through the existing control frameworks, they are usually designed and adopted in specific application scenarios, and their effectiveness is mainly dependent on the accuracy of the mathematical models and control rules. Deterministic methods lack learning ability and cannot quickly adapt to dynamic changes, resulting in poor or unsatisfactory solutions in open and complex environments.

Heuristic methods, which simplify complex problems by utilizing the existing knowledge and inspiration to quickly find approximate or satisfactory solutions [16], are frequently used for problems that cannot be solved using traditional exact algorithms, such as combinatorial optimization problems and the traveling salesman problem. In multi-UAV cooperative target encirclement, heuristic methods not only incorporate heuristic information and rules to guide the encirclement process but also rapidly adapt to dynamically changing environments and efficiently adjust their strategies to achieve the mission. Inspired by the cooperative hunting behaviors of falcons, Bin et al. [17] proposed a multi-UAV grouped attack strategy composed of target selection and cooperative capture success mechanisms to handle the target assignment and team-mate assignment problem during the confrontation in real-time systems [18]. Phung and Ha [19] designed a spherical vector-based particle swarm optimization (SPSO) method to produce feasible and safe flight paths for UAVs in complex and dynamic environments with multiple threats. Altan [20] proposed a proportional-integral-derivative algorithm to achieve both attitude and altitude control of quad-rotors by optimizing the parameters with the Harris Hawks Optimization (HHO). Zhang et al. [21] improved the particle swarm optimization algorithm by using a specific particle coding method and a population updating strategy and achieved a multi-UAV dynamic mission while taking into account a variety of complex constraints. Although heuristic methods have obvious advantages in efficiency, simplicity, practicality, and adaptability [22], they cannot effectively explore the entire solution space and easily fall into local optimum due to the over-reliance on initial parameter settings. Meanwhile, heuristic methods usually have strong dependencies on the assumptions and specific problem structures. If the problem features and assumptions are not satisfied, the effectiveness of heuristic methods may be significantly compromised.

Reinforcement learning methods, which allow agents continuously to learn and optimize their decision-making policies through real-time interactions with environments [23], have attracted lots of attention in seeking the optimal solutions for the cooperative target encirclement problem of multiple UAVs. In these methods, the agent does not need to make any assumptions or establish an accurate model of the environment and can obtain the optimal behavior policy through trial and error while quickly adapting to different environments and tasks. Ebrahimi et al. [24] designed a multi-agent reinforcement learning-based framework for multiple UAVs to autonomously produce their flight paths and improve the localization accuracy of distributed objects. Xiong and Zhang [25] proposed a reinforcement learning-based control approach to enable UAVs to efficiently track the targets while keeping formation and avoiding collision. Inspired by the way of human thinking that situation prediction is always before decision-making, Zhang et al. [26] improved the multi-agent deep deterministic policy gradient formulation with a target prediction network and proposed a pursuit-evasion scenario framework to enhance the cooperation effectiveness of multiple quad-copters. Reinforcement learning methods have high demands on the state space of the environment and the action space of the agents. Especially in complex optimization problems where continuous and high-dimensional state spaces are observed by a large number of agents, these methods may involve combinatorial explosion, and have difficulties in efficiently finding the optimal solution. Meanwhile, traditional reinforcement learning methods struggle with insufficient exploration and poor convergence performance when addressing the target encirclement problem of multiple UAVs, particularly in complex environments with sparse or imprecise rewards. The design of efficient learning frameworks and significant reward functions has been the focus of current research in reinforcement learning methods.

III. System Model and Problem Formulation

In this section, we first build the system model used in the multi-UAV cooperative target encirclement problem, then analyze the constraints and objectives and abstract the studied problem as a multi-constraint combinatorial optimization one.

A. System Model

In this work, n heterogeneous UAVs $U = \{U_1, U_2, \ldots, U_n\}$ are adopted to fly to and encircle a moving target G in an environment containing m static obstacles $O = \{O_1, O_2, \ldots, O_m\}$. UAVs are autonomous and have different flight speeds and energy consumption performances. In order to simplify the UAV model, we assume that all UAVs fly at the same height and the UAVs' physical attributes, such as sharps, sizes, and colors, are neglected [27]. The dynamics model of a UAV U_i can be defined as [28]:

$$\begin{cases} \dot{\boldsymbol{p}}_i(t) = \boldsymbol{v}_i(t) \\ \dot{\boldsymbol{v}}_i(t) = \alpha_p \boldsymbol{p}_i(t) + \alpha_v \boldsymbol{v}_i(t) + \boldsymbol{u}_i(t) \end{cases} \tag{1}$$

where $\boldsymbol{p}_i(t) \in \mathbb{R}^2$, $\boldsymbol{v}_i(t) \in \mathbb{R}^2$, and $\boldsymbol{u}_i(t) \in \mathbb{R}^2$ respectively denote the position, velocity, and control input vector of UAV

U_i at a time point t. α_p and α_v are damping factors. $\boldsymbol{p}_i^S$ and d_i^s are used to represent the starting position of UAV U_i when the target encirclement mission begins, and the safety distance of UAV U_i to obstacles or other UAVs, respectively.

In practical applications, propulsion energy is required for UAVs to remain aloft and fly to the specified positions. As introduced in [29] and [30], the propulsion power of UAVs is composed of three aspects: the blade profile power spent in overcoming the profile drag resulting from blade rotation, the induced power used to overcome the induced drag and produce a lifting force, and the parasite power spent in overcoming the parasite friction drag resulting from moving in the air. Thus, the propulsion power of UAVs could be modeled as:

$$P(V) = \underbrace{c_1(1+c_2V^2)}_{\text{blade profile}} + \underbrace{c_3\left(\sqrt{1+\frac{V^4}{c_4^2}}-\frac{V^2}{c_4}\right)^{\frac{1}{2}}}_{\text{induced}} + \underbrace{c_5V^3}_{\text{parasite}} \tag{2}$$

where V is the flying speed of UAVs, and factors c_1 to c_5 represent the modeling parameters depending on both the inherent characteristics of UAVs and the environmental conditions. E_i^{max} is used to denote the maximum available energy consumption of U_i in the target encirclement mission.

In this work, static obstacles are composed of enemy anti-aircraft artilleries, radar threat areas, and natural terrains. These obstacles can be characterized by a fixed geographical position and would not move during the encirclement task. To simplify the obstacle model, we abstract static obstacles as circles with fixed radii, and each static obstacle O_j is represented by a tuple $O_j = \langle \boldsymbol{p}_j, r_j \rangle$, where $\boldsymbol{p}_j$ and r_j respectively denote the two-dimensional coordinate of the center and the radius of the obstacle O_j. Meanwhile, the moving target G is regarded as a point mass with an initial position $\boldsymbol{p}_G$ and a fixed moving speed $\boldsymbol{v}_G$. Then the position of the target at time point t can be calculated via $\boldsymbol{p}_G(t) = \boldsymbol{p}_G + \boldsymbol{v}_G t$.

B. Problem Formulation

As shown in Fig. 2, the goal of the multi-UAV cooperative target encirclement problem is to design an effective flight path for each UAV such that the moving target would be evenly and efficiently surrounded while avoiding collision with obstacles. When flight paths of UAVs are designed, the following constraints should be met:

1) Collision avoidance constraint: UAVs should not collide with obstacles, other UAVs, or the target, i.e., at any time point, the distance between each two UAVs or between a UAV and an obstacle should be longer than the predefined safety distance. We use T to denote the time consumption in achieving the target encirclement mission, and this constraint can be expressed via:

$$\begin{cases} \forall t \in [0,T],\ \forall i \in [1,n], \\ D\big(\boldsymbol{p}_i(t), \boldsymbol{p}_j(t)\big) > d_i^s,\ \forall j \in [1,n],\ j \neq i \\ D\big(\boldsymbol{p}_i(t), \boldsymbol{p}_j\big) > d_i^s + r_j,\ \forall j \in [1,m] \\ D\big(\boldsymbol{p}_i(t), \boldsymbol{p}_G(t)\big) > d_i^s \end{cases} \tag{3}$$

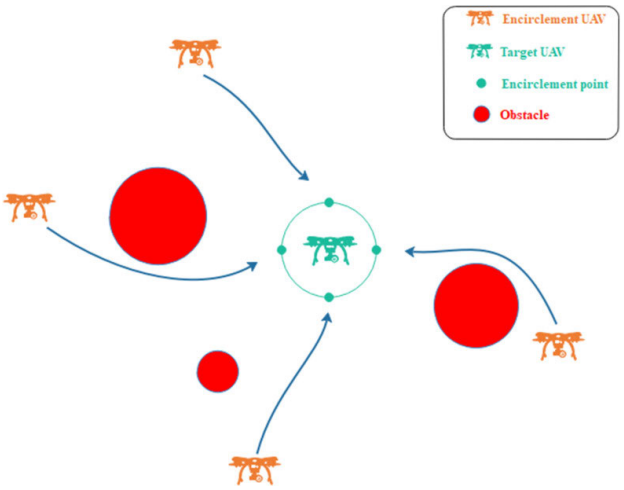

Fig. 2. A target encirclement mission achieved by four UAVs.

where $D(x, y)$ represents the Euclidean distance between the two-dimensional positions x and y.

2) Motion continuity constraint: Each UAV should take off from its starting position and continuously update its motion states until achieving the target encirclement mission. When the moving target is encircled, all UAVs should keep a fixed distance from the target and have the same moving speeds as the target. Since UAVs and the target are modeled as particles, this constraint can be expressed as:

$$\begin{cases} \forall t \in [0,T],\ \forall i \in [1,n], \\ \boldsymbol{p}_i(0) = \boldsymbol{p}_i^S,\ \boldsymbol{v}_i(0) = 0 \\ D\big(\boldsymbol{p}_i(T), \boldsymbol{p}_G(T)\big) = D_f,\ \boldsymbol{v}_i(T) = \boldsymbol{v}_G \\ \boldsymbol{v}_i(t+\Delta t) = \boldsymbol{v}_i(t) + \boldsymbol{u}_i(t)\Delta t \\ \boldsymbol{p}_i(t+\Delta t) = \boldsymbol{p}_i(t) + \frac{1}{2}\big(\boldsymbol{v}_i(t) + \boldsymbol{v}_i(t+\Delta t)\big)\Delta t \end{cases} \tag{4}$$

where D_f represents the predefined encirclement distance from UAVs to the target, and Δt denotes a very short time interval.

3) Energy consumption constraint: Each heterogeneous UAV should complete its target encirclement mission before its energy runs out. As we pointed out in Sect. III-A, propulsion energy is required for UAVs to remain aloft and fly to their destinations. We need to calculate the total energy of each UAV spent in the encirclement mission and ensure the energy consumption does not exceed the available value. With the propulsion power model built in Sect. III-A, this constraint can be expressed via:

$$\forall i \in [1,n],\ \int_0^T P\big(\boldsymbol{v}_i(t)\big)dt \leq E_i^{max} \tag{5}$$

The multi-UAV cooperative target encirclement problem studied in this work is defined as seeking a safe flight path for each UAV such that the moving target would be quickly encircled while satisfying the collision avoidance, motion continuity, and energy consumption constraints, which can be formally formulated as:

$$\begin{aligned} &\min \quad T \\ &s.t.\ (3)\ ,\ (4)\ ,\ (5) \end{aligned} \tag{6}$$

Although the above formulation can find an optimal flight path for each UAV to encircle the moving target efficiently, it has to search all possible solutions and requires a great deal of time, which is unacceptable in practical applications. Meanwhile, the above formulation cannot adapt to the uncertain change of tasks and environments, resulting in unsatisfactory solutions that cannot be efficiently adopted in complex and dynamic scenarios. In the following, an extrinsic-and-intrinsic reward-based multi-agent reinforcement approach is proposed to improve the efficiency and adaptability of problem-solving.

IV. Extrinsic-and-Intrinsic Reward-Based Multi-Agent Reinforcement Learning Framework

In this section, we first transform the multi-UAV cooperative target encirclement problem into a Markov game and then propose an extrinsic-and-intrinsic reward-based multi-agent reinforcement learning framework to settle the problem and efficiently achieve the encirclement mission of moving targets.

A. Problem Transformation

The target encirclement problem of multiple heterogeneous UAVs can be described as a Markov game and is denoted by a quintuple $\langle \boldsymbol{S}, \boldsymbol{A}, \boldsymbol{R}, \boldsymbol{P}, \gamma \rangle$. $\boldsymbol{S} = S_1 \times S_2 \times \ldots \times S_n$ represents the state space of all heterogeneous UAVs, and S_i signifies the state space of UAV U_i. $\boldsymbol{A} = A_1 \times A_2 \times \ldots \times A_n$ denotes the action space of all UAVs in the environment, and A_i is the action space of UAV U_i. $\boldsymbol{R} = \{R_1, R_2, \ldots, R_n\}$ is the collection of reward functions of each UAV. $\boldsymbol{P} = \boldsymbol{S} \times \boldsymbol{A} \times \boldsymbol{S}$ signifies the state transition probabilities of all UAVs, and γ denotes the discount factor for the reward functions. Each UAV is regarded as an independent agent in this work, and its state space, action space, and reward function are defined as follows.

1) State Space S: The environmental information observed by every UAV is composed of four aspects: its own position and velocity, the positions and velocities of its neighboring UAVs, the positions of detected obstacles, and the position and speed of the moving target. We use s_i^t to represent the environmental states of UAV i at time point t, then s_i^t could be calculated via:

$$s_i^t = \Big\{ \underbrace{\boldsymbol{p}_i(t), \boldsymbol{v}_i(t)}_{\text{itself}}, \underbrace{\{\boldsymbol{p}_j(t), \boldsymbol{v}_i(t)\}_{j \neq i}}_{\text{neighbouring UAVs}}, \underbrace{\{\boldsymbol{p}_k\}_{k \in [1,m]}}_{\text{obstacles}}, \underbrace{\boldsymbol{p}_G(t), \boldsymbol{v}_G(t)}_{\text{target}} \Big\} \tag{7}$$

2) Action Space A: As introduced in Sect. III-A, UAVs can update their motion states (i.e., positions and velocities) by quantitatively setting the parameters in the control input vector $\boldsymbol{u}$ according to Eq. (1). Therefore, the action of any UAV U_i at time point t, denoted by a_i^t, can be written as:

$$a_i^t = \{\boldsymbol{u}_i(t)\} \tag{8}$$

3) Reward Function R: Reward functions should be reasonably constructed by fully considering the constraints and objectives of the studied problem. The goal of the multi-UAV cooperative target encirclement problem in this work is to seek an effective flight path for each UAV such that the moving target would be efficiently surrounded while satisfying the collision avoidance, motion continuity, and energy consumption constraints. Therefore, we analyze the three constraints and the objective and use $r_i^{ca}(t)$, $r_i^{mc}(t)$, $r_i^{ec}(t)$, and $r_i^{ob}(t)$, to represent the rewards derived from the collision avoidance constraint, motion continuity constraint, energy consumption constraint, and the objective, respectively, at the time point t. Now, we analyze the calculation process of these four types of rewards by using a relative motion model, which is a mathematical model describing the motion laws and characteristics of objects in relative motion. In this model, the motion state of UAVs is described by determining the positions, velocities, and accelerations relative to the reference object.

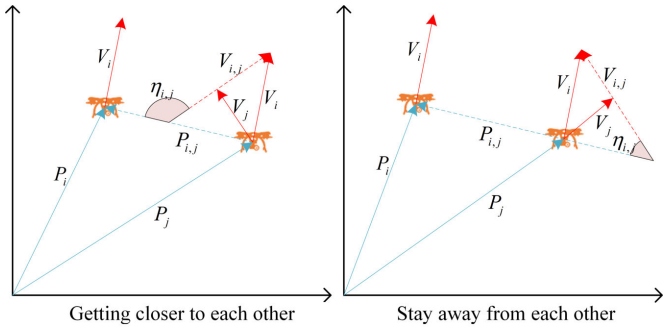

Fig. 3. The relative yaw angle in the relative motion model of two UAVs.

We use $\boldsymbol{P}_{i,j}(t)$, $\boldsymbol{V}_{i,j}(t)$, and $\eta_{i,j}(t)$ to respectively denote the relative position, the relative velocity, and the relative yaw angle of UAV U_i with respect to UAV U_j at the time point t, i.e.,

$$\begin{cases} \boldsymbol{P}_{i,j}(t) = \boldsymbol{p}_i(t) - \boldsymbol{p}_j(t) \\ \boldsymbol{V}_{i,j} = \boldsymbol{v}_i(t) - \boldsymbol{v}_j(t) \\ \eta_{i,j} = \arccos\left(\dfrac{\boldsymbol{P}_{i,j}(t) \cdot \boldsymbol{V}_{i,j}(t)}{|\boldsymbol{P}_{i,j}(t)| \times |\boldsymbol{V}_{i,j}(t)|} \right) \end{cases} \tag{9}$$

The probability of collisions between UAVs is influenced not only by their relative positions but also by the relative yaw angles. As depicted in Fig. 3, when the relative yaw angle is bigger than $-\frac{\pi}{2}$ and smaller than $\frac{\pi}{2}$, i.e., $-\frac{\pi}{2} \leq \eta_{i,j}(t) \leq \frac{\pi}{2}$, UAV U_i moves away from UAV U_j and the probability of collision reduces; otherwise, when $\eta_{i,j}(t) \in [-\pi, -\frac{\pi}{2}] \cup [\frac{\pi}{2}, \pi]$, UAV U_i flies towards to UAV U_j and the two UAVs are more likely to collide. Therefore, the reward resulting from the collisions between UAV U_i and other UAVs (or dynamic obstacles) at time point t, denoted by $r_i^{uav}(t)$, can be calculated via:

$$r_i^{uav}(t) = \sum_{j=1, j \neq i}^{n} \left(\left(1 - \exp^{-|\boldsymbol{P}_{i,j}|}\right) + \cos \eta_{i,j} \right) \tag{10}$$

Since the static obstacles can be regarded as a special UAV with a fixed velocity, the reward resulting from the collisions between UAV U_i and static obstacles at time point t, denoted by $r_i^{obs}(t)$, can be calculated via:

$$r_i^{obs}(t) = \sum_{k=1}^{m} \left(\left(1 - \exp^{-|\boldsymbol{P}_{i,k}|}\right) + \cos \eta_{i,k} \right) \tag{11}$$

Combining Eq. (10) and Eq. (11) together, the reward of UAV U_i derived from the collision avoidance constraint can be computed via:

$$r_i^{ca} = \alpha_1 r_i^{uav}(t) + \alpha_2 r_i^{obs}(t) \tag{12}$$

where α_1 and α_2 are adjustable weighting parameters.

Since the goal of UAVs is to fly to and encircle the moving target, the desired distances between UAVs and the target are the predefined encirclement distance D_f, and the desired velocities of UAVs are the target's current velocity. Therefore, the reward resulting from the motion continuity constraint, denoted by $r_i^{mc}(t)$, can be expressed via:

$$r_i^{mc}(t) = \underbrace{\varphi_1 \exp^{D_f - |\boldsymbol{P}_{i,j}|}}_{\text{position reward}} + \underbrace{\varphi_2 \exp^{-|\boldsymbol{V}_{i,j}|}}_{\text{velocity reward}} \tag{13}$$

where φ_1 and φ_2 are predefined weight factors. Please note that the values of factors φ_1 and φ_2 may change with an objective of quickly approaching the target or following the movement of the target.

The energy consumption constraint requires that all UAVs carry out the target encirclement mission before their energies run out. Since the maximum allowable energy can be viewed as a static obstacle, the reward resulting from the energy consumption constraint can be stated as:

$$r_i^{ec}(t) = \begin{cases} -\frac{1}{2}\xi\left(\frac{1}{E_i^t} - \frac{1}{E_i^{max}}\right)^2, & E_i^t > E_i^{max} \\ 0, & E_i^t \le E_i^{max} \end{cases} \tag{14}$$

where ξ is a fixed repulsive coefficient for energy consumption, and E_i^t is currently allowable energy of UAV i at time point t.

In order to encourage UAVs to successfully complete the target encirclement task, we give a very attractive reward $r_i^{ob}(t)$ to UAVs that correctly fly to the desired position and follow the movement of the target:

$$r_i^{ob}(t) = \begin{cases} R_c, & \text{UAV } U_i \text{ successfully encircles the target} \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

where R_c is a positive real number such as 10000.

With the above discussion, the overall extrinsic reward of each UAV at time point t, denoted by r_i^{ex}, is calculated via:

$$r_i^{ex} = k_1 r_i^{ca}(t) + k_2 r_i^{mc}(t) + k_3 r_i^{ec}(t) + k_4 r_i^{ob}(t) \tag{16}$$

where k_1, k_2, k_3, and k_4 are adjustable weighting parameters that represent the positive weight of each reward.

B. Extrinsic-and-Intrinsic Reward Mechanism

In the practical applications of heterogeneous UAVs, it is often difficult to provide stable and effective rewards to continuously improve the policies and guide the learning steps. Especially in the scenario of multi-UAV cooperative target encirclements, the reinforcement signals of each agent are very sparse and cannot be exactly provided by fully evaluating the behaviors of each individual UAV in every step. This may lead to a slow learning problem or even make learning impossible.

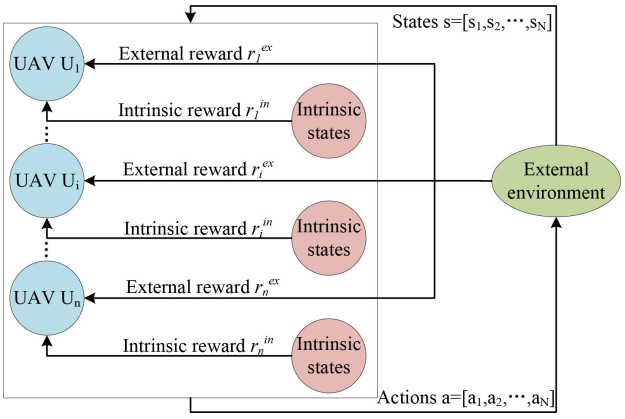

Fig. 4. External-and-intrinsic reward mechanism in the reinforcement learning.

Meanwhile, decisions in the large-scale UAV swarm often involve uncertainty and complexity. Traditional approaches, which produce the reinforcement signals solely by evaluating the effect of actions in the external environment, may be biased and incorrect. The setting of reinforcement signals should fully analyze the interaction and collaboration between intelligent agents to optimize the overall objectives of the UAV swarms. Therefore, in this work, we design an external-and-intrinsic reward mechanism to overcome the problems of sparse and inaccurate reinforcement signals in the learning process.

As shown in Fig. 4, the extrinsic-and-intrinsic reward mechanism emphasizes that the reward signals that guide the learning of an agent come from both extrinsic and intrinsic reward components. Extrinsic rewards are derived from the individual behaviors of agents contributing to the achievement of group goals (i.e., encircling the target here), representing the benefits of the UAV swarm. They are used to enhance the collaboration among agents to achieve the overall goals of UAVs. On the other hand, intrinsic rewards are the feedback signals that agents obtain from their own behavior based on changes in their states. They represent the benefits of each individual agent and are used to guide agents in autonomously exploring the environment and acquiring new knowledge.

As illustrated in Fig. 5, the framework of the proposed extrinsic-and-intrinsic reward-based multi-agent reinforcement learning approach is composed of n agents, where each agent controls the corresponding UAV and is implemented by the actor and the critic components similar to those used in the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm. By taking full advantage of the policy gradient, each agent i selects an action for each state according to the policy network μ_i with a parameter setting θ^{μ_i}. Then, the critic component adopts a state-action value network Q_i with a parameter setting θ^{Q_i} to evaluate the action selected by the actor. As in MADDPG, target networks and experience replay technology are used to enhance the stabilization and reliability in the learning process [31]. The parameter θ^{μ_i} of the policy network μ_i of agent i is updated by using gradient ascent to maximize the expected reward via:

$$\begin{aligned} \nabla_{\theta^{\mu_i}} J_i = \mathbb{E}_{\boldsymbol{o},\boldsymbol{a}\sim\mathcal{D}}[\nabla_{\theta^{\mu_i}} \mu_i(s_i|\theta^{\mu_i}) \\ \cdot \nabla_{ai} Q_i(\boldsymbol{o}, a_1, a_2, \ldots, a_n|\theta^{Q_i})|_{a_i=\mu_i(s_i)}] \end{aligned} \tag{17}$$

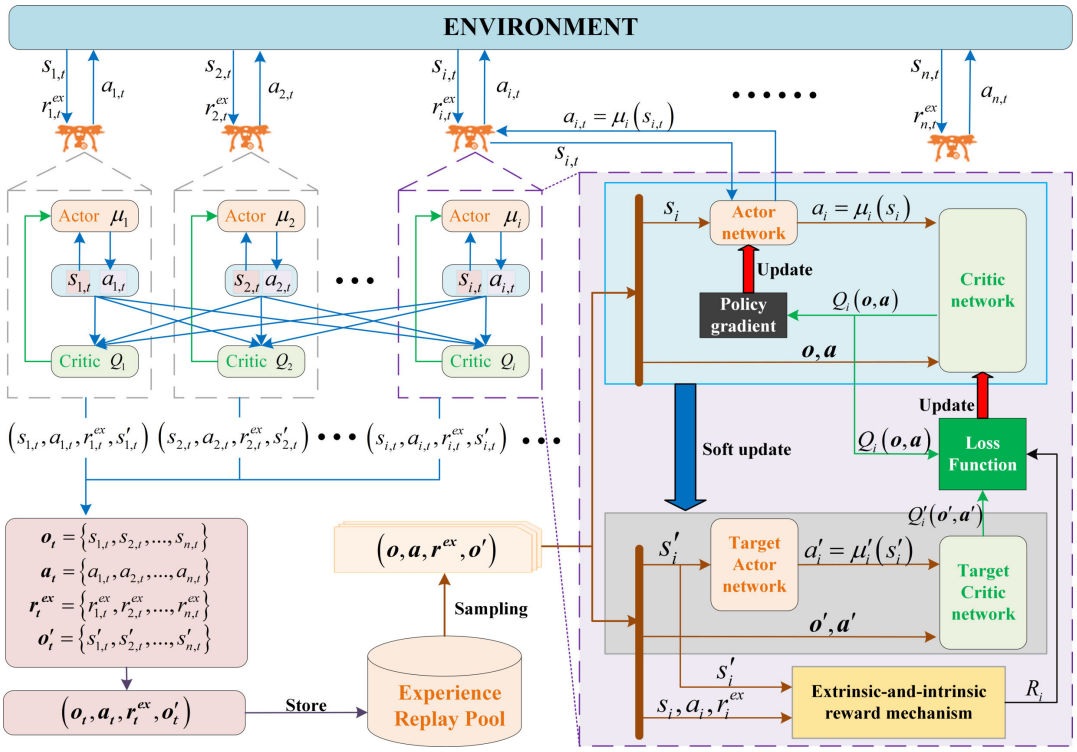

Fig. 5. Framework of the proposed extrinsic-and-intrinsic reward-based multi-agent reinforcement learning approach.

where $\mathcal{D}$ is the experience replay pool containing a substantial amount of experience sequences $(\boldsymbol{o}, \boldsymbol{a}, \boldsymbol{r}^{ex}, \boldsymbol{o}')$ from the UAVs. $\boldsymbol{o} = \{s_1, s_2, \ldots, s_n\}$ is the global states information of UAVs, $\boldsymbol{a} = \{a_1, a_2, \ldots, a_n\}$ represents the actions of all UAVs, $\boldsymbol{r}^{ex} = \{r_1^{ex}, r_2^{ex}, \ldots, r_n^{ex}\}$ denotes the external rewards obtained by UAVs when actions $\boldsymbol{a}$ are taken under the states $\boldsymbol{o}$. $\boldsymbol{o}' = \{s_1', s_2', \ldots, s_n'\}$ is the global states information at the next time step. Please note that in Eq. (17), all data, except the action a_i selected by agent i, comes from the experience replay pool $\mathcal{D}$. Correspondingly, the value network Q_i of agent i needs to be iteratively updated using time-series differencing with the following loss function:

$$\mathcal{L}(\theta^{Q_i}) = \mathbb{E}_{\boldsymbol{o},\boldsymbol{a},\boldsymbol{r}^{ex},\boldsymbol{o}'\sim\mathcal{D}}[(Q_i(\boldsymbol{o}, a_1, a_2, \ldots, a_n|\theta^{Q_i}) - \mathcal{Y}_i)^2] \tag{18}$$

$$\mathcal{Y}_i = R_i + \gamma \cdot Q_i'(\boldsymbol{o}', \mu_1'(s_1'|\theta^{\mu_1'}), \mu_2'(s_2'|\theta^{\mu_2'}), \ldots, \mu_n'(s_n'|\theta^{\mu_n'})|\theta^{Q_i'}) \tag{19}$$

where $Q_i'(\cdot|\theta^{Q_i'})$ and $\mu_i'(\cdot|\theta^{\mu_i'})$ represent the target value network and target policy network of agent i, respectively, and $\theta^{Q_i'}$ and $\theta^{\mu_i'}$ are their corresponding parameters. The target networks are used to mitigate the overestimation problem in reinforcement learning. R_i is the total reward value obtained by UAV U_i, which is generated by the extrinsic-and-intrinsic reward mechanism and is comprised of the extrinsic reward r_i^{ex} from the environment and the intrinsic one r_i^{in} from the intrinsic states.

Now, we introduce the base structure of the extrinsic-and-intrinsic reward mechanism, which fully exploits the advantages of the extrinsic environment and the intrinsic states. As shown in Fig. 6, our extrinsic-and-intrinsic reward

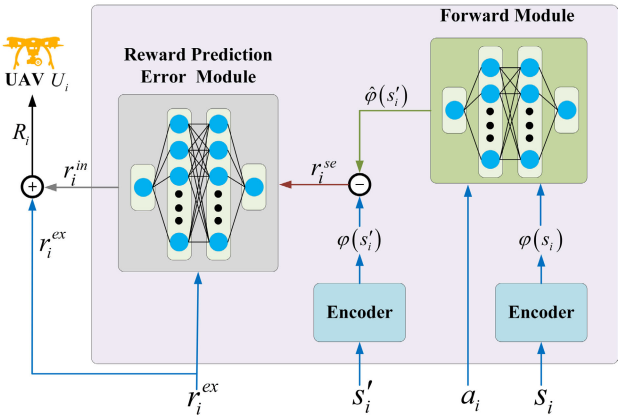

Fig. 6. Structure of the extrinsic-and-intrinsic reward mechanism.

mechanism is constituted of three aspects: an *encoder* which extracts and compresses features from the original state information for further processing and analysis, a *forward module* which is a neural network to predict the next state of the environment and a *reward prediction error module* that produces the intrinsic reward by analyzing the prediction errors.

As illustrated in Fig. 6, the extrinsic-and-intrinsic reward mechanism samples the current state s_i, action a_i, extrinsic reward r_i^{ex}, and the next state s_i' from the experience replay pool, and outputs the final reward R_i for agent i to update its networks. First, encoders are used to map the states s_i and s_i' to the corresponding features $\varphi(s_i)$ and $\varphi(s_i')$. Then, $\varphi(s_i)$ and a_i are input into the forward module to predict the state of agent i, denoted by $\varphi(\hat{s}_i')$, in the next step, i.e.,

$$\varphi(\hat{s}_i') = f_i(\varphi_i(s_i), a_i) \tag{20}$$

where f_i is a single-layer fully connected network for agent i and its parameters are updated by minimizing the difference between the features $\hat{\varphi}(s_i')$ and $\varphi(s_i')$. In this work, the loss function of the network f_i is defined as follows:

$$L(f_i) = \frac{1}{2}\left\|\hat{\varphi}(s_i') - \varphi(s_i')\right\|^2 \tag{21}$$

By comparing the similarity between $\varphi(s_i')$ and $\hat{\varphi}(s_i')$, the current state error reward r_i^{se} is calculated through the Euclidean norm (L2 norm):

$$r_i^{se} = \left\|\hat{\varphi}_i(s_i') - \varphi_i(s_i')\right\|_2^2 \tag{22}$$

Subsequently, the reward prediction error module outputs the intrinsic reward r_i^{in} by taking the external reward r_i^{ex} and the state error reward r_i^{se} as input:

$$r_i^{in} = g_i\left(r_i^{se}, r_i^{ex}\right) \tag{23}$$

where g_i is also a single-layer fully connected network for agent i to compare the differences between rewards r_i^{se} and r_i^{ex} and generates the intrinsic reward. Since the optimization objective of the reward prediction error module is to minimize the error between the external reward r_i^{ex} and the intrinsic reward r_i^{in}, the loss function of f_i can be expressed as:

$$L(g_i) = \frac{1}{2}\left\|r_i^{in} - r_i^{ex}\right\|^2 \tag{24}$$

Combining Eqs. (21) and (24) together, the loss function of the extrinsic-and-intrinsic reward mechanism can be stated as:

$$L_i = \lambda \cdot L(f_i) + (1-\lambda)\cdot L(g_i) \tag{25}$$

where $\lambda \in [0, 1]$ is used to balance the importance of the forward module and the reward prediction error module. The parameters of networks f_i and g_i are updated by minimizing the loss function L_i.

Finally, the reward R_i for agent i can be obtained by weighted summing the intrinsic reward r_i^{in} and the external reward r_i^{ex}:

$$R_i = \beta_{ex}\cdot r_i^{ex} + \beta_{in}\cdot r_i^{in} \tag{26}$$

where β_{ex} and β_{in} are represent the positive weight factors. Please note that although multiple reward functions are designed in our work, they are ultimately fused into a signal reward through a weighted combination.

According to the above discussion, the proposed extrinsic-and-intrinsic reward-based multi-agent reinforcement learning approach is summarized as Algorithm 1. We start by initializing the parameters of the value network, the policy network, the corresponding target networks (line 2), the networks f_i and g_i in the extrinsic-and-intrinsic reward mechanism (line 3), and the experience replay pool $\mathcal{D}$ (line 4). Then, the training process begins with a fixed number of episodes. At the beginning of each episode, the current environment state is obtained (line 7), and each episode is divided into several steps to continuously train the networks of each UAV (lines 8 to 25). In every step, each UAV selects an optimal action with its policy network (lines 9 to 11). After the joint action is performed (line 12), the extrinsic reward $\boldsymbol{r}_t^{ex}$ and the next environmental state $\boldsymbol{o}_{t+1}$ are obtained (line 13), and the newly generated transition $(\boldsymbol{o}_t, \boldsymbol{a}_t, \boldsymbol{r}_t^{ex}, \boldsymbol{o}_{t+1})$ is stored into the experience replay pool $\mathcal{D}$ (line 14). Finally, a batch of transitions is randomly chosen from the experience replay pool (line 15), and the parameters of the networks of each UAV are gradually updated (lines 16 to 24). The training process stops after the predefined iteration number is reached.

Algorithm 1 Pseudo Code of the Extrinsic-and-Intrinsic Reward-Based Multi-Agent Reinforcement Learning Approach (EIC-MARL)

```
1  for i = 1 to n do
2      Initialize the parameters of the value network, the policy network and the
       corresponding target networks of each UAV U_i;
3      Initialize the networks f_i and g_i in the extrinsic-and-intrinsic reward
       mechanism of each UAV;
4      Initialize the experience replay pool D;
5  end
6  for episode = 1 to M do
7      Get the current environmental state o_0;
8      for t = 0 to T do
9          for i = 1 to n do
10             Select an action a_{i,t} according to the environmental state
               o_t = {s_{1,t}, ..., s_{n,t}};
11         end
12         Perform the joint action a_t = {a_{1,t}, ..., a_{n,t}};
13         Obtains the extrinsic reward r_t^{ex} and the next environmental state o_{t+1};
14         Store (o_t, a_t, r_t^{ex}, o_{t+1}) into the experience replay pool D;
15         Sample a batch of N_b transitions from the experience replay pool D;
16         for i = 1 to n do
17             Calculate the intrinsic reward according to Eq. (23);
18             Produce the overall reward according to Eq. (26);
19             Update the value network' parameters with Eqs. (18) and (19);
20             Update the parameters of the policy network with Eq. (17);
21             Softly update the parameters of the target networks;
22             Compute the loss function of the extrinsic-and-intrinsic reward
               mechanism according to Eq. (25);
23             Update the parameters of networks f_i and g_i in the
               extrinsic-and-intrinsic reward mechanism;
24         end
25     end
26 end
```

V. Simulation and Results

In this section, we establish a simulation environment to verify the performance of the proposed extrinsic-and-intrinsic reward-based multi-agent reinforcement learning approach by comparing it with three classic reinforcement learning algorithms, including MATD3 [32], MADDPG [33], and SAC [34]. In the following, we first introduce the experimental environment and parameter settings and then evaluate the effectiveness and practicality of our approach in terms of average reward, encirclement success rate, encirclement time, and encirclement energy consumption.

A. Experimental Environment and Parameter Settings

In our experiments, Pytorch 1.10 is chosen as the training environment for neural networks in reinforcement learning algorithms. All UAVs are heterogeneous, with different flight speeds and energy capabilities. The area where UAVs perform the target encirclement task is defined as a rectangular map with a length and width of 400 meters. In this map, the coordinates of UAVs, moving targets, and static obstacles are randomly generated. Furthermore, two hidden layers with 400 neurons are used for both the policy network and the value network of each UAV, and one hidden layer with 256 neurons is adopted for the networks in the extrinsic-and-intrinsic reward mechanism. Please note that all hidden

TABLE I
HYPER PARAMETERS USED IN THE SIMULATION

Parameters	Values
Number of UAVs	4
Number of obstacles	4
Learning rate of networks	0.001
Experience replay pool size	1000000
Batch size	128
Soft-update parameter	0.001
Number of episodes	10000
Number of steps in each episode	250

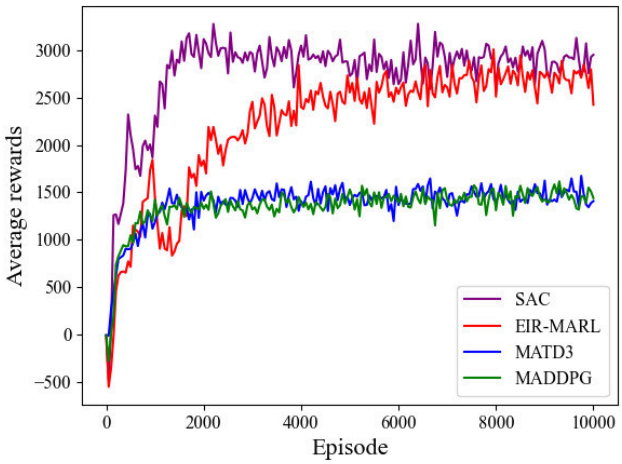

Fig. 7. Average rewards obtained by four approaches per episode in the training procedure.

layers adopt the Leaky Relu to achieve the activation function. The values of the remaining parameters in the experiment are shown in Table I.

B. Average Reward Evaluation

Average reward, which represents the mean value of rewards obtained by agents in each episode during the training process, is an outstanding feature of the convergence speed and learning efficiency of reinforcement learning algorithms [35]. The learning states of agents can be intuitively observed from the changes in average rewards, and a higher average reward indicates that the reinforcement learning algorithms are more efficient and convenient.

In the experiments of Fig. 7, the average rewards of the four approaches are measured per episode in the training procedure. As seen in Fig. 7, all four curves denoting the average rewards of the four approaches, have the same changing trend in that they first grow up with the episodes and then converge to an approximate equilibrium value. As expected, our approach exhibits a faster convergence speed and a higher average reward than the two multi-agent approaches, MATD3 and MADDPG. This is because the external-and-intrinsic reward mechanism adopted in our approach can stimulate the curiosity of UAVs about new states in the environment and encourage UAVs to fully explore the environment and find reasonable flight paths to encircle the target, leading to a fast convergence speed and avoiding falling into a local optimum.

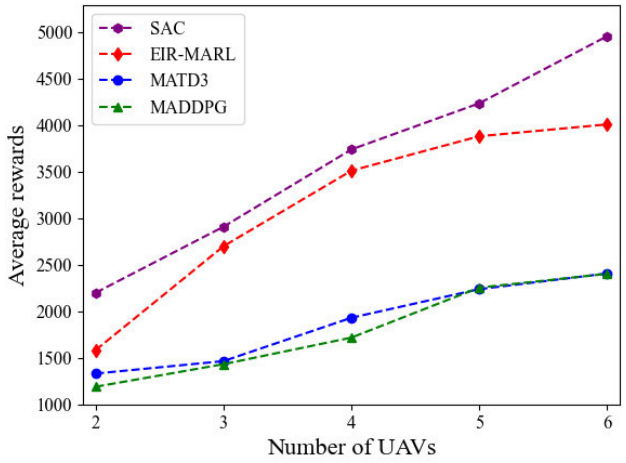

Fig. 8. Average rewards of four algorithms when the number of UAVs increases from 2 to 6.

Fig. 8 shows the average rewards of the four approaches with different numbers of UAVs. It is easy to find from Fig. 8 that as the number of UAVs increases, the average rewards of all approaches grow. One possible reason for this trend is that a larger UAV number leads to the generation of higher rewards in each training step to fly towards or encircle the moving target. When the number of UAVs is fixed, our approach obtains a higher average reward than the MADDPG and MATD3 approach. When three UAVs are adopted in the simulation experiments, the average reward value obtained by our approach is 2704.84, which is 1267.73 and 1234.79 higher than those of MADDPG and MATD3, respectively. Experiment results shown in Fig. 7 and Fig. 8 prove that our approach can obtain an excellent performance in terms of average reward.

C. Encirclement Success Rate Evaluation

In this subsection, we begin to evaluate the performance of our approach in terms of encirclement success rate, i.e., the percentage of task sets where the moving target can be successfully encircled according to the control strategies provided by learning-based approaches. Some tasks that cannot be successfully carried out by one method may be determined feasible and enforceable by other approaches. The encirclement success rate is an intuitive metric for measuring the efficacy and reliability of approaches in multi-UAV cooperative target encirclement tasks. A more reliable and accurate approach would produce a higher encirclement success rate under the same experimental conditions.

In the experiments of Fig. 9, the encirclement success rates of all four approaches are evaluated by varying the number of obstacles from 1 to 10. As can be seen, the encirclement success rates decrease gradually with the increase of the number of obstacles. One of the possible reasons is that a larger number of obstacles in environments results in a higher probability of collisions between UAVs and obstacles. Meanwhile, when the number of obstacles remains unchanged, our approach can achieve a relatively higher success rate than the others in most situations. This indicates that our

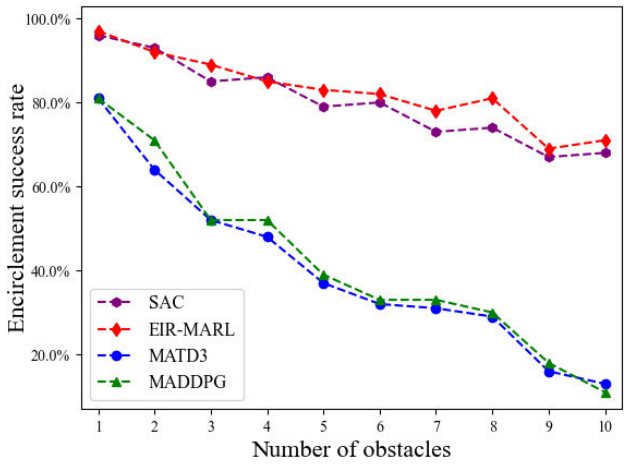

Fig. 9. Encirclement success rates of four approaches when the number of obstacles increases from 1 to 10.

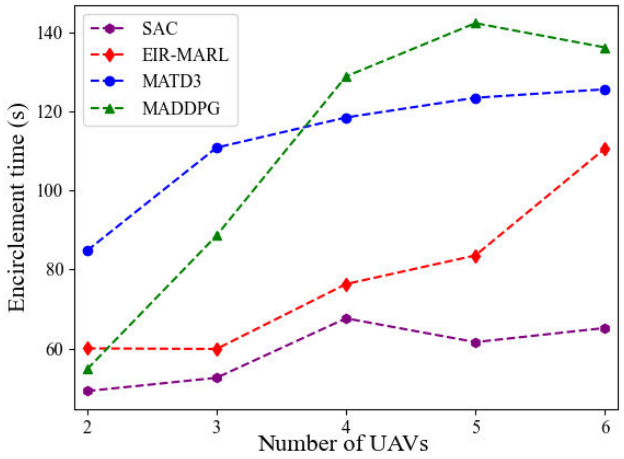

Fig. 11. Encirclement time of the four approaches when the number of UAVs changes.

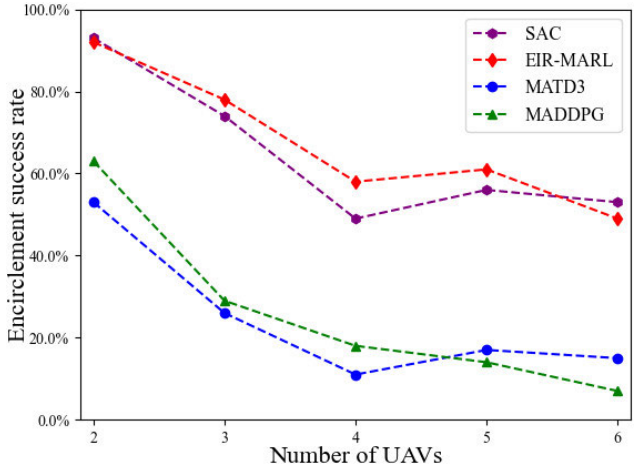

Fig. 10. Encirclement success rates of four approaches with different numbers of UAVs.

EIR-MARL approach can help UAVs in effectively and efficiently accomplishing the encirclement task of moving targets. When five obstacles are used in the target encirclement task, the encirclement success rate of our approach is 83%, which is 4%, 46% and 44% higher than those of SAC, MATD3, and MADDPG, respectively.

In Fig. 10, the encirclement success rates of the four approaches are evaluated by increasing the number of UAVs from 2 to 6. The maximum flight speeds of the UAVs are two times more than that of the moving target, and five obstacles with a radius of 15 meters are randomly generated during the initialization of the environment. From Fig. 10, we can find that the encirclement success rates of all four approaches decrease gradually with the number of UAVs. The reason is that with the increasing number of UAVs, the simultaneous arrival of all UAVs at the designated points becomes more difficult, resulting in a reduction in successfully encircling the moving targets. When the number of UAVs remains fixed, our approach is able to obtain a better encirclement success rate than the other approaches in most cases. For example, when four UAVs are tested in experiments, the encirclement success rate of our approach is 58%, which is 9%, 47%, and 40% higher than those of SAC, MATD3, and MADDPG, respectively. Results in Fig. 9 and Fig. 10 demonstrate that our approach performs better in the field of encirclement success rate.

D. Encirclement Time Evaluation

Now, we evaluate the performance of our approach by measuring the encirclement time, which is defined as the time consumption of UAVs in achieving the target encirclement tasks. As the optimization objective in the studied problem, encirclement time is a crucial symbol for evaluating the efficiency of the encirclement algorithms, and a solution that requires a shorter time to finish the target encirclement tasks is regarded to have stronger effectiveness and practicability.

Fig. 11 shows the encirclement time of the four approaches with different numbers of UAVs. In these experiments, the maximum flight speeds of the UAVs are two times more than that of the moving target, and four obstacles with a radius of 15 meters are randomly generated during the initialization of the environment. As shown in Fig. 11, the general trend of all curves representing the encirclement time of approaches increases with the number of UAVs. This trend can be attributed to the fact that with an increasing number of UAVs, the simultaneous arrival of all UAVs at the designated points becomes more difficult, resulting in an increase in the time required to complete the target encirclement tasks. However, when the number of UAVs remains unchanged, our approach can produce good enough solutions, and the encirclement time of our approach is shorter than that of MATD3 and MADDPG.

In the experiments of Fig. 12, the encirclement time of the four approaches is evaluated by varying the radius of the static obstacles. The maximum flight speeds of the UAVs are three times more than that of the moving target, and the initial positions of the four static obstacles are randomly generated. As expected, the encirclement time achieved by the four approaches has a similar trend and gradually grows with the increase of the obstacle radius. The reason is that with the increasing radius of the obstacles in the environment, UAVs need to plan a longer encirclement path to bypass the obstacles

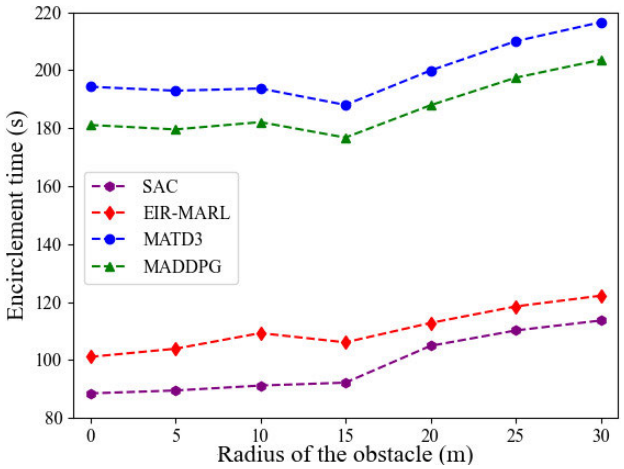

Fig. 12. Encirclement time of the four approaches when the radius of the static obstacles changes.

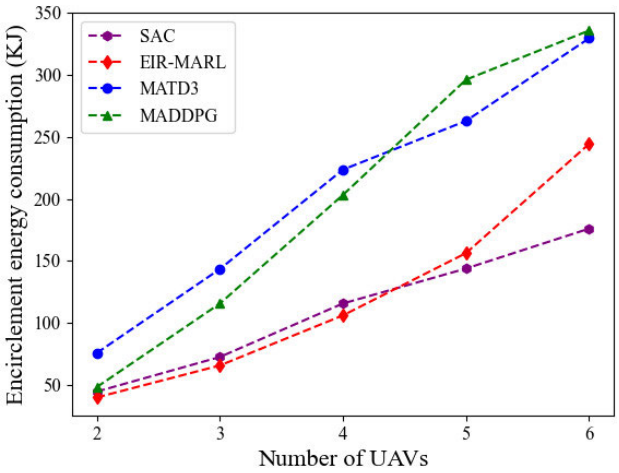

Fig. 13. Encirclement energy consumption of the four approaches when different numbers of UAVs are adopted.

and fly to the desired point while avoiding any collision with the obstacles. Therefore, UAVs should spend a longer time completing the target encirclement tasks.

Although the encirclement time of our approach is longer than that of the SAC algorithm, our approach is able to successfully finish the target encirclement task in a shorter time than the remaining two multi-agent reinforcement learning algorithms, MADDPG and MATD3. When the radius of the static obstacle is 20 meters, the encircling time achieved by our approach is 112.8 seconds, which is 39.97% and 43.57% shorter than that of MADDPG and MATD3, respectively. Experiment results shown in Fig. 11 and Fig. 12 demonstrate that our approach can perform better in terms of encirclement time.

E. Encirclement Energy Consumption Evaluation

Now, we evaluate the performance of our method by measuring the encirclement energy consumption, which is defined as the total energy consumption of all UAVs when they are adopted to complete the target encirclement tasks. Encirclement energy consumption is a crucial metric for evaluating the efficiency of encirclement algorithms, and a solution that requires less energy to finish the target encirclement tasks is considered to be more efficient and practical.

Fig. 13 shows the encirclement energy consumption of the four approaches with different numbers of UAVs. In these experiments, the maximum flight speeds of the UAVs are two times more than that of the moving target, and four obstacles with a radius of 15 meters are randomly generated during the initialization of the environment. We can see from Fig. 13 that the encirclement energy consumption of all four algorithms increases gradually with the number of UAVs from 2 to 6. Although the encirclement energy consumption of our algorithm is higher than the SAC algorithm in the case of more than four UAVs, the total energy consumption for the encirclement of our algorithm is close to the lowest compared to the other three algorithms when the number of UAVs does not exceed four. For example, when the number of UAVs is 4, the encirclement energy consumption of the

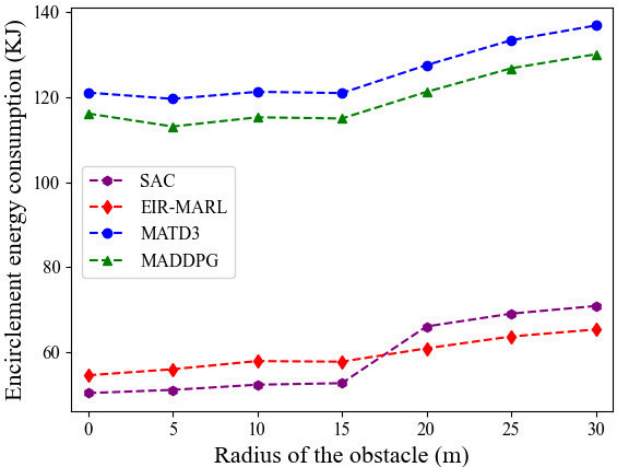

Fig. 14. Encirclement energy consumption of the four approaches when the radius of the static obstacles varies from 0 to 30.

EIR-MARL algorithm is 106.15 kilojoules, which is 8.29%, 47.76%, and 52.52% lower than that of SAC, MADDPG, and MATD3, respectively.

In the experiments of Fig. 14, the encirclement energy consumption of the four approaches is evaluated by varying the radius of the static obstacles from 0 meters to 30 meters. The maximum flight speeds of UAVs are three times more than that of the moving target, and the initial positions of the four static obstacles are randomly generated. From Fig. 14, we can find that the encirclement energy consumption of all approaches has the same trend that they grow gradually along with the increase of the obstacle radius. Similar to the results shown in Fig. 13, our approach has a lower energy consumption than the others when the obstacle radius is less than or equal to 15 meters and requires slightly higher energy than SAC when the obstacle radius is more than 15 meters. When the obstacle radius is 20 meters, the total energy consumption of our approach is 60.81 kilojoules, which is 7.81%, 49.85%, and 52.33% lower than that of SAC, MADDPG, and MATD3, respectively. Experiment results shown in Fig. 13 and Fig. 14 demonstrate that our approach can perform better in terms of encirclement energy consumption.

VI. Conclusion

This paper focuses on the cooperative target encirclement problem of heterogeneous UAVs and proposes a multi-agent reinforcement learning approach to provide optimal encirclement strategies. First, the constraints and objective functions of the multi-UAV cooperative target encirclement problem are analyzed, and the problem is formulated as a combinatorial optimization one with multiple constraints. Then, an extrinsic-and-intrinsic reward-based multi-agent reinforcement learning framework is proposed to seek an optimal and safe path for each UAV from the starting position to the target encirclement point, efficiently solving under-exploration and poor convergence problems of traditional reinforcement learning algorithms. Finally, experimental results validate the rationality of our analysis and demonstrate the effectiveness and practicality of our approach.

Although our approach can obtain a reasonable solution to solve the target encirclement problem, it still suffers from the space explosion problem when the number of UAVs and obstacles sharply increases. In the future, we will develop more technologies to compress the combined action and state space of agents and enhance the effectiveness of the proposed models and frameworks in large-scale applications. Meanwhile, we will collect more data from practical scenarios to train and optimize our models and try to provide technical support for the multi-field applications of intelligent unmanned systems.

References

[1] L. Zhang, X. Ma, Z. Zhuang, H. Xu, V. Sharma, and Z. Han, "Q-learning aided intelligent routing with maximum utility in cognitive UAV swarm for emergency communications," *IEEE Trans. Veh. Technol.*, vol. 72, no. 3, pp. 3707–3723, Mar. 2023.

[2] B. Yang, H. Shi, and X. Xia, "Federated imitation learning for UAV swarm coordination in urban traffic monitoring," *IEEE Trans. Ind. Informat.*, vol. 19, no. 4, pp. 6037–6046, Apr. 2023.

[3] Q. Luo, T. H. Luan, W. Shi, and P. Fan, "Edge computing enabled energy-efficient multi-UAV cooperative target search," *IEEE Trans. Veh. Technol.*, vol. 72, no. 6, pp. 7757–7771, Jun. 2023.

[4] Y. Gao, C. Bai, L. Zhang, and Q. Quan, "Multi-UAV cooperative target encirclement within an annular virtual tube," *Aerosp. Sci. Technol.*, vol. 128, Sep. 2022, Art. no. 107800.

[5] D. Schvartzman, J. D. Díaz, D. Zrnic, M. Herndon, M. B. Yeary, and R. D. Palmer, "Holographic back-projection method for calibration of fully digital polarimetric phased array radar," *IEEE Trans. Radar Syst.*, vol. 1, pp. 295–307, 2023.

[6] X. Fu, Q. Pan, and X. Huang, "AoI-Energy-Aware collaborative data collection in UAV-enabled wireless powered sensor networks," *IEEE Sensors J.*, vol. 23, no. 24, pp. 31307–31324, Dec. 2023.

[7] B. R. Kiran et al., "Deep reinforcement learning for autonomous driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4909–4926, Jun. 2022.

[8] M. Ran and L. Xie, "Adaptive observation-based efficient reinforcement learning for uncertain systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 10, pp. 5492–5503, Oct. 2022.

[9] B. Zhang, X. Sun, S. Liu, M. Lv, and X. Deng, "Event-triggered adaptive fault-tolerant synchronization tracking control for multiple 6-DOF fixed-wing UAVs," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 148–161, Jan. 2022.

[10] J. Chen, C. Du, Y. Zhang, P. Han, and W. Wei, "A clustering-based coverage path planning method for autonomous heterogeneous UAVs," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25546–25556, Dec. 2022.

[11] J. Chuzhoy and R. Zhang, "A new deterministic algorithm for fully dynamic all-pairs shortest paths," in *Proc. 55th Annu. ACM Symp. Theory Comput.*, Jun. 2023, pp. 1159–1172.

[12] H. Shen, N. Li, S. Rojas, and L. Zhang, "Multi-robot cooperative hunting," in *Proc. Int. Conf. Collaboration Technol. Syst. (CTS)*, Oct. 2016, pp. 349–353.

[13] A. T. Hafez, A. J. Marasco, S. N. Givigi, M. Iskandarani, S. Yousefi, and C. A. Rabbath, "Solving multi-UAV dynamic encirclement via model predictive control," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 6, pp. 2251–2265, Nov. 2015.

[14] A. Altan and R. Hacıoğlu, "Model predictive control of three-axis gimbal system mounted on UAV for real-time target tracking under external disturbances," *Mech. Syst. Signal Process.*, vol. 138, Apr. 2020, Art. no. 106548.

[15] J. Liu, Z. Wu, J. Yu, and Z. Xue, "Cooperative target tracking in aquatic environment using dual robotic dolphins," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 8, pp. 4782–4792, Aug. 2021.

[16] C. Zhang, W. Zhou, W. Qin, and W. Tang, "A novel UAV path planning approach: Heuristic crossing search and rescue optimization algorithm," *Expert Syst. Appl.*, vol. 215, Apr. 2023, Art. no. 119243.

[17] L. Bin, C. Wei, and H. Duan, "Grouped attack strategy of multi-UAV imitating hawk hunting behaviors," in *Proc. IEEE Int. Conf. Unmanned Syst. (ICUS)*, Oct. 2022, pp. 1437–1442.

[18] J. Chen, P. Han, Y. Zhang, T. You, and P. Zheng, "Scheduling energy consumption-constrained workflows in heterogeneous multiprocessor embedded systems," *J. Syst. Archit.*, vol. 142, Sep. 2023, Art. no. 102938.

[19] M. D. Phung and Q. P. Ha, "Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization," *Appl. Soft Comput.*, vol. 107, Aug. 2021, Art. no. 107376.

[20] A. Altan, "Performance of metaheuristic optimization algorithms based on swarm intelligence in attitude and altitude control of unmanned aerial vehicle for path following," in *Proc. 4th Int. Symp. Multidisciplinary Stud. Innov. Technol. (ISMSIT)*, Oct. 2020, pp. 1–6.

[21] J. Zhang, Y. Cui, and J. Ren, "Dynamic mission planning algorithm for UAV formation in battlefield environment," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 4, pp. 3750–3765, Aug. 2023.

[22] J. Chen, F. Ling, Y. Zhang, T. You, Y. Liu, and X. Du, "Coverage path planning of heterogeneous unmanned aerial vehicles based on ant colony system," *Swarm Evol. Comput.*, vol. 69, Mar. 2022, Art. no. 101005.

[23] W. Zhang, K. Song, X. Rong, and Y. Li, "Coarse-to-fine UAV target tracking with deep reinforcement learning," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 4, pp. 1522–1530, Oct. 2019.

[24] D. Ebrahimi, S. Sharafeddine, P.-H. Ho, and C. Assi, "Autonomous UAV trajectory for localizing ground objects: A reinforcement learning approach," *IEEE Trans. Mobile Comput.*, vol. 20, no. 4, pp. 1312–1324, Apr. 2021.

[25] H. Xiong and Y. Zhang, "Reinforcement learning-based formation-surrounding control for multiple quadrotor UAVs pursuit-evasion games," *ISA Trans.*, vol. 145, pp. 205–224, Feb. 2024.

[26] R. Zhang, Q. Zong, X. Zhang, L. Dou, and B. Tian, "Game of drones: Multi-UAV pursuit-evasion game with online motion planning by deep reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 10, pp. 7900–7909, Oct. 2023.

[27] J. Chen et al., "Global-and-Local attention-based reinforcement learning for cooperative behaviour control of multiple UAVs," *IEEE Trans. Veh. Technol.*, vol. 73, no. 3, pp. 4194–4206, Mar. 2024.

[28] J. Hu, M. Wang, C. Zhao, Q. Pan, and C. Du, "Formation control and collision avoidance for multi-UAV systems based on Voronoi partition," *Sci. China Technological Sci.*, vol. 63, no. 1, pp. 65–72, Jan. 2020.

[29] N. Gao et al., "Energy model for UAV communications: Experimental validation and model generalization," *China Commun.*, vol. 18, no. 7, pp. 253–264, Jul. 2021.

[30] Z. Yang, W. Xu, and M. Shikh-Bahaei, "Energy efficient UAV communication with energy harvesting," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1913–1927, Feb. 2020.

[31] H. Peng and X. Shen, "Multi-agent reinforcement learning based resource management in MEC- and UAV-assisted vehicular networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 131–141, Jan. 2021.

[32] S. Zhou, Y. Cheng, X. Lei, Q. Peng, J. Wang, and S. Li, "Resource allocation in UAV-assisted networks: A clustering-aided reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 71, no. 11, pp. 12088–12103, Nov. 2022.

[33] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, 2017, pp. 6382–6393.

[34] J. Duan, Y. Guan, S. E. Li, Y. Ren, Q. Sun, and B. Cheng, "Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6584–6598, Nov. 2022.

[35] J. Chen, Y. Zhang, L. Wu, T. You, and X. Ning, "An adaptive clustering-based algorithm for automatic path planning of heterogeneous UAVs," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16842–16853, Sep. 2022.

Jinchao Chen (Member, IEEE) received the Ph.D. degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2016. He is currently an Associate Professor with the School of Computer Science, Northwestern Polytechnical University. He has published four highly-cited papers and more than 70 technical research papers in top-cited journals and conferences which are cited more than 1000 times at well-known researchers. His research interests include multi-core and multi-processor scheduling, embedded and real-time systems, simulation and verification, decision-making, and intelligent control of unmanned aerial vehicles. He works as the editor-in-chief of two internal research journals and the editorial board member of several high-reputed journals.

Yang Wang received the bachelor's degree from the School of Computer Science and Technology, Guizhou University, in 2023. He is currently a Research Assistant and a Postgraduate Student with the School of Computer Science, Northwestern Polytechnical University. His research interests include path planning, multi-agent reinforcement learning, intelligent control of unmanned aerial vehicles, autonomous control, situation awareness, real-time distributed computing systems, and workflow scheduling.

Ying Zhang (Member, IEEE) received the Ph.D. degree in computer science and technology from the College of Computer Science and Electronic Engineering, Hunan University, in 2020. He is currently an Associate Professor with the School of Computer Science, Northwestern Polytechnical University, China. His research interests include energy-aware scheduling, real-time scheduling, autonomous control, situation awareness, real-time distributed computing systems, and intelligent control of unmanned systems.

Yantao Lu (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Syracuse University in 2020. He is currently an Associate Professor with the School of Computer Science, Northwestern Polytechnical University, China. His research interests include perception of autonomous driving, complex networks, intelligent control of unmanned systems, adversarial examples in deep neural networks, and computer vision.

Qiuhao Shu received the bachelor's degree from the School of Computer Science, Northwestern Polytechnical University, in 2023. He is currently a Research Assistant and a Postgraduate Student with the School of Computer Science, Northwestern Polytechnical University. His research interests include collaborative perception of unmanned systems and computer vision.

Yujiao Hu (Member, IEEE) received the bachelor's and Ph.D. degrees from the Department of Computer Science, Northwestern Polytechnical University, Xi'an, China, in 2016 and 2021, respectively. From November 2018 to March 2020, she was a Visiting Ph.D. Student with the National University of Singapore. Currently, she is a Faculty Member with Purple Mountain Laboratories. Her research interests include deep learning, edge computing, multi-agent cooperation problems, and time-sensitive networks.