

Article

Multi-Agent Reinforcement Learning for Multi-UAV Pursuit with Full Planar Motion and a Limited Detectable Region

Soobin Huh , Sungwon Lim , Hyeokjae Jang , Woohyun Byun, Suhyeong Yu  and Woonchul Nam * 

Department of Mechanical Engineering, Chung-Ang University, Seoul 06974, Republic of Korea; hsb820@cau.ac.kr (S.H.); tj028@cau.ac.kr (S.L.); skmsgurwoek@cau.ac.kr (H.J.); imbwh@cau.ac.kr (W.B.); dbtngud@cau.ac.kr (S.Y.)

* Correspondence: wcnam@cau.ac.kr

Abstract

Although previous studies have considered sensing constraints and UAV dynamics, most of them have used unrealistic sensing limitations and simplified dynamic models. Thus, these approaches can suffer from a significant discrepancy between simulation results and real-world deployment. To address this issue, this study incorporates high-fidelity sensing constraints and UAV dynamics into a multi-agent reinforcement learning approach, focusing on the practical interplay between FOV limitations and pursuit strategies. First, the proposed reward considers the sensing constraints via a gaze-alignment reward, which varies with the field-of-view condition, and a capturability reward that encourages transitions toward a capturable region. Second, realistic UAV dynamics, including lateral motion, forward motion, and yawing, are modeled in a simulation environment to reduce the sim-to-real gap. Quantitative evaluations demonstrated that the proposed formulation significantly improved the capture performance under diverse sensing conditions. The capturability reward increases the capture success rate by 11.4%. When the maximum speed of the evading UAV was 2 m/s faster than that of the pursuing UAVs, all capture trials failed when lateral motion was not used. However, when lateral motion was enabled, the success rate increased to 99.2%, highlighting the need for lateral motion.

Keywords: multi-UAV; cooperative pursuit; multi-agent reinforcement learning; MAPPO; pursuit evasion; limited detectable region

1. Introduction

Unmanned aerial vehicles (UAVs) have been widely adopted across both military and civilian domains, driven by the miniaturization and enhancement of computing hardware, as well as advances in artificial intelligence (AI)-enabled autonomous flight technologies [1–3]. Additionally, decreasing UAV costs have made them easily accessible to the public for recreational activities, including aerial photography and drone racing [1,4,5]. However, as UAV accessibility and usage continue to expand, potential threats involving UAVs, such as privacy violations, illegal filming, air traffic disruptions around airports, and terrorist attacks, are also increasing [6–8]. Therefore, there is an increasing need for anti-drone technologies that can rapidly and safely track and neutralize unauthorized or hostile drones.

This study considered a capture-based anti-drone scenario in which a pursuer UAV is equipped with a capture device, such as a net gun, to neutralize a target UAV during



Academic Editor: Tao Li

Received: 27 February 2026

Revised: 28 March 2026

Accepted: 6 April 2026

Published: 8 April 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

flight [9,10]. However, pursuing a target UAV using only a single pursuer UAV faces fundamental limitations owing to limited field-of-view (FOV) and finite detection ranges [11]. As the target UAV becomes faster and more maneuverable, it becomes increasingly challenging to maintain continuous target detection within the detectable region. Under such constraints, single-UAV interception is prone to target loss and can suffer substantial performance degradation against fast and agile targets [12]. These limitations motivate the use of cooperative multi-UAV pursuit-and-capture strategies.

Multi-UAV target pursuit is typically formulated as a pursuit–evasion game [13–19], which provides a natural framework for defining agent interactions, objectives, success and failure conditions, and optimization targets [20]. Among the various approaches for solving pursuit–evasion games, multi-agent reinforcement learning (MARL) is particularly suitable because it enables decentralized decision-making under partial observations while optimizing team-level objectives such as capture success rate and capture time [21,22]. In addition, MARL can learn cooperative interaction patterns directly from simulated experience, making it a promising approach for cooperative multi-UAV pursuit and capture problems [22].

With rapid advancements in MARL algorithms, MARL-based target searches or pursuits have been extensively studied recently. However, many of these studies have adopted simplified environmental abstractions and sensing or motion models. Some studies have formulated search-and-track tasks for grid-based worlds. Wei et al. [23] proposed an efficient minimum-time swarm search framework for dynamic targets, and Xu and Dang [24] analyzed emergent behaviors in a grid-world pursuit–evasion setting. Although such formulations are useful for studying high-level coordination, their discrete state–action representations limit realism for continuous UAV motion and real-world transfer.

Other studies have considered continuous action spaces to better reflect motion dynamics. De Souza et al. [15] and Chen et al. [18] investigated cooperative capture scenarios in which multiple pursuers surrounded an evader. They reported improved performance and training stability via reward designs, tailored learning schemes, and predictive components. Zhang et al. [17] further considered an urban setting and introduced a long short-term memory (LSTM)-based architecture with a more realistic drone dynamics model. Nevertheless, many of these studies still rely on simplified sensing and motion assumptions, such as direct access to evader position information or the omission of lateral motion through unicycle-based dynamics. As a result, pursuit under limited FOV and full multi-rotor maneuverability remains insufficiently explored.

More recently, Peng et al. [19] considered multi-agent capture under limited FOVs and designed phase-dependent rewards within a unified framework. They also introduced a Normalizing Flow Actor with a Graph Attention Critic to represent complex policy distributions and spatial interactions. However, the motion model remained simplified, without jointly considering lateral motion and yaw control, and the assumed visual field was relatively broad compared with what is typically achievable using practical onboard sensing and processing.

Motivated by these limitations, this study investigates cooperative pursuit-and-capture scenarios under more practical sensing constraints and UAV dynamics, with particular emphasis on limited FOV conditions and lateral motion. For improved training efficiency, an NVIDIA Isaac-Lab-based environment that supports large-scale GPU-accelerated parallel simulation was developed [25]. Within this environment, reward shaping was designed to reflect limited-FOV tracking conditions, and the effect of enabling lateral motion in a limited detectable region was quantitatively analyzed. Through these designs, this study aims to clarify how FOV-aware reward formulation and rotary-wing lateral maneuverability affect tracking stability and capture performance.

The main contributions of this paper can be summarized as follows:

- FOV-dependent and capturability rewards are proposed to optimize team objectives for cooperative pursuit and capture in a limited detectable region.
- The effect of lateral motion under detectable-region constraints was analyzed by comparing forward–yaw and forward–lateral–yaw control structures in anti-drone scenarios.
- An Isaac-Lab-based multi-UAV pursuit–evasion environment was developed, incorporating realistic detectable regions and net-gun-based capture conditions.
- To reduce sim-to-real gap, a cascaded proportional–integral–derivative (PID) controller, which is widely used in various commercial flight controllers, and a 6 degree-of-freedom dynamics model of quadcopters were used for the MARL.

The remainder of this paper is organized as follows. Section 2 reviews related works. The simulation environment is described in Section 3. Section 4 describes the proposed method in detail, including the limited detectable region, reward shaping, lateral motion, and training setup. Section 5 presents the quantitative and qualitative analyses of the results. Finally, Section 6 concludes the paper and outlines directions for future research.

2. Related Work

2.1. Anti-Drone Methods

Counter-drone measures can be categorized into non-destructive and destructive approaches [8]. Non-destructive approaches aim to disrupt UAV missions or deprive them of control authority without causing fatal physical damage to the airframe, whereas destructive approaches physically neutralize the target using means such as laser weapons, missiles, or interceptor drones. However, in civilian environments, many counter-drone measures entail the risk of secondary damage, including interference with communication and navigation infrastructure, as well as debris hazards following neutralization [26,27]. Accordingly, capture-based countermeasures, which offer relatively higher safety with reduced secondary damage, have attracted increasing attention as practical alternatives [8,28]. Aerial capture systems that equip UAVs with nets or net guns to safely capture and neutralize target UAVs during flight have been actively studied [9,10,29–31].

2.2. UAV Detection and Tracking

In UAV-based target tracking, target state information is typically inferred from measurements acquired by onboard sensors [32–34]. Consequently, in practical systems, the target position information is directly constrained by the identifiable range and FOV of the sensors mounted on the UAV. Such visibility constraints go beyond mere detection failures and critically influence the difficulty of downstream objectives, such as persistent tracking and capture; therefore, they should be explicitly incorporated in the scenario design stage.

Among the various UAV detection modalities, this study focused on vision-based UAV detection. For vision-based UAV detection, numerous image datasets (e.g., anti-UAV [35], drone vs. bird [36], MAV-VID [37], and DUT [38]) have been developed along with efficient detectors tailored to these datasets. Previous studies have reported that UAV detection is particularly challenging because the target occupies a very small pixel area in the image, unlike general-purpose datasets such as Microsoft COCO [39]. Furthermore, owing to the computational limitations of onboard computers, the use of low-resolution inputs or lightweight models may further reduce the effective detection range [40]. Accordingly, in the present study, a pixel footprint of the target was computed as a function of distance using a pinhole camera model and a reasonable detectable range was set based on this analysis.

2.3. MARL

Multi-UAV tracking and capturing inherently involve multiple decision-making entities that act simultaneously. Consequently, naively reducing the problem to single-agent deep reinforcement learning (DRL) brings fundamental challenges to the forefront, including non-stationarity, credit assignment, partial observability, and scalability [41]. To overcome these challenges, centralized training with decentralized execution (CTDE) has become the most common learning paradigm for cooperative MARL [42].

Value-decomposition methods such as the value-decomposition network (VDN) [43] and QMIX [44] leverage centralized value representations to reduce the coordination difficulty in cooperative tasks while enabling decentralized action selection during execution. These methods explicitly note that independent learners, such as implicit Q-learning (IQL) [45], may exhibit unstable convergence under non-stationarity and propose alternatives accordingly. CTDE also plays a central role in policy gradient and actor–critic methods. MADDPG [46] is a representative approach that combines decentralized policies with centralized critics to address learning in mixed cooperative–competitive environments. COMA [47], which uses a counterfactual baseline to alleviate credit assignment, is another well-known centralized critic approach. Recent studies have further reported that Proximal Policy Optimization (PPO)-based multi-agent methods (MAPPO) [48] can serve as strong baselines and achieve competitive performance across cooperative benchmark tasks. Accordingly, MAPPO was adopted in this study because previous benchmarking results have shown stable and competitive performance across diverse cooperative multi-agent tasks [49].

3. Environment Configuration

3.1. Scenario Setup

This study considered a cooperative pursuit–evasion game in which three pursuer UAVs, each operating in a limited detectable region, attempt to capture a single-evader UAV. Episodes begin immediately after the first detection to focus on the pursuit and capture stages. The detectable region uses a maximum detection range of 30 m. The selection of this value is explained in Section 4.2.

The initial geometry was designed to emulate an approximately uniform spatial distribution of pursuers during the search. The angular separation was fixed at 120° . The initial radial distances were assigned as follows. The detecting pursuer spawned 20 m from the evader, with its initial heading oriented toward the evader. The remaining two pursuers spawned at random distances uniformly sampled from 70–80 to 100–110 m. The initial headings of the two non-detecting pursuers were randomized.

An episode terminates upon successful capture or when a termination condition is triggered. To define capture success, the capturable region was specified as the effective engagement envelope of the net-gun device. The radial constraint was set to 3–10 m, based on the effective range [31] and net deployment distance. The angular constraint was set to a central angle of 10° to keep the target within approximately 2 m of the aiming center at 10 m. Let $\mathbf{p}_{p_i} \in \mathbb{R}^2$ and ψ_{p_i} denote the position and heading of pursuer i , respectively, and let $\mathbf{p}_e \in \mathbb{R}^2$ denote the evader position. Then, the capturable region $\mathcal{C}_i$ is defined as

$$\mathcal{C}_i = \left\{ \mathbf{p}_e \in \mathbb{R}^2 \mid d_{\min} \leq \|\mathbf{p}_e - \mathbf{p}_{p_i}\| \leq d_{\max} \wedge \angle(\mathbf{p}_e - \mathbf{p}_{p_i}, \psi_{p_i}) \leq \frac{\theta_c}{2} \right\}, \quad (1)$$

where $d_{\min} = 3$ m, $d_{\max} = 10$ m, and $\theta_c = 10^\circ$.

Capture was considered successful if at least one pursuer maintained the evader within $\mathcal{C}_i$ for 0.5 s. This episode terminates upon a collision, target loss, or timeout. A

collision occurs if the distance between the centers of any two UAVs is less than 1 m. The evader position is shared among all UAVs when it is within the detectable region of at least one pursuer; otherwise, it is not updated. Target loss was declared if the evader remained outside the detectable regions of all pursuers for more than 3 s. This is appropriate for an open, unbounded environment in which the evader has a higher maximum speed than the pursuers. A timeout occurred when the episode duration reached 60 s. The overall environmental configuration is shown in Figure 1.

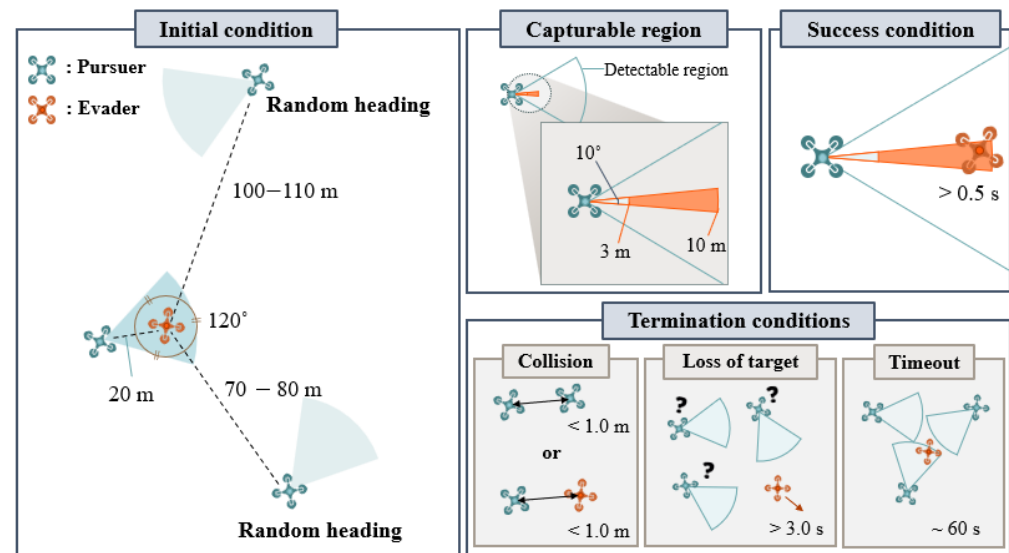


Figure 1. Schematic of the environment setup, including initial conditions, the capturable region, and success and termination conditions.

3.2. UAV Dynamics and Cascaded Controller

Each UAV j is driven by a high-level command input

$$\mathbf{u}_j = \left(v_{x,j}^{ref}, v_{y,j}^{ref}, \omega_j^{ref} \right), \quad (2)$$

where $v_{x,j}^{ref}$ and $v_{y,j}^{ref}$ are components of the horizontal velocity command in the body frame, and ω_j^{ref} is the yaw rate command. To focus on planar motion, the vertical velocity command was set to zero, i.e., $v_{z,j}^{ref} = 0$, and the altitude was regulated by the low-level controller. Although the high-level command is restricted to planar motion, the UAVs are simulated using three-dimensional (3D) dynamics to allow realistic attitude changes and low-level flight behavior. Command $\mathbf{u}_j$ is provided by different high-level decision modules, depending on the UAV role. Specifically, $\mathbf{u}_j$ is generated by the MARL policy for pursuers and by an artificial potential field (APF)-based strategy for evaders.

To model the UAV motion more realistically under these velocity commands, a cascaded PID control architecture was employed to convert high-level commands $\mathbf{u}_j$ into low-level wrench inputs. Specifically, the controller computes the collective thrust T_j applied along the body z -axis and body-axis torques $\boldsymbol{\tau}_j = [\tau_{x,j}, \tau_{y,j}, \tau_{z,j}]^T$, which are applied to the simulated UAV rigid body as external force and torque. This wrench input is passed to the PhysX rigid-body solver, which numerically integrates the resulting six-degree-of-freedom (DoF) dynamics using Newton–Euler equations [50]. Let $\mathbf{p}_j, \mathbf{v}_j \in \mathbb{R}^3$ denote the position and linear velocity in the world frame, respectively; $\mathbf{q}_j = [q_w, q_x, q_y, q_z] \in \mathbb{R}^4$ denote the attitude quaternion; and $\boldsymbol{\omega}_j = [\omega_{x,j}, \omega_{y,j}, \omega_{z,j}] \in \mathbb{R}^3$ denote the angular velocity expressed in the body frame. The mass and inertia parameters used in the simulation were obtained through system identification of the real UAV platform. With mass m and

diagonal inertia matrix $I = \text{diag}(I_x, I_y, I_z)$, the translational and rotational dynamics are respectively given by

$$\dot{p}_j = v_j, \quad m\dot{v}_j = mg + R(q_j) \begin{bmatrix} 0 \\ 0 \\ T_j \end{bmatrix}, \quad (3)$$

$$\dot{q}_j = \frac{1}{2}\Omega(\omega_j)q_j, \quad I\dot{\omega}_j = \tau_j - \omega_j \times I\omega_j, \quad (4)$$

where $g = [0, 0, -9.81]^\top$ and $R(q_j) \in SO(3)$ is the rotation matrix corresponding to q_j . According to the differential formulation in [51], the Matrix $\Omega(\omega_j) \in \mathbb{R}^{4 \times 4}$ is defined as:

$$\Omega(\omega_j) = \begin{bmatrix} 0 & -\omega_{x,j} & -\omega_{y,j} & -\omega_{z,j} \\ \omega_{x,j} & 0 & \omega_{z,j} & -\omega_{y,j} \\ \omega_{y,j} & -\omega_{z,j} & 0 & \omega_{x,j} \\ \omega_{z,j} & \omega_{y,j} & -\omega_{x,j} & 0 \end{bmatrix}. \quad (5)$$

The overall control framework is illustrated in Figure 2. Additionally, to ensure computational efficiency in a large number of parallel environments, the control computations are implemented in a vectorized manner using GPU tensor operations, and TorchScript in PyTorch 2.5.1 is used to reduce the runtime overhead and optimize the execution speed.

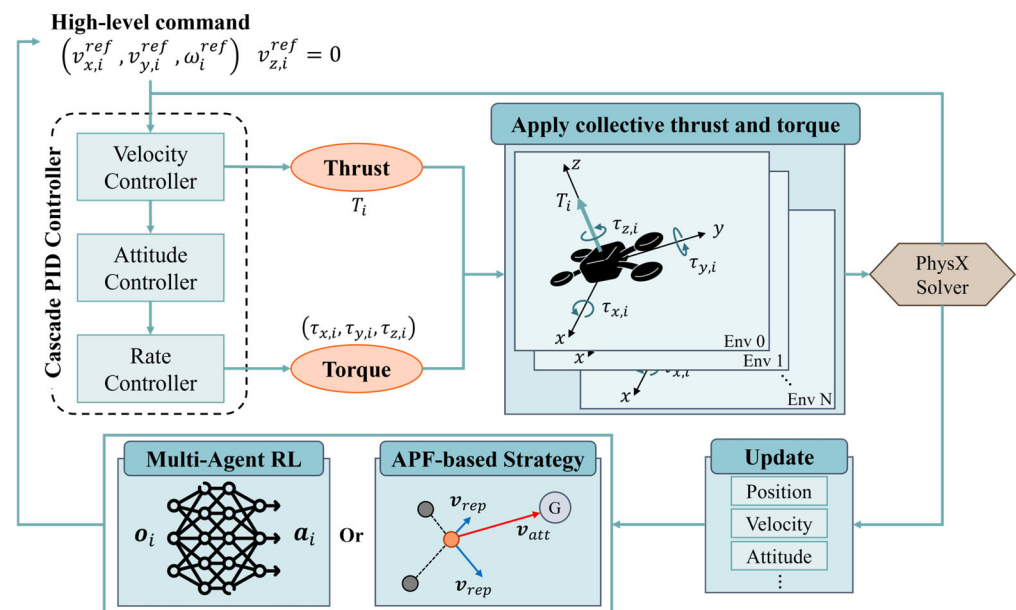


Figure 2. Closed-loop control architecture with a MARL-based policy and cascaded PID controllers.

3.3. Evasion Strategy

To adjust for task difficulty, the maximum speed was set to 10 and 12 m/s for the pursuer and evader, respectively. The yaw rate limit was identical for both the pursuer and evader, i.e., $[-\pi/2, \pi/2]$. Note that the maximum speed of the evader was varied to study how pursuit difficulty affects the performance metrics in the ablation study for lateral motion.

Evader behavior was modeled by modifying the APF-based strategy introduced in [18]. The evader was assumed to have omnidirectional awareness of all pursuers within awareness radius r_{rep} , which further increases the pursuit difficulty. In this study, $r_{rep} = 30$ m, which matches the pursuer detection range. This value was used as the repulsion range in the APF term.

At each step, the evader generates a horizontal velocity command by combining an attractive component toward the goal point and a repulsive component that drives it away from nearby pursuers. Additionally, a yaw rate command was computed to rotate the heading toward the goal point. Let p_e and p_g denote the evader and goal-point positions, respectively, and let p_{p_i} denote the position of pursuer i . For a compact notation, the relative position vectors, distances, and corresponding unit vectors are respectively defined as

$$\begin{aligned} r_{e,g} &= p_g - p_e, \quad d_{e,g} = \|r_{e,g}\|, \quad \hat{r}_{e,g} = \frac{r_{e,g}}{d_{e,g}}, \\ r_{e,p_i} &= p_e - p_{p_i}, \quad d_{e,p_i} = \|r_{e,p_i}\|, \quad \hat{r}_{e,p_i} = \frac{r_{e,p_i}}{d_{e,p_i}}. \end{aligned} \quad (6)$$

Using these definitions, the attractive and repulsive velocity components are given by

$$\begin{aligned} v_{att} &= k_{att} d_{e,g}^2 \hat{r}_{e,g}, \\ v_{rep} &= \sum_i v_{rep,i}, \quad v_{rep,i} = k_{rep} \left(\frac{1}{d_{e,p_i}} - \frac{1}{r_{rep}} \right)^2 \hat{r}_{e,p_i}, \quad d_{e,p_i} \leq r_{rep} \end{aligned} \quad (7)$$

where k_{att} and k_{rep} are positive gains. The evader yaw rate command is computed from the heading error e_ψ between the evader heading and direction toward the goal point as follows:

$$\omega_{cmd} = \text{sgn}(e_\psi) \cdot e_\psi^2. \quad (8)$$

The goal point is regenerated every 1 s at a location 30 m from the evader. Its position is determined by the relative configuration of nearby pursuers to encourage escape from encirclement. The following cases were considered:

- Case 1 (single pursuer within r_{rep}): The goal point is placed in the direction opposite to that of the detected pursuer.
- Case 2 (two or more pursuers within r_{rep}): The goal point is placed along the bisector of the largest angular gap between neighboring pursuers, as observed by the evader, encouraging escape through the widest opening in the formation.
- Case 3 (no pursuer within r_{rep}): The evader maintains the direction of the previous goal point, assuming that it escaped along that direction.

An overview of the evader behavior is summarized in Figure 3.

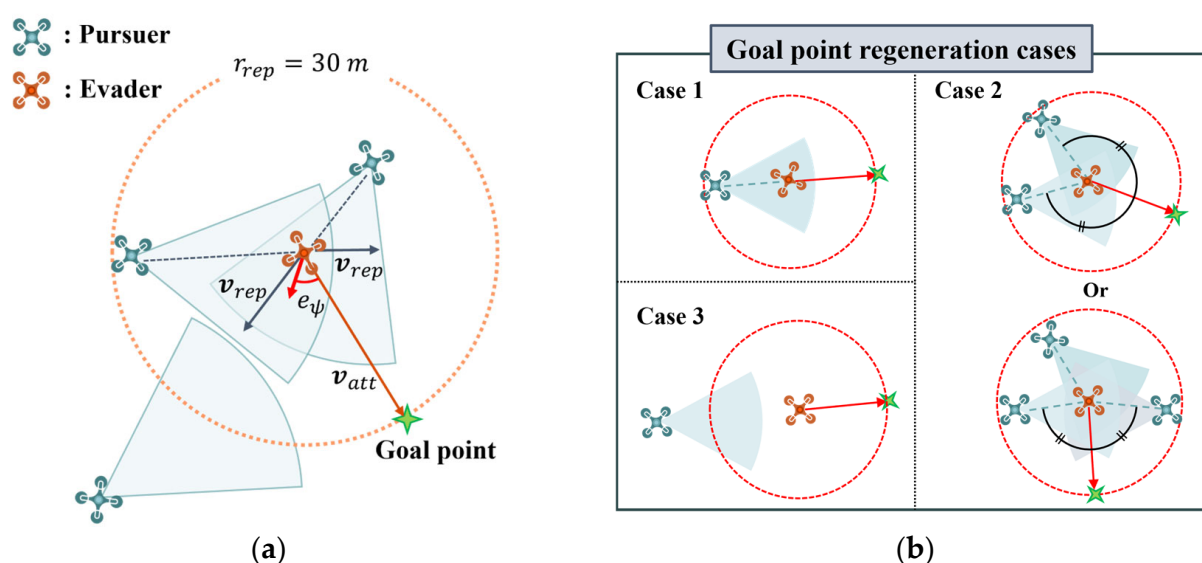


Figure 3. (a) APF-based velocity command generation for the evader; (b) Goal-point regeneration cases in the APF-based evader strategy.

4. Methods

4.1. Problem Formulation

From a MARL perspective, the objective of the proposed environment is to maximize the team reward by enabling pursuers to cooperate and capture the evader. As each pursuer cannot fully observe the global state due to its limited detectable region, the task can be modeled as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP). Dec-POMDP is represented by the tuple $\mathcal{M} = (\mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \{\mathcal{O}_i\}_{i \in \mathcal{N}}, \gamma)$. Here, $\mathcal{N} = \{1, 2, \dots, n\}$ denotes the set of pursuer agents, and $n = 3$ in this study. $\mathcal{S}$ is the global state space that includes the motion states of all UAVs and their relative geometric relationships, and $\mathcal{O}_i$ is the local observation space available to pursuer i . Additionally, $\mathcal{A}$ denotes the joint action space of all pursuers. $\mathcal{P}$ is the state transition probability. $\mathcal{R}$ is the reward function, and γ is the discount factor. Under this formulation, the key environmental elements are defined as follows: Let $\mathbf{o}_{i,t} \in \mathcal{O}_i$ denote the local observation of pursuer i at timestep t , and let $\mathbf{a}_{i,t} \in \mathcal{A}$, $\mathbf{s}_t \in \mathcal{S}$, and $\mathbf{r}_{i,t}$ denote the corresponding action, global state, and individual reward, respectively.

1. **Observation** $\mathbf{o}_{i,t}$ includes the velocity and acceleration of pursuer i , yaw rate, previous action $\mathbf{a}_{i,t-1}$, and a binary indicator of whether the evader lies within the detectable region of pursuer i . Limited inter-pursuer communication is assumed, allowing each pursuer to share its own position and evader-related information. Under this assumption, $\mathbf{o}_{i,t}$ also contains the distances and bearings relative to the other pursuers, as well as the most recently detected evader position. To ensure permutation invariance with respect to agent indices, the relative information of other pursuers included in $\mathbf{o}_{i,t}$ is sorted by relative bearing before being concatenated into the observation vector.
2. **The global state** $\mathbf{s}_t$ is constructed by aggregating the local observations of all pursuers, i.e., $\{\mathbf{o}_{i,t}\}_{i \in \mathcal{N}}$. In addition, $\mathbf{s}_t$ includes the pursuer–evader relative geometry computed from the ground-truth evader position, regardless of whether the evader is currently detected by any pursuer. The evader velocity, yaw rate, and heading angle are also included in $\mathbf{s}_t$ to provide richer information to the critic under the CTDE framework.
3. **Joint action** is defined as $\mathbf{a}_t = \{\mathbf{a}_{i,t}\}_{i \in \mathcal{N}} \in \mathcal{A} = \prod_{i \in \mathcal{N}} \mathcal{A}_i$. As described in Section 3.2, each UAV is driven by a high-level command input $u_j = (v_{x,j}^{ref}, v_{y,j}^{ref}, \omega_j^{ref})$, where $v_{x,j}^{ref}$ and $v_{y,j}^{ref}$ are the reference velocity components and ω_j^{ref} is the reference yaw rate.
4. **The team reward** r_t is computed by averaging the individual rewards, as follows:

$$r_t = \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} r_{i,t}. \quad (9)$$

The reward is designed to encourage the pursuers to approach the evader, maintain it within the capturable region, and avoid undesirable events. The detailed reward formulation is presented in Section 4.3.

4.2. Limited Detectable Region

As discussed in Section 2.2, UAV tracking is typically performed based on observations obtained from onboard sensors and the number of pixels occupied by a UAV bounding box is often significantly smaller than that in conventional object-detection datasets. Therefore, to address this limitation, it is necessary to define a more conservative detectable region. Herein, the detectable region refers to a sector-shaped area defined by the camera FOV and detection range. Accordingly, the detectable region is determined conservatively using a pixel-based approximation.

Let $W(Z)$ denote the physical horizontal span of the camera frustum at distance Z . Under the pinhole camera model,

$$W(Z) = 2Z \tan\left(\frac{\theta_h}{2}\right), \quad (10)$$

where θ_h denotes the horizontal FOV. As planar motion was assumed in this study, only the horizontal FOV was considered. Let N_x be the horizontal image resolution. Then, the number of pixels corresponding to a 1-m length at distance Z , denoted by $w(Z)$, can be expressed as

$$w(Z) = \frac{N_x}{W(Z)} = \frac{N_x}{2Z \tan(\theta_h/2)}. \quad (11)$$

Accordingly, if L denotes the UAV frame width, the apparent pixel width of the UAV at a distance Z , denoted by $n_x(Z)$, is given by

$$n_x(Z) = w(Z)L = \frac{N_x L}{2Z \tan(\theta_h/2)}. \quad (12)$$

For the FOV, $\theta_h = 60^\circ$ was conservatively assumed based on commercially available camera specifications, and the image resolution was set to 1280×720 . The UAV model used in this study had a frame width of 650 mm; thus, $n_x(Z) = 720.53/Z$.

Previous studies reported that most detection errors occur when the object size is less than 32×32 pixels. They also indicated that drones in aerial imagery often appear smaller than 20×20 pixels, which substantially increases the detection difficulty [36,40]. Although small objects are commonly characterized using area-based measures, the detectable region considered in this study was parameterized by the horizontal FOV and range. Accordingly, the bounding-box horizontal extent $n_x(Z)$ was adopted as a conservative proxy for small-object detectability. Under the pinhole-model-based approximation, the distances at which the UAV attained horizontal extents of 32 and 20 pixels were approximately 22.5 and 36.0 m, respectively. Based on these observations, objects with horizontal extents of less than 20 pixels were assumed to be unreliable. Considering an additional margin, the detection range was set to 30 m. Consequently, the detectable region is defined as a circular sector with a maximum range of $Z_{\max} = 30$ m and horizontal FOV of $\theta_h = 60^\circ$, within which targets are assumed to be detectable. The resulting detectable region is shown in Figure 4.

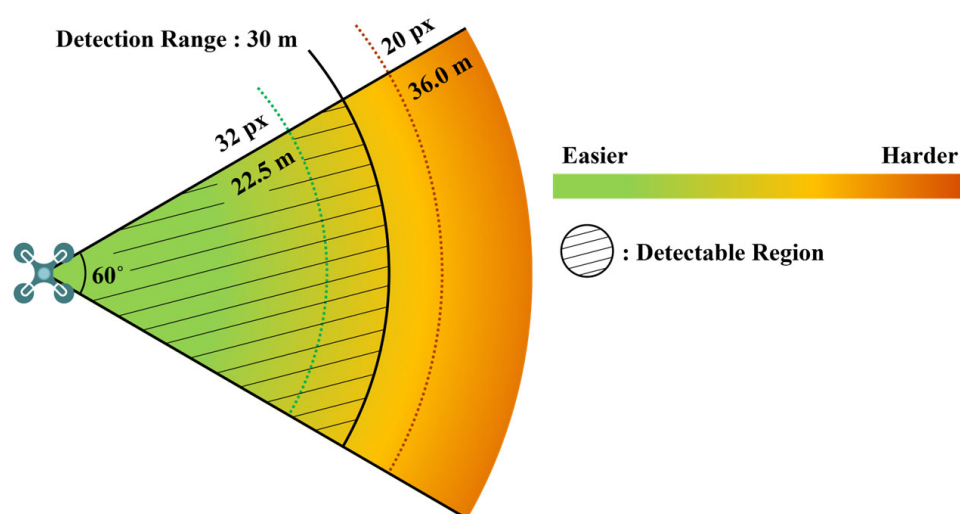


Figure 4. Detectable region definition using the pinhole camera approximation.

4.3. Reward Shaping

A reward function was designed for pursuit and capture under a limited-FOV setting. Most MARL-based pursuit studies and forward-looking FOV constraints were neither explicitly modeled nor directly reflected in the reward design. To address this issue, an angle-based shaping term that varies with the viewing angle (i.e., relative bearing) was incorporated into the reward structure. The per-agent reward r_i is defined as follows:

$$r_i = C_{dist}r_{d,i} + C_{safe}r_{safe,i} + C_{cap}r_{cap,i} + C_{gaze}r_{gaze,i} + r_{sparse}. \quad (13)$$

The weighting factors in (13) were set to $C_{dist} = 0.35$, $C_{safe} = 0.30$, $C_{gaze} = 0.35$, and $C_{cap} = 2.0$, based on empirical tuning in preliminary experiments. The coefficients C_{dist} , C_{safe} , and C_{gaze} were chosen so that their sum is 1, thereby balancing the relative contributions of the dense shaping terms. A larger value was assigned to C_{cap} to more strongly emphasize capture-oriented alignment.

4.3.1. Distance Reward $r_{d,i}$

For pursuer i , the distance reward $r_{d,i}$ is computed from the relative distance between the evader and pursuer i , denoted by d_{e,p_i} , and the maximum detection range Z_{max} . As the pursuer approaches the evader, $r_{d,i}$ increases, and the reward is designed to become positive when the evader is within the detection range of the pursuer as:

$$r_{d,i} = -\frac{d_{e,p_i} - Z_{max}}{Z_{max}}. \quad (14)$$

4.3.2. Safe Distance Penalty $r_{safe,i}$

The safe distance penalty $r_{safe,i}$ is defined using the minimum separation distance $d_{close,i}$ between pursuer i and any other UAV; it is given by

$$r_{safe,i} = -2e^{(-0.3d_{close,i})}. \quad (15)$$

This formulation yields a more negative reward when a pursuer is excessively close to other UAVs, thereby discouraging risky flights that may lead to collisions.

4.3.3. Capturability Reward $r_{cap,i}$

An additional capturability reward, $r_{cap,i}$, is provided for the pursuer closest to the evader. This term is defined using the distance $d_{cap,i}$ between the evader and the center of a capturable region located in front of pursuer i . This reward is given by

$$r_{cap,i} = e^{(-\frac{d_{cap,i}}{2})}. \quad (16)$$

This term encourages the pursuer closest to the evader to keep the evader within a capturable region. As $d_{cap,i}$ measures the distance to the center of the capturable region, translational deviation and yaw-induced misalignment can increase $d_{cap,i}$. Consequently, the capturability reward promotes yaw regulation, encouraging the policy to keep the evader near the center of the capturable region. For clarity, Figure 5 shows the capturable region and distance $d_{cap,i}$, highlighting how translational deviation and yaw misalignment affect $d_{cap,i}$.

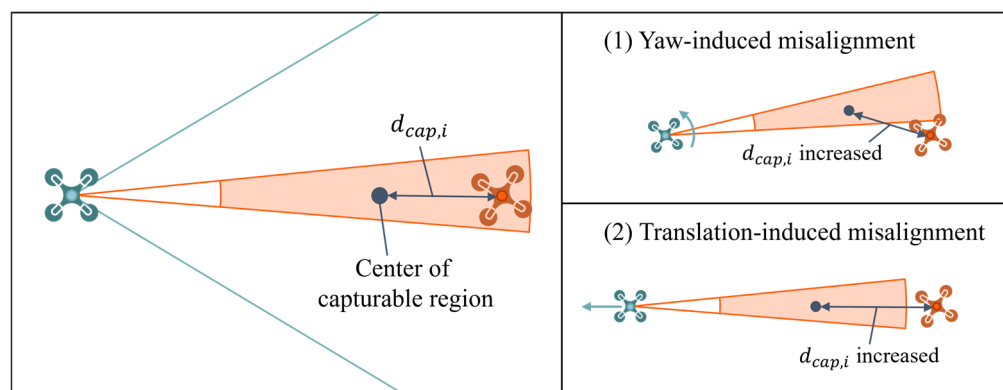


Figure 5. Schematics of the capturable region and distance $d_{cap,i}$ used for $r_{cap,i}$.

4.3.4. Gaze-Alignment Reward $r_{gaze,i}$

The gaze-alignment reward $r_{gaze,i}$ is defined as a function of the relative bearing angle between the pursuer heading and the line-of-sight direction to the evader, denoted by $\psi_{gaze,i}$. In this study, three reward functions were considered for gaze alignment.

- **Case 1: Piecewise-ramp (PR) reward**

$$r_{gaze,i} = \begin{cases} -\frac{2|\psi_{gaze,i}|}{\theta_h} + 1, & |\psi_{gaze,i}| \leq \frac{\theta_h}{2} \\ -\frac{2|\psi_{gaze,i}| - \theta_h}{2\pi - \theta_h}, & |\psi_{gaze,i}| > \frac{\theta_h}{2} \end{cases} \quad (17)$$

Within the pursuer FOV, the reward remains positive, whereas it becomes negative when the evader lies outside the FOV. This sign change encourages the policy to keep the evader inside the FOV.

- **Case 2: Ramp reward**

$$r_{gaze,i} = -\left| \frac{2\psi_{gaze,i}}{\pi} \right| + 1 \quad (18)$$

This absolute-value linear function attains its maximum value of 1 at $\psi_{gaze,i} = 0$, becomes zero at $|\psi_{gaze,i}| = \pi/2$, and decreases to negative values beyond that. However, it does not explicitly reflect the pursuer FOV and may remain positive even when the evader is already outside the FOV.

- **Case 3: Cosine reward**

$$r_{gaze,i} = \cos(\psi_{gaze,i}) \quad (19)$$

This function attains its maximum value of 1 at $\psi_{gaze,i} = 0$ and its minimum value of -1 at $\psi_{gaze,i} = \pm\pi$, providing a smooth and continuous variation with respect to the relative bearing angle. However, similar to Case 2, it does not explicitly reflect the pursuer FOV and may therefore provide insufficient penalty when the evader is outside the FOV.

4.3.5. Sparse Reward r_{sparse}

The sparse reward r_{sparse} is an event-based term that is added besides continuous shaping rewards, such as distance and gaze alignment. A reward of $+30$ is assigned when the capture is successful, whereas a penalty of -30 is assigned when collisions occur between pursuers. Additionally, a penalty of -10 is applied when a pursuer collides with the evader, and a penalty of -20 is applied when the evader is lost from view for more

than 3 s. These event-based signals were designed to promote successful capture while maintaining safe and persistent tracking behavior.

4.4. UAV Lateral Motion

Previous UAV pursuit–evasion studies have approximated planar UAV motion using a two-dimensional (2D) unicycle model in which the control inputs consist only of a forward motion command and a yaw-rate command. Under this assumption, the travel direction of the pursuer is always aligned with its heading, making it difficult for the pursuer to reposition laterally while simultaneously maintaining the evader within the FOV.

Some previous studies extended the control inputs to include both forward and lateral velocity commands, enabling planar translation. However, these settings often assumed continuous access to the evader position or did not explicitly impose an FOV constraint, making visibility maintenance less critical. By contrast, in a limited detectable region, yawing and lateral motion serve distinct roles, with yaw helping maintain visibility and lateral motion supporting spatial repositioning. Therefore, their joint use enables more flexible pursuit maneuvers under sensing constraints.

Table 1 summarizes the assumptions made in previous studies regarding access to the evader position, employed dynamics models, and resulting action spaces. As can be seen in Table 1, some previous studies considered partial sensing constraints, such as range, FOV, or occlusion. To the best of authors' knowledge, however, no previous study considered both practical visual constraints (i.e., range and FOV) and various motion commands (i.e., forward, yaw, and lateral motion). Accordingly, this study quantitatively evaluated the need and benefits of jointly considering yaw and lateral motions in pursuer tracking scenarios with a limited detectable region by analyzing their impacts on tracking stability and capture performance.

Table 1. Summary of previous UAV pursuit–evasion studies: Evader position availability, dynamics models, and actions.

Ref.	Evader Position	Simulation Dynamics Model	Actions *
[15]	Always available	2D unicycle kinematics	(ω_i) and $(v_{x,i}, \omega_i)$
[16]	Always available	2D particle kinematics	$(v_{x,i}, v_{y,i})$
[17]	Always available	Simplified quadcopter dynamics (Fixed altitude)	$(a_{x,i}, a_{y,i})$
[18]	Range constraint	2D kinematics	$(v_{x,i}, v_{y,i})$
[19]	Range + FOV + occlusion constraint	2D unicycle-like kinematics	$(\omega_{z,i}, a_{x,i})$
[52]	Range constraint	2D unicycle-like kinematics	$(\omega_{z,i}, a_{x,i})$
[53]	Occlusion constraint	6-DoF quadrotor dynamics	$(T, \omega_{z,i}, \omega_{z,i}, \omega_{z,i})$
Ours	Range + FOV constraint	6-DoF quadcopter dynamics (Planar command)	$(v_{x,i}, v_{y,i}, \omega_i)$

* $v_{x,i}, v_{y,i}$: planar velocity commands; $\omega_{x,i}, \omega_{y,i}, \omega_{z,i}$: angular rate command; $a_{x,i}, a_{y,i}$: planar acceleration commands; T : collective thrust.

4.5. Training Setup

The training environment was implemented on top of the NVIDIA Isaac Lab framework, in which data were collected from a large set of GPU-parallel vectorized environments. In the simulation, the physics engine was stepped at 400 Hz, while the high-level action commands were updated at 20 Hz and held constant between action updates. As on-policy algorithms such as MAPPO update policies using rollouts collected under the latest policy, increasing the interaction throughput via parallel simulation is advantageous for reducing the overall training time.

Training was built on the MAPPO implementation provided by the SKRL library [54]. However, a default configuration of SKRL primarily targets heterogeneous-agent settings. Therefore, to match the homogeneous pursuer setup considered in this study, the training configuration was modified slightly to support parameter sharing. Specifically, all pursuers were configured to use a single shared policy, and the experience collected from multiple agents was aggregated such that data collection and the training loop proceeded in a consistent manner. The overall training pipeline is illustrated in Figure 6.

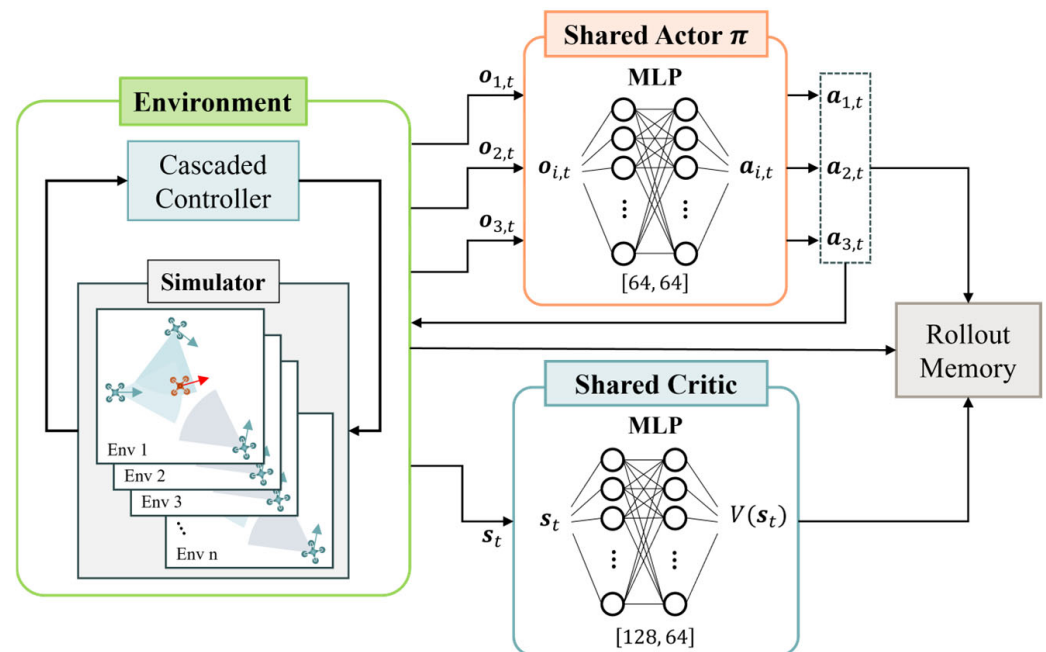


Figure 6. Parameter-sharing MAPPO training framework.

All experiments were conducted on an AMD EPYC 7413 CPU and an NVIDIA A100 GPU. Training was performed in 16,384 parallel environments for 80,000 time steps, resulting in $80,000 \times 16,384$ environment transitions collected during training. The main learning hyperparameters are listed in Table 2. To ensure that the observed performance differences were primarily attributable to changes in reward design and environment settings rather than the optimization configuration, the hyperparameters were kept fixed across experiments.

Table 2. Learning hyperparameters used in training.

Hyperparameter	Value
Number of environments (Training)	16,384
Number of environments (Validation)	50,000
Rollout	16
Learning epochs	8
Mini batches	16
Discount factor	0.99
Learning rate	0.0001
Value loss scale	2.0
Entropy loss scale	0.01
Policy network layers	[64, 64]
Value network layers	[128, 64]
Total learning timesteps	80,000

5. Results and Discussion

This section analyzes the effect of the proposed reward function on learning performance. Pursuit performance under a limited-detectable-region condition was also evaluated with respect to the maximum speed of the evader and the presence or absence of lateral maneuvers. For validation, episodes were executed across 50,000 environments, and the success rate and number of failed episodes were logged for comparison. To avoid premature termination of ongoing pursuits, the episode time limit during validation was extended to 120 s, which reduced the timeouts while tracking remained active.

5.1. Effect of Reward Design

To evaluate the effect of reward design quantitatively, training was conducted with four different random seeds for each reward configuration. For each configuration, the policy achieving the highest success rate during training was selected and evaluated in the same validation environment for comparison. The evaluation considers the three gaze-alignment reward functions (PR, ramp, and cosine) described in Section 4.3.4. For each variant, cases with and without $r_{cap,i}$ were examined, resulting in six reward configurations. The validation results are summarized in Table 3. First, we consider cases without $r_{cap,i}$. Among the three gaze-alignment rewards, the PR reward achieved the highest success rate (87.79%), whereas the ramp and cosine rewards yielded success rates of 84.90% and 66.45%, respectively. Particularly, the cosine-based reward produced a large number of lost-target terminations, with 15,893 episodes in which the evader was lost. This behavior can be attributed to the cosine function property that its gradient becomes very small as $\psi_{gaze,i}$ approaches zero. Once a certain level of alignment is achieved, the reward provides only a weak learning signal and therefore does not sufficiently encourage further policy updates. Notably, a similar trend was observed when $r_{cap,i}$ was enabled. Therefore, a cosine-based reward is insufficient to simultaneously promote stable distance regulation and successful capture under limited-detectable-region pursuit conditions.

Table 3. Validation results for gaze-alignment reward variants under capturability reward ablation. Each cell reports the number of validation environments that terminate for the corresponding reason.

Reward Shape		Without Capturability Reward			with Capturability Reward		
		PR	Ramp	Cosine	PR	Ramp	Cosine
Success env		43,893	42,452	33,224	49,617	43,262	41,304
Lost env		476	6518	15,893	144	4796	7014
Crashed	w/pursuer	0	0	59	0	0	0
	w/evader	399	231	824	234	1597	702
Timeout env		5232	799	0	5	345	980
Success rate		87.79%	84.90%	66.45%	99.23%	86.52%	82.61%

When $r_{cap,i}$ was applied, the success rates increased for all gaze-alignment reward settings, and the number of lost-target terminations decreased consistently. The largest improvement was observed with the cosine reward, for which the success rate increased from 66.45% to 82.61%. This suggests that in regimes in which the gaze-alignment signal becomes weak, $r_{cap,i}$ acts as a complementary distance-based learning signal that enhances pursuit stability. Additionally, for the PR reward, enabling the capturability reward substantially reduced the number of timeout terminations. This indicates that $r_{cap,i}$ encourages the evader to remain within a capturable region, thereby promoting a behavioral transition from merely sustaining tracking to completing the capture.

Overall, combining the PR reward, which is designed with FOV-dependent gradients and reward signs, with the capture-promoting capturability reward improved both the pursuit and capture performance. To further validate the generality of the proposed rewards beyond the 60° FOV setting, additional training and evaluation were conducted under $\theta_h = 30^\circ$ and $\theta_h = 90^\circ$, where θ_h denotes the horizontal FOV. For each θ_h , all six reward configurations were trained with three different random seeds, and the success rates logged during training were compared across configurations. The resulting learning curves, showing the success-rate changes across reward formulations, are presented in Figure 7. As can be seen, the PR-reward settings and those incorporating capturability reward generally achieve higher training success rates, even under different θ_h conditions. Notably, at $\theta_h = 90^\circ$, the performance gap attributable to the PR reward becomes relatively smaller, whereas the improvement from enabling the capturability reward becomes more pronounced. This can be interpreted as a consequence of increasing θ_h , which makes the piecewise slope characteristics of the PR shaping less steep and thus renders its overall profile more similar to that of the ramp reward. Nevertheless, the PR reward still maintains a higher success rate, and the capturability reward complements it by further improving performance. These results suggest that the proposed FOV-dependent reward shaping and capturability reward design can enhance the learning performance for pursuit-and-capture tasks within a limited detectable region.

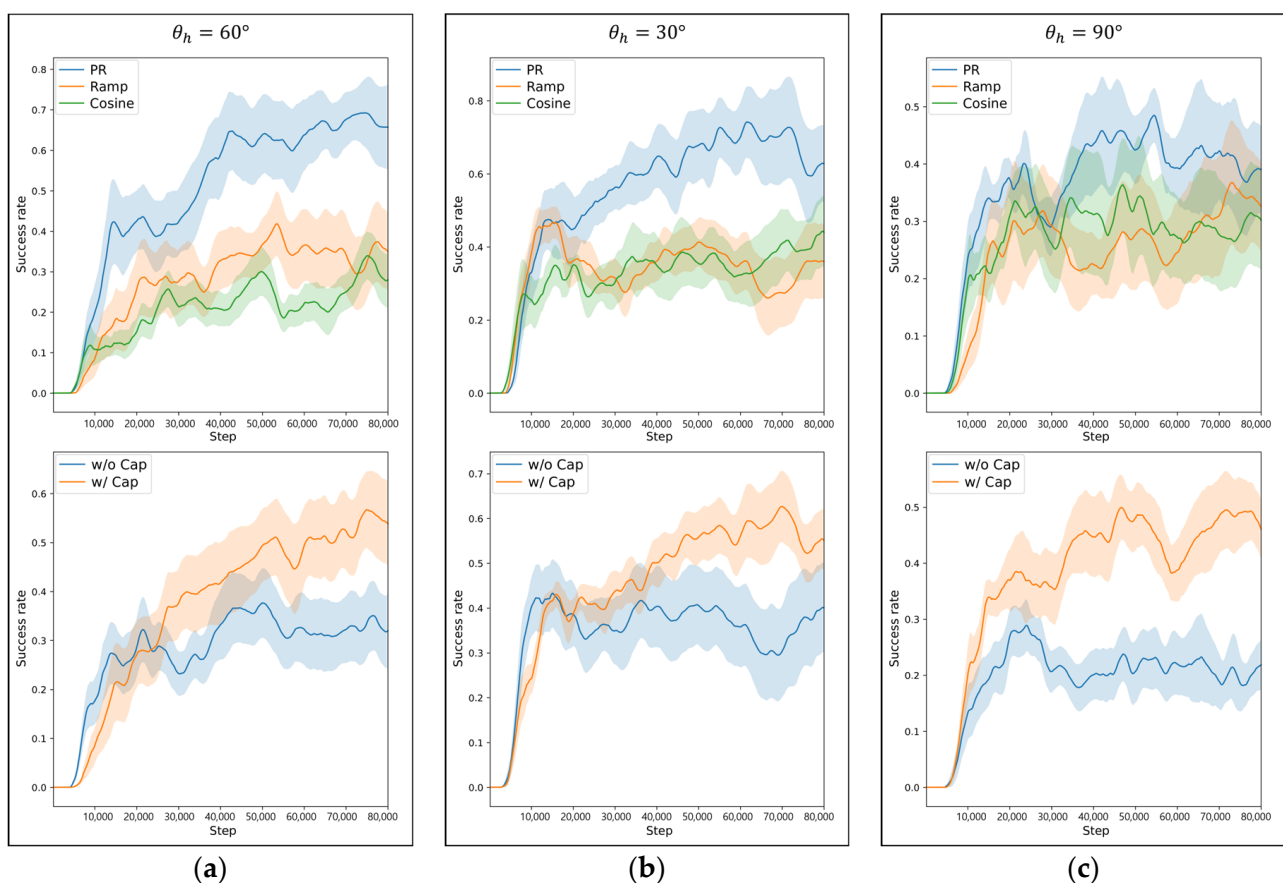


Figure 7. Training success-rate curves across reward formulations for different horizontal FOV θ_h . Top: PR/ramp/cosine comparison. Bottom: with vs. without capturability reward. (a) $\theta_h = 60^\circ$; (b) $\theta_h = 30^\circ$; (c) $\theta_h = 90^\circ$.

5.2. Effects of Lateral Motion

This section investigates the effect of augmenting the pursuer action space with lateral-velocity command v_y under varying evader speed limits. Using the validation method discussed in Section 5.1, policies were trained separately for maximum speed setting of each evader and subsequently evaluated by selecting the checkpoint with the highest success rate. For this evaluation, training was performed using the proposed PR and capturability reward design. If a configuration does not successfully capture episodes during training, the policy with the highest episodic return is selected for validation. The validation results are listed in Table 4.

Table 4. Validation results for the effect of lateral motion across different evader maximum speed settings.

Evader Max Speed		8 m/s		10 m/s		12 m/s	
Pursuer Max Speed		10 m/s		10 m/s		10 m/s	
Action Space		(v_x, ω)	(v_x, v_y, ω)	(v_x, ω)	(v_x, v_y, ω)	(v_x, ω)	(v_x, v_y, ω)
Success env		48,859	49,894	0	46,554	0	49,617
Lost env		1	87	33,102	1659	49,947	144
Crashed	w/pursuer	1140	1	11,089	181	53	0
	w/evader	0	18	0	195	0	234
Timeout env		0	0	5809	1411	0	5
Success rate		97.72%	99.79%	0.0%	93.11%	0.0%	99.23%

5.2.1. Evader Is Slower than Pursuers

When the maximum speed of the evader was lower than that of the pursuer, a high success rate (97.72%) was achieved even without lateral velocity control. This indicates that under relatively easy settings, pursuit and capture can be accomplished without lateral DoF. However, when lateral motion is enabled, the number of episodes terminating owing to inter-pursuer collisions decreases sharply from 1140 to 1, and the success rate further increases to 99.79%. These results suggest that although lateral motion is not a strict requirement for capture when the evader is slower than the pursuers, it plays an important role in improving the overall performance by maintaining safe inter-agent separation and substantially reducing collisions during pursuit.

To examine the corresponding behavior, a representative validation episode was selected and visualized, as shown in Figure 8. In Figure 8a, where lateral motion is not available, the pursuers approach the evader predominantly from behind and tend to align in a nearly collinear formation while attempting to keep the evader within the limited detectable region. Such an alignment makes it difficult to maintain safe inter-agent separation and increases the likelihood of collisions, thereby degrading cooperative positioning during pursuit. By contrast, Figure 8b shows that when lateral motion is available, the pursuers can maintain pursuit while approaching from different directions and occupy distinct lateral positions, thereby preserving the inter-agent spacing. This spatial dispersion and increased maneuvering flexibility also enabled capture over a shorter time horizon. Overall, these observations demonstrate that lateral motion can facilitate safer and more effective pursuit and capture even when the evader is slower.

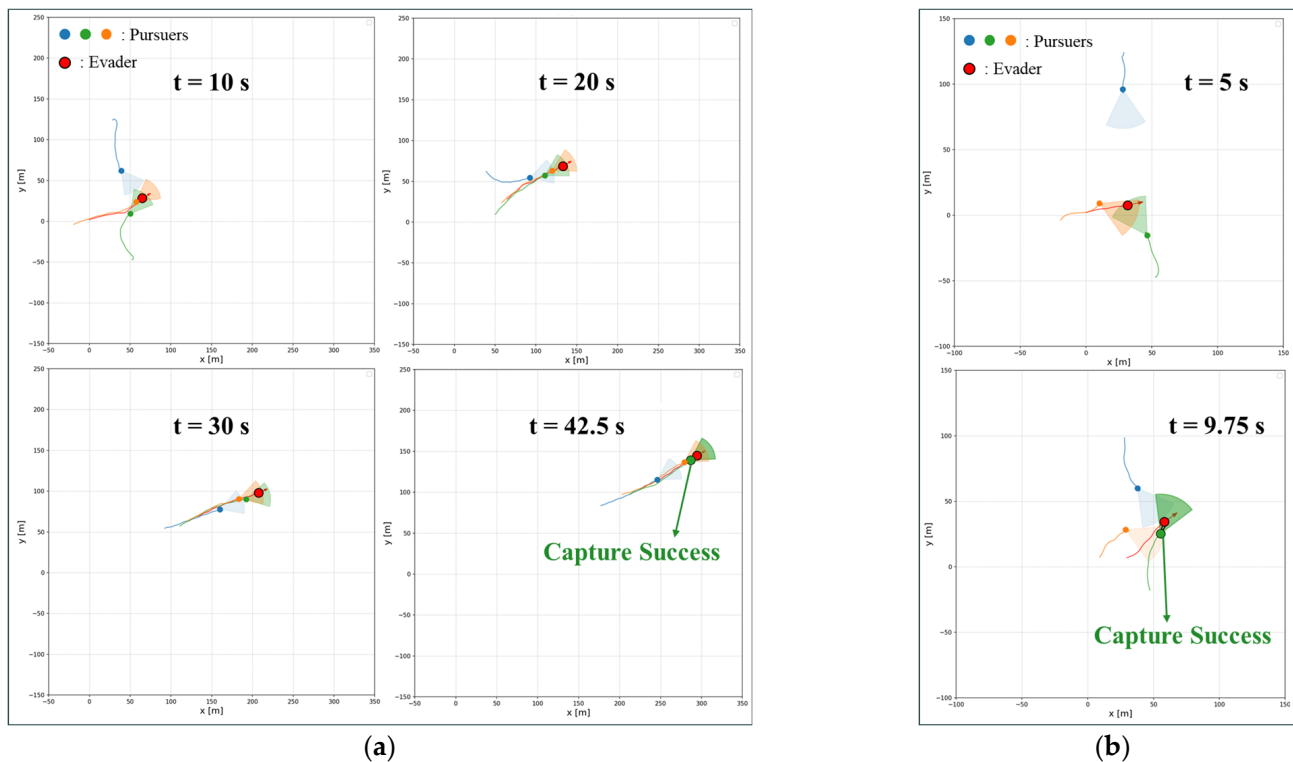


Figure 8. Pursuit trajectory snapshots for a slower evader: (a) without and (b) with lateral motion.

5.2.2. Same Maximum Speed for Evader and Pursuers

When the evader and pursuer share the same maximum speed, the benefit of lateral motion becomes more pronounced. Without lateral motion, no capture was achieved, and most episodes were terminated owing to evader loss and inter-pursuer collisions. These results highlight the fundamental limitations of pursuit without lateral velocity control. In the absence of lateral DoF, pursuers have a limited ability to approach the evader from the sides and constrain its maneuvering. Furthermore, under identical speed bounds, a purely tail-chasing strategy could not effectively reduce the separation distance. Such tail-chasing behavior also increases the likelihood of inter-agent collisions and makes it difficult to maintain the evader consistently within the limited detectable region, ultimately leading to frequent target-loss terminations. This behavior is illustrated in Figure 9a, where the pursuers follow similar paths from behind and the episode ends with target loss at $t = 20$ s. Overall, in the no-lateral setting, inefficient pursuit tends to persist, resulting in frequent collision- or target-loss-based terminations, with some episodes reaching the time limit.

By contrast, allowing lateral motion substantially recovers the capture performance. With lateral velocity control, the number of evader-loss terminations and inter-pursuer collisions decreased markedly, increasing the success rate to 93.11%. Notably, collisions with the evader were slightly higher than those in the slower-evader case, suggesting that lateral maneuverability encourages more aggressive close-range blocking maneuvers for capture. The resulting cooperative behavior is shown in Figure 9b, where the pursuers maintain their pursuit while approaching from different directions and occupying distinct lateral positions, thereby forming a more spatially dispersed enclosing geometry. Particularly, the initially detecting pursuer implicitly guides the evader toward another pursuer, strengthening the interception configuration of the team, and ultimately enabling successful capture.

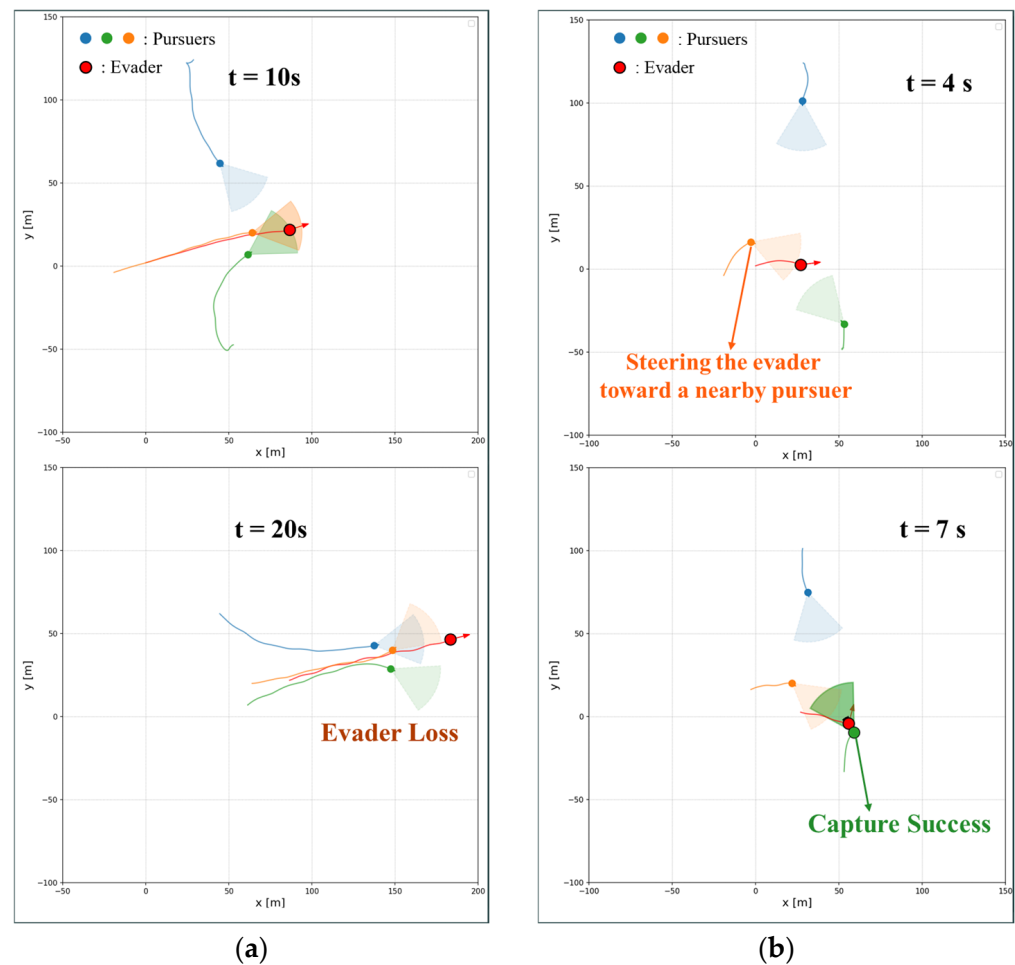


Figure 9. Pursuit trajectory snapshots under equal maximum speed: (a) without and (b) with lateral motion.

5.2.3. Evader Is Faster than Pursuers

As no capture is achieved without lateral maneuverability, even when the evader and pursuers share the same maximum speed, the success rate is expected to remain at 0% when the evader is faster. In this setting, no timeout terminations were observed, and most episodes ended because of target loss. This indicates that pursuit cannot be sustained, and the evader is typically lost at an early stage. When the evader has a speed advantage, sustaining pursuit and transitioning to capture require cooperative strategies that constrain the evader maneuvering and block escape routes. However, without lateral motion, the pursuers tend to converge into a tail-chasing formation, which prevents persistent tracking and leads to early target-loss terminations, as illustrated in Figure 10a.

By contrast, enabling lateral motion yielded a success rate of 99.23%, indicating that effective cooperation was learned despite the higher maximum speed of the evader. Similar to the behavior shown in Figure 9b, early capture is sometimes achieved when the initially detected pursuer implicitly guides the evader toward another pursuer. Nonetheless, failed early interception attempts can also occur, as shown at $t = 10s$ in Figure 10b. In such cases, pursuers subsequently approach from different directions and progressively constrain the evader maneuvering while blocking escape routes. This re-established containment allows the pursuit to be sustained and ultimately leads to successful capture. These results demonstrate that lateral maneuverability is essential for enabling cooperative strategies such as escape-route blocking under speed-disadvantaged conditions.

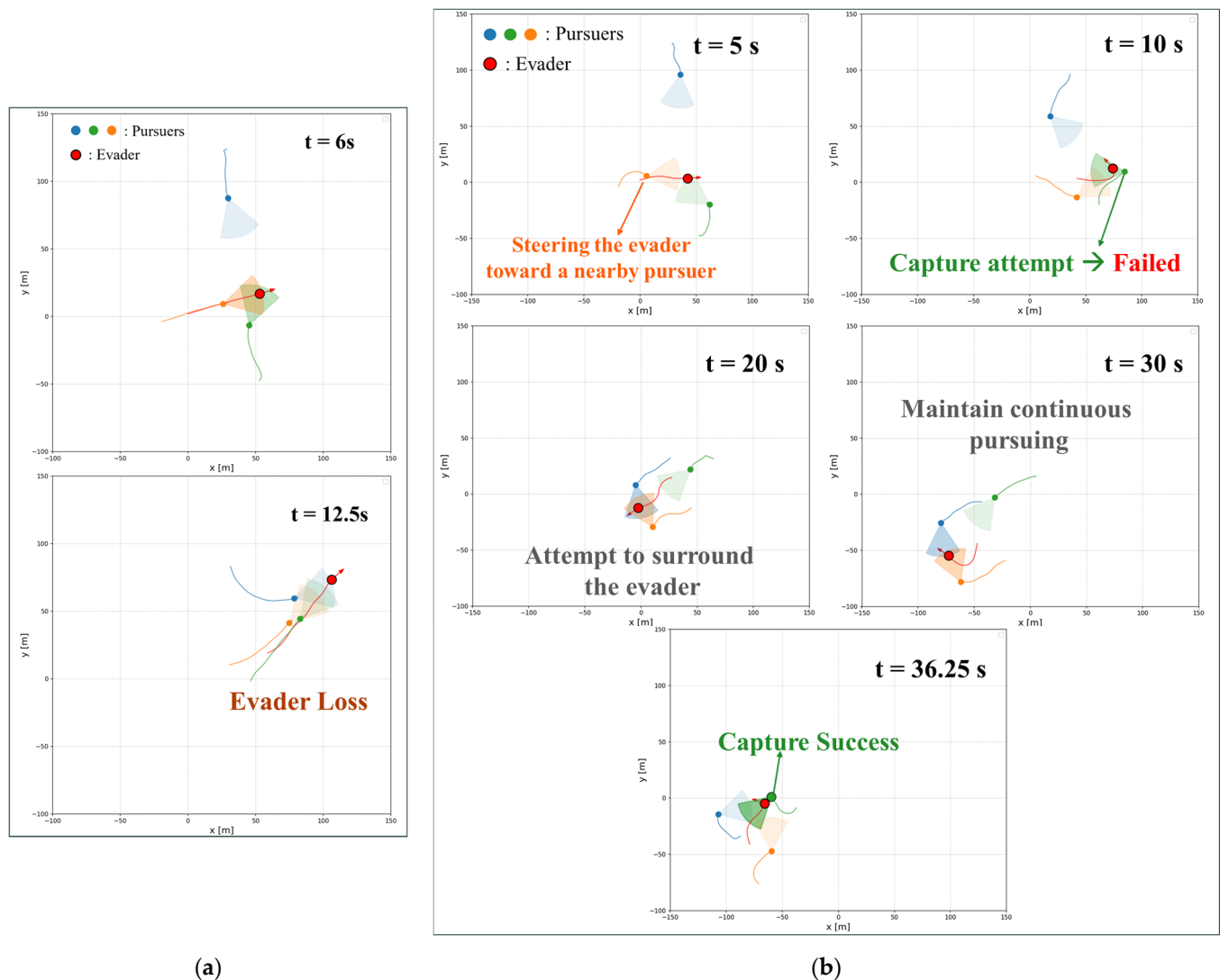


Figure 10. Pursuit trajectory snapshots for a faster evader: (a) without and (b) with lateral motion.

Overall, the success rate and behavior analyses suggest that, when modeling pursuit-and-capture scenarios with rotary-wing UAVs, it is important to explicitly include lateral motion in the action space and control inputs. Rotary-wing UAVs can directly generate a lateral velocity component in the body frame; excluding this degree of freedom limits the ability of the policy to learn cooperative maneuvers, such as forming enclosing geometries and blocking escape routes. Under limited-detectable-region conditions, yawing is required to maintain the target within view, which further constrains the feasible motion directions. Incorporating lateral velocity control alleviates this constraint by allowing a multi-directional approach and blocking maneuvers while maintaining gaze alignment, thereby improving pursuit stability and capture performance.

6. Conclusions

This paper proposes a MARL approach to solve a pursuit–evasion game in which multiple pursuer UAVs track and capture a single evader UAV under a limited-detectable-region constraint. Practical operational constraints were incorporated into the simulation environment by considering previous studies on vision-based detection performance and the effective engagement range of a net-gun device. To improve performance, reward shaping was designed to encourage FOV alignment and tracking stability, and an additional capturability reward was introduced to guide pursuers toward a capturable region.

Experimental results verified that these designs improved both the tracking and capture success rates. Furthermore, training and evaluation under multiple FOV conditions demonstrated that the proposed reward formulation can be generalized across diverse sensing settings. The results also showed that considering the lateral motion characteristics of rotary-wing UAVs can significantly enhance the tracking and capture performance under limited sensing, suggesting that lateral motion should be explicitly considered in target-tracking problems with constrained detection ranges.

Future work will extend the environment to include obstacles and resulting occlusions, enabling evaluation under more realistic operational conditions. It will also be valuable to incorporate background-dependent detectability so that the effective detection range can vary according to scene complexity and target pixel size. Additionally, as the implemented cascaded controller supports altitude regulation, the environment can be expanded to 3D scenarios by incorporating vertical FOV and 3D sensing characteristics. Moreover, the MARL policy can be integrated with robust and adaptive control techniques to improve robustness against external disturbances and modeling uncertainty, which can enhance reliability in real-world deployment.

Author Contributions: Conceptualization, W.N. and S.H.; methodology, S.H.; software, S.H. and S.L.; validation, S.H., S.L. and H.J.; formal analysis, S.H. and S.Y.; investigation, S.H. and W.B.; resources, W.N.; data curation, S.L.; writing—original draft preparation, S.H.; writing—review and editing, W.N.; visualization, S.H.; supervision, W.N.; project administration, W.N.; funding acquisition, W.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government Ministry of Science and ICT (MSIT) under Grant 2023R1A2C1006655, and by the Chung-Ang University Graduate Research Scholarship.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

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

Abbreviations

The following abbreviations are used in this manuscript:

UAVs	Unmanned Aerial Vehicles
AI	Artificial Intelligence
MARL	Multi-Agent Reinforcement Learning
LSTM	Long Short-Term Memory
PID	Proportional–Integral–Derivative
DRL	Deep Reinforcement Learning
CTDE	Centralized Training with Decentralized Execution
VDN	Value-Decomposition Network
IQL	Implicit Q-Learning
PPO	Proximal Policy Optimization
APF	Artificial Potential Field
DoF	Degree-of-Freedom
Dec-POMDP	Decentralized Partially Observable Markov Decision Process
PR	Piecewise-Ramp
2D	Two-dimensional
3D	Three-dimensional

References

- González-Jorge, H.; Martínez-Sánchez, J.; Bueno, M.; Arias, P. Unmanned aerial systems for civil applications: A review. *Drones* **2017**, *1*, 2. [\[CrossRef\]](#)
- Sarkar, N.I.; Gul, S. Artificial intelligence-based autonomous UAV networks: A survey. *Drones* **2023**, *7*, 322. [\[CrossRef\]](#)
- Katkuri, A.V.R.; Madan, H.; Khatri, N.; Abdul-Qawy, A.S.H.; Patnaik, K.S. Autonomous UAV navigation using deep learning-based computer vision frameworks: A systematic literature review. *Array* **2024**, *23*, 100361. [\[CrossRef\]](#)
- Rao, B.; Gopi, A.G.; Maione, R. The societal impact of commercial drones. *Technol. Soc.* **2016**, *45*, 83–90. [\[CrossRef\]](#)
- Kaufmann, E.; Gehrig, M.; Foehn, P.; Ranftl, R.; Dosovitskiy, A.; Koltun, V.; Scaramuzza, D. Beauty and the beast: Optimal methods meet learning for drone racing. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 690–696.
- Lykou, G.; Moustakas, D.; Gritzalis, D. Defending airports from UAS: A survey on cyber-attacks and counter-drone sensing technologies. *Sensors* **2020**, *20*, 3537. [\[CrossRef\]](#)
- Chamola, V.; Kotesch, P.; Agarwal, A.; Gupta, N.; Guizani, M. A comprehensive review of unmanned aerial vehicle attacks and neutralization techniques. *Ad Hoc Netw.* **2021**, *111*, 102324. [\[CrossRef\]](#)
- Park, S.; Kim, H.T.; Lee, S.; Joo, H.; Kim, H. Survey on anti-drone systems: Components, designs, and challenges. *IEEE Access* **2021**, *9*, 42635–42659. [\[CrossRef\]](#)
- Chen, Y.; Li, Z.; Li, L.; Ma, S.; Zhang, F.; Fan, C. An anti-drone device based on capture technology. *Biomim. Intell. Robot.* **2022**, *2*, 100060. [\[CrossRef\]](#)
- Zhang, T.; Lu, R.; Yang, X.; Xie, X.; Fan, J.; Tang, B. UAV Hunter: A Net-Capturing UAV System with Improved Detection and Tracking Methods for Anti-UAV Defense. *Drones* **2024**, *8*, 573. [\[CrossRef\]](#)
- Sun, P.; Li, S.; Zhu, B.; Zheng, Z.; Zuo, Z. Vision-based finite-time prescribed performance control for uncooperative UAV target-tracking subject to field of view constraints. *ISA Trans.* **2024**, *149*, 168–177. [\[CrossRef\]](#)
- Liu, X.; Yang, Y.; Ma, C.; Li, J.; Zhang, S. Real-time visual tracking of moving targets using a low-cost unmanned aerial vehicle with a 3-axis stabilized gimbal system. *Appl. Sci.* **2020**, *10*, 5064. [\[CrossRef\]](#)
- Vidal, R.; Shakernia, O.; Kim, H.J.; Shim, D.H.; Sastry, S. Probabilistic pursuit-evasion games: Theory, implementation, and experimental evaluation. *IEEE Trans. Robot. Autom.* **2002**, *18*, 662–669. [\[CrossRef\]](#)
- Chung, T.H.; Hollinger, G.A.; Isler, V. Search and pursuit-evasion in mobile robotics: A survey. *Auton. Robot.* **2011**, *31*, 299–316. [\[CrossRef\]](#)
- de Souza, C.; Newbury, R.; Cosgun, A.; Castillo, P.; Vidolov, B.; Kulić, D. Decentralized Multi-Agent Pursuit Using Deep Reinforcement Learning. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4552–4559. [\[CrossRef\]](#)
- Kouzeghar, M.; Song, Y.; Meghjani, M.; Bouffanais, R. Multi-Target Pursuit by a Decentralized Heterogeneous UAV Swarm using Deep Multi-Agent Reinforcement Learning. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 3289–3295.
- Zhang, R.; Zong, Q.; Zhang, X.; Dou, L.; Tian, B. Game of Drones: Multi-UAV Pursuit-Evasion Game With Online Motion Planning by Deep Reinforcement Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 7900–7909. [\[CrossRef\]](#)
- Chen, Y.; Shi, Y.; Dai, X.; Meng, Q.; Yu, T. Pursuit-evasion game with online planning using deep reinforcement learning. *Appl. Intell.* **2025**, *55*, 512. [\[CrossRef\]](#)
- Peng, Z.; Wu, G.; Luo, B.; Wang, L. Multi-UAV cooperative pursuit strategy with limited visual field in urban airspace: A multi-agent reinforcement learning approach. *IEEE/CAA J. Autom. Sin.* **2025**, *12*, 1350–1367. [\[CrossRef\]](#)
- Weintraub, I.E.; Pachter, M.; Garcia, E. An introduction to pursuit-evasion differential games. In Proceedings of the 2020 American Control Conference (ACC), Denver, CO, USA, 1–3 July 2020; pp. 1049–1066.
- Gronauer, S.; Diepold, K. Multi-agent deep reinforcement learning: A survey. *Artif. Intell. Rev.* **2022**, *55*, 895–943. [\[CrossRef\]](#)
- Yang, K.; Shen, A.; Xu, N.; Deng, F.; Lu, M.; Chen, C. A review of reinforcement learning approaches for pursuit-evasion games. *Chin. J. Aeronaut.* **2025**; in press. [\[CrossRef\]](#)
- Wei, D.; Zhang, L.; Liu, Q.; Chen, H.; Huang, J. UAV Swarm Cooperative Dynamic Target Search: A MAPPO-Based Discrete Optimal Control Method. *Drones* **2024**, *8*, 214. [\[CrossRef\]](#)
- Xu, S.; Dang, Z. Emergent behaviors in multiagent pursuit evasion games within a bounded 2D grid world. *Sci. Rep.* **2025**, *15*, 29376. [\[CrossRef\]](#)
- Mittal, M.; Roth, P.; Tigue, J.; Richard, A.; Zhang, O.; Du, P.; Serrano-Muñoz, A.; Yao, X.; Zurbrugg, R.; Rudin, N. Isaac Lab: A GPU-Accelerated Simulation Framework for Multi-Modal Robot Learning. *arXiv* **2025**, arXiv:2511.04831.
- Castrillo, V.U.; Manco, A.; Pascarella, D.; Gigante, G. A review of counter-UAS technologies for cooperative defensive teams of drones. *Drones* **2022**, *6*, 65. [\[CrossRef\]](#)

27. Zidane, Y.; Silva, J.S.; Tavares, G. Jamming and spoofing techniques for drone neutralization: An experimental study. *Drones* **2024**, *8*, 743. [\[CrossRef\]](#)
28. Kang, H.; Joung, J.; Kim, J.; Kang, J.; Cho, Y.S. Protect your sky: A survey of counter unmanned aerial vehicle systems. *IEEE Access* **2020**, *8*, 168671–168710. [\[CrossRef\]](#)
29. Guvenc, I.; Koohifar, F.; Singh, S.; Sichitiu, M.L.; Matolak, D. Detection, tracking, and interdiction for amateur drones. *IEEE Commun. Mag.* **2018**, *56*, 75–81. [\[CrossRef\]](#)
30. Rothe, J.; Strohmeier, M.; Montenegro, S. A concept for catching drones with a net carried by cooperative UAVs. In Proceedings of the 2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Würzburg, Germany, 2–4 September 2019; pp. 126–132.
31. Dynamics, D. DroneCatcher. Available online: <https://dronecatcher.nl/> (accessed on 10 January 2026).
32. Kim, K.; Kim, J.; Lee, H.-G.; Choi, J.; Fan, J.; Joung, J. UAV chasing based on YOLOv3 and object tracker for counter UAV systems. *IEEE Access* **2023**, *11*, 34659–34673. [\[CrossRef\]](#)
33. Lee, D.; Park, W.; Yi, J.; Byun, W.; Huh, S.; Nam, W. UAV Control for Close Tracking of a Flying Object Using Search Region Focused Detector and Target-Visibility Enhancing Control. *IEEE Access* **2023**, *11*, 139326–139334. [\[CrossRef\]](#)
34. Wang, Y.; Huang, Z.; Laganière, R.; Zhang, H.; Ding, L. A UAV to UAV tracking benchmark. *Knowl.-Based Syst.* **2023**, *261*, 110197. [\[CrossRef\]](#)
35. Jiang, N.; Wang, K.; Peng, X.; Yu, X.; Wang, Q.; Xing, J.; Li, G.; Guo, G.; Ye, Q.; Jiao, J. Anti-UAV: A large-scale benchmark for vision-based UAV tracking. *IEEE Trans. Multimed.* **2021**, *25*, 486–500. [\[CrossRef\]](#)
36. Coluccia, A.; Fascista, A.; Schumann, A.; Sommer, L.; Dimou, A.; Zarpalas, D.; Akyon, F.C.; Eryuksel, O.; Ozfuttu, K.A.; Altinuc, S.O. Drone-vs-bird detection challenge at IEEE AVSS2021. In Proceedings of the 2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Washington, DC, USA, 16–19 November 2021; pp. 1–8.
37. Rodriguez-Ramos, A.; Rodriguez-Vazquez, J.; Sampedro, C.; Campoy, P. Adaptive inattentional framework for video object detection with reward-conditional training. *IEEE Access* **2020**, *8*, 124451–124466. [\[CrossRef\]](#)
38. Zhao, J.; Zhang, J.; Li, D.; Wang, D. Vision-based anti-UAV detection and tracking. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 25323–25334. [\[CrossRef\]](#)
39. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
40. Nikouei, M.; Baroutian, B.; Nabavi, S.; Taraghi, F.; Aghaei, A.; Sajedi, A.; Moghaddam, M.E. Small Object Detection: A Comprehensive Survey on Challenges, Techniques and Real-World Applications. *arXiv* **2025**, arXiv:2503.20516. [\[CrossRef\]](#)
41. 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. [\[CrossRef\]](#)
42. Amato, C. An introduction to centralized training for decentralized execution in cooperative multi-agent reinforcement learning. *arXiv* **2024**, arXiv:2409.03052. [\[CrossRef\]](#)
43. Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W.M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J.Z.; Tuyls, K. Value-decomposition networks for cooperative multi-agent learning. *arXiv* **2017**, arXiv:1706.05296.
44. Rashid, T.; Samvelyan, M.; De Witt, C.S.; Farquhar, G.; Foerster, J.; Whiteson, S. Monotonic value function factorisation for deep multi-agent reinforcement learning. *J. Mach. Learn. Res.* **2020**, *21*, 1–51.
45. Tan, M. Multi-agent reinforcement learning: Independent vs. cooperative agents. In Proceedings of the Tenth International Conference on Machine Learning, Amherst, MA, USA, 27–29 June 1993; pp. 330–337.
46. Lowe, R.; Wu, Y.I.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6379–6390.
47. Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; Whiteson, S. Counterfactual multi-agent policy gradients. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 2974–2982.
48. Yu, C.; Velu, A.; Vinitisky, E.; Gao, J.; Wang, Y.; Bayen, A.; Wu, Y. The surprising effectiveness of PPO in cooperative multi-agent games. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24611–24624.
49. Papoudakis, G.; Christianos, F.; Schäfer, L.; Albrecht, S.V. Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. *arXiv* **2020**, arXiv:2006.07869.
50. Bouabdallah, S.; Murrieri, P.; Siegwart, R. Design and control of an indoor micro quadrotor. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), New Orleans, LA, USA, 26 April–1 May 2004; pp. 4393–4398.
51. Trawny, N.; Roumeliotis, S.I. Indirect Kalman filter for 3D attitude estimation. *Univ. Minn. Dept. Comp. Sci. Eng. Tech. Rep.* **2005**, *2*, 2005.
52. Xia, Z.; Du, J.; Wang, J.; Jiang, C.; Ren, Y.; Li, G.; Han, Z. Multi-Agent Reinforcement Learning Aided Intelligent UAV Swarm for Target Tracking. *IEEE Trans. Veh. Technol.* **2022**, *71*, 931–945. [\[CrossRef\]](#)

53. Chen, J.; Yu, C.; Li, G.; Tang, W.; Ji, S.; Yang, X.; Xu, B.; Yang, H.; Wang, Y. Online planning for multi-uav pursuit-evasion in unknown environments using deep reinforcement learning. *IEEE Robot. Autom. Lett.* **2025**, *10*, 8196–8203. [\[CrossRef\]](#)
54. Serrano-Munoz, A.; Chrysostomou, D.; Bøgh, S.; Arana-Arexolaleiba, N. skrl: Modular and flexible library for reinforcement learning. *J. Mach. Learn. Res.* **2023**, *24*, 1–9.

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