machines

MDPI

Article

Target-Aware Safety-Residual Reinforcement Learning for Cooperative Multi-UAV Pursuit in Complex Environments

Shun Li [1,2,*], Bo Yu [1,2], Dongying Liu [1,2], Dayu Gao [1], Peizheng He [1,2], Gongbo Chen [1] and Lin Xu [1]

[1] Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China; yubo@sict.ac.cn (B.Y.); liudongying22@mails.ucas.ac.cn (D.L.); gaodayu@sict.ac.cn (D.G.); hepeizheng23@mails.ucas.ac.cn (P.H.); chengb@sict.ac.cn (G.C.); xulin@sict.ac.cn (L.X.)
[2] University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: lishun22@mails.ucas.ac.cn

Abstract

Multi-UAV cooperative persistent tracking in complex obstacle environments requires agents to approach dynamic targets while ensuring obstacle avoidance and flight safety; however, standard multi-agent reinforcement learning (MARL) methods typically rely on a single policy to implicitly handle both objectives, making it difficult to balance task performance and risk control. To address this issue, this paper proposes a Target-Aware Safety-Residual Pursuit Reinforcement Learning (TASRP) framework for constrained three-dimensional environments. A continuous-control 3D tracking environment is constructed in IsaacLab, where two multirotor UAVs cooperatively track a dynamic target under random, target-blocking, and gate-like obstacle layouts, boundary constraints, and inter-agent collision risks, with each UAV producing a four-dimensional action composed of normalized thrust and body-frame torques. TASRP adopts a dual-head residual policy in which a pursuit branch generates nominal actions, and a safety branch predicts corrective residuals, together with a risk-aware gating mechanism, a target-guided teacher for obstacle detouring, and a dual-critic safety-constrained optimization scheme. Under clean observations, TASRP achieves task success rates of 75–79%, obstacle crash rates of 13–15%, and boundary crash rates of 1–2% across three representative scenarios. Under noisy observations, TASRP achieves 72.1% task success, 20.3% obstacle crash, and 2.8% boundary crash, outperforming MAPPO (61.2%, 61.2%, 5.6%) and HAPPO (58.1%, 73.5%, 4.1%). These results indicate that explicitly decoupling target-oriented control and safety correction enables a more effective and robust performance–safety trade-off under both clean and moderately noisy observations.

Keywords: multi-agent reinforcement learning; cooperative pursuit–evasion; obstacle avoidance; residual policy learning; dynamic target tracking

Check for updates

Academic Editors: Dan Zhang, Vito Antonio Nardi, Salvatore Rosario Bassolillo and Giuseppe Tricomi

Received: 20 April 2026
Revised: 20 June 2026
Accepted: 23 June 2026
Published: 29 June 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) license.

1. Introduction

Cooperative persistent tracking with multiple unmanned aerial vehicles (Multi-UAVs) in complex obstacle environments has significant research value in applications such as airspace security, target surveillance, emergency response, and autonomous confrontation [1,2]. Compared with single-UAV systems, multi-UAV platforms can significantly improve spatial coverage, operational redundancy, and target acquisition efficiency. Recent advancements in area coverage path planning (CPP) and dynamic task allocation demonstrate that cooperative decision-making frameworks—ranging from market-based

consensus algorithms to enhanced deep reinforcement learning—enable swarms to optimize mission success metrics while mitigating redundant visits in highly constrained environments [3–5].

The existing studies on multi-UAV tracking and pursuit–evasion primarily focus on target approaching, cooperative control, and task allocation [6,7]. However, when obstacles directly block the target-oriented motion, the nature of the problem changes significantly. In such cases, agents can neither rely solely on direct motion toward the target nor achieve effective decisions by simply avoiding obstacles; instead, they must generate reasonable detouring behaviors while maintaining pursuit intent [8,9]. In other words, multi-UAV tracking in complex obstacle environments is neither a pure target-approaching problem nor an independent safety-avoidance problem but, rather, a continuous control decision-making problem with deeply coupled objectives of task progression and risk avoidance.

Multi-Agent Reinforcement Learning (MARL) provides an effective approach for addressing such complex sequential decision-making problems [10]. Representative methods based on the centralized training with decentralized execution (CTDE) paradigm, such as Multi-Agent Proximal Policy Optimization (MAPPO), Multi-Agent Constrained Policy Optimization (MACPO), Heterogeneous-Agent Proximal Policy Optimization (HAPPO), and Heterogeneous-Agent Trust Region Policy Optimization (HATRPO) [11–13], have achieved promising performance in tasks, including cooperative control, formation flight, and multi-agent decision-making. However, these methods typically employ a single policy representation to generate actions, implicitly coupling target-oriented motion and safety-aware avoidance within a unified policy network. For the cooperative tracking task considered in this work under complex obstacle layouts, such a coupled representation exhibits two key limitations. First, the learned policy tends to favor aggressive pursuit behaviors with higher rewards, thereby reducing safety margins and increasing collision risks. Second, relying solely on environmental reward signals makes it difficult to stably learn high-quality detouring strategies, particularly in blocking obstacle scenarios, where "moving away from obstacles" does not necessarily correspond to "bypassing obstacles while maintaining pursuit of the target" [14].

From the perspective of safety control, existing studies have attempted to enhance the safety of reinforcement learning policies through techniques such as reward shaping, constrained optimization, safety critics, and action correction layers [12,15,16]. Although these approaches have achieved certain improvements in reducing collision risks, most of them model safety as a passive suppression of hazardous states, with a limited explicit consideration of the coordination between target-oriented motion and safety correction [17]. For tracking tasks with blocking obstacles, the key challenge lies not merely in avoiding collisions but in generating corrective actions that both mitigate risks and preserve the target-oriented motion as much as possible under high-risk conditions. Therefore, there is a need for a policy learning framework that can explicitly decouple nominal task actions from safety corrective actions while adaptively coordinating them according to the risk level [18,19].

Based on the above considerations, this paper investigates a dual-UAV cooperative persistent tracking problem within a bounded three-dimensional space, where two UAVs are required to cooperatively track a dynamic target under random, target-blocking, and gate-like obstacle layouts, together with boundary constraints and collision risks. Different from conventional settings that only consider randomly distributed obstacles, the target-blocking layout explicitly places obstacles along the target-oriented direction, making "bypassing obstacles while continuously approaching the target" a necessary component of task execution, while the gate-like layout further requires precise motion through narrow feasible passages. These layouts do not define separate optimization objectives; rather,

they induce different layout-conditioned feasible regions and return-cost trade-offs within the same safety-constrained decentralized partially observable Markov decision process (Dec-POMDP)formulation. To more realistically model UAV control, a four-dimensional continuous action space is adopted, where each UAV outputs a normalized thrust command and three body-frame torque commands. This environment endows the studied problem with three key characteristics: continuous control, cooperative tracking, and safety-constrained optimization. From a system perspective, this work is formulated at the decision-and-control interface of the UAV autonomy stack. The learned policy operates on structured local observations and outputs command-level continuous actions. Accordingly, the focus of this study is on cooperative target pursuit and safety-aware correction under complex obstacle layouts, while the full onboard sensing, perception, and low-level flight-control pipeline is outside the scope of the present formulation.

The main contributions of this paper are summarized as follows:

1. A unified Target-Aware Safety-Residual Pursuit Reinforcement Learning (TASRP) framework is proposed for cooperative multi-UAV persistent tracking in complex three-dimensional environments, where target advancement and safety correction are explicitly decoupled within a structured policy representation.
2. A risk-adaptive residual-fusion mechanism and a target-oriented blocking-obstacle teacher are introduced to enable task-consistent detouring behaviors, so that the learned policy does not merely avoid obstacles but also bypasses critical blockages while preserving pursuit intent.
3. A dual-critic constrained optimization scheme is developed to connect the structured policy design with safety-constrained Dec-POMDP optimization, jointly improving persistent tracking performance and safety robustness under CTDE training.
4. Extensive experiments in random-obstacle, target-blocking-obstacle, and gate-like-obstacle scenarios show that TASRP achieves task success rates of 75–79%, obstacle crash rates of 13–15%, and boundary crash rates of 1–2% under clean observations. Under noisy observations, TASRP further achieves 72.1% task success, 20.3% obstacle crash, and 2.8% boundary crash, outperforming MAPPO (61.2%, 61.2%, 5.6%) and HAPPO (58.1%, 73.5%, 4.1%). These results demonstrate that TASRP provides a more favorable and more robust performance–safety trade-off under both clean and moderately noisy observations.

The remainder of this paper is organized as follows. Section 2 reviews related work on multi-UAV cooperative tracking, multi-agent reinforcement learning, safe reinforcement learning, and structured policy learning. Section 3 presents the problem formulation and environment setup. Section 4 details the proposed TASRP framework. Section 5 provides comparative experiments, ablation studies, and result analysis. Section 6 concludes the paper and discusses future research directions.

2. Related Work

We review four research directions that are most relevant to this work: cooperative multi-UAV pursuit and tracking, MARL for cooperative UAV control, safe reinforcement learning for UAV obstacle avoidance, and structured policy learning with residual action modeling.

2.1. Cooperative Multi-UAV Pursuit and Tracking

Cooperative multi-UAV pursuit and tracking is an important research direction in multi-agent unmanned system control, with existing studies primarily focusing on target approaching, cooperative encirclement, and dynamic target tracking. Ren et al. proposed a communication-aware distributed actor–critic reinforcement learning framework, in which

bidirectional recurrent networks are employed to enable information sharing among UAVs, and a multi-stage reward design is introduced to improve cooperative pursuit performance, demonstrating the effectiveness of communication-enhanced mechanisms in such tasks [20]. Zhou et al. developed the MAAC-R method for multi-target tracking, incorporating a maximum mutual-reward mechanism to strengthen inter-agent cooperation and improve tracking performance [21]. Chen et al. further incorporated UAV dynamics and control inputs explicitly into the pursuit–evasion modeling process, highlighting the importance of continuous-control formulations for multi-UAV-tracking tasks [22].

However, most of the existing studies on multi-UAV tracking primarily emphasize target approaching efficiency, cooperative mechanisms, or task completion rates, while giving limited consideration to safety constraints in complex obstacle environments [6]. Many current pursuit–evasion studies are still based on simplified two-dimensional waypoint representations or weak dynamic assumptions, and they have not been fully extended to three-dimensional continuous-control settings with obstacle-coupled tracking tasks [23–27]. Therefore, multi-UAV persistent tracking in complex environments is more appropriately characterized as a continuous decision-making problem with deeply coupled objectives of task progression and safety avoidance, rather than a conventional target pursuit problem.

2.2. MARL for Cooperative UAV Control

Multi-agent reinforcement learning (MARL) has been widely applied to tasks such as cooperative UAV control, collaborative search, and target tracking. Methods based on the centralized training with decentralized execution (CTDE) paradigm have become the mainstream approach for multi-UAV control, as they leverage global information during training while maintaining distributed decision-making during execution [10]. Within this framework, representative methods such as MAPPO, as well as trust-region-based extensions, including HAPPO and HATRPO, have demonstrated strong performance in cooperative control and partially observable tasks [11,13,28]. Zhao et al. model UAVs, targets, and obstacles as nodes in a graph structure and employ graph attention networks to enhance cooperative search and tracking in dynamic environments, highlighting the importance of relational modeling for multi-agent decision-making [29].

Despite these advances in cooperation efficiency and environment representation, most existing MARL-based UAV methods still adopt a single policy to generate actions, implicitly coupling target-oriented motion and safety-aware avoidance within a unified decision network. In tracking tasks under complex obstacle layouts, this often leads to notable limitations: on the one hand, the policy tends to favor aggressive target-approaching behaviors with higher returns; on the other hand, local avoidance actions may disrupt the directional consistency required for persistent tracking [30]. Therefore, how to explicitly coordinate task-oriented control and safety correction within the MARL framework remains an open challenge for multi-UAV tracking in complex obstacle environments.

2.3. Safe RL for UAV Obstacle Avoidance

Safe reinforcement learning and constrained reinforcement learning have become important research directions in autonomous UAV control in recent years. Gu and Wang combined obstacle perception with constrained reinforcement learning by introducing explicit safety penalties into the reward design, achieving safe multi-UAV obstacle avoidance in densely cluttered environments [31]. Shi et al. incorporated safety critics into multi-UAV tasks and applied safety cost–constrained optimization to reduce the probability of risk violations, demonstrating improved training stability in high-risk scenarios [32]. In addition, Dawood et al. designed a distributed safety filtering mechanism in multi-robot formation tasks to mask potentially dangerous actions, thereby achieving collision-free control [33].

These studies indicate that explicitly incorporating safety penalties, safety critics, or safety filtering mechanisms into the reinforcement learning process can effectively reduce policy failure in high-risk environments. However, most of the existing approaches model safety as suppressing or correcting unsafe actions, with limited consideration of how to maintain task progression after obstacle avoidance [12,15].

2.4. Structured Policy Learning and Residual Action Modeling

Structured policy learning, residual policy methods, and hierarchical control provide promising modeling paradigms for complex control tasks. Sharma et al. proposed a residual policy learning framework that focuses learning on local corrections to an existing control policy, laying the foundation for subsequent research [34]. Abbas et al. further applied residual modeling to safe reinforcement learning in continuous control settings, showing that residual corrections can improve control accuracy and robustness [35]. In terms of hierarchical control, Pang et al. designed a hierarchical architecture for multi-UAV air combat tasks to enhance cooperative learning efficiency [36]. Tan et al. combined imitation learning with safe reinforcement learning to construct a hierarchical quadrotor control system, demonstrating the effectiveness of integrating teacher guidance with structured control [37].

Despite substantial progress in cooperative UAV tracking, safe multi-agent reinforcement learning, and structured policy learning, the existing methods still face limitations in complex multi-obstacle environments. In particular, prior tracking approaches provide limited consideration of blocking obstacles, boundary-aware pursuit behaviors, and inter-agent safety interactions, while safe reinforcement learning methods mainly emphasize risk suppression rather than pursuit-consistent avoidance. Moreover, structured residual policies have not been fully integrated into cooperative multi-UAV persistent tracking tasks.

More specifically, the difference between TASRP and earlier research can be clarified from three aspects. First, compared with safety-constrained MARL methods that mainly improve safety through reward shaping, safety penalties, safety critics, or constrained updates [12,15,16,31,32], TASRP does not treat safety only as an optimization-side regularizer. Instead, it explicitly embeds safety intervention into the action-generation process by separating target-oriented nominal pursuit from safety-oriented local correction and then modulating their interaction through risk-adaptive residual fusion. Second, compared with residual policy learning approaches [18,19,34,35], TASRP is not a generic residual correction scheme for continuous control. Rather, the residual branch in TASRP is task-structured: it is activated according to explicit obstacle and boundary risk, and it is trained to preserve target-pursuit consistency instead of merely improving local control accuracy. Third, compared with obstacle-avoidance guiding or heuristic detouring strategies [14,17,33], TASRP does not simply guide the UAV away from the nearest obstacle or suppress risky motion in a reactive manner. Instead, it introduces a target-oriented blocking-obstacle teacher that identifies obstacles that directly hinder target advancement and provides supervision for pursuit-consistent bypassing. Therefore, the essential novelty of TASRP lies in integrating structured target-safety decoupling, explicit risk-conditioned residual fusion, target-aware blocking guidance, and dual-critic constrained optimization into a unified framework for cooperative multi-UAV pursuit in complex environments.

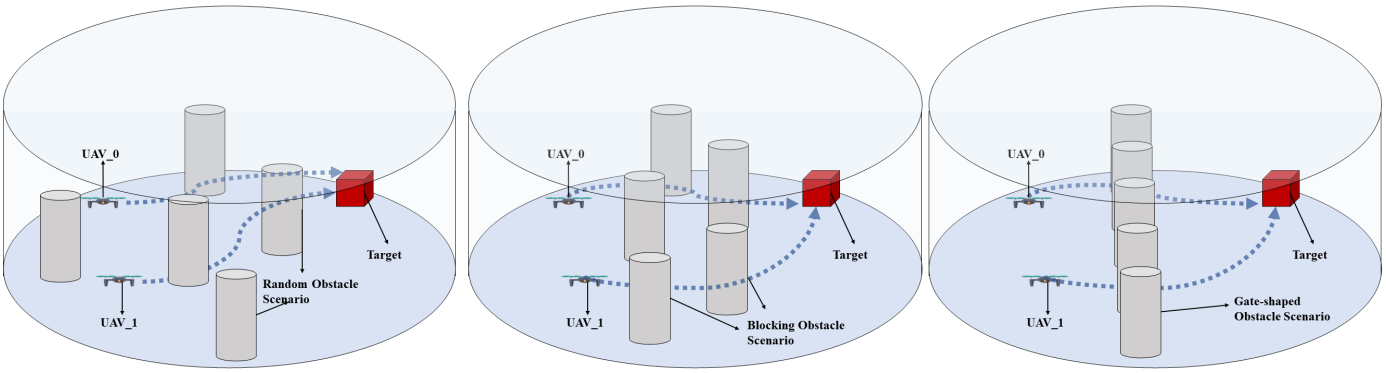
Based on the above observations, this paper proposes a Target-Aware Safety-Residual Pursuit Reinforcement Learning (TASRP) framework for cooperative multi-UAV persistent tracking in complex environments. The core methodological novelty of TASRP is not an isolated modification to an existing MARL backbone but a structured policy-learning framework specifically designed for cooperative multi-UAV pursuit in obstacle-constrained environments. Specifically, TASRP explicitly separates target-oriented nominal pursuit

from safety-oriented local correction through a dual-head residual policy, adaptively regulates the intervention of the safety branch through explicit risk-aware fusion, and introduces a target-aware blocking-obstacle teacher so that the learned corrective behavior supports obstacle bypassing while preserving pursuit intent. These components are further integrated with dual-critic constrained optimization under CTDE training, yielding a unified framework that differs from conventional single-policy MARL pursuit methods, generic safe reinforcement learning approaches, and standard residual reinforcement learning formulations.

3. Problem Formulation and Environment Definition

This paper investigates a dual-UAV cooperative persistent tracking problem in complex obstacle environments. In the considered task, two pursuing UAVs are required to continuously approach and track a dynamically maneuvering target within a bounded three-dimensional space while satisfying constraints including boundary limits, obstacle avoidance, and inter-agent safety separation. Unlike conventional target-approaching tasks in open environments, the proposed setting incorporates random, target-blocking, and gate-like cylindrical obstacle layouts. Here, the complexity is defined at the environment level rather than at the level of any single obstacle geometry: each obstacle is a simple cylinder, while the overall arrangement of multiple obstacles determines whether direct target-oriented motion is blocked, redirected, or funneled through narrow passages. In particular, target-blocking layouts may place obstacles directly between the UAVs and the target, making "bypassing obstacles while maintaining target tracking" a central challenge of the task.

Based on these considerations, the problem is formulated as a safety-constrained cooperative decentralized partially observable Markov decision process (Dec-POMDP), and the control policies are learned under the centralized training with decentralized execution (CTDE) framework [38]. The overall scenario setup is illustrated in Figure 1.



(**a**) (**b**) (**c**)

Figure 1. Representative obstacle configurations used in the experimental environments: (**a**) Random obstacle scenario with stochastically distributed obstacles. (**b**) Target-blocking obstacle scenario featuring structured occlusions that impede direct target-oriented motion. (**c**) Gate-like obstacle scenario forming constrained passageways. Note that obstacle positions are randomly initialized in each episode rather than fixed, thereby increasing scenario variability and improving policy robustness.

3.1. Problem Formulation and System Modeling

We define the task space as a bounded three-dimensional flight region,

$$\Omega = \left\{ p = [x, y, z]^\top \in \mathbb{R}^3 \;\middle|\; \|p_{xy} - c_{xy}\|_2 \le R_{\text{env}},\ z_{\min} \le z \le z_{\max} \right\} \tag{1}$$

where $p_{xy} = [x, y]^\top$ denotes the projection of the position onto the horizontal plane, $c_{xy} \in \mathbb{R}^2$ is the environment center in the horizontal plane, $R_{\text{env}} > 0$ is the boundary radius, and $z_{\min}$ and $z_{\max}$ denote the minimum and maximum allowable flight altitudes, respectively.

Two pursuing UAVs operate in this region, denoted as $\mathcal{U} = \{U_1, U_2\}$. In this study, the pursuing agents are multirotor UAVs. For any UAV $U_i \in \mathcal{U}$, its state at time t is defined as $x_{i,t} = (p_{i,t}, v_{i,t}, q_{i,t}, \omega_{i,t})$, where $p_{i,t} \in \mathbb{R}^3$ is the position of UAV i, $v_{i,t} \in \mathbb{R}^3$ is the linear velocity, $q_{i,t} \in \mathbb{S}^3$ is the attitude quaternion, and $\omega_{i,t} \in \mathbb{R}^3$ is the body angular velocity.

The target state at time t is defined as $x_t^{\text{tar}} = \left(p_t^{\text{tar}}, v_t^{\text{tar}}\right)$, where $p_t^{\text{tar}} \in \mathbb{R}^3$ and $v_t^{\text{tar}} \in \mathbb{R}^3$ denote the position and velocity of the target, respectively. Since the target exhibits time-varying motion, UAVs must continuously adapt their policies rather than relying on static planning.

To characterize geometric constraints, we define the obstacle set $\mathcal{O}_t = \{O_m\}_{m=1}^{M}$, where M is the maximum number of obstacles. Each obstacle is defined as $O_m = (p_m, r_m, h_m, \xi_m)$, where $p_m \in \mathbb{R}^3$ is the center position, $r_m > 0$ and $h_m > 0$ denote the radius and height of the cylindrical obstacle, respectively, and $\xi_m \in \{0, 1\}$ is a binary activation flag. The occupied region of the m-th active obstacle is defined as $\mathcal{C}_m = \{p \in \mathbb{R}^3 \mid \|p^{xy} - p_m^{xy}\|_2 \le r_m, 0 \le p^z \le h_m,\ \xi_m = 1\}$.

Accordingly, the overall obstacle region is represented as the union $\mathcal{X}_t^{\text{obs}} = \bigcup_{m=1}^{M} \mathcal{C}_m$.

For clarity, this work does not model the difficulty of individual obstacle geometry or perception. Each obstacle is represented by a simple cylindrical primitive. The task difficulty instead comes from the overall multi-obstacle layout, where multiple active obstacles jointly form different scenario instances. We denote by ω an obstacle-layout instance sampled for each episode, where ω specifies the arrangement of the active obstacles and may correspond to random, target-blocking, or gate-like layouts. Under this formulation, the challenge comes from how the multi-obstacle layout blocks direct target-oriented motion, creates constrained passages, and requires detouring during tracking.

By integrating UAVs, target, and obstacle information, the global state at time t is defined as

$$s_t = \left(x_{1,t}, x_{2,t}, x_t^{\text{tar}}, \mathcal{O}_t, b\right), \tag{2}$$

where b denotes constant environmental parameters, including boundary constraints, altitude limits, and safety thresholds. Therefore, the problem can be summarized as learning a cooperative control policy for multirotor UAVs in a dynamic target and obstacle-constrained 3D space, such that agents maintain stable tracking performance while achieving effective approach and successful capture. In particular, the detouring and collision-avoidance decisions considered in this work are fully three-dimensional rather than restricted to planar motion, because the policy is allowed to adjust both horizontal motion and altitude when bypassing obstacles and maintaining inter-agent safety. Equations (1)–(4) are introduced in this work as task/environment definitions and simulator-level control interface specifications for the considered cooperative UAV tracking setting. It should be noted that the present formulation does not aim to model the full raw-sensor-to-actuator pipeline. Instead, the sensing side is abstracted into structured local observations, and the action side is abstracted into continuous command-level control variables. This abstraction is adopted so that the learning problem can focus on cooperative pursuit, detouring, and safety-aware decision making in complex obstacle environments.

3.2. UAV Dynamics Modeling and Action Space Design

To more realistically capture the flight dynamics and control characteristics of UAVs, this paper adopts a continuous low-level control formulation. For each UAV, $U_i \in \mathcal{U}$, the control input at time t is defined as a four-dimensional continuous action vector,

$$a_{i,t} = \left[u_{T,i,t},\ u_{\phi,i,t},\ u_{\theta,i,t},\ u_{\psi,i,t}\right]^\top \in [-1,1]^4 \tag{3}$$

where $u_{T,i,t}$ denotes the normalized thrust command, while $u_{\phi,i,t}$, $u_{\theta,i,t}$, and $u_{\psi,i,t}$ represent attitude-related control inputs corresponding to roll, pitch, and yaw, respectively. The joint action is denoted as $a_t = \{a_{1,t}, a_{2,t}\}$.

During environment execution, the normalized action of each UAV is first clipped element-wise to $[-1,1]$, yielding the bounded action $\bar{a}_{i,t} = \left[\bar{u}_{T,i,t},\ \bar{u}_{\phi,i,t},\ \bar{u}_{\theta,i,t},\ \bar{u}_{\psi,i,t}\right]^\top = \mathrm{clip}(a_{i,t}, -1, 1)$, where any component exceeding the admissible range is truncated to the nearest boundary value. The clipped thrust and attitude commands are then mapped to physical control inputs. In the simulator, the total thrust and body-frame torque of UAV i at time t are defined as

$$T_{i,t} = f_T(\bar{u}_{T,i,t}) = \lambda_T m_i g \frac{\alpha_T \bar{u}_{T,i,t} + 1}{2}, \qquad \tau_{i,t} = k_M \begin{bmatrix} \bar{u}_{\phi,i,t} \\ \bar{u}_{\theta,i,t} \\ \bar{u}_{\psi,i,t} \end{bmatrix}, \tag{4}$$

where m_i denotes the mass of UAV i, g is the gravitational acceleration, $\lambda_T = 1.9$ is the thrust-to-weight ratio, $\alpha_T = 0.75$ is the thrust scaling factor, and $k_M = 0.01\ \mathrm{N \cdot m}$ is the torque scaling coefficient. Accordingly, the thrust command covers the range $T_{i,t} \in [0.2375\, m_i g,\ 1.6625\, m_i g]$, which places the hover operating point close to the zero-centered action region while preserving sufficient upward acceleration margin for aggressive obstacle-avoidance and target-tracking maneuvers. The torque coefficient k_M is chosen according to the Crazyflie-scale quadrotor dynamics used in the simulator and further tuned empirically to provide sufficient roll, pitch, and yaw control authority without causing excessively aggressive or unstable attitude responses.

Accordingly, the discrete-time state transition of each UAV can be abstracted as $x_{i,t+1} = f_U(x_{i,t}, T_{i,t}, \tau_{i,t})$, where $f_U(\cdot)$ represents the state transition function jointly determined by the rigid-body dynamics model and the control execution process in the simulation environment. The reason for adopting continuous control modeling is that, in complex obstacle environments, UAV detouring and tracking behaviors rely on fine-grained thrust modulation and attitude adjustment. Discrete action representations are, therefore, insufficient to stably capture the continuous, high-dynamic, and high-risk maneuvers required in such scenarios.

3.3. Decentralized Partially Observable Model

Under the centralized training with decentralized execution (CTDE) framework, the critic has access to the global state during training, while each UAV executes decentralized policies based solely on its local observation during execution. The local observation of UAV i at time t is denoted as $o_{i,t} \in \mathcal{O}_i$. To jointly capture target tracking and safety constraints, the local observation is decomposed into five components: ego-state, target-related information, teammate-related information, obstacle information, and boundary information.

In addition to the default clean simulator-interface observation used in the main formulation, this study also considers a noisy-observation variant for robustness evaluation. Specifically, controlled perturbations are injected into selected local observation channels to emulate moderate uncertainty at the state-estimation interface, including relative-position, velocity, angular-velocity, and boundary-related features. For a clean observation vector,

$o_{i,t}$, the perturbed observation is defined as $\tilde{o}_{i,t} = \mathrm{clip}(o_{i,t} + \nu_{i,t}, \underline{o}, \overline{o})$, where $\nu_{i,t} \sim \mathcal{N}(0, \Sigma)$ denotes the injected observation perturbation, and $\underline{o}$ and $\overline{o}$ denote the component-wise lower and upper bounds of the selected observation channels. The perturbation magnitudes are selected to approximate moderate estimation and relative-localization errors while preserving physically consistent scene semantics. Detailed experimental settings and comparative results under this noisy-observation protocol are reported in Section 5.

$$o_{i,t} = \left[o_{i,t}^{\mathrm{self}}, o_{i,t}^{\mathrm{tar}}, o_{i,t}^{\mathrm{team}}, o_{i,t}^{\mathrm{obs}}, o_{i,t}^{\mathrm{bd}}\right] \tag{5}$$

Since both pursuit and obstacle avoidance decisions depend on local geometric relationships in each UAV's reference frame, all relative positions and velocities are transformed into the body frame. Let $R(q_{i,t})$ denote the rotation matrix corresponding to the attitude quaternion $q_{i,t}$. Then, any relative vector, Δp, expressed in the world frame can be transformed into the body frame as

$$\Delta p_{i,t}^{b} = R(q_{i,t})^{\top} \Delta p \tag{6}$$

First, the ego-state feature is defined as $o_{i,t}^{\mathrm{self}} = \left[v_{i,t}^{b},\ \omega_{i,t}^{b},\ g_{i,t}^{b}\right]$, where $v_{i,t}^{b} \in \mathbb{R}^3$ and $\omega_{i,t}^{b} \in \mathbb{R}^3$ denote the linear velocity and angular velocity in the body frame, respectively, and $g_{i,t}^{b} \in \mathbb{R}^3$ is the gravity vector projected in the body frame.

The target-relative position and velocity in the body frame are defined as

$$\Delta p_{i,t}^{\mathrm{tar},b} = R(q_{i,t})^{\top}\left(p_{t}^{\mathrm{tar}} - p_{i,t}\right) \tag{7}$$

$$v_{t}^{\mathrm{tar},b} = R(q_{i,t})^{\top} v_{t}^{\mathrm{tar}} \tag{8}$$

Accordingly, the teammate feature is defined as $o_{i,t}^{\mathrm{team}} = \left[\Delta p_{i,j,t}^{b},\ v_{j,t}^{b}\right]$.

To accommodate varying numbers of obstacles across episodes while maintaining a fixed observation dimension, a fixed-length encoding scheme is adopted. Let M denote the maximum number of obstacles in the environment. The local feature corresponding to the m-th obstacle is defined as $z_{i,m,t} = \left[\Delta p_{i,m,t}^{b},\ \widetilde{r}_{m},\ \widetilde{h}_{m},\ \xi_{m}\right]$. where

$$\Delta p_{i,m,t}^{b} = R(q_{i,t})^{\top}\left(p_{m} - p_{i,t}\right) \tag{9}$$

Here, $\widetilde{r}_{m}$ and $\widetilde{h}_{m}$ denote the normalized radius and height of the obstacle, respectively. In the simulator implementation, the obstacle radius is normalized with respect to the environment radius R_{env}, and the obstacle height is normalized with respect to the maximum flight altitude $z_{\max}$. Meanwhile, the obstacle-relative position is scaled by a fixed distance normalization factor and clipped into $[-1, 1]$ before being concatenated with the geometric attributes. The variable $\xi_{m} \in \{0, 1\}$ is the activation flag indicating whether the corresponding obstacle slot is valid. All obstacle features are concatenated as $o_{i,t}^{\mathrm{obs}} = [z_{i,1,t},\ z_{i,2,t},\ \ldots,\ z_{i,M,t}]$, where inactive obstacles are zero-masked to maintain a consistent input dimension. Since the policy network in this work uses an MLP-based encoder rather than an explicit attention module, the fixed-length zero-masking strategy does not introduce a separate learned attention score over obstacle slots. Instead, it provides structurally consistent placeholder entries for nonexistent obstacles, so that padded slots contribute negligibly to the hidden representation, while valid obstacles retain nonzero normalized geometry together with $\xi_{m} = 1$ and can, therefore, be implicitly emphasized by the encoder and subsequent risk-evaluation modules during training. This design preserves a fixed observation dimension, avoids spurious responses caused by arbitrary padding values, and improves training stability when the number of obstacles varies across episodes.

Boundary information is used to describe the relative geometric relationship and safety margin between the UAV and the feasible flight region. The horizontal boundary margin is defined as

$$m_{i,t}^{r} = R_{\text{env}} - \left\| p_{i,t}^{xy} - c_{xy} \right\|_2 \tag{10}$$

The lower and upper altitude margins are defined as $m_{i,t}^{z,\min} = p_{i,t}^{z} - z_{\min}, \quad m_{i,t}^{z,\max} = z_{\max} - p_{i,t}^{z}$, where $p_{i,t}^{z}$ denotes the altitude of UAV i at time t. The overall boundary safety margin is defined as $m_{i,t}^{\text{bd}} = \min\left(m_{i,t}^{r},\ m_{i,t}^{z,\min},\ m_{i,t}^{z,\max}\right)$. The boundary center expressed in the body frame is given by $c_{i,t}^{b} = R(q_{i,t})^{\top}(c - p_{i,t})$, where $c = \left[c_x, c_y, c_z\right]^{\top}$ denotes the environment center. Therefore, the boundary feature is defined as $o_{i,t}^{\text{bd}} = \left[c_{i,t}^{b},\ m_{i,t}^{\text{bd}}\right]$.

It should be noted that the minimum operator in $m_{i,t}^{\text{bd}}$ yields a continuous but piecewise-defined boundary-margin representation. Switching only occurs when the active limiting factor changes among the radial margin, the lower altitude margin, and the upper altitude margin. In the present work, this quantity is introduced as a geometric observation feature to indicate the most critical boundary bottleneck at the current state, rather than as a control variable in a model-based differentiable optimization process.

Overall, the local observation $o_{i,t}$ contains the essential information required for both target pursuit and safety-aware decision-making, providing a unified and structured input representation for subsequent policy learning.

3.4. Safety Constraints and Termination Conditions

This study explicitly considers boundary constraints, inter-UAV safe-separation constraints, and obstacle-collision constraints in the task design. First, the boundary constraint requires all UAVs to remain within the task space Ω at all times, that is, $p_{i,t} \in \Omega, \quad \forall \mathcal{U}_i \in \mathcal{U}$. Next, let the Euclidean distance between two UAVs be $d_{12,t}^{UU} = \|p_{1,t} - p_{2,t}\|_2$. Then, the corresponding safe-separation constraint can be written as $d_{12,t}^{UU} \geq d_{\min}^{UU}$, where $d_{\min}^{UU}$ denotes the minimum allowable safe distance. When the distance further decreases below the collision threshold d_{col}^{UU}, an intra-team collision is considered to have occurred.

It should be noted that the minimum allowable safe distance, $d_{\min}^{UU}$, is not selected arbitrarily. Instead, it is set as an engineering safety-margin parameter according to the physical size and rotor-envelope clearance of the multirotor UAVs and the confined spatial scale of the environment. In this way, $d_{\min}^{UU}$ serves as a preventive separation threshold before the actual collision boundary is reached. Moreover, the safety constraints and termination conditions in this work are defined mainly through geometric violation events, including inter-UAV separation, obstacle contact, and boundary violation. Velocity is, therefore, not introduced as an additional explicit termination variable in this subsection, although it is already included in the system state and local observation and influences collision risk through the UAV dynamics and state evolution.

For obstacle constraints, define the horizontal-plane distance between the i-th UAV and the m-th obstacle as $d_{i,m,t}^{xy} = \left\| p_{i,t}^{xy} - p_{m}^{xy} \right\|_2$. Then, the distance from the UAV to the obstacle surface is given by $d_{i,m,t}^{\text{surf}} = d_{i,m,t}^{xy} - (r_m + r_U)$, where r_U denotes the effective collision radius of the UAV. When $d_{i,m,t}^{\text{surf}} \leq 0 \quad \text{and} \quad z_{i,t} \leq h_m$ holds, the UAV is regarded as colliding with the obstacle.

In terms of task completion measurement, a successful capture event is defined according to the target-neighborhood entry criterion. Let r_c denote the capture radius. Then,

$$\text{Capture}(t) \iff \exists \mathcal{U}_i \in \mathcal{U}, \left\| p_{i,t} - p_{t}^{\text{tar}} \right\|_2 \leq r_c \tag{11}$$

This condition is used for task-success determination and reward design. Since the environment focuses on persistent tracking capability, a capture event does not necessarily terminate the episode.

Episode termination events mainly include two categories: collision failure and time truncation. Specifically, when an obstacle collision, boundary violation, or inter-UAV collision occurs, the episode is marked as failure termination; when the maximum interaction step number $T_{\max}$ is reached, it is marked as time truncation. Therefore, the episode termination criterion can be uniformly written as $I_{\text{done},t} = I_{\text{obs},t} \vee I_{\text{bd},t} \vee I_{\text{col},t} \vee I_{t=T_{\max}}$, where $I_{\text{obs},t}$, $I_{\text{bd},t}$, and $I_{\text{col},t}$ denote the indicator functions of obstacle collisions, boundary violations, and inter-UAV collisions, respectively. The above constraints and termination conditions jointly constitute the safety foundation of the task, forcing the policy to balance obstacle avoidance, boundary maintenance, and cooperative safety while optimizing target-tracking performance.

3.5. Constrained Optimization Problem Formulation

Based on the above system definition, continuous-control interface, decentralized observation model, and safety constraints, this study formulates the dual-UAV cooperative persistent tracking problem in complex obstacle environments as a cooperative Dec-POMDP subject to safety constraints:

$$\mathcal{G} = \langle \mathcal{U}, \mathcal{S}, \{\mathcal{O}_i\}_{i\in\mathcal{U}}, \{\mathcal{A}_i\}_{i\in\mathcal{U}}, P, R, C, \gamma \rangle \tag{12}$$

where $\mathcal{U}$ denotes the UAV set, $\mathcal{S}$ denotes the global state space, and $\mathcal{O}_i$ and $\mathcal{A}_i$ denote the local observation space and action space of the i-th UAV, respectively. Moreover, P is the environment transition function, R is the task reward function, C is the safety cost function, and $\gamma \in (0,1)$ is the discount factor. Let ω denote an obstacle-layout instance sampled for each episode. The sampled layout determines the active obstacle configuration and, therefore, changes the feasible motion corridors, the likelihood of safety violations, and the detouring effort required to maintain target-oriented tracking. Accordingly, under a given layout instance, ω, the transition kernel, task reward, and safety cost can be written more explicitly as P_ω, R_ω, and C_ω, respectively, although the dependence on ω is omitted in the tuple notation above for brevity.

Under the CTDE framework, during execution, the i-th UAV produces an action based on its local observation $o_{i,t}$, and its local policy is defined as $\pi_i(a_{i,t} \mid o_{i,t})$. The joint policy can be expressed as

$$\pi(a_t \mid o_t) = \prod_{i\in\mathcal{U}} \pi_i(a_{i,t} \mid o_{i,t}) \tag{13}$$

where $o_t = \{o_{1,t}, o_{2,t}\}$. During training, the critic can access the global state s_t, whereas during execution each UAV makes decisions solely based on its own local observation, thereby satisfying the decentralized-execution requirement.

The task objective of this study can be summarized as follows: under the obstacle region and safety constraints, a set of cooperative continuous-control policies is learned for two UAVs, so that they can stably and persistently track a dynamic target while improving target-approach efficiency, reducing collision risk, and maintaining flight safety throughout the interaction process. For a sampled obstacle-layout instance ω, the obstacle-avoidance constraint is defined by the corresponding layout-conditioned obstacle region $\mathcal{X}_t^{\text{obs}}(\omega)$, so that the state constraint can be written as $p_{i,t} \in \Omega \setminus \mathcal{X}_t^{\text{obs}}(\omega)$ for all $i \in \mathcal{U}$ and all t.

Equivalently, for each active obstacle whose height overlaps the UAV altitude, the UAV is required to satisfy $\|p_{i,t}^{xy} - p_m^{xy}\|_2 > r_m + r_U$ whenever $z_{i,t} \le h_m$.

Moreover, let $\overline{p_{i,t}^{xy} p_t^{\mathrm{tar},xy}}$ denote the line segment from UAV i to the target in the horizontal plane. A layout is termed target-blocking if at least one active obstacle lies sufficiently close to this target-oriented line segment and its projection falls within the segment interior, so that direct pursuit is likely to intersect the obstacle field and detouring becomes necessary. By contrast, random layouts mainly correspond to stochastic obstacle placements under the same feasibility constraints, whereas gate-like layouts correspond to cases in which multiple obstacles jointly form a narrow traversable passage.

Under a given layout instance, ω, let the cumulative task return and cumulative safety cost be denoted by $J_R(\pi;\omega)$ and $J_C(\pi;\omega)$, respectively, where

$$J_R(\pi;\omega) = \mathbb{E}_{P_\omega,\pi}\left[\sum_{t=0}^{T_{\max}} \gamma^t r_t(s_t,a_t;\omega)\right], \tag{14}$$

and

$$J_C(\pi;\omega) = \mathbb{E}_{P_\omega,\pi}\left[\sum_{t=0}^{T_{\max}} \gamma^t c_t(s_t,a_t;\omega)\right]. \tag{15}$$

Then, the optimization objective can be written as

$$\max_{\pi}\ \mathbb{E}_{\omega\sim p(\omega)}[J_R(\pi;\omega)] \quad \text{s.t.} \quad \mathbb{E}_{\omega\sim p(\omega)}[J_C(\pi;\omega)] \le d. \tag{16}$$

where d denotes the allowable upper bound of the safety cost. This formulation clarifies that random, target-blocking, and gate-like layouts do not introduce different optimization objectives; instead, they instantiate different obstacle environments under the same constrained optimization framework. Their difference lies in how they modify feasible motion corridors, the likelihood of safety violations, and the detouring effort required to maintain target-oriented progress. As a result, the layout instance ω changes the reward-cost trade-off encountered by the policy, even though the form of Equation (16) remains unchanged. This optimization objective provides the theoretical basis for the subsequent TASRP framework: on the one hand, stable target-pursuit capability must be maintained; on the other hand, an effective safety-correction mechanism must be introduced under high-risk states.

4. Architecture and Algorithm Design

For the dual-UAV cooperative persistent tracking task in complex obstacle environments defined in Section 3, this study proposes the Target-Aware Safety-Residual Pursuit (TASRP) framework to model the coupled decision-making problem between target pursuit and safe avoidance under continuous control.

The essential methodological novelty of TASRP is not that each of its components is individually unprecedented but that it provides a unified structured policy-learning framework specifically designed for cooperative multi-UAV pursuit in obstacle-constrained environments. In contrast to CTDE/MAPPO-based pursuit methods that usually rely on a single monolithic policy to jointly absorb target tracking and safety avoidance, TASRP explicitly decomposes target-oriented nominal pursuit and safety-oriented local correction and coordinates them through risk-adaptive intervention. In contrast to conventional safe reinforcement learning approaches that mainly introduce safety through reward penalties, safety critics, or constrained updates, TASRP further embeds safety intervention directly into the action-generation process through risk-adaptive residual fusion. In contrast to standard residual reinforcement learning methods that mainly perform generic local action correction, TASRP introduces a target-oriented blocking-obstacle teacher to guide pursuit-consistent detouring around obstacles that hinder target advancement.

Therefore, the dual-head residual policy, risk-adaptive fusion, target-oriented blocking-obstacle teacher, and dual-critic constrained optimization should be understood as coordinated realizations of a unified methodological idea rather than as four independent novelties.

After clarifying that different obstacle layouts affect the constrained objective by changing feasible motion corridors, safety risk, and detouring difficulty, Table 1 further shows how the major components of TASRP realize different parts of Equation (16) during policy learning. Note that the mapping includes not only the safety-related modules but also the target-tracking branch because Equation (16) contains both a return-maximization term and a safety-cost constraint.

Table 1. Mapping from the constrained objective in Equation (16) under different obstacle layouts to TASRP components.

Component	Implementation in TASRP	Relation to Equation (16)
Target-tracking branch	Main action mean $\mu_{i,t}^{\mathrm{trk}}$ in Equations (17), (21) and (22)	Preserves the dominant reward-oriented behavior and mainly contributes to maximizing the expected persistent tracking return term $\mathbb{E}_{\omega}[J_R(\pi;\omega)]$.
Safety-residual branch	Raw and effective residuals in Equations (19)–(22)	Introduces local corrective degrees of freedom for reducing obstacle, boundary, and local safety violations without fully replacing the nominal target-pursuit policy.
Risk-adaptive gate	Explicit risk modeling and fusion weights in Equations (23)–(25)	Realizes state-dependent constraint enforcement by strengthening safety correction in high-risk states and suppressing unnecessary intervention in low-risk states, thereby approximating the control of the expected safety-cost term $\mathbb{E}_{\omega}[J_C(\pi;\omega)]$.
Target-oriented blocking-obstacle teacher	Blocking score, teacher residual, and auxiliary supervision in Equations (26)–(29)	Improves the directionality of safety correction under blockage, helping the policy reduce safety violations while preserving target-oriented progress instead of learning purely repulsive detours.
Dual critics and constrained update	Reward critic, cost critic, PPO reward/cost losses, and Lagrangian update in Equations (30)–(37)	Directly implement the layout-conditioned constrained optimization in Equation (16) by jointly estimating task return, safety cost, and their corresponding policy-update signals during training.

The target-tracking branch is included because Equation (16) contains both the expected return-maximization term and the safety-cost constraint; without this row, the mapping to the reward side would remain incomplete.

As shown in Table 1, TASRP implements Equation (16) in a layered manner rather than through a single undifferentiated policy representation. The target-tracking branch mainly preserves the return-oriented behavior required for persistent pursuit, the safety-residual branch and risk-adaptive gate provide state-dependent corrective control for reducing safety cost under different obstacle layouts, the target-oriented teacher improves the directionality of such correction under target-blocking situations, and the dual critics directly optimize the return-cost trade-off during training. Therefore, the following subsections should be understood as a component-wise realization of the constrained Dec-POMDP objective rather than as a descriptive list of architectural modules.

4.1. Overall TASRP Framework

Based on the Dec-POMDP $\mathcal{G} = \langle \mathcal{U}, \mathcal{S}, \{\mathcal{O}_i\}_{i\in\mathcal{U}}, \{\mathcal{A}_i\}_{i\in\mathcal{U}}, P, R, C, \gamma \rangle$ defined in Section 3, TASRP constructs a unified structured local policy for each UAV. For the i-th UAV, the local observation $o_{i,t}$ at time t is first encoded into a feature representation, $h_{i,t} = \phi(o_{i,t})$, where $\phi(\cdot)$ is a parameter-shared multilayer perceptron (MLP) encoder. Based on the encoded feature $h_{i,t}$, the policy network produces three outputs: the target-pursuit main action, the safety-residual correction, and the fusion-gating coefficient. Accordingly, the generation of the fused mean action $\mu_{i,t}$ in TASRP can be written as

$$\mu_{i,t} = \Psi(o_{i,t}) = \mathcal{F}\left(\mu_{i,t}^{\mathrm{trk}}, \delta_{i,t}^{\mathrm{raw}}, \alpha_{i,t}\right) \tag{17}$$

where $\mu_{i,t}^{\mathrm{trk}}$ is the continuous action mean generated by the main tracking branch, $\delta_{i,t}^{\mathrm{raw}}$ is the raw residual generated by the safety branch, $\alpha_{i,t}$ is the fusion gating coefficient, and $\mathcal{F}(\cdot)$ denotes the subsequent risk-adaptive action-fusion process. Based on the fused mean $\mu_{i,t}$, the local execution policy is modeled as a Gaussian distribution:

$$\pi_{\theta_i}(a_{i,t} \mid o_{i,t}) = \mathcal{N}\left(\mu_{i,t}, \mathrm{diag}\left(\sigma_\theta^2\right)\right) \tag{18}$$

where $\mathrm{diag}(\sigma_\theta^2)$ denotes the diagonal covariance matrix constructed from learnable log-standard-deviation parameters. Equations (17) and (18) show that TASRP does not directly regard the fused result as a deterministic control command. Instead, it first generates a structured mean action and then samples actions from a continuous stochastic policy, thereby remaining consistent with the optimization framework of PPO/MAPPO-style policy-gradient methods.

As illustrated in Figure 2, TASRP under the CTDE paradigm forms a unified learning framework for complex-obstacle pursuit tasks. For each UAV agent, the local observation $o_{i,t}$ is fed into the encoder to extract the feature representation $h_{i,t}$ used for action decision-making. On this basis, the TASRP actor is designed as a dual-head policy, where the target-tracking head outputs the mean action $\mu_{i,t}^{\mathrm{trk}}$, the safety-residual head outputs the raw safety residual $\delta_{i,t}^{\mathrm{raw}}$, and the fusion head further produces the base fusion coefficient $\overline{\alpha}_{i,t}$. Meanwhile, the algorithm constructs explicit risk gating according to obstacle risk and boundary risk, and combines it with the fusion coefficient to obtain the final fusion weight $\alpha_{i,t}$, thereby yielding the effective safety residual $\widetilde{\delta}_{i,t}$. Then, the main tracking action and the effective safety residual are fused through the risk-adaptive action-fusion module to generate the final mean action $\mu_{i,t}$, from which the Gaussian policy samples and executes the continuous action.

During training, TASRP further introduces a target-oriented blocking-obstacle teacher to provide auxiliary supervision for the safety-residual head and improve its obstacle-bypassing efficiency in target-blocking-obstacle scenarios. Meanwhile, the PPO critic is composed of a reward critic and a safety-cost critic, which estimate task return and safety cost, respectively, and jointly optimize policy learning through constrained updates. Thus, TASRP forms a complete pipeline from local-state perception and structured action generation to risk-adaptive fusion and constrained policy optimization.

4.2. Target-Aware Safety-Residual Policy

In pursuit tasks under complex obstacle layouts, if a single policy directly outputs the full continuous action, the two types of control demands, namely target advancement and safe avoidance, are usually entangled in the same representation, which easily leads to conflicting objectives during training. To address this issue, this study proposes a target-aware safety-residual policy that decomposes action generation into a main tracking action and a local safety correction.

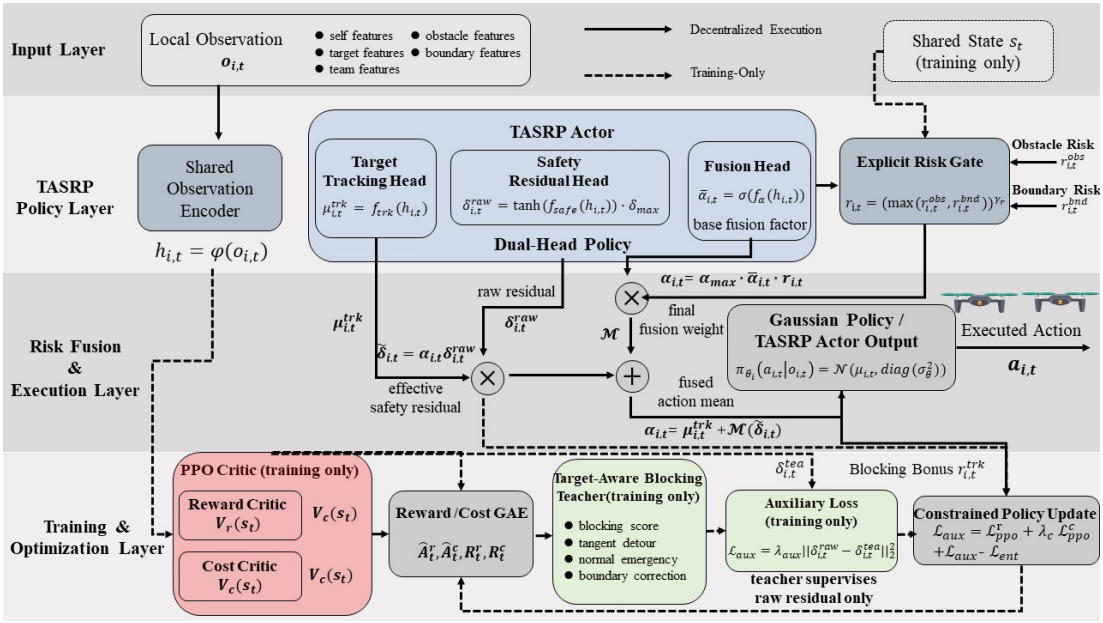


Figure 2. Overview of the TASRP framework under the CTDE paradigm. Local observations are encoded and processed by a dual-head actor to generate a target-tracking action, a safety residual, and a fusion factor. An explicit risk gate modulates residual intervention before decentralized action execution. During training, a reward critic, a safety-cost critic, and a target-oriented blocking-obstacle teacher jointly guide learning. The figure summarizes the information flow and the functional roles of the main TASRP modules.

For the i-th UAV, based on the encoded feature $h_{i,t}$, the main tracking branch outputs the full action mean $\mu_{i,t}^{\mathrm{trk}} = f_{\mathrm{trk}}(h_{i,t}) \in \mathbb{R}^{d_a}$. Here, $d_a = 4$, corresponding to the continuous action dimensions defined in Section 3, namely total thrust, roll-, pitch-, and yaw-related control components. This branch is responsible for the dominant control intent required for target approach and persistent tracking. Meanwhile, instead of replacing the full action, the safety branch only outputs a residual correction acting on a local maneuver subspace:

$$\delta_{i,t}^{\mathrm{raw}} = \tanh(f_{\mathrm{safe}}(h_{i,t})) \cdot \delta_{\max} \tag{19}$$

where $\delta_{\max}$ is the upper bound on the safety-residual magnitude. Through this bounded residual design, the role of the safety branch is limited to local compensation of the main tracking action rather than full replacement, thus preventing the stable target-advancement structure learned by the main tracking branch from being destroyed. Let the full action-dimension set be $\{1, \ldots, d_a\}$, and let the subset of dimensions actually affected by the safety residual be $\mathcal{S} \subset \{1, \ldots, d_a\}$. In the considered four-dimensional action space, the action components correspond to total thrust, roll torque, pitch torque, and yaw torque. In this work, $\mathcal{S}$ is specified as the roll- and pitch-related dimensions, meaning that the safety residual is only applied to the two attitude channels most directly associated with short-horizon obstacle-bypassing maneuvers. The physical rationale is that local collision avoidance for the considered multirotor UAV is mainly achieved by adjusting roll and pitch so as to redirect the thrust vector and generate immediate lateral detouring motion. By contrast, the thrust dimension primarily determines overall lift and altitude support, and allowing the safety branch to directly perturb thrust would more easily interfere with altitude stability and the nominal target-approach behavior maintained by the main tracking branch. Similarly, the yaw channel mainly affects heading orientation rather than producing immediate translational clearance from nearby obstacles, and is, therefore, kept under the dominant control of the main tracking branch. Accordingly, the mapping operator $M(\cdot)$ in Equation (22) should be understood as embedding the low-dimensional

safety residual only into the roll and pitch components of the full action, while leaving the thrust and yaw components unchanged. Under risk gating, the raw safety residual is modulated into an effective safety residual:

$$\widetilde{\delta}_{i,t} = \alpha_{i,t}\delta_{i,t}^{\mathrm{raw}} \tag{20}$$

Thus, the final fused action mean can be written as

$$\mu_{i,t}^{(k)} = \begin{cases} \mu_{i,t}^{\mathrm{trk},(k)} + \widetilde{\delta}_{i,t}^{(k)}, & k \in \mathcal{S}, \\ \mu_{i,t}^{\mathrm{trk},(k)}, & k \notin \mathcal{S}, \end{cases} \tag{21}$$

or, equivalently, in a compact form:

$$\mu_{i,t} = \mu_{i,t}^{\mathrm{trk}} + M\left(\widetilde{\delta}_{i,t}\right) \tag{22}$$

where $M(\cdot)$ denotes the mapping operator that embeds the low-dimensional residual into the corresponding dimensions of the full action space. Equations (21) and (22) indicate that TASRP does not construct a full obstacle-avoidance controller parallel to the main policy. Instead, it only allows the safety branch to correct those action dimensions that are more closely related to local obstacle-bypassing maneuvers, so that the main tracking branch always maintains dominance over the overall flight behavior.

It should be emphasized that the term "target-aware" does not mean that the safety branch directly outputs an action pointing toward the target. Rather, it means that the learning objective of the safety residual remains aligned with the tracking task itself. In other words, the purpose of the safety correction is not simply to push the UAV away from obstacles but to provide local compensation under high-risk states while preserving the target-advancement tendency as much as possible. In this way, the UAV can form a small-scale continuous obstacle-bypassing tendency when encountering blocking obstacles and continue to pursue the target after bypassing them. This idea is further embodied in the teacher design in Section 4.4.

In terms of action-distribution modeling, TASRP maintains the Gaussian-policy form of standard continuous-control policy-gradient methods, while introducing an explicit structural separation between nominal target pursuit and local safety correction.

4.3. Risk-Adaptive Fusion Mechanism

After obtaining the main tracking action $\mu_{i,t}^{\mathrm{trk}}$ and the safety residual $\delta_{i,t}^{\mathrm{raw}}$, how to determine the intervention strength of the safety correction under different states becomes a key factor affecting policy performance and stability. If the safety residual always acts on the final action with a fixed strength, it may cause unnecessary interference in low-risk regions, while in high-risk regions, it may fail to prevent collisions in time because the correction is too weak. To address this issue, this study designs a risk-adaptive fusion mechanism that dynamically regulates the actual effect of the safety residual according to the obstacle proximity and boundary safety margin in local observations.

First, according to the obstacle geometry defined in Section 3, let the surface distance between the k-th obstacle and the UAV be $d_{i,k,t}^{\mathrm{surf}} = \left\| p_{i,k,t}^{\mathrm{obs}} \right\|_2 - r_{k,t}^{\mathrm{obs}}$, where $p_{i,k,t}^{\mathrm{obs}}$ denotes the relative obstacle position, and $r_{k,t}^{\mathrm{obs}}$ denotes the equivalent obstacle radius. Given an obstacle influence threshold, $d_{\mathrm{inf}}^{\mathrm{obs}} > 0$, the normalized risk corresponding to the k-th obstacle is defined as

$$r_{i,k,t}^{\mathrm{obs}} = \mathrm{clip}_{[0,1]}\left(\frac{d_{\mathrm{inf}}^{\mathrm{obs}} - d_{i,k,t}^{\mathrm{surf}}}{d_{\mathrm{inf}}^{\mathrm{obs}}}\right) \tag{23}$$

The overall obstacle risk is then obtained by taking the maximum over all valid obstacles, i.e., $r_{i,t}^{\mathrm{obs}} = \max_k r_{i,k,t}^{\mathrm{obs}}$. Next, according to the boundary safety margin $m_{i,t}^{\mathrm{bd}}$ defined in Section 3 and a boundary influence threshold $m_{\mathrm{inf}}^{\mathrm{bnd}} > 0$, the boundary risk is defined as $r_{i,t}^{\mathrm{bnd}} = \mathrm{clip}_{[0,1]}\left(\frac{m_{\mathrm{inf}}^{\mathrm{bnd}} - m_{i,t}^{\mathrm{bd}}}{m_{\mathrm{inf}}^{\mathrm{bnd}}}\right)$. When the UAV is far from the boundary, $r_{i,t}^{\mathrm{bnd}}$ is close to 0; when the UAV approaches the boundary, this value gradually increases. By combining obstacle and boundary factors, the explicit geometric risk at the current time step is defined as $r_{i,t} = \left(\max(r_{i,t}^{\mathrm{obs}}, r_{i,t}^{\mathrm{bnd}})\right)^{\gamma_r}$, where $\gamma_r > 0$ is a risk exponent used to enhance the discrimination between high-risk states and medium-to-low-risk states.

From the optimization viewpoint, the possible effect of this piecewise definition on PPO is limited. In our implementation, PPO optimizes the policy with respect to network parameters from sampled observations, rather than differentiating through the handcrafted boundary-margin computation or the environment transition. Therefore, the non-smoothness of the minimum operator does not enter policy optimization as a direct end-to-end gradient path. Its practical influence is mainly reflected in the occasional switching of the active boundary indicator near the intersection of different limiting surfaces. This influence is further moderated by the clipped boundary-risk mapping, the risk-adaptive residual fusion, and the clipped surrogate objective in PPO. Hence, the minimum-based boundary margin serves as a conservative safety indicator, while its non-smoothness is not expected to be a dominant source of optimization instability.

On this basis, the gating branch outputs a base fusion coefficient according to the encoded feature, i.e., $\overline{\alpha}_{i,t} = \sigma(f_\alpha(h_{i,t}))$, where $\sigma(\cdot)$ is the sigmoid function. By combining the explicit risk with the base gate, the final fusion weight is obtained as

$$\alpha_{i,t} = \alpha_{\max} \cdot \overline{\alpha}_{i,t} \cdot r_{i,t} \tag{24}$$

where $\alpha_{\max} \in (0, 1]$ denotes the maximum intervention ratio of the safety residual. Combined with Equation (20), the effective safety residual can be further written as

$$\widetilde{\delta}_{i,t} = \alpha_{i,t} \cdot \delta_{i,t}^{\mathrm{raw}} \tag{25}$$

When Equations (22) and (25) are combined, the final fused action mean can be written as $\mu_{i,t} = \mu_{i,t}^{\mathrm{trk}} + M(\widetilde{\delta}_{i,t})$. In this way, the risk-adaptive gate strengthens safety correction in high-risk states while avoiding unnecessary intervention in low-risk states.

4.4. Target-Oriented Blocking-Obstacle Teacher

Although Sections 4.2 and 4.3 improve the specificity of safety correction from the perspectives of policy representation and action fusion, the safety-residual branch may still suffer from sparse learning signals and unstable correction directions if only environmental rewards and collision penalties are used for end-to-end reinforcement learning. In particular, in the blocking-obstacle and gate-like obstacle scenarios considered in this study, merely learning a repulsive correction that moves away from obstacles is insufficient for achieving good tracking performance, because the UAV may avoid collisions while deviating from the target-advancement direction and thus degrading persistent tracking ability. To address this issue, this study introduces a target-oriented blocking-obstacle teacher during training to provide auxiliary supervision with geometric meaning for the safety-residual branch.

Let the planar projected direction from the i-th UAV to the target be $\widehat{g}_{i,t} = \frac{p_{i,t}^{\mathrm{tar}}}{\left\|p_{i,t}^{\mathrm{tar}}\right\|_2 + \varepsilon}$, where $p_{i,t}^{\mathrm{tar}}$ is the projection of the relative target position onto the roll-pitch safety-correction subspace defined by the action dimensions affected by the safety residual and $\varepsilon > 0$ is a stabilization term. For the k-th obstacle in the local observation, its proximity strength

is first defined as $s_{i,k,t} = \mathrm{clip}_{[0,1]}\left(\frac{d_{\mathrm{inf}}^{\mathrm{tea}} - d_{i,k,t}^{\mathrm{surf}}}{d_{\mathrm{inf}}^{\mathrm{tea}}}\right)$, where $d_{\mathrm{inf}}^{\mathrm{tea}}$ is the obstacle influence-distance threshold used by the teacher. To quantify the degree to which an obstacle blocks target advancement, define the unit vector of the obstacle-relative direction as $\hat{p}_{i,k,t}^{\mathrm{obs}} = \frac{p_{i,k,t}^{\mathrm{obs}}}{\left\|p_{i,k,t}^{\mathrm{obs}}\right\|_2 + \varepsilon}$, and construct the blocking projection coefficient $b_{i,k,t} = \mathrm{clip}_{[0,1]}(\hat{p}_{i,k,t}^{\mathrm{obs}} \cdot \hat{g}_{i,t})$. The target-oriented blocking score of the k-th obstacle is then defined as $q_{i,k,t} = s_{i,k,t} \cdot b_{i,k,t}^{\eta_b}$, where $\eta_b > 0$ is the blocking-projection weight. The teacher then selects the obstacle with the largest blocking score as the current principal blocking obstacle:

$$k_i^* = \arg\max_k q_{i,k,t} \tag{26}$$

After determining the principal blocking obstacle, the teacher does not simply output a radial correction away from the obstacle. Instead, it constructs a composite correction consisting of a tangential bypassing term and a normal emergency term. Let the unit normal direction away from the principal blocking obstacle be $\hat{n}_{i,t} = -\frac{p_{i,k_i^*,t}^{\mathrm{obs}}}{\left\|p_{i,k_i^*,t}^{\mathrm{obs}}\right\|_2 + \varepsilon}$. In the planar subspace, two candidate tangential directions can be constructed from $\hat{n}_{i,t}$, namely $\hat{t}_{i,t}^{(1)} = (-\hat{n}_y, \hat{n}_x)$ and $\hat{t}_{i,t}^{(2)} = (\hat{n}_y, -\hat{n}_x)$. By comparing their consistency with the target direction $\hat{g}_{i,t}$, the bypassing direction that is more favorable for continued pursuit is selected as

$$\hat{t}_{i,t} = \arg\max_{\hat{t} \in \{\hat{t}_{i,t}^{(1)}, \hat{t}_{i,t}^{(2)}\}} \hat{t} \cdot \hat{g}_{i,t} \tag{27}$$

Based on this, the tangential bypassing correction, the normal emergency correction, and the boundary-guidance correction together form the target-oriented safety-residual reference. Let their sum be $\delta_{i,t}^{\mathrm{tea,pre}}$. Consistently, this teacher reference is also constructed in the same roll-pitch subspace, so that the auxiliary supervision remains aligned with the action dimensions actually corrected by the safety-residual branch. The final teacher reference residual is then defined as

$$\delta_{i,t}^{\mathrm{tea}} = \mathrm{Proj}_{\|\cdot\| \le \delta_{\mathrm{tea}}^{\max}}\left(\delta_{i,t}^{\mathrm{tea,pre}}\right) \tag{28}$$

where $\mathrm{Proj}_{\|\cdot\| \le \delta_{\mathrm{tea}}^{\max}}$ denotes magnitude truncation, which limits the amplitude of the teacher output. During training, the teacher does not directly replace the policy action but only serves as an auxiliary supervision signal for the safety-residual branch. Let the raw safety residual be $\delta_{i,t}^{\mathrm{raw}}$. Then, the auxiliary supervision loss is defined as

$$\mathcal{L}_{\mathrm{aux}} = \lambda_{\mathrm{aux}} \cdot \mathbb{E}\left[\left\|\delta_{i,t}^{\mathrm{raw}} - \delta_{i,t}^{\mathrm{tea}}\right\|_2^2\right] \tag{29}$$

where λ_{aux} is the auxiliary-loss weight. Through Equation (29), the teacher provides geometrically meaningful supervision for pursuit-consistent obstacle bypassing rather than generic obstacle repulsion.

It should be noted that the auxiliary supervision in Equation (29) may introduce additional gradients that interact with the reward-advantage-driven policy update. However, this interaction is structurally localized rather than global. Specifically, Equation (29) is applied to the raw safety residual $\delta_{i,t}^{\mathrm{raw}}$ rather than directly to the nominal target-tracking action. Therefore, its direct effect is mainly concentrated on the safety-residual branch and the shared representation, instead of directly overriding the target-tracking head. In this sense, the auxiliary term should be understood as a training-time regularizer for local corrective behavior rather than as a separate control objective that replaces return maximization.

4.5. Dual-Critic Safe Constrained Optimization

According to Equation (16) in Section 3, the objective is to maximize the expected persistent tracking return over sampled obstacle layouts while keeping the expected safety cost within a given threshold. To this end, TASRP introduces a dual-critic architecture under the CTDE training framework and jointly learns a task-value function and a safety-cost value function. For the joint state s_t, the task-value critic and the safety-cost critic are denoted as $V_r(s_t)$ and $V_c(s_t)$, respectively. Here, V_r estimates future cumulative task return, while V_c estimates future cumulative safety cost. After a joint trajectory is sampled, the reward advantage and the cost advantage are computed from the reward sequence $\{r_t\}$ and the cost sequence $\{c_t\}$, respectively. Let the joint policy be π_θ. Then, the generalized advantage estimators can be written as

$$\widehat{A}_t^r = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^r, \quad \delta_t^r = r_t + \gamma V_r(s_{t+1}) - V_r(s_t) \tag{30}$$

$$\widehat{A}_t^c = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^c, \quad \delta_t^c = c_t + \gamma V_c(s_{t+1}) - V_c(s_t) \tag{31}$$

where λ is the GAE coefficient. Correspondingly, the reward return and cost return are denoted as $\widehat{R}_t^r$ and $\widehat{R}_t^c$, respectively.

During policy optimization, this study still adopts the clipped policy-update scheme based on the importance-sampling ratio. Let the old policy be $\pi_{\theta_{\text{old}}}$. Then,

$$\rho_t(\theta) = \frac{\pi_\theta(a_t \mid o_t)}{\pi_{\theta_{\text{old}}}(a_t \mid o_t)} \tag{32}$$

Based on this ratio, the reward-driven PPO minimization loss is

$$\mathcal{L}_{\text{ppo}}^r = -\mathbb{E}\Big[\min\Big(\rho_t(\theta)\widehat{A}_t^r,\ \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\widehat{A}_t^r\Big)\Big] \tag{33}$$

where ϵ is the clipping threshold. Correspondingly, the safety-cost constraint term is defined as

$$\mathcal{L}_{\text{ppo}}^c = \mathbb{E}\Big[\max\Big(\rho_t(\theta)\widehat{A}_t^c,\ \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\widehat{A}_t^c\Big)\Big] \tag{34}$$

Meanwhile, the task critic and the safety critic are updated by minimizing the mean-squared-error losses:

$$\mathcal{L}_{V_r} = \eta_r\, \mathbb{E}\left[\left(V_r(s_t) - \widehat{R}_t^r\right)^2\right], \quad \mathcal{L}_{V_c} = \eta_c\, \mathbb{E}\left[\left(V_c(s_t) - \widehat{R}_t^c\right)^2\right] \tag{35}$$

It should be clarified that the safety budget C and the multiplier update rate β do not directly appear in the regression objectives of the two critics in Equation (35). The task critic and the safety critic are trained by supervised value fitting to the reward return $\widehat{R}_t^r$ and the cost return $\widehat{R}_t^c$, respectively. Therefore, their optimization targets are determined by sampled return estimation, whereas C and β mainly act on the adaptive update of the Lagrange multiplier in Equation (37). In this sense, these two hyperparameters influence the policy-side reward-safety trade-off more directly than the functional form of the dual-critic losses themselves.

Where η_r and η_c are the weighting coefficients for the task-critic loss and the safety-critic loss, respectively. To explicitly control the safety budget, a Lagrange multiplier $\lambda_c \geq 0$ is introduced, and the joint policy-optimization objective is constructed as

$$\mathcal{L}_\pi^{\min} = \mathcal{L}_{\text{ppo}}^r + \lambda_c \mathcal{L}_{\text{ppo}}^c + \mathcal{L}_{\text{aux}} - \mathcal{L}_{\text{ent}} \tag{36}$$

where $\mathcal{L}_{\text{aux}}$ is the teacher-assisted supervision term in Equation (29), and $\mathcal{L}_{\text{ent}} = \beta_H \mathbb{E}[\mathcal{H}(\pi_\theta(\cdot \mid o_t))]$ is the entropy bonus, which is subtracted from the total loss to encourage sufficient exploration. Thus, Equation (36) combines reward improvement, safety-cost suppression, and teacher-assisted supervision in a unified policy-update objective.

Accordingly, the total objective in Equation (36) combines two complementary learning signals. The PPO reward term and the safety-cost term still determine the main policy-improvement direction under the return-safety trade-off, whereas $\mathcal{L}_{\text{aux}}$ only regularizes the direction of the raw safety residual. Moreover, the possible gradient competition is mitigated by the structural design of TASRP: the residual correction is restricted to a low-dimensional action subspace with bounded magnitude, and its effective contribution is further modulated by the risk-adaptive gate in Equations (24) and (25). In addition, the teacher residual is constructed to preserve target-consistent bypassing rather than to encourage purely repulsive obstacle avoidance. Therefore, the auxiliary term may introduce localized gradient interaction, but it is constrained by design and is intended to improve pursuit-consistent safety correction rather than dominate the reward-driven policy update.

After each update round, the Lagrange multiplier is adaptively adjusted according to the average safety cost $\bar{c}$ of the current batch and the given safety budget C:

$$\lambda_c \leftarrow \text{Proj}_{[0,\lambda_{\max}]}(\lambda_c + \beta(\bar{c} - C)) \tag{37}$$

where β is the multiplier learning rate, and $\text{Proj}_{[0,\lambda_{\max}]}$ denotes projection onto a non-negative bounded interval. Equation (37) adjusts the strength of the safety constraint according to the current cost level, thereby avoiding both under-constrained and overly conservative updates.

Accordingly, changing C or β mainly affects how quickly λ_c reacts to the observed batch cost and how strongly the policy update is biased toward safety-cost suppression. A smaller safety budget or a larger multiplier learning rate generally leads to a more aggressive increase in λ_c, which may make the policy update more conservative; conversely, a larger safety budget or a smaller multiplier learning rate relaxes this correction. However, this effect is indirect with respect to the dual critics: it primarily alters the actor-side optimization balance and the resulting data distribution, rather than directly changing the critic update equations. For this reason, in the present study, we use a fixed safety budget and multiplier learning rate and focus the empirical evaluation on the overall effectiveness of the TASRP framework under a unified training protocol.

In addition to the explicit safety cost, this study further introduces a blocking-improvement reward to enhance the effective advancement capability of the policy in blocking-obstacle environments. According to the target-oriented blocking score $q_{i,k,t}$ defined in Section 4.4, let the local blocking intensity of the i-th UAV at time t be $B_{i,t} = \max_k q_{i,k,t}$, and let the relative target distance be $D_{i,t} = \left\| p_{i,t}^{\text{tar}} \right\|_2$. Then, the blocking-improvement reward is defined as

$$r_{i,t}^{\text{blk}} = \lambda_{\text{blk}}[B_{i,t} - B_{i,t+1}]_+ + \lambda_{\text{prog}}[D_{i,t} - D_{i,t+1}]_+ \cdot \mathbf{1}(B_{i,t} > \tau_{\text{blk}}) \tag{38}$$

where $[x]_+ = \max(x, 0)$ and τ_{blk} is the blocking-activation threshold. Equation (38) encourages the policy to effectively reduce blockage through obstacle bypassing while maintaining net progress toward the target when obvious blockage exists.

Finally, the environmental reward term becomes

$$\tilde{r}_{i,t} = r_{i,t} + r_{i,t}^{\text{blk}} \tag{39}$$

Therefore, the training process of TASRP does not rely solely on collision penalties to learn passive obstacle avoidance. Instead, under the joint action of constrained optimization

and blocking-improvement incentives, the policy learns cooperative behavior with the property of "safely bypassing obstacles and then continuing pursuit."

4.6. Training Procedure and Algorithm Implementation

By integrating the above modules, the overall training procedure of TASRP is as follows. First, the environment generates an interactive scenario composed of dual UAVs, a dynamic target, and multiple obstacles arranged under sampled layouts according to the task model defined in Section 3. At each time step, the i-th UAV receives the local observation $o_{i,t}$. After passing through the shared encoder, the main tracking branch outputs $\mu_{i,t}^{\mathrm{trk}}$, the safety branch outputs $\delta_{i,t}^{\mathrm{raw}}$, and the gating branch produces the base fusion coefficient. Meanwhile, the algorithm computes the explicit risk $r_{i,t}$ according to the relative obstacle positions, obstacle sizes, and boundary margin in the local observation, then obtains the fusion weight $\alpha_{i,t}$ via Equation (24), and finally constructs the fused action distribution according to Equation (25), from which continuous actions are sampled and executed in the environment.

Subsequently, the environment returns the reward, next state, termination flag, and safety-cost information. During experience recording, the blocking-improvement reward is further computed according to Equation (38) and added to the original task reward, while the corresponding safety cost is also extracted for subsequent estimation of reward and cost advantages. During training, the task-value critic V_r and the safety-cost critic V_c use the global state for value estimation, after which the reward and cost advantages are computed via Equations (30) and (31), respectively. The policy network is then updated according to Equation (36), where the main tracking branch is mainly driven by the persistent tracking return, while the safety-residual branch is jointly influenced by the reward term, the constraint term, and teacher-assisted supervision. Finally, the Lagrange multiplier λ_c is updated according to Equation (37) to dynamically adjust the sensitivity of the policy to safety cost.

It should be noted that TASRP is trained from online interactions rather than from a fixed offline dataset. Therefore, in the present reinforcement-learning setting, the effective training data size is characterized by the number of sampled transitions. Let P denote the number of parallel environments, H the rollout length, and T the total training horizon in environment steps. Then, each rollout-update cycle collects $P \times H$ environment transitions, while the overall training process uses approximately $P \times T$ environment transitions in total. Under the training configuration used in this paper, we set $P = 1024$, $H = 24$, and $T = 400{,}000$, corresponding to 24,576 environment transitions per rollout and approximately 4.096×10^8 environment transitions in total. For the two-UAV setting, this is equivalent to approximately 8.192×10^8 agent-level transitions. On the experimental platform equipped with an NVIDIA A30 GPU, training one TASRP model for 400,000 environment steps requires approximately 32.76 h (about 1.365 days) under the setting used in this paper.

In this way, the TASRP training process can be summarized as a closed-loop learning framework jointly formed by local-observation modeling, action generation, risk modulation, and constrained optimization. Specifically, the policy first encodes local observations into features and then generates a structured action representation. It subsequently performs adaptive action fusion by combining environmental risk information, thereby enabling online correction of potentially unsafe behavior. At the same time, auxiliary supervision signals are introduced to strengthen the target-oriented capability of the policy, while the dual-critic architecture jointly evaluates and constrains task return and safety cost.

These mechanisms allow the learned policy to balance target-advancement efficiency and local-safety regulation in continuous control spaces and, under the synergy of explicit

constraints and auxiliary learning, improve its stability and interpretability in complex blocking environments.

By integrating the aforementioned structured policy representation, explicit risk gating, teacher-assisted supervision, and safe constrained updating, the training of TASRP is not a single policy-optimization process, but rather a joint learning procedure consisting of environment interaction, action fusion, auxiliary-target construction, and constrained parameter updating. For the clarity of the overall implementation flow, the training steps of TASRP under the CTDE framework are summarized in Algorithm 1.

Algorithm 1 Training procedure of TASRP under CTDE.

Require: Parallel environments $\{E_j\}_{j=1}^{P}$, agent set $\mathcal{U}$, policy parameters θ, task critic parameters ψ_r, safety critic parameters ψ_c, Lagrange multiplier λ_c, rollout length H, total training horizon T, update epochs K, safety budget C

Ensure: Trained decentralized execution policy π_θ

Initialize shared encoder, target-tracking branch, safety-residual branch, and gating branch
Initialize task critic V_r and safety critic V_c
Initialize replay buffer D and Lagrange multiplier λ_c
Initialize accumulated environment steps $t_{\text{tot}} \leftarrow 0$
while $t_{\text{tot}} < T$ **do**
    Clear buffer D
    for $t = 1, 2, \ldots, H$ **do**
        Collect local observations $o_{i,t}$ and global state s_t from all parallel environments
        For each agent i, compute feature representation $h_{i,t}$
        Compute tracking action mean $\mu_{i,t}^{\text{trk}}$ via the tracking branch
        Compute raw safety residual $\delta_{i,t}^{\text{raw}}$ via the safety branch
        Compute base gating coefficient $\bar{\alpha}_{i,t}$ via the gating branch
        Compute explicit risk $r_{i,t}$ based on obstacle surface distance and boundary margin
        Compute fusion weight $\alpha_{i,t}$
        Compute effective safety residual $\widetilde{\delta}_{i,t} = \alpha_{i,t}\delta_{i,t}^{\text{raw}}$
        Compute fused action mean $\mu_{i,t} = \mu_{i,t}^{\text{trk}} + M(\widetilde{\delta}_{i,t})$
        Sample continuous action $a_{i,t}$ from the Gaussian policy
        Execute joint action a_t, and obtain reward, next state, done flag, and safety cost
        Construct teacher residual $\delta_{i,t}^{\text{tea}}$ based on target direction and local obstacles
        Compute local blocking metric $B_{i,t}$ and target distance $D_{i,t}$
        Compute blocking-aware reward $r_{i,t}^{\text{blk}}$ via Equation (38)
        Compute shaped reward $\widetilde{r}_{i,t} = r_{i,t} + r_{i,t}^{\text{blk}}$
        Store $\left(o_{i,t}, s_t, a_{i,t}, \widetilde{r}_{i,t}, c_{i,t}, o_{i,t+1}, s_{t+1}, done_t, \delta_{i,t}^{\text{tea}}\right)$ into buffer D
    end for
    Compute reward returns, cost returns, reward advantages, and cost advantages from D
    for $k = 1, 2, \ldots, K$ **do**
        Sample a mini-batch from D
        Update task critic V_r
        Update safety critic V_c
        Compute reward PPO loss, cost constraint loss, teacher auxiliary loss, and entropy regularization
        Update policy parameters θ via Equation (36)
    end for
    Compute average safety cost $\bar{c}$ over the current batch
    Update Lagrange multiplier λ_c via Equation (37)
    Update accumulated environment steps $t_{\text{tot}} \leftarrow t_{\text{tot}} + H$
end while
return trained decentralized execution policy π_θ

5. Experimental Results and Discussion

After completing the task formulation and algorithm design, this study conducts comprehensive comparative experiments to evaluate the performance of the proposed TASRP framework in multi-UAV cooperative persistent tracking tasks under complex obstacle environments. The evaluation is carried out with five aspects. First, the clean-observation task success rates of TASRP are compared with those of representative multi-agent reinforcement learning methods. Second, a dedicated noisy-observation robustness evaluation is conducted to examine sensitivity to observation uncertainty. Third, safety-related metrics, including obstacle collisions, boundary violations, and inter-UAV collisions, are analyzed. Fourth, the training stability and convergence characteristics are evaluated. Fifth, ablation studies are conducted to examine the effectiveness of key components, including the risk-aware gating mechanism, target-oriented teacher, dual-branch residual policy, and safety-constrained optimization.

The experimental results demonstrate that TASRP consistently achieves a more favorable balance between task success and safety-related performance than the representative baselines across the three validation scenarios under clean observations. A high-level summary of the main result ranges is provided in Table 2. In addition, a separate noisy-observation robustness evaluation is further conducted to assess the sensitivity of the learned policies to moderate observation uncertainty.

Table 2. High-level performance summary of TASRP and baseline methods over the three validation scenarios under clean observations.

Method	Success Rate ↑	Obstacle Crash ↓	Boundary Crash ↓	Inter-UAV Collision Count ↓
MAPPO	0.68–0.72	0.51–0.54	0.02–0.04	0–6
HAPPO	0.66–0.68	0.63–0.68	0.02–0.03	0–0
HATRPO	0.70–0.74	0.35–0.39	0.02–0.04	1–4
MACPO	0.55–0.58	0.48–0.52	0.03–0.05	0–0
TASRP	**0.75–0.79**	**0.13–0.15**	**0.01–0.02**	**0–0**

Note: ↑ indicates that a higher value is better, while ↓ indicates that a lower value is better. Bold values indicate the best performance in each column.

The remainder of this section is organized as follows. Section 5.1 introduces the experimental setup. Section 5.2 presents comparative results under clean observations. Section 5.3 reports a robustness evaluation under noisy observations. Section 5.4 analyzes the trade-off between performance and safety. Section 5.5 investigates cooperative tracking behaviors in complex scenarios. Section 5.6 provides ablation studies. Section 5.7 summarizes the experimental findings.

5.1. Experimental Setup

This section presents the experimental setup from five aspects: environment construction, selection of comparison baselines, evaluation metrics, training implementation details, and noisy-observation evaluation settings. A unified comparison protocol is adopted throughout all experiments, where all methods are trained and evaluated under identical task environments, observation and action spaces, and training procedures. This ensures that the performance differences can be attributed to the underlying policy structures and optimization mechanisms, rather than discrepancies in environment settings or hyperparameter configurations. The detailed experimental settings are described as follows.

5.1.1. Simulation Scenario

The experiments are conducted on a dual-UAV cooperative persistent tracking task in complex obstacle environments. Compared with open-space scenarios, blocking obstacles

and gate-like obstacles are introduced to strengthen the coupling between target pursuit and safety-aware avoidance.

In each episode, two UAVs are required to cooperatively approach a dynamic target in a three-dimensional space while improving tracking success under flight safety constraints. The environment contains a fixed number of obstacles that significantly interfere with the direct target-approach path. Meanwhile, UAVs must satisfy boundary constraints and maintain safe inter-UAV distances. Therefore, the task requires not only target-oriented control capability but also effective obstacle-avoidance adjustments in locally high-risk regions.

To ensure a fair comparison, all methods are trained and evaluated under identical environment configurations, observation dimensions, action spaces, and training steps. The detailed environmental parameters are summarized in Table 3.

Table 3. Environmental parameters.

Parameter	Value
Number of UAVs	2
Scene description	Enclosed three-dimensional environment
Environment range	Cylindrical space with radius 3 m and height 3 m
Episode duration	20.0 s
Number of obstacles	4–5
Obstacle size range	Cylinders with radius in $[0.22, 0.36]$ m and height 2.8 m
Blocking-layout trigger probability	0.85
Gate-layout trigger probability	0.55
Teammate safe distance	0.20 m
Capture radius	0.25 m
Action definition	Continuous four-dimensional control, including total thrust, roll, pitch, and yaw; $a_{i,t} \in [-1, 1]^4$
Initial UAV positions	$(x, y) \in [-1.5, -0.5], \quad z \in [0.5, 2.0]$
UAV task-space altitude bounds	$z_{\min} = 0.3,\ z_{\max} = 3.0$
Initial target position	$(x, y) \in [0.8, 1.6], \quad z \in [1.5, 2.0]$
Target speed	$[0.97, 1.36]$

Instead of using a large-scale open environment, we adopt a confined three-dimensional space, mainly to strengthen the coupling among target advancement, obstacle blockage, and boundary maintenance, so that the policy must accomplish persistent tracking under relatively high local risk. This setting better captures the core challenges of safe multi-UAV tracking in complex obstacle environments and is, therefore, well aligned with the objective of this study.

It should be noted that, during training, the trigger probabilities of the target-blocking layout and the gate-like layout are not normalized probabilities over mutually exclusive scene categories. Rather, they are independent triggering parameters for two obstacle-layout rules during scene generation.

5.1.2. Baselines and Evaluation Metrics

To systematically evaluate the effectiveness of the proposed method, this study considers MAPPO, HAPPO, HATRPO, and MACPO as baseline methods, which represent typical multi-agent reinforcement-learning paradigms with parameter sharing, independent policy updates, hierarchical optimization, trust-region optimization, and safety-constrained optimization, respectively. All methods are trained and evaluated under the same task environment, observation and action spaces, and unified training protocol to ensure a

fair comparison. In addition to the full clean-observation benchmark, a focused noisy-observation robustness comparison is further conducted for TASRP, MAPPO, and HAPPO, which allows a direct comparison of robustness among representative continuous-control policy-gradient methods under the same perturbation protocol.

From the perspectives of task performance and flight safety, four core metrics are adopted: the task success rate, the obstacle crash rate, the boundary crash rate, and the inter-UAV collision count. These metrics jointly characterize tracking capability and safety performance in complex obstacle environments. For clarity, the expected value range and desirable direction of each metric are also specified: success-rate and crash-rate metrics are defined in $[0, 1]$, where a higher success rate and lower crash rates indicate better performance, while the inter-UAV collision count is a nonnegative quantity for which smaller values are preferable and zero is ideal. The baselines and evaluation metrics are summarized in Table 4.

Table 4. Baselines and evaluation metrics.

Category	Item	Description	Typical Range/ Desirable Value
Baseline	MAPPO	Parameter sharing under CTDE	N/A
Baseline	HAPPO	Hierarchical policy optimization	N/A
Baseline	HATRPO	Trust-region constrained update	N/A
Baseline	MACPO	Safety-constrained policy optimization baseline	N/A
Metric	Success Rate	Task completion performance	$[0, 1]$; higher is better; 1 is ideal
Metric	Obstacle Crash	Obstacle-avoidance capability	$[0, 1]$; lower is better; 0 is ideal
Metric	Boundary Crash	Boundary-constraint handling	$[0, 1]$; lower is better; 0 is ideal
Metric	Inter-UAV Collision Count	Multi-agent safety	nonnegative; lower is better; 0 is ideal

5.1.3. Implementation Details

All experiments are conducted under a unified software and hardware environment. The training framework is implemented based on IsaacLab (version 2.3.2) and skrl (version 1.4.3), and the underlying networks are built with PyTorch (version 2.7.0+cu128).To reduce the influence of randomness, all methods are trained with multiple random seeds, and the mean results together with their statistical fluctuations are reported. The key hyperparameters include the learning rate, rollout length, number of mini-batches, number of update epochs, discount factor, GAE coefficient, and number of parallel environments. The training hyperparameters reported in the source document are listed in Table 5.

In all reported experiments, the safety budget and multiplier update rate are fixed to a unified setting to ensure fair comparison across methods and ablation variants. A systematic sensitivity study on different safety budgets and multiplier update rates, especially regarding their influence on the reward-safety trade-off and the long-horizon training dynamics, is an important direction for future work.

Table 5. Hyperparameter settings for training.

Parameter	Value
Agent set ($\mathcal{U}$)	{`drone_0`, `drone_1`}
Total training steps; number of parallel environments	400,000; 1024
Random seed	42
Rollout length (H); number of mini-batches	24; 8
Update epochs (K)	4
Learning rate; discount factor; GAE coefficient	2×10^{-4}; 0.99; 0.95
Policy network (θ)	Dual-head Gaussian actor, hidden layers [256, 256], ELU activations
Task critic (ψ_r)	MLP critic, hidden layers [256, 256], ELU activations, scalar output
Safety critic (ψ_c)	MLP critic, hidden layers [256, 256], ELU activations, scalar output
Action dimension; residual-controlled dimension	4; 2
Maximum obstacle-avoidance residual	0.22
Obstacle influence distance; boundary influence margin; risk exponent	1.4; 0.35; 1.4
Auxiliary-loss weight; teacher influence distance	0.30; 1.60
Blocking-reward gain; progress-reward gain	0.35; 0.20
Safety budget (C)	0.25
Lagrange multiplier (λ_c)	initialized to 0.0, learning rate 0.05, clipped to $[0, 10.0]$

5.1.4. Noisy-Observation Evaluation Settings

For the noisy-observation evaluation, zero-mean Gaussian perturbations are injected into selected observation channels during both training and validation. Standard deviations of 0.05 m, 0.03 m/s, and 0.02 rad/s are applied to relative-position channels, linear-/relative-velocity channels, and angular-velocity channels, respectively. A smaller perturbation with standard deviation 0.03 is applied to normalized boundary-related distance features. Obstacle activation flags and obstacle size parameters are kept unchanged in order to preserve physically consistent scene descriptions. Under this protocol, TASRP, MAPPO, and HAPPO are retrained and re-evaluated using the same statistical procedure as in the clean-observation benchmark.

The adopted perturbation model is designed to approximate uncertainty at the state-estimation interface rather than to emulate a complete raw-sensor stack. Relative-position perturbations mimic target-localization, teammate-localization, and obstacle-localization errors, velocity perturbations reflect state-estimation uncertainty in motion states, and boundary-feature perturbations emulate imperfect awareness of safety margins. The selected magnitudes are moderate with respect to the 3 m-scale workspace, yet non-negligible relative to the 0.20 m safe-separation threshold and the 0.25 m capture radius, thereby providing a meaningful robustness test without making the task physically unrealistic. Therefore, the noisy-observation evaluation should be interpreted as a robustness test at the observation and state-estimation interface, rather than as an end-to-end validation of a complete onboard sensing pipeline.

5.2. Comparative Performance Evaluation

To comprehensively evaluate the performance of the algorithms in different complex obstacle environments, we consider three representative scenarios: random obstacles, target-blocking obstacles, and gate-like obstacles. For each scenario type, 512 task instances are generated per evaluation round, and 10 independent validation experiments are conducted. Based on these 10 runs, we report the mean and standard deviation of task success rate and safety-related metrics. In addition, 95% confidence intervals are reported for the main comparative metrics to better reflect the statistical reliability of the observed differences among methods. The clean-observation comparative results across different scenarios are summarized in Table 6, the corresponding confidence intervals are summarized in Table 7, and the wall-clock training-time comparison is summarized in Table 8. The additional noisy-observation robustness results are summarized in Table 9.

It should be noted that the wall-clock time per training step is not constant during learning. In the early stage of training, UAVs frequently crash or violate constraints, which leads to more frequent environment resets and relatively low effective throughput. After the policies gradually learn basic obstacle-avoidance and target-tracking behaviors, the interaction process becomes more stable and the training efficiency improves accordingly. Therefore, the per-step computational cost is better understood as a stage-dependent quantity rather than a fixed scalar. On the adopted experimental platform, the observed training throughput typically varies within a range of approximately 3–16 iterations per second, depending on the training stage, scenario complexity, and random initialization. For this reason, besides the comparative task and safety metrics, we further report the overall wall-clock training time of each method on the same hardware platform.

Table 6. Main comparative results over three validation scenarios.

Scenario	Metric	TASRP (Ours)	MAPPO *	HAPPO *	HATRPO *	MACPO *
1	Task success rate	0.79 ± 0.02	0.72 ± 0.02	0.68 ± 0.01	0.74 ± 0.01	0.58 ± 0.14
	Obstacle crash rate	0.13 ± 0.02	0.51 ± 0.02	0.63 ± 0.02	0.35 ± 0.02	0.48 ± 0.15
	Boundary crash rate	0.01 ± 0.01	0.02 ± 0.01	0.02 ± 0.00	0.02 ± 0.00	0.03 ± 0.06
	Inter-UAV collision count	0	3	0	1	0
2	Task success rate	0.78 ± 0.02	0.70 ± 0.02	0.68 ± 0.01	0.72 ± 0.01	0.56 ± 0.14
	Obstacle crash rate	0.15 ± 0.02	0.54 ± 0.02	0.68 ± 0.02	0.35 ± 0.02	0.49 ± 0.15
	Boundary crash rate	0.02 ± 0.01	0.04 ± 0.01	0.02 ± 0.00	0.03 ± 0.01	0.03 ± 0.06
	Inter-UAV collision count	0	0	0	4	0
3	Task success rate	0.75 ± 0.03	0.68 ± 0.02	0.66 ± 0.01	0.70 ± 0.02	0.55 ± 0.14
	Obstacle crash rate	0.14 ± 0.01	0.54 ± 0.02	0.65 ± 0.02	0.39 ± 0.02	0.52 ± 0.15
	Boundary crash rate	0.02 ± 0.01	0.04 ± 0.01	0.03 ± 0.01	0.04 ± 0.01	0.05 ± 0.06
	Inter-UAV collision count	0	6	0	4	0

* The algorithms marked with * were reproduced according to the logic described in their original papers. The results are reported as mean $\pm$ standard deviation over 10 independent validation runs. The corresponding 95% confidence intervals for the main comparative metrics are reported in Table 7.

As shown in Table 8, the proposed TASRP framework introduces additional computational overhead compared with standard MAPPO due to its structured policy design, risk-adaptive fusion mechanism, teacher-assisted supervision, and dual-critic constrained optimization. However, the additional training cost remains moderate and is accompanied by consistently better task-success and safety performance across the three validation scenarios. Compared with HATRPO, which incurs substantially higher wall-clock training time, TASRP achieves a more favorable balance between computational efficiency and overall control effectiveness.

Table 7. 95% confidence intervals of the main comparative metrics over three validation scenarios.

Scenario	Metric	TASRP (Ours)	MAPPO *	HAPPO *	HATRPO *	MACPO *
1	Task success rate	[0.776, 0.804]	[0.706, 0.734]	[0.673, 0.687]	[0.733, 0.747]	[0.480, 0.680]
	Obstacle crash rate	[0.116, 0.144]	[0.496, 0.524]	[0.616, 0.644]	[0.336, 0.364]	[0.373, 0.587]
2	Task success rate	[0.766, 0.794]	[0.686, 0.714]	[0.673, 0.687]	[0.713, 0.727]	[0.460, 0.660]
	Obstacle crash rate	[0.136, 0.164]	[0.526, 0.554]	[0.666, 0.694]	[0.336, 0.364]	[0.383, 0.597]
3	Task success rate	[0.729, 0.771]	[0.666, 0.694]	[0.653, 0.667]	[0.686, 0.714]	[0.450, 0.650]
	Obstacle crash rate	[0.133, 0.147]	[0.526, 0.554]	[0.636, 0.664]	[0.376, 0.404]	[0.413, 0.627]

The algorithms marked with * were reproduced according to the methodological descriptions provided in their original papers. The confidence intervals are estimated from the reported mean and standard deviation over 10 independent validation runs using the 95% t-interval approximation.

Table 8. Overall training time comparison of different methods.

Method	Overall Training Time (h)	Overall Training Time (Days)
TASRP (ours)	32.76	1.365
MAPPO *	30.55	1.273
HAPPO *	33.65	1.402
HATRPO *	56.11	2.338
MACPO *	20.89	0.870

* Reproduced according to the methodological descriptions provided in the original papers. All methods were trained on the same experimental platform equipped with an NVIDIA A30 GPU (NVIDIA Corporation, Santa Clara, CA, USA).

Table 9. Robustness comparison under clean and noisy observations.

Method	Observation Setting	Success Rate ↑	Obstacle Crash ↓	Boundary Crash ↓	Inter-UAV Collision Count ↓
TASRP (ours)	Clean	0.773	0.140	0.017	0
	Noisy	**0.721**	**0.203**	**0.028**	**0**
MAPPO *	Clean	0.700	0.530	0.033	9
	Noisy	0.612	0.612	0.056	5
HAPPO *	Clean	0.673	0.653	0.023	0
	Noisy	0.581	0.735	0.041	1

* Reproduced according to the methodological descriptions provided in the original papers. Rate-based metrics are averaged over the three validation scenarios. Inter-UAV collision count is reported as the accumulated number of collision events over the same three scenarios. ↑ indicates that a higher value is better, while ↓ indicates that a lower value is better.

We further analyze the task success rate of the proposed TASRP framework during training under clean observations, with emphasis on its comparative performance and learning efficiency against the baseline algorithms, as shown in Figure 3.

As shown in Figure 3, TASRP exhibits the fastest and most stable improvement among the compared methods. Its success rate rises to about 0.65–0.70 within roughly 2×10^4–3×10^4 steps and then remains mostly within 0.69–0.78 with only small oscillations. This trend is consistent with Equations (17)–(25): the main branch provides the dominant pursuit action, whereas the safety branch only adds a bounded residual through Equation (19), modulated by the explicit risk gate in Equations (24) and (25). Physically, this is important because obstacle bypassing mainly requires roll-pitch-induced lateral redirection of the thrust vector, while thrust and yaw should remain comparatively stable. By restricting correction to the local maneuver subspace, TASRP reduces interference between target advancement and safety regulation, which explains both its rapid early improvement and its stable late-stage plateau.

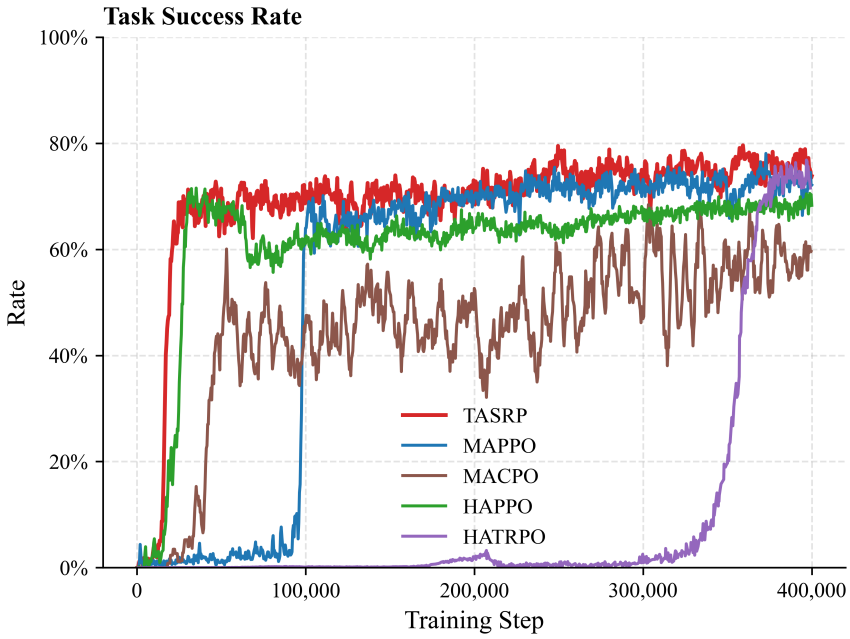


Figure 3. Performance comparison of different algorithms in terms of task success rate over the reported training horizon.

The baselines learn the same coupling much more slowly. HAPPO rises early but drops around 6×10^4 steps, suggesting that the policy enters a transition stage in which obstacle-aware maneuvering begins to influence the control behavior more strongly. In multirotor flight, near-obstacle motion requires repeated roll-pitch adjustments to redirect the thrust vector and preserve local clearance, rather than maintaining purely target-directed motion. Before this avoidance-pursuit coordination becomes sufficiently stable, part of the control effort is effectively diverted from direct target closing to short-horizon safety-preserving maneuvers, which temporarily reduces pursuit efficiency and, therefore, leads to the observed decrease in the success rate. With further training, HAPPO partially improves this coordination, but its success rate remains below its earlier local peak, indicating that the learned compromise is still suboptimal for persistent target pursuit. MACPO remains conservative but oscillatory, which is consistent with cost regularization without explicit geometric detour guidance. HATRPO is the most delayed, remaining near zero until about 1.8×10^5 steps and rising sharply only after about 3.4×10^5 steps. Its pronounced fluctuations are consistent with trust-region-based updates being conservative in step size yet sensitive to changes in the sampled multi-agent state distribution. In this task, once the policy partially reduces one type of failure, the rollout distribution shifts, and another failure mode can temporarily dominate, producing visible oscillations in the reported curve.

Two broader insights can be drawn from Figure 3. First, in cluttered cooperative pursuit, sample efficiency is governed not only by how quickly a policy learns to move toward the target but also by how early it discovers target-consistent detours around blocking obstacles. This explains why HAPPO can show a reasonable early success level yet still lose performance once obstacle interactions become dominant, and why HATRPO remains ineffective for a long period before abruptly improving only after enough feasible detour experience has been accumulated. Second, the late-stage curves suggest that simply extending training is unlikely to remove the gap between TASRP and the baselines: after their main transitions occur, MAPPO, MACPO, and HAPPO exhibit either only shallow improvements or persistent oscillations. Therefore, the advantage of TASRP is not merely faster convergence under the same learning process but the fact that it imposes a more suitable control structure for obstacle-constrained pursuit from the beginning.

Overall, in target-blocking layouts, the physically feasible solution is not pure attraction to the target or pure obstacle repulsion but tangentially biased lateral correction that preserves future target access. TASRP encodes this explicitly through the fused action structure, whereas the baselines must discover it implicitly through trial and error, which explains the slower or less stable learning process.

5.3. Robustness Evaluation Under Noisy Observations

In addition to the clean-observation benchmark, this study further evaluates TASRP, MAPPO, and HAPPO under the noisy-observation protocol described in Sections 3 and 5.1. Table 9 summarizes the average rate-based metrics over the three validation scenarios, together with the accumulated inter-UAV collision counts across the same scenarios.

As expected, observation noise degrades the performance of all methods. Nevertheless, TASRP still achieves the highest task success rate and the lowest obstacle crash rate under the noisy protocol, while maintaining stable boundary-safety performance. Compared with MAPPO and HAPPO, TASRP shows a smaller reduction in task success and a more moderate increase in crash-related metrics, indicating stronger robustness to moderate estimation and relative-localization errors.

Although this setting does not represent a full raw-sensor model, it provides additional evidence that TASRP remains effective under moderate observation uncertainty.

Table 9 shows that introducing observation noise degrades the performance of all compared methods, confirming that the noisy setting is more challenging than the clean simulator-interface benchmark. However, TASRP still maintains the best overall balance between persistent tracking and safety.

More specifically, the average task success rate of TASRP decreases from 0.773 to 0.721, corresponding to a relative reduction of about 6.7%, whereas MAPPO and HAPPO decrease by about 12.6% and 13.7%, respectively. In terms of safety, TASRP still yields the lowest average obstacle crash rate under noisy observations, increasing from 0.140 to 0.203, while MAPPO and HAPPO rise to 0.612 and 0.735, respectively. Boundary-crash performance shows a similar trend, with TASRP remaining the most stable among the compared methods. Moreover, TASRP produces no inter-UAV collisions in either the clean or noisy setting. For the baseline methods, accumulated inter-UAV collision events are still observed, indicating that observation uncertainty can also affect cooperative separation safety, although the more prominent degradation is reflected by the rate-based task and safety metrics.

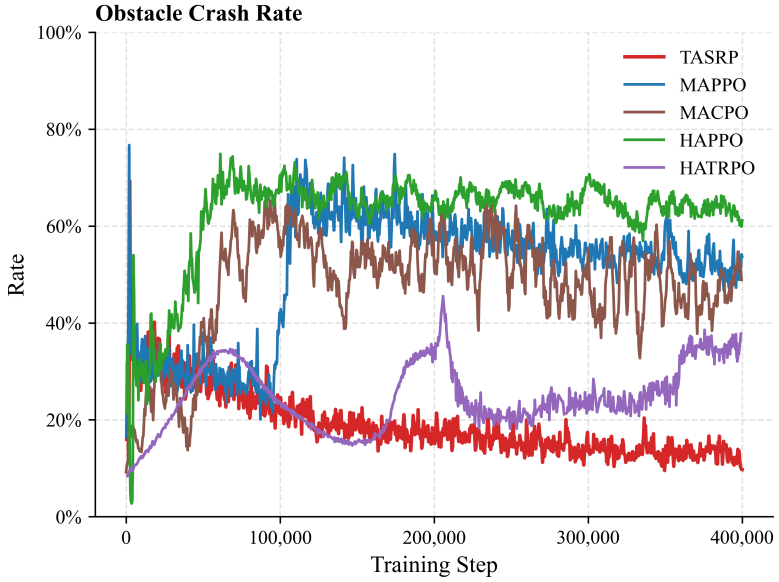
These results suggest that the structured separation between nominal target pursuit and safety-oriented residual correction improves not only clean-setting performance but also robustness under imperfect observations. Even when the observation quality deteriorates, TASRP preserves a clearer task-safety coordination mechanism than standard single-policy baselines.

5.4. Performance-Safety Trade-Off Analysis

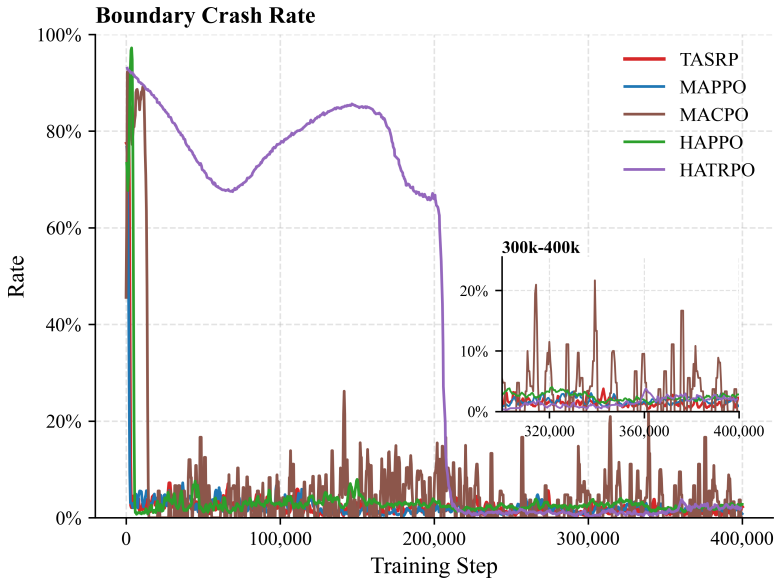
Under clean observations, we compare three safety-related indicators during training: the obstacle crash rate, the boundary crash rate, and the inter-UAV collision count. By further combining success rate with safety metrics, we analyze the performance-safety trade-off of different methods.

Figure 4 should be interpreted jointly. In Figure 4a, TASRP reduces the obstacle crash rate from about 0.30 in the initial stage to about 0.12–0.16 after 2.5×10^5 steps. This trend is consistent with Equations (23)–(25): as the obstacle surface distance $d_{i,m,t}^{\mathrm{surf}}$ decreases, the obstacle risk and fusion weight increase, so the roll-pitch residual provides stronger lateral clearance. By contrast, MAPPO and HAPPO maintain high obstacle-crash levels despite

relatively low boundary-crash rates, indicating that their dominant failure mode is obstacle collision rather than global boundary escape. For MAPPO, the curve still shows a mild downward tendency after 4×10^5 steps, but the residual decrease is already very limited, so its practical minimum under the present budget is reached only near the end of the reported horizon. TASRP reaches this low-crash plateau earlier, around 2.5×10^5 steps, which is consistent with faster formation of stable obstacle-bypassing behavior. MACPO is more mixed: the low average boundary-crash rate suggests effective cost suppression, but the broad obstacle-crash fluctuation and occasional boundary spikes indicate conservative yet unstable local maneuvering. HATRPO shows the clearest failure-mode transition: early in training it is dominated by boundary violation, whereas after the sharp drop in boundary crash near 2.1×10^5 steps its dominant failure mode becomes obstacle collision.



(**a**)



(**b**)

Figure 4. Safety performance during training: (**a**) Obstacle crash rate. (**b**) Boundary crash rate.

These differences are consistent with the governing geometry and control physics. Boundary failure is associated with the loss of the positive boundary margin, whereas obstacle failure occurs when $d_{i,m,t}^{\mathrm{surf}} \leq 0$ under overlapping altitude. For the multirotor model, local bypassing mainly depends on roll-pitch-induced redirection of the thrust vector. TASRP exploits this explicitly by restricting residual correction to the roll-pitch subspace in Equations (19)–(22), scaling it with geometric risk in Equations (23)–(25), and guiding its direction through Equations (26)–(29) and Equation (38). The remaining TASRP failures are, therefore, mainly low-frequency obstacle contacts in highly cluttered states, rather than boundary escape or inter-UAV contact. MAPPO lacks explicit risk-conditioned residual correction, so shrinking clearance does not necessarily produce proportional lateral correction; obstacle collision is, therefore, its dominant failure mode. HAPPO learns effective pursuit but not stable post-approach bypassing, so it can achieve non-negligible success while still failing mainly through obstacle collisions. MACPO suppresses obviously unsafe behavior through the constrained objective in Equation (16), but it still lacks explicit local geometric detour guidance. HATRPO lacks all three structural elements above, so after it eventually learns global containment inside Ω, it still cannot stably generate target-consistent tangential clearance around blocking obstacles; obstacle collision, therefore, remains its final dominant failure mode.

This joint interpretation also explains the apparent inconsistency around 2×10^5 steps. The noticeable change of HATRPO near that stage is mainly a redistribution of failure types rather than a uniform improvement of all safety indicators. Once boundary violations decrease, more trajectories survive longer inside the feasible region and continue into cluttered areas, where insufficient tangential clearance control leads to obstacle contacts instead. For this reason, a sudden change in the boundary-crash curve does not need to be matched by an equally strong upward rebound in the success-rate or obstacle-crash trend. The same mechanism also helps explain why HATRPO exhibits significant fluctuations overall: under trust-region updates, the policy changes conservatively from one iteration to the next, but the sampled rollout distribution can still shift noticeably when the dominant failure mode switches between boundary violation and obstacle collision.

Two additional insights can be drawn from Figure 4. First, a low boundary-crash rate alone is not sufficient to indicate strong overall safety in this task. MACPO and late-stage HATRPO can keep boundary violations relatively low, yet both still show clear weakness in obstacle avoidance, which indicates that local geometric navigation around clutter is the harder bottleneck than simply remaining inside the global workspace. Second, the figure reveals a stage-dependent failure evolution: early training is dominated by learning to remain within the admissible region, whereas later performance is increasingly determined by whether the policy can generate stable tangential clearance around obstacles without sacrificing target access. From this perspective, TASRP is stronger not only because it lowers both crash metrics but also because it suppresses the shift of dominant failure modes from boundary escape to obstacle collision more effectively than the baseline methods.

Finally, capture does not necessarily terminate the episode in this task. A method may, therefore, improve the success rate by reaching the target region more often while still incurring obstacle collisions later in the same rollout. This explains why HAPPO and, in late training, HATRPO can show non-negligible success together with high obstacle-crash levels. Overall, the joint shape of the success, obstacle-crash, and boundary-crash curves indicates that TASRP learns a more coherent mechanism for balancing persistent tracking and safety in target-blocking and gate-like environments.

As shown in Figure 5, TASRP yields zero average inter-UAV collisions across the three validation scenarios, which is the same as MACPO and HAPPO. Physically, this indicates that the learned policies are able to maintain sufficient relative separation while

coordinating target-oriented motion in the bounded three-dimensional space. In the present setting, such behavior is supported by teammate-related information in the local observation, explicit inter-UAV safe-separation constraints during training, and the additional geometric freedom offered by three-dimensional maneuvering. For TASRP, the result is further consistent with its explicit safety-correction mechanism; for MACPO, it is consistent with the conservative tendency induced by safety-constrained optimization; and for HAPPO, it suggests that inter-UAV collision is not the dominant failure mode of the learned cooperative policy in this task. Together with Figures 3 and 4, this result indicates that TASRP achieves strong overall safety in obstacle avoidance, boundary maintenance, and inter-agent safety interaction.

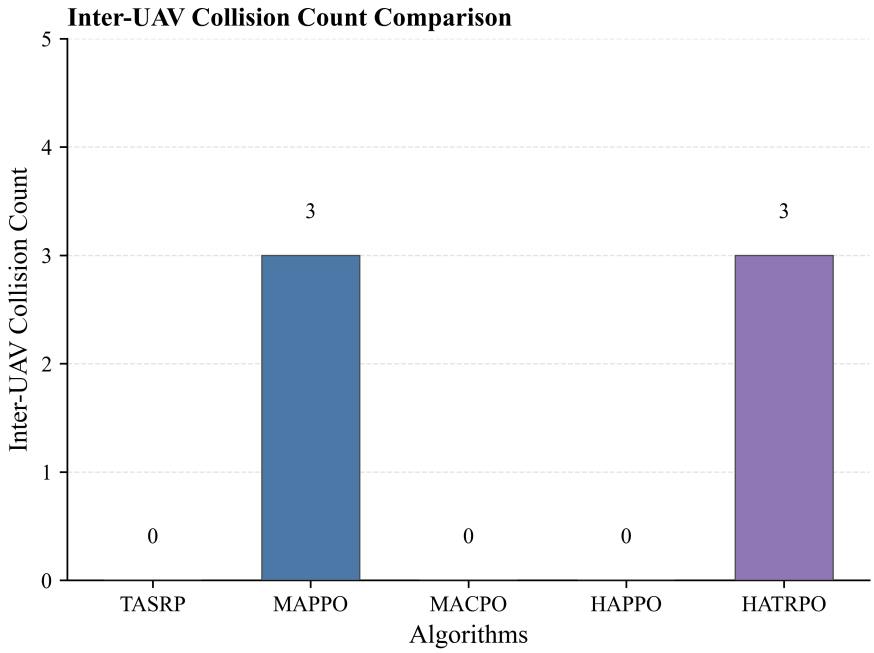
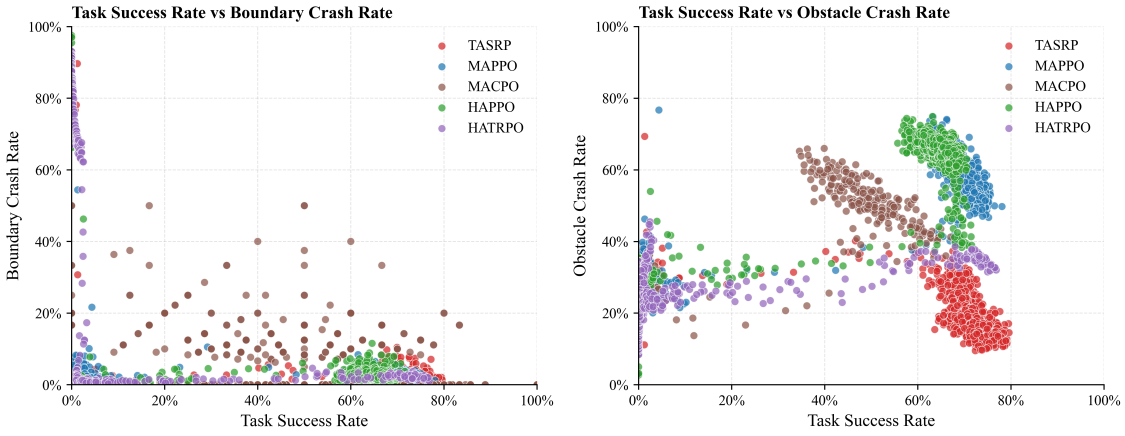


Figure 5. Inter-UAV collision count verification.

As shown in Figure 6a, TASRP is mainly distributed in the region with high task success rate, about 65–80%, and low obstacle crash rate, about 10–30%. Its overall distribution is concentrated and exhibits relatively low dispersion. By contrast, although the other methods can reach medium-to-high success rates in some ranges, they are generally accompanied by higher obstacle crash rates and more scattered distributions, implying inferior stability. This result shows that TASRP improves task success while effectively reducing obstacle-collision risk.

As shown in Figure 6b, TASRP exhibits a lower and more stable obstacle crash rate within the high-success region, roughly 60–80%, and its samples cluster in the low-crash region. This indicates that the method provides stronger obstacle-safety control while maintaining task completion capability. MAPPO and HAPPO also appear in the medium-to-high success region, but their obstacle crash rates fluctuate more noticeably. In particular, the relatively high task-success and obstacle-crash rates of HAPPO are not contradictory under the present evaluation protocol. In this work, task success reflects whether the UAVs can effectively reach and persistently track the dynamic target, whereas obstacle crash rate reflects safety performance over the full motion process of the episode. Since a successful capture/tracking event does not necessarily terminate the episode, a policy may still achieve a relatively high success rate while later incurring obstacle collisions during subsequent motion. Moreover, this observation is also consistent with Figure 5, which reports inter-UAV collision counts rather than obstacle collisions. Therefore, the

zero inter-UAV collision result of HAPPO indicates that direct teammate collisions are rare, but it does not imply low obstacle-collision risk. MACPO is more dispersed over the full success-rate range, especially with relatively high crash rates in the medium-success regime, suggesting limited effectiveness in obstacle-safety control. HATRPO clusters in the low-success region together with high obstacle crash rates, indicating that it struggles to balance task completion and safety in complex constrained environments.



(**a**) (**b**)

Figure 6. Safety-performance trade-off analysis: (**a**) Task success rate vs. boundary crash rate. (**b**) Task success rate vs. obstacle crash rate.

Considering Figures 3–6 together with Table 6 reveals that TASRP not only achieves a high task success rate but also maintains low levels in obstacle crashes, boundary violations, and inter-UAV collisions. More importantly, the practical benefit of TASRP over conventional MARL baselines is not limited to better numerical results on a single metric but lies in the fact that it provides a more reliable operating pattern for cooperative UAV pursuit in complex environments.

First, from the perspective of robustness, TASRP remains concentrated in the high-success and low-crash-rate region across random-obstacle, target-blocking-obstacle, and gate-like-obstacle scenarios, while its training curves also exhibit lower fluctuation in obstacle and boundary crash rates. This indicates that the learned policy is less sensitive to changes in obstacle layout and can maintain a more stable balance between target advancement and safety regulation under different scene structures. For cooperative UAV chase tasks, such robustness is practically important because real deployments rarely encounter a single fixed obstacle configuration.

Second, from the perspective of safety interpretability, TASRP offers a clearer control logic than standard single-policy MARL baselines. In the proposed framework, target-oriented nominal pursuit and safety-oriented residual correction are explicitly separated, and the effective intervention of the safety branch is further modulated by the risk-adaptive gate. As a result, when the UAV deviates from a direct pursuit path near obstacles or boundaries, such behavior can be interpreted as a structured safety correction rather than an opaque action change from an undifferentiated policy. This property is beneficial for analyzing policy behavior, diagnosing failure cases, and improving the transparency of safety-related decisions in practical UAV systems.

Third, from the perspective of generalization, the consistent advantage of TASRP across three representative three-dimensional scenarios suggests that its performance does not depend on one specific geometric setting. Because the method combines explicit safety correction with target-consistent detour learning, it is better suited to environments in

which obstacle relationships become denser and local feasible paths change frequently. Therefore, although broader validation in even more crowded scenes is still needed, the present results already indicate that TASRP has better potential than conventional MARL baselines for extension to crowded three-dimensional pursuit environments.

5.5. Trajectory and Behavior Visualization in Complex Scenarios

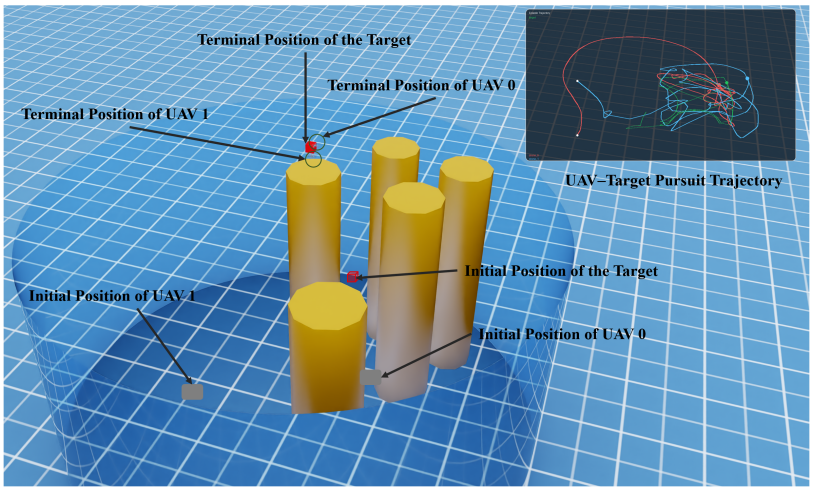
To further verify the effectiveness of TASRP from the behavioral perspective, this subsection visualizes the decision characteristics of dual-UAV cooperative persistent tracking by combining scene-layout illustrations and three-dimensional trajectory plots. Figure 7 presents the spatial configurations of three representative complex scenes, namely the random-obstacle scene, the target-blocking-obstacle scene, and the gate-like-obstacle scene. Figure 8 further shows the corresponding three-dimensional trajectories of UAV 0, UAV 1, and the target. By jointly examining the initial positions, terminal positions, obstacle distributions, and trajectory-evolution process, it becomes possible to more intuitively analyze the policy's ability to balance target pursuit, obstacle avoidance, and cooperative maneuvering.

As shown in Figure 7, in all three scenarios, the two UAVs and the target start from different initial positions and gradually evolve toward the terminal state under obstacle constraints. Figure 7a corresponds to the random-obstacle scene, where the obstacle distribution is relatively scattered but the local traversable space is irregular, requiring continuous dynamic detours during pursuit. Figure 7b corresponds to the target-blocking-obstacle scene, where the obstacles substantially block the target-approach path and force the UAVs to re-plan their approach direction within limited space. Figure 7c corresponds to the gate-like-obstacle scene, where the narrow passage further increases path-selection difficulty and imposes stricter requirements on spatial traversal and cooperative coordination. In all three layouts, both UAV 0 and UAV 1 eventually converge toward the target terminal region without an evident loss of control, prolonged stagnation, or ineffective circling, indicating strong environmental adaptability and target-oriented behavior.

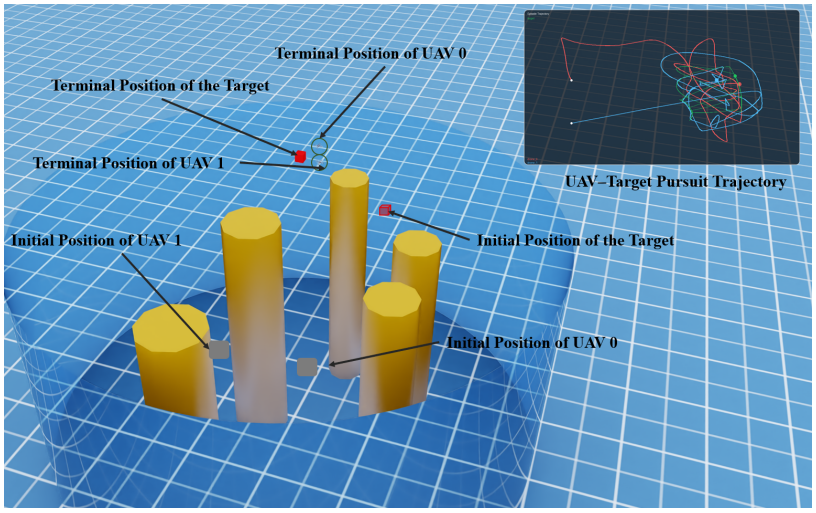
The three-dimensional trajectories in Figure 8 further show that the proposed method produces continuous and interpretable cooperative pursuit behavior in complex obstacle environments. Overall, the target trajectory exhibits avoidance and turning tendencies near obstacles, while the two UAVs can adjust their flight directions and altitudes in time according to changes in spatial structure and continuously maintain a tendency to approach the target. When entering obstacle-dense regions, the UAV trajectories do not directly pass through obstacles. Instead, they detour along obstacle boundaries or through traversable corridors, indicating that the learned policy has acquired explicit risk-avoidance behavior. Meanwhile, the two UAVs exhibit differentiated trajectory distributions: one tends to perform rapid approach, whereas the other forms a complementary encirclement from the lateral or rear-side region. Such role differentiation is beneficial for increasing the probability of sustained target containment and reducing the risk of tracking failure caused by single-agent occlusion.

A further comparison across different scenarios shows that trajectory adjustment is smoother in the random-obstacle scene, indicating that the policy can maintain favorable continuous control under relatively scattered obstacle layouts. In the target-blocking-obstacle scene, the UAVs exhibit more evident turning and approach-path reconstruction in front of obstacles, reflecting adaptability to large-scale blocking structures. In the gate-like-obstacle scene, the trajectories become more contracted and aligned near the passage, showing that the UAVs can exploit limited traversable space to accomplish safe passage and then re-establish stable tracking of the target after crossing. These phenomena indicate

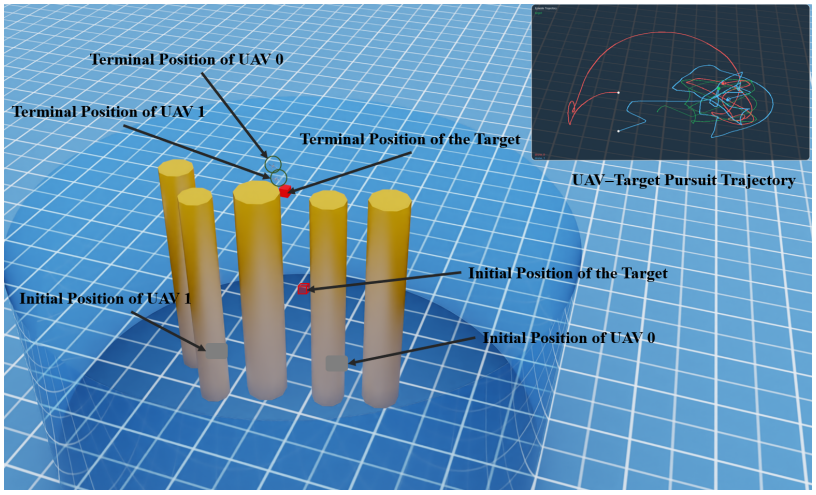
that TASRP not only improves the final task success rate but also forms safety maneuvers with consistent policy behavior under complex geometric constraints.



(**a**)

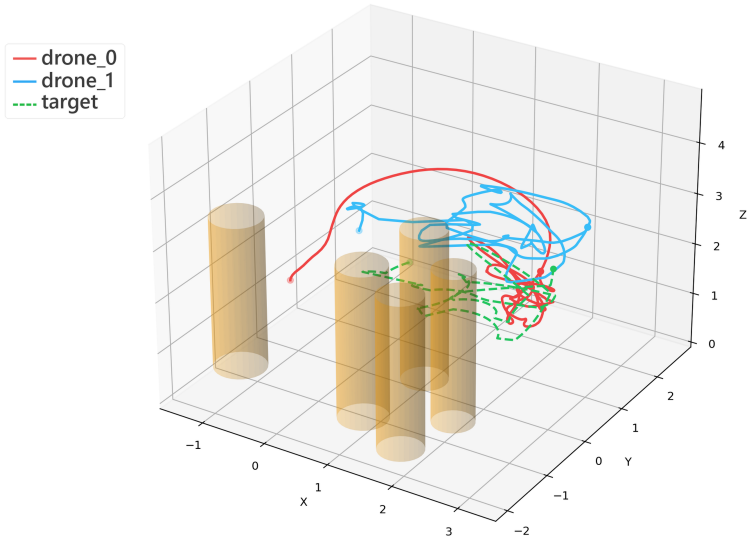


(**b**)

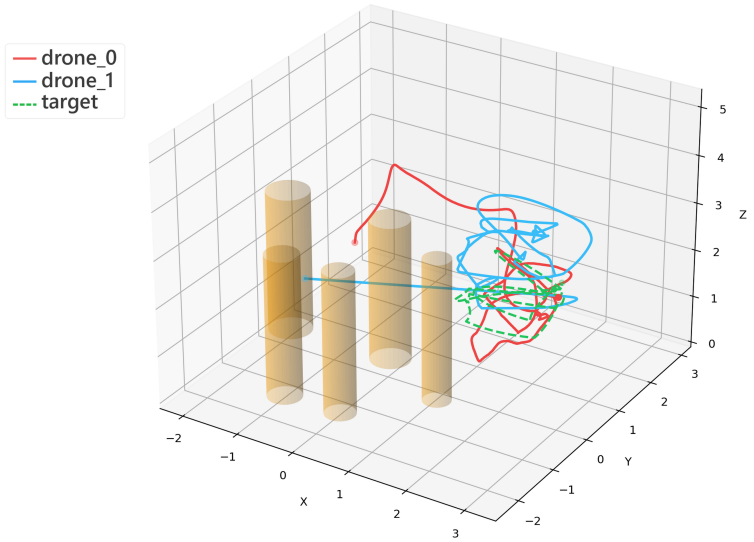


(**c**)

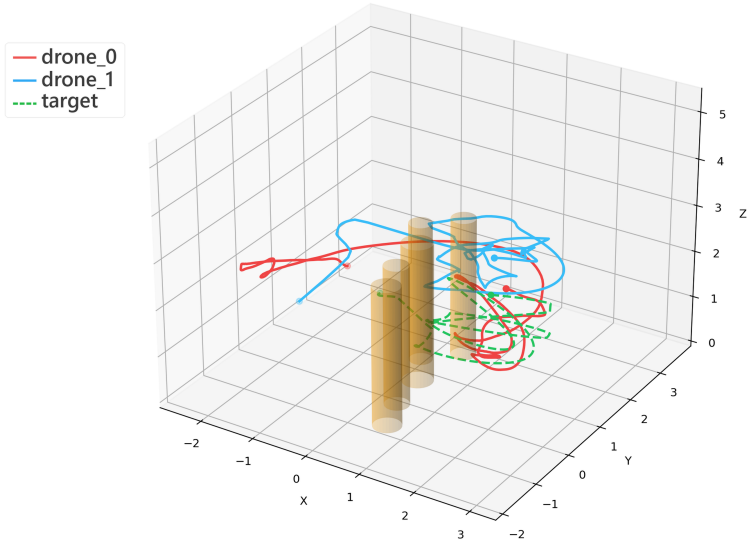
Figure 7. Visualization of complex obstacle scenarios: (**a**) Random obstacle scenario. (**b**) Target-blocking obstacle scenario. (**c**) Gate-like obstacle scenario.



(**a**)



(**b**)



(**c**)

Figure 8. Three-dimensional trajectory visualization of UAV pursuit behaviors in complex scenarios: (**a**) Random obstacle scenario. (**b**) Target-blocking obstacle scenario. (**c**) Gate-like obstacle scenario.

Taken together, Figures 7 and 8 show that the proposed method achieves favorable target-approach capability, obstacle-avoidance capability, and dual-UAV cooperation across different complex obstacle scenarios. The visualization results are consistent with the quantitative conclusions drawn from the task success rate and safety metrics, further verifying that TASRP can realize stable, continuous, and safe multi-UAV persistent tracking control in complex three-dimensional environments.

5.6. Ablation Experiments and Analysis

To provide a clearer estimate of how each TASRP component contributes to task performance and safety robustness, this study conducts ablation experiments by removing the goal-oriented teacher, the risk-gating mechanism, the dual-head residual structure, and the safety-cost critic, respectively. All ablation variants are evaluated under the same environment settings as those used in the main experiments. Table 10 reports the scenario-wise validation results together with the average changes relative to the complete TASRP model, while Figures 9 and 10 illustrate the corresponding training dynamics and performance-safety trade-off characteristics. Based on these results, we further analyze which modules mainly support persistent tracking and which modules mainly contribute to safety-oriented regulation.

Table 10. Final validation results of the TASRP ablation study.

Scenario	Metric	TASRP (Ours)	w/o Goal Teacher	w/o Risk Gating	w/o Dual-Head Residual	w/o Safety-Cost Critic	w/o Risk Gating + Safety-Cost Critic
1	Task success rate	0.79	0.15	0.76	0.16	0.74	0.74
	Obstacle crash rate	0.13	0.10	0.34	0.26	0.22	0.38
	Boundary crash rate	0.01	0.02	0.02	0.02	0.02	0.15
	Inter-UAV collision count	0	0	3	0	3	4
2	Task success rate	0.78	0.14	0.74	0.14	0.71	0.72
	Obstacle crash rate	0.15	0.11	0.37	0.30	0.25	0.41
	Boundary crash rate	0.02	0.02	0.03	0.03	0.03	0.17
	Inter-UAV collision count	0	2	5	0	9	7
3	Task success rate	0.75	0.13	0.73	0.13	0.69	0.71
	Obstacle crash rate	0.14	0.11	0.39	0.33	0.29	0.42
	Boundary crash rate	0.02	0.03	0.04	0.04	0.04	0.19
	Inter-UAV collision count	0	4	4	0	3	5

Table 10 shows that different TASRP components contribute to performance and safety in clearly different ways rather than providing redundant gains. In addition to the single-component ablations, we further include a representative joint ablation variant without the risk-gating mechanism and the safety-cost critic to better reveal the combined effect of removing multiple safety-oriented components. It should be noted that all ablation variants are retrained end-to-end under the same protocol as the full TASRP model, so the results reflect performance changes caused by removing specific components from the unified framework rather than combining separately trained submodules.

The most significant reductions in the task success rate occur when the goal-oriented teacher or the dual-head residual structure is removed. Across all three scenarios, the success rate drops from 0.79, 0.78, and 0.75 for the complete TASRP model to 0.15, 0.14, and 0.13 without the goal teacher, and to 0.16, 0.14, and 0.13 without the dual-head residual structure. These results indicate that the goal-oriented teacher and the dual-head residual structure are the two components that contribute most directly to maintaining effective target-oriented pursuit under obstacle blockage.

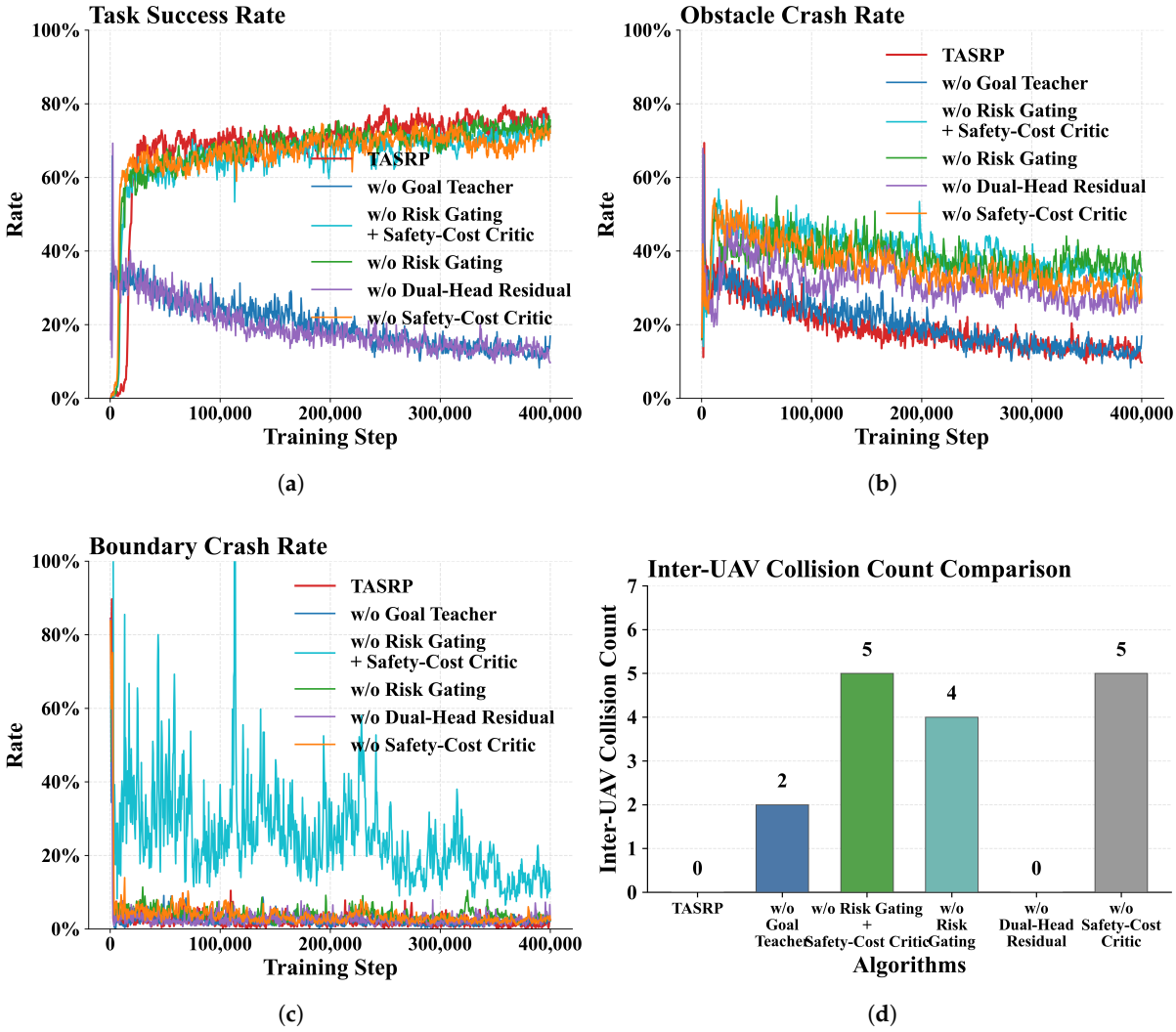


Figure 9. Training curves and validation collision comparison of TASRP and its ablation variants: (**a**) Task success rate. (**b**) Obstacle crash rate. (**c**) Boundary crash rate. (**d**) Validation inter-UAV collision count.



Figure 10. Safety-performance trade-off analysis of different ablation variants: (**a**) Task success rate vs. obstacle crash rate. (**b**) Task success rate vs. boundary crash rate.

By contrast, the strongest degradation in safety performance is observed when the risk-gating mechanism and/or the safety-cost critic are removed. In all three scenarios, removing only risk gating increases the obstacle crash rate from 0.13–0.15 to 0.34–0.39, whereas removing only the safety-cost critic increases it to 0.22–0.29. When both components are removed simultaneously, the obstacle crash rate further rises to 0.38–0.42, and the boundary crash rate increases much more substantially to 0.15–0.19. These observations show that the risk-gating mechanism and the safety-cost critic play complementary roles in suppressing safety violations, and their joint removal causes a broader deterioration in safety robustness than removing either component alone.

An important phenomenon in Table 10 is that the newly added joint ablation without risk gating and the safety-cost critic does not cause a drastic collapse in task success rate. Its success rate remains at 0.74, 0.72, and 0.71 across the three scenarios, which is only slightly lower than that of the complete TASRP model and remains close to the variants with only one of the two safety-related components removed. This result suggests that the target-oriented pursuit capability of the framework is still largely preserved even after the two safety-oriented components are removed jointly. However, the corresponding obstacle crash rate, boundary crash rate, and inter-UAV collision count all become clearly worse, indicating that the main effect of these two modules lies not in creating the target-reaching ability itself but in regulating how safely that ability is executed in cluttered environments.

Another important phenomenon in Table 10 is that removing the goal-oriented teacher or the dual-head residual structure causes a drastic drop in success rate, while the obstacle crash rate does not increase proportionally. This behavior indicates a conservative failure mode rather than an aggressive unsafe mode. Without the goal-oriented teacher, the safety branch loses task-consistent geometric supervision and tends to generate weak or poorly aligned corrections under obstacle blockage. Without the dual-head residual structure, the policy can no longer explicitly separate target advancement from local safety correction, making it difficult to maintain stable pursuit when the direct path is blocked. As a result, the learned behavior becomes hesitant and less effective in making forward progress toward the target. Such conservative behaviors may avoid some severe collisions, but they fail to sustain efficient target approach and, therefore, lead to very low task success.

The training curves in Figure 9a further support this interpretation. The complete TASRP model converges to the highest success rate and remains stable during the late training stage, whereas the variants without the goal teacher or without the dual-head residual structure remain at persistently low success levels. Figure 9b,c show that the complete TASRP model maintains low obstacle crash rate and boundary crash rate throughout training while removing the risk gate or the safety-cost critic causes a clear increase in safety violations. More importantly, the joint variant without risk gating and the safety-cost critic exhibits the most pronounced increase in boundary crash rate and one of the highest obstacle crash-rate trends, which further confirms the coordinated contribution of these two modules to safety regulation. Figure 9d also indicates that removing safety-related mechanisms leads to more inter-UAV collision events, suggesting that these modules benefit not only obstacle avoidance and boundary maintenance but also cooperative interaction safety.

Figure 10 reveals the same contribution pattern from the perspective of performance-safety balance. The complete TASRP model is concentrated in the high-success and low-safety-cost region. By contrast, the variants without the goal teacher or without the dual-head residual structure mainly shift toward lower-success regions, indicating a degraded task capability. Meanwhile, the variants without risk gating or without the safety-cost critic shift toward higher obstacle-crash and/or boundary-crash regions, indicating degraded safety robustness. The additional joint ablation without risk gating and the safety-cost critic shows an even broader departure from the favorable region, especially in the boundary-

related trade-off distribution, which further demonstrates that these two safety-oriented components act cooperatively rather than redundantly.

As shown in Figure 10a, TASRP is mainly concentrated in the high-success and low-obstacle-crash region, demonstrating a more favorable trade-off between performance and safety. By contrast, after removing risk gating or the safety-cost critic, the obstacle crash rate increases substantially. The joint removal of both components pushes the distribution even farther away from the ideal region, confirming that the safety-related gain of TASRP is not produced by only one of them in isolation. Meanwhile, removing the dual-head residual structure or the goal-oriented teacher leads to a clear decline in task success rate together with a more dispersed distribution. These observations indicate that risk gating and safety-cost constraints are crucial for safety regulation, whereas the structured residual policy and goal-oriented supervision are essential for persistent tracking performance.

As shown in Figure 10b, the boundary crash rates of the single-component ablations remain relatively low overall, suggesting that boundary violation is not the dominant failure source for those variants. However, the joint ablation without risk gating and the safety-cost critic produces a clearly wider and higher boundary-crash distribution, showing that the coordinated absence of online risk modulation and safety-constrained optimization significantly weakens boundary-safety control. In comparison, TASRP remains more concentrated in the region of high task success rate and near-zero boundary crash rate. These results indicate that TASRP not only improves the task success rate but also preserves boundary safety more stably through the collaborative design of its safety-oriented modules.

Taken together, Figure 10a,b show that the complete TASRP model achieves a more favorable balance between task performance and safety constraints than all ablation variants. The ablation results consistently suggest that the goal-oriented teacher and the dual-head residual structure mainly support task completion, whereas the risk-gating mechanism and the safety-cost critic mainly regulate safe behavior. The additional joint ablation further confirms that the overall advantage of TASRP arises from the coordinated interaction of these modules rather than from a local gain introduced by any single design choice.

5.7. Comprehensive Analysis of Experimental Results

By synthesizing all experimental results in Section 5, it can be seen that the proposed TASRP method exhibits clear advantages for multi-UAV persistent tracking in complex obstacle environments. Under clean observations, TASRP achieves higher task success rate, lower safety violations, and more stable convergence across the three representative scenarios than the baseline methods. Under the additional noisy-observation protocol, all compared methods experience performance degradation, but TASRP still maintains the most favorable balance between persistent tracking and safety. These findings indicate that the advantage of TASRP does not rely solely on the clean simulator-interface setting, but also extends to moderate observation uncertainty at the state-estimation interface.

The joint analysis of success rate and safety metrics shows that the advantage of TASRP does not lie in a single indicator but in a more effective balance between persistent pursuit and safety regulation. In addition, the ablation experiments confirm that the performance gain of TASRP arises from the complementary interaction of structured target pursuit, risk-aware safety correction, teacher-guided detour learning, and safety-constrained optimization.

In addition, the ablation experiments verify the necessity of each key module in TASRP. Specifically, the goal-oriented teacher and the dual-head residual structure mainly influence task completion capability, whereas risk gating and the safety-cost critic play important roles in suppressing obstacle collisions, boundary failures, and inter-UAV collisions. The complete model consistently outperforms the ablation variants in terms of both task success

rate and safety indicators, showing that the performance improvement of TASRP stems from the synergistic effect of multiple modules rather than from a local gain introduced by a single design choice.

In summary, the experiments in Section 5 jointly demonstrate from four aspects, namely comparative evaluation, safety analysis, behavior visualization, and ablation validation, that TASRP can realize more efficient, safer, and more stable multi-UAV persistent tracking control in complex three-dimensional obstacle environments, thereby confirming the effectiveness, robustness, and practical value of the proposed method.

6. Conclusions

This study has addressed the coupled conflict between target advancement and flight safety in multi-UAV cooperative persistent tracking tasks under complex obstacle environments and proposes a Target-Aware Safety-Residual Pursuit reinforcement-learning framework, namely TASRP. By explicitly separating target-oriented nominal pursuit from safety-oriented local correction and coordinating them through risk-adaptive fusion, target-oriented teacher supervision, and dual-critic constrained optimization, the proposed method provides a structured solution to cooperative tracking in cluttered three-dimensional environments.

Experimental results in random-obstacle, target-blocking-obstacle, and gate-like-obstacle scenarios show that TASRP consistently outperforms representative MARL baselines in task success and safety-related metrics while maintaining stable training behavior. In addition, under a controlled noisy-observation protocol that perturbs key local-state channels, TASRP remains more robust than MAPPO and HAPPO, indicating that the proposed structured policy retains a favorable performance-safety trade-off even under moderate observation uncertainty.

In conclusion, the proposed TASRP framework provides an effective and interpretable solution for the joint optimization of cooperative tracking and safe avoidance in complex obstacle environments. More broadly, this work offers a structured perspective for handling the coupling between task-driven objectives and safety constraints in multi-agent systems. In practical terms, TASRP has potential application value in real-world tasks such as multi-UAV target surveillance, airspace security, emergency response, and autonomous cooperative missions in cluttered environments, where both persistent tracking capability and flight safety are required.

Although the present study includes an additional noisy-observation robustness evaluation, several aspects still require further investigation. First, the proposed framework is developed at the state-estimation-to-command abstraction layer rather than as a full raw-sensor-to-actuator pipeline. Therefore, raw perception, sensor fusion, motor-level actuation, and embedded low-level control effects are not explicitly modeled in the current study. Second, only moderate independent observation perturbations are considered; richer imperfections such as temporally correlated noise, delayed communication, missed detections, and occlusion-induced partial observability are not yet modeled. Third, the current validation remains limited to two UAVs and static obstacle layouts in simulation. Therefore, further investigation is still needed before deployment in larger-scale, highly noisy, partially observable, and real-world scenarios.

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

Funding: This work was supported by the Liaoning Provincial Science and Technology Plan Joint Programme (Key R&D Programme), "Research on Key Technologies for Adaptive Command and Dispatch of Unmanned Aerial Vehicle Equipment", under Project No. 2025JH2/101800088.

Institutional Review Board Statement: Ethical review and approval were not required for this study, as it does not involve humans or animals.

Informed Consent Statement: Informed consent was not required for this study, as it does not involve human participants.

Data Availability Statement: The original data supporting the findings of this study are included in the experimental section of the article. Further inquiries can be directed to the corresponding author.

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

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
MARL	Multi-Agent Reinforcement Learning
RL	Reinforcement Learning
CTDE	Centralized Training with Decentralized Execution
Dec-POMDP	Decentralized Partially Observable Markov Decision Process
TASRP	Target-Aware Safety-Residual Pursuit Reinforcement Learning
MAPPO	Multi-Agent Proximal Policy Optimization
MACPO	Multi-Agent Constrained Policy Optimization
HAPPO	Heterogeneous-Agent Proximal Policy Optimization
HATRPO	Heterogeneous-Agent Trust Region Policy Optimization
MLP	Multilayer Perceptron

References

1. Liu, D.; Zhu, X.; Bao, W.; Fei, B.; Wu, J. SMART: Vision-Based Method of Cooperative Surveillance and Tracking by Multiple UAVs in the Urban Environment. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 24941–24956. [CrossRef]
2. Luo, C.; Miao, W.; Ullah, H.; McClean, S.; Parr, G.; Min, G. Unmanned Aerial Vehicles for Disaster Management. In *Geological Disaster Monitoring Based on Sensor Networks*; Durrani, T.S., Wang, W., Forbes, S.M., Eds.; Springer: Singapore, 2019; pp. 83–107. [CrossRef]
3. Alqefari, S.; Menai, M.E.B. Multi-UAV Task Assignment in Dynamic Environments: Current Trends and Future Directions. *Drones* **2025**, *9*, 75. [CrossRef]
4. Ni, J.; Ge, Y.; Zhao, Y.; Gu, Y. An Improved Multi-UAV Area Coverage Path Planning Approach Based on Deep Q-Networks. *Appl. Sci.* **2025**, *15*, 11211. [CrossRef]
5. Zeng, H.; Tong, L.; Xia, X. Multi-UAV Cooperative Coverage Search for Various Regions Based on Differential Evolution Algorithm. *Biomimetics* **2024**, *9*, 384. [CrossRef] [PubMed]
6. Zhang, Y.; Ding, M.; Yuan, Y.; Zhang, J.; Yang, Q.; Shi, G.; Jiang, F.; Lu, M. Multi-UAV Cooperative Pursuit of a Fast-Moving Target UAV Based on the GM-TD3 Algorithm. *Drones* **2024**, *8*, 557. [CrossRef]
7. Su, K.; Qian, F. Multi-UAV Cooperative Searching and Tracking for Moving Targets Based on Multi-Agent Reinforcement Learning. *Appl. Sci.* **2023**, *13*, 11905. [CrossRef]
8. Pan, H.; Han, L.; Yan, J.; Liu, R. Action Correction-Enhanced Multi-Agent Reinforcement Learning for Path Planning in Urban Environments. *Unmanned Syst.* **2026**, *14*, 461–479. [CrossRef]
9. 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]
10. Ekechi, C.C.; Elfouly, T.; Alouani, A.; Khattab, T. A Survey on UAV Control with Multi-Agent Reinforcement Learning. *Drones* **2025**, *9*, 484. [CrossRef]
11. Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; Wu, Y. The surprising effectiveness of PPO in cooperative multi-agent games. In Proceedings of the 36th International Conference on Neural Information Processing Systems (NIPS '22), Red Hook, NY, USA, 28 November–9 December 2022.

12. Gu, S.; Kuba, J.G.; Wen, M.; Chen, R.; Wang, Z.; Tian, Z.; Wang, J.; Knoll, A.; Yang, Y. Multi-agent constrained policy optimisation. *arXiv* **2021**, arXiv:2110.02793.
13. Zhong, Y.; Kuba, J.G.; Feng, X.; Hu, S.; Ji, J.; Yang, Y. Heterogeneous-agent reinforcement learning. *J. Mach. Learn. Res.* **2024**, *25*, 1–67. [CrossRef]
14. Hu, J.; Yang, X.; Wang, W.; Wei, P.; Ying, L.; Liu, Y. Obstacle Avoidance for UAS in Continuous Action Space Using Deep Reinforcement Learning. *IEEE Access* **2022**, *10*, 90623–90634. [CrossRef]
15. Zhang, L.; Li, L.; Wei, W.; Song, H.; Yang, Y.; Liang, J. Scalable Constrained Policy Optimization for Safe Multi-agent Reinforcement Learning. In *Proceedings of the Advances in Neural Information Processing Systems*; Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2024; Volume 37, pp. 138698–138730. [CrossRef]
16. Dalal, G.; Dvijotham, K.D.; Vecerík, M.; Hester, T.; Paduraru, C.; Tassa, Y. Safe Exploration in Continuous Action Spaces. *arXiv* **2018**, arXiv:1801.08757.
17. Zhang, J.; Lei, C.; Dai, C.; Wang, L.; Han, Z.; Gao, F. High-Speed Vision-Based Flight in Clutter with Safety-Shielded Reinforcement Learning. *arXiv* **2026**, arXiv:2602.08653. [CrossRef]
18. Johannink, T.; Bahl, S.; Nair, A.; Luo, J.; Kumar, A.; Loskyll, M.; Ojea, J.A.; Solowjow, E.; Levine, S. Residual Reinforcement Learning for Robot Control. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 6023–6029. [CrossRef]
19. Kim, M.G.; Kang, D.; Kim, H.; Park, H.W. A Modular Residual Learning Framework to Enhance Model-Based Approach for Robust Locomotion. *IEEE Robot. Autom. Lett.* **2025**, *10*, 9072–9079. [CrossRef]
20. Ren, H.; Han, C.; Pan, H.; Sun, J.; Li, S.; An, D.; Hu, K. Multi-UAV Cooperative Pursuit Planning via Communication-Aware Multi-Agent Reinforcement Learning. *Aerospace* **2025**, *12*, 993. [CrossRef]
21. Wenhong, Z.; Li, J.; Liu, Z.; Shen, L. Improving multi-target cooperative tracking guidance for UAV swarms using multi-agent reinforcement learning. *Chin. J. Aeronaut.* **2021**, *35*, 100–112. [CrossRef]
22. 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]
23. Wang, Y.; Dong, L.; Sun, C. Cooperative Control for Multi-Player Pursuit-Evasion Games With Reinforcement Learning. *Neurocomputing* **2020**, *412*, 101–114. [CrossRef]
24. Xu, L.; Hu, B.; Guan, Z.; Cheng, X.; Li, T.; Xiao, J. Multi-agent Deep Reinforcement Learning for Pursuit-Evasion Game Scalability. In *Proceedings of the 2019 Chinese Intelligent Systems Conference*; Jia, Y., Du, J., Zhang, W., Eds.; Springer: Singapore, 2020; pp. 658–669.
25. Hüttenrauch, M.; Šošić, A.; Neumann, G. Deep reinforcement learning for swarm systems. *J. Mach. Learn. Res.* **2019**, *20*, 1966–1996.
26. Kokolakis, N.M.T.; Vamvoudakis, K.G. Safety-Aware Pursuit-Evasion Games in Unknown Environments Using Gaussian Processes and Finite-Time Convergent Reinforcement Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 3130–3143. [CrossRef] [PubMed]
27. Liu, D.; Zong, Q.; Zhang, X.; Zhang, R.; Dou, L.; Tian, B. Game of Drones: Intelligent Online Decision Making of Multi-UAV Confrontation. *IEEE Trans. Emerg. Top. Comput. Intell.* **2024**, *8*, 2086–2100. [CrossRef]
28. Witt, C.S.D.; Gupta, T.; Makoviichuk, D.; Makoviychuk, V.; Torr, P.H.S.; Sun, M.; Whiteson, S. Is Independent Learning All You Need in the StarCraft Multi-Agent Challenge? *arXiv* **2020**, arXiv:2011.09533.
29. Zhao, B.; Huo, M.; Li, Z.; Feng, W.; Yu, Z.; Qi, N.; Wang, S. Graph-based multi-agent reinforcement learning for collaborative search and tracking of multiple UAVs. *Chin. J. Aeronaut.* **2025**, *38*, 103214. [CrossRef]
30. Zhang, R.; Hou, J.; Chen, G.; Li, Z.; Chen, J.; Knoll, A. Residual Policy Learning Facilitates Efficient Model-Free Autonomous Racing. *IEEE Robot. Autom. Lett.* **2022**, *7*, 11625–11632. [CrossRef]
31. Gu, J.; Wang, Y. A constrained reinforcement learning based approach for cooperative control of multi-UAV in dense obstacle environments. *Sci. China Technol. Sci.* **2026**, *69*, 1120601. [CrossRef]
32. Shi, J.; Li, Z.; Sun, W.; Bai, Z.; Wang, F.; Quek, T. Multi-agent SAC approach aided joint trajectory and power optimization for multi-UAV assisted wireless networks with safety constraints. *Chin. J. Aeronaut.* **2025**, *38*, 103368. [CrossRef]
33. Dawood, M.; Pan, S.; Dengler, N.; Zhou, S.; Schoellig, A.P.; Bennewitz, M. Safe Multi-Agent Reinforcement Learning for Behavior-Based Cooperative Navigation. *IEEE Robot. Autom. Lett.* **2025**, *10*, 6256–6263. [CrossRef]
34. Silver, T.; Allen, K.R.; Tenenbaum, J.B.; Kaelbling, L.P. Residual Policy Learning. *arXiv* **2018**, arXiv:1812.06298.
35. Abbas, A.N.; Chasparis, G.C.; Kelleher, J.D. Specialized Deep Residual Policy Reinforcement Learning Framework for Safe and Adaptive Continuous Control. *IET Control Theory Appl.* **2026**, *20*, e70099. [CrossRef]
36. Pang, J.; He, J.; Mohamed, N.M.A.A.; Lin, C.; Zhang, Z.; Hao, X. A hierarchical reinforcement learning framework for multi-UAV combat using leader–follower strategy. *Knowl.-Based Syst.* **2025**, *316*, 113387. [CrossRef]

37. Tan, J.; Xue, S.; Guo, Z.; Li, H.; Zheng, X.; Cao, H. Adaptive hierarchical control of quadcopters via safe reinforcement learning from human demonstration. *Eng. Appl. Artif. Intell.* **2026**, *163*, 113013. [CrossRef]
38. Koops, W.; Junges, S.; Jansen, N. Approximate Dec-POMDP Solving Using Multi-Agent A*. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*; Larson, K., Ed.; International Joint Conferences on Artificial Intelligence Organization: Montpellier, France, 2024; pp. 6743–6751. [CrossRef] [PubMed]

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.