applied sciences

MDPI

Article

Enhanced Soft Actor–Critic with Dual-Path Channel Attention for UAV Autonomous Navigation in Complex Environments

Yufei Wang [1], Tong Zhang [1], Fan Zhou [1,*], Fang Liu [1], Bo Qian [1] and Jun Liu [2]

[1] School of Information Science and Engineering, Shenyang Ligong University, Shenyang 110159, China; yfwang@sylu.edu.cn (Y.W.); tongz714@163.com (T.Z.); liufang@sylu.edu.cn (F.L.); keen_xp@163.com (B.Q.)
[2] School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China; liujun@cse.neu.edu.cn
* Correspondence: zhoufan@sylu.edu.cn

Abstract

In complex and unknown environments, unmanned aerial vehicle (UAV) autonomous navigation still faces issues such as insufficient representation of state characteristics, fixed reward guidance, and low efficiency in utilizing key experience samples. To address these problems, this paper proposes an improved soft actor–critic (SAC) method that integrates dual-path channel attention (DPCA), adaptive reward feedback (ARF), and prioritized experience replay (PER); this method is named DPCA-ARF-PER-SAC. The proposed DPCA module is introduced into the actor network to recalibrate one-dimensional navigation state features and enhance the representation ability of key decision-making information. At the same time, the ARF mechanism can dynamically adjust the reward weights according to the training progress, while PER is used to improve the utilization efficiency of key samples. The experiments are conducted in a two-stage structure, including module-level ablation verification in the two-dimensional (2D) SimpleAvoid scenario and main performance comparison in the three-dimensional (3D) NH_center scenario. The experimental results show that the success rate of this method reaches 1.00 in the 2D scenario, and the collision rate is 0.00. In the 3D scenario, the success rate is 0.76, the collision rate is 0.24, and the average episode length is 232.8 steps. Compared with the baseline SAC, the success rate is increased by 2 percentage points, the collision rate is reduced by 2 percentage points, and the average episode length is reduced by 4.3 steps. Compared with twin delayed deep deterministic policy gradient (TD3), SAC, and representative reinforcement learning methods such as attention-mechanism SAC (AM-SAC) based on attention mechanism enhancement, the proposed DPCA-ARF-PER-SAC method shows more balanced performance in task completion, navigation safety, and path efficiency. These results indicate that DPCA-ARF-PER-SAC provides a more robust navigation strategy for complex 3D UAV autonomous navigation tasks.

Keywords: unmanned aerial vehicle (UAV) autonomous navigation; obstacle avoidance; path planning; soft actor–critic (SAC); deep reinforcement learning; dual-path channel attention (DPCA); adaptive reward feedback (ARF); prioritized experience replay (PER)

Academic Editor: Yutaka Ishibashi

Received: 22 June 2026
Revised: 9 July 2026
Accepted: 13 July 2026
Published: 17 July 2026

1. Introduction

Due to the high mobility, flexible deployment, and strong task adaptability of unmanned aerial vehicles (UAVs), they have been widely applied in surveillance and reconnaissance, remote sensing and mapping, power line inspection, emergency rescue, logistics

transportation, and detection tasks in complex environments [1–3]. As the application scenarios of UAVs gradually expand from structured and static environments to unknown, dynamic, and complex three-dimensional environments, autonomous navigation and obstacle avoidance technologies have become the key technologies to ensure the safety and efficiency of tasks [4–6]. Especially for small UAVs, their onboard computing resources, perception capabilities, and flight endurance are usually limited, so achieving real-time, stable, and reliable autonomous navigation in complex environments remains a challenging research issue [4,5].

Traditional methods for UAV path planning include graph search algorithms, sampling methods, artificial potential field methods, and swarm intelligence optimization algorithms [1,2]. These methods can achieve satisfactory performance in known environments or structured scenarios. However, they usually rely on pre-acquired maps, environment modeling, and repeated optimization processes. When UAVs operate in unknown, dynamic, or complex three-dimensional environments, traditional methods often exhibit insufficient real-time performance, poor robustness, and limited generalization capabilities [1–3]. Therefore, improving the autonomous decision-making ability of drones in complex environments through data-driven methods has become an important research direction in the field of intelligent drone navigation.

In recent years, deep reinforcement learning (DRL) has provided a promising solution for UAV autonomous navigation. By modeling the navigation process of UAVs as a Markov decision process, the intelligent agent can learn the mapping from states to actions through interaction with the environment, thereby achieving end-to-end autonomous decision making [4,6–8]. Existing studies have applied algorithms such as deep Q-network (DQN), deep deterministic policy gradient (DDPG), double delayed deep deterministic policy gradient (TD3), approximate policy optimization (PPO), and soft actor–critic (SAC) to UAV path planning, obstacle avoidance control, target tracking, and task decision making, and have achieved certain results [6–8]. Among them, value function-based methods such as DQN are mainly applicable to discrete action spaces and have limitations when dealing with continuous control variables such as UAV speed, yaw angle, and attitude deviation. DDPG and TD3 can handle continuous action spaces, but the determinism of the policy is prone to exploration insufficiency and local optima in complex environments. In contrast, SAC adopts the maximum entropy reinforcement learning framework, and enhances the exploration ability of the policy while maximizing the expected return, and is therefore more suitable for handling complex continuous control tasks [9].

However, there is still room for improvement in autonomous navigation of UAVs based on SAC. Firstly, the state of the UAV usually contains various types of information, such as target distance, obstacle risk, speed, attitude deviation, and heading error. Different state dimensions have different importance in different decision stages. If all state dimensions are directly input into the actor network with equal weights, it may lead to insufficient representation of key navigation features. Some studies have introduced self-attention or Transformer structures to enhance the understanding ability of SAC on states [10–12]. However, these methods usually increase the network size and training cost, which is not conducive to achieving lightweight deployment on small unmanned aerial vehicles. Secondly, the fixed-weight reward function is difficult to balance the proximity of the target and the obstacle avoidance, flight safety, and posture stability in the early training stage, which may affect the convergence efficiency of the strategy and the final control quality [10,11]. Finally, in complex navigation tasks, collision samples, samples of approaching obstacles, and samples with high time difference errors are usually unevenly distributed. Uniform experience replay may lead to insufficient utilization of key samples, thereby reducing sample efficiency and training stability [13].

To address the aforementioned issues, this paper proposes an enhanced SAC-based unmanned aerial vehicle (UAV) autonomous navigation method that integrates dual-path channel attention, adaptive reward feedback, and prioritized experience replay. While maintaining the maximum entropy optimization framework and Bellman update form of SAC unchanged, this paper first introduces a dual-path channel attention module inspired by the convolution block attention module [14] into the actor network, aiming to enhance the expression ability of key decision-making features in the one-dimensional navigation state vector; secondly, it designs a training progress-driven adaptive reward feedback mechanism, allowing the reward weights to gradually transition from the goal guidance in the early training stage to the safety constraints and fine control in the later training stage; at the same time, it combines the prioritized experience replay mechanism [15] to improve the utilization efficiency of key samples. The experiment adopts a two-layer structure of "2D SimpleAvoid ablation verification + 3D NH_center main experiment", respectively verifying the contribution of each module and the comprehensive navigation performance of the complete method in complex three-dimensional scenarios.

The main contributions of this paper are summarized as follows:

- A dual-path channel attention module was proposed for the one-dimensional unmanned aerial vehicle navigation state vector and embedded into the actuator branch of SAC to improve the adaptive representation of key navigation features.
- An adaptive reward feedback mechanism based on training progress was designed to dynamically adjust the weights of reward items related to approaching the target, obstacle avoidance, heading correction, and flight stability, thereby enhancing the learning effect of the strategy in different training stages.
- By integrating the dual-path channel attention, adaptive reward feedback, and prioritized experience replay, an enhanced SAC framework named DPCA-ARF-PER-SAC was constructed. Its effectiveness was verified through module-level ablation experiments in the 2D SimpleAvoid scenario and main comparative experiments with TD3, SAC, and AM-SAC in the 3D NH_center scenario.

The remainder of this paper is organized as follows. Section 2 reviews the related work on UAV autonomous navigation and SAC-based reinforcement learning methods. Section 3 formulates the UAV autonomous navigation problem as a Markov decision process. Section 4 presents the proposed DPCA-ARF-PER-SAC method, including the dual-path channel attention module, adaptive reward feedback mechanism, and prioritized experience replay. Section 5 describes the experimental settings, evaluation metrics, ablation study, and comparative results, and further discusses the limitations and practical applicability of the proposed method. Finally, Section 6 concludes this paper and outlines future research directions.

2. Related Work

2.1. Deep Reinforcement Learning for UAV Autonomous Navigation

Autonomous navigation and obstacle avoidance of unmanned aerial vehicles (UAVs) are important research directions in the field of intelligent control of unmanned systems. Although traditional path planning methods have good stability in known environments, they usually rely on precise maps, environmental modeling, or repetitive planning processes, making them difficult to meet the requirements of real-time autonomous navigation in unknown, dynamic, and complex three-dimensional environments. Therefore, deep reinforcement learning has gradually been introduced into the tasks of UAV path planning and obstacle avoidance control. Huang et al. applied DQN to the UAV navigation problem, achieving navigation decision optimization based on deep reinforcement learning by learn-

ing the mapping relationship between environmental states and flight actions [16]. Guo et al. addressed the problem of difficult convergence of UAV navigation in high-dynamic environments by proposing a hierarchical deep reinforcement learning framework, decomposing the obstacle avoidance and target approach processes into different sub-tasks to improve learning efficiency in complex dynamic environments [7]. Wang et al. addressed the problem of autonomous navigation and obstacle avoidance of UAVs based on images in unknown environments by combining the target detection model with DQN and designing a data storage mechanism to improve training effectiveness [6]. Liu et al. addressed the path planning problem of UAVs in complex dynamic environments by using deep reinforcement learning methods to enhance the autonomous planning ability of UAVs in dynamic obstacle environments [17]. Yin et al. further studied the adaptive control method of UAVs based on deep reinforcement learning to enhance the autonomous navigation ability of UAVs in three-dimensional environments [18]. More recent studies have further explored DRL-based UAV navigation from the perspectives of visual perception, training efficiency, and generalization ability. Wu et al. studied model-free UAV navigation in unknown complex environments using vision-based reinforcement learning [19]. McEnroe et al. focused on efficient DRL-based UAV obstacle avoidance at the edge by reducing training time and inference latency [20], while Sun et al. introduced causal reinforcement learning to improve obstacle-avoidance robustness and generalization in unfamiliar environments [21].

In addition to value function-based methods, actor–critic algorithms have also been widely used in continuous control tasks of UAVs. He et al. modeled the reactive navigation of UAVs as a Markov decision process and used the TD3 algorithm to train the path planning strategy, while combining interpretability analysis methods to explain the basis of network decisions [8]. Li et al. proposed an expert experience-assisted SAC method for UAV air combat maneuver decision-making problems, improving the exploration efficiency and sample utilization ability in the early training stage by introducing a small amount of expert experience into the experience replay pool [22]. These studies show that deep reinforcement learning can improve the autonomous decision-making ability of UAVs in complex environments to a certain extent, but DQN-based methods have difficulty directly handling continuous action spaces, and deterministic strategy methods such as DDPG and TD3 may still have insufficient exploration problems in complex environments. Therefore, SAC with the maximum entropy exploration mechanism is more suitable for continuous action UAV navigation tasks.

2.2. SAC-Based Improvements for UAV Tasks

SAC, due to its adoption of the maximum entropy reinforcement learning framework, can maximize the expected return while enhancing the exploration of the strategy. Therefore, it has been gradually applied in scenarios such as three-dimensional trajectory planning for unmanned aerial vehicles, target tracking, and complex task control in recent years. Zhou et al. addressed the issues of high environmental state dimensions and sparse rewards in three-dimensional online trajectory planning, introducing self-attention mechanisms into the actor network of SAC and integrating the artificial potential field method into the reward function, thereby improving the convergence speed and path planning success rate of the algorithm [10]. Song et al. proposed the TW-SAC method, introducing the three-way decision-making idea into SAC, setting reward weights according to different task states, and applying it to the planning of unmanned aerial vehicles tracking moving target tasks, improving the adaptive ability of unmanned aerial vehicles in complex task environments [11]. Huang et al. proposed the GTrXL-SAC method, combining Gated Transformer-XL with SAC, and enhancing the obstacle perception and control decision-

making ability of unmanned aerial vehicles through joint modeling of historical states and current observations [12].

Recent studies have also explored Transformer-based reinforcement learning and multi-agent reinforcement learning for UAV planning and control, further indicating that more complex DRL architectures can improve decision-making capability but usually introduce additional model complexity and training cost [23–26].

In addition, SAC has also been applied to complex decision-making tasks such as unmanned aerial vehicle communication, edge computing, and resource optimization. Xu et al. proposed the AD-SAC method for a mixed action space, decoupling discrete actions and continuous actions, and applying it to unmanned aerial vehicle path planning and gimbal scanning tasks [27]. Xie et al. proposed the GeoAgg-HSAC framework, combining the mixed action space SAC with geographic information state aggregation, and applying it to trajectory and resource optimization in the communication and positioning network of unmanned aerial vehicles in mountainous areas [28]. Goudarzi et al. applied SAC to the unmanned aerial vehicle-assisted vehicular network edge computing scenario, jointly optimizing the unmanned aerial vehicle trajectory, user association, and task offloading decisions to reduce information age and energy consumption [29]. Yang et al. studied the multi-unmanned aerial vehicle cooperative search problem in partially observable low-altitude environments and applied deep reinforcement learning to cooperative decision-making in complex environments [30]. These studies demonstrate that SAC has strong continuous control capabilities and task adaptability, but most methods focus on complex task modeling, multimodal state processing, or multi-unmanned aerial vehicle collaborative scenarios, while the attention to lightweight single unmanned aerial vehicle autonomous navigation problems based on one-dimensional navigation state vectors is still insufficient.

2.3. Research Gap and Motivation

In the drone navigation tasks based on DRL, the representation ability of state features directly affects the performance of policy learning. In complex environments, the state of the drone usually contains various types of information, such as target distance, obstacle risk, speed, attitude error, and heading deviation. The importance of different features varies at the decision-making stage. The convolution block attention module (CBAM) proposed by Woo et al. adaptively recalibrates the features through channel attention and spatial attention, providing an effective reference for lightweight feature enhancement [14]. However, CBAM was originally designed for two-dimensional convolution feature maps, while the drone navigation state used in this study is a compact one-dimensional state vector. Therefore, this paper did not directly adopt the original CBAM structure but focused on designing a lightweight dual-path channel attention structure suitable for one-dimensional navigation state vectors.

In drone tasks, the studies by Zhou et al. and Huang et al. also show that the attention mechanism can enhance the agent's ability to focus on key state information, thereby improving the performance of policy learning in complex environments [10,12]. Jiang et al. further introduced the Transformer into the drone obstacle avoidance and target tracking tasks, enhancing the understanding of the dynamic environment through observation sequence modeling [23]. Dong et al. proposed a data-driven reinforcement learning method based on Transformer for the joint planning and control of multiple drones without collision, further verifying the potential of the attention mechanism in complex drone decision-making tasks [24]. However, complex attention structures like Transformer usually increase the network size and training cost, and may not be suitable for resource-constrained small drones. Therefore, designing a lightweight dual-path channel

attention module for one-dimensional navigation state vectors has significant practical research significance.

The design of reward functions and the experience replay mechanism are also important factors affecting the navigation performance based on reinforcement learning. Schaul et al. proposed the prioritized experience replay method, which improves the sampling probability of samples with larger time difference errors, thereby enhancing the utilization efficiency of important experiences [13]. In the drone obstacle avoidance task, the collision state, the state of approaching obstacles, and the state of reaching the target usually have higher learning value than ordinary transfer samples. Zhang et al. introduced safety constraints into the reinforcement learning decision-making process for the trajectory optimization of emergency communication in drone, indicating the importance of balancing safety and task rewards in drone control tasks [31]. Chen et al. designed a robust DQN framework for drone relay communication optimization under heterogeneous tasks and service quality constraints, improving the learning ability in complex constraint environments through improved sampling strategies and network structures [32]. Chen et al. also proposed a semi-dual DQN resource optimization strategy to alleviate the Q-value overestimation problem in traditional DQN and improve the energy efficiency optimization ability in drone-assisted networks [33]. Wang et al. proposed the HAP-DQN method for multi-drone collaborative inspection, introducing hierarchical exploration, reward shaping, and adaptive prioritized replay mechanisms to improve the exploration efficiency and key sample learning ability in path planning [34]. Han et al. applied DDQN, priority DDQN, distributed DQN, and noise DQN to the path planning for maritime search and rescue coverage, verifying the effectiveness of various deep reinforcement learning improvement mechanisms in improving path planning quality and search efficiency [25,26,35].

In summary, the existing research has promoted the development of unmanned aerial vehicle (UAV) autonomous navigation from aspects such as navigation based on deep reinforcement learning (DRL), continuous control based on policy control (SAC), attention mechanism, reward function design, and experience replay. However, there are still three major limitations: First, although the complex attention structure can enhance the understanding of the state, it may introduce unnecessary model overhead for tasks based on compact one-dimensional navigation state vectors. Second, fixed reward functions are difficult to adapt to the constantly changing learning goals in different training stages. Third, unified experience replay cannot fully utilize key samples such as collisions, approaching obstacles, and high temporal difference errors. To address these issues, this paper proposes an improved SAC method—DPCA-ARF-PER-SAC—which integrates dual-path channel attention, adaptive reward feedback, and prioritized experience replay to improve the success rate, safety, path efficiency, and training stability of UAV autonomous navigation in complex environments.

3. Problem Formulation

Before formulating the UAV autonomous navigation task as a Markov decision process, a situational sketch of the considered navigation scenario is presented in Figure 1. The UAV starts from an initial position and is required to reach the target region while avoiding obstacles and remaining within the feasible workspace. At each time step, the UAV receives a compact state vector containing obstacle-risk and target-related navigation information, and the policy outputs continuous control actions to guide the UAV toward the goal.

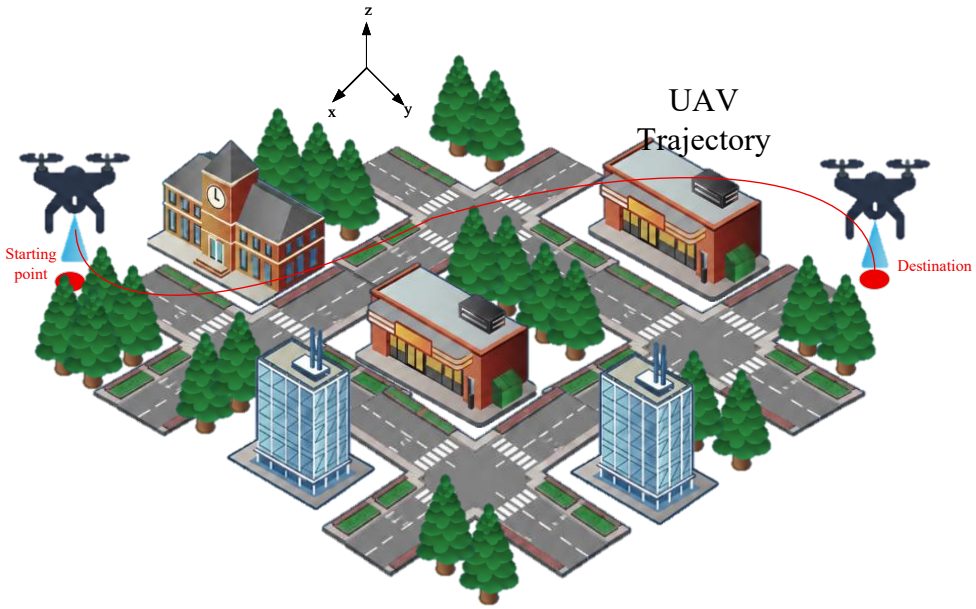


Figure 1. Situational sketch of UAV autonomous navigation in a complex environment.

3.1. Task Description

This paper focuses on the problem of autonomous navigation and obstacle avoidance for unmanned aerial vehicles (UAVs) in complex and unknown environments. Given the initial position and the target point, the intelligent agent needs to continuously output flight control actions based on the current local perception information and its own state, so as to enable the UAV to reach the target area as efficiently as possible while avoiding collisions and exceeding boundaries. This paper adopts a reactive navigation method based on state vector input. The policy network only generates control actions based on the current observed state, thereby reducing the reliance on complete prior information of the environment. In the experiments, the 2D SimpleAvoid scenario is used for ablation verification, and the 3D NH_center scenario is used for the verification of complex three-dimensional navigation performance.

3.2. Markov Decision Process Formulation

To describe the UAV autonomous navigation process using a deep reinforcement learning framework, the task is formulated as a Markov decision process (MDP) with a continuous action space. The MDP is defined as

$$\mathcal{M} = \{\mathcal{S}, \mathcal{A}, P, R, \gamma\}, \tag{1}$$

where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space, P represents the state transition probability of the environment, R denotes the reward function, and $\gamma \in (0, 1)$ is the discount factor used to balance immediate and future rewards.

At time step t, the agent observes the current state $s_t \in \mathcal{S}$ and selects an action $a_t \in \mathcal{A}$ according to the policy π_ϕ. After interacting with the environment, the agent receives the next state s_{t+1} and an immediate reward r_t. This process can be expressed as

$$a_t \sim \pi_\phi(\cdot|s_t), \tag{2}$$

$$s_{t+1} \sim P(\cdot|s_t, a_t), \tag{3}$$

$$r_t = R(s_t, a_t, s_{t+1}). \tag{4}$$

Here, $a_t \sim \pi_\phi(\cdot|s_t)$ indicates that the action is sampled from the current stochastic policy under state s_t, s_{t+1} denotes the next state after executing action a_t, and r_t is the corresponding immediate reward.

To describe whether an episode is terminated, a terminal indicator d_t is defined as

$$d_t = \begin{cases} 1, & \text{if the episode terminates,} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

In this study, an episode terminates when the UAV reaches the target region, collides with an obstacle, violates the flight boundary, or reaches the maximum number of steps. Otherwise, $d_t = 0$ and the interaction continues.

3.3. State Space Design

In this work, a compact state vector is used as the input of the policy network. The state vector consists of two parts: local obstacle perception features and target-related navigation features. It is defined as

$$s_t = [p_t, q_t], \tag{6}$$

where p_t denotes the local obstacle perception feature vector, and q_t denotes the target-related navigation feature vector.

According to the experimental setting, the local perception information is compressed into five regional obstacle-risk features, which can be expressed as

$$p_t = [p_{1,t}, p_{2,t}, p_{3,t}, p_{4,t}, p_{5,t}]. \tag{7}$$

Here, $p_{i,t}$ denotes the obstacle-risk response in the i-th perception region at time step t. A larger value indicates that the obstacle in the corresponding region is closer to the UAV and that the potential collision risk is higher, whereas a smaller value indicates that the region is relatively safe. Specifically, $p_{1,t}, \ldots, p_{5,t}$ do not represent action categories or fixed danger levels. Instead, they are compact spatial risk descriptors used to encode the local obstacle distribution around the UAV. The surrounding space of the UAV is divided into five direction-related perception sectors according to the current heading direction. Each $p_{i,t}$ corresponds to the obstacle-risk response of the i-th sector.

When an obstacle is closer to the UAV in a certain sector, the corresponding risk value becomes larger. When no obstacle exists in the sector or the obstacle is far away from the UAV, the corresponding risk value becomes smaller. Therefore, $p_{1,t}, \ldots, p_{5,t}$ provide a compact representation of local obstacle distribution without directly using a full environmental map. This design reduces the input dimension of the policy network and is suitable for lightweight UAV navigation based on one-dimensional state vectors.

In the 3D NH_center scenario, the target-related navigation and motion-state vector is defined as

$$q_t = [d_{xy,t}, \Delta z_t, \Delta\psi_t, v_{xy,t}, v_{z,t}], \tag{8}$$

where $d_{xy,t}$ denotes the horizontal distance between the UAV and the target point, Δz_t denotes the altitude difference between the UAV and the target point, $\Delta\psi_t$ denotes the heading deviation between the current heading direction and the target direction, $v_{xy,t}$ denotes the horizontal velocity-related motion state, and $v_{z,t}$ denotes the vertical velocity-related motion state. The introduction of velocity-related information enables the state vector to better reflect the current motion tendency of the UAV, which is important for continuous control and approximate Markov modeling.

Therefore, the complete state vector for the 3D navigation task can be written as

$$s_t = [p_{1,t}, p_{2,t}, p_{3,t}, p_{4,t}, p_{5,t}, d_{xy,t}, \Delta z_t, \Delta\psi_t, v_{xy,t}, v_{z,t}]. \tag{9}$$

The horizontal distance is calculated as

$$d_{xy,t} = \sqrt{(x_t - x_g)^2 + (y_t - y_g)^2}, \tag{10}$$

where (x_t, y_t, z_t) denotes the current position of the UAV, and (x_g, y_g, z_g) denotes the position of the target point.

The altitude difference is defined as

$$\Delta z_t = z_t - z_g. \tag{11}$$

A positive value of Δz_t indicates that the UAV is higher than the target point, while a negative value indicates that the UAV is lower than the target point.

The heading deviation is calculated as

$$\Delta\psi_t = \text{wrap}\big(\text{atan2}(y_g - y_t, x_g - x_t) - \psi_t\big), \tag{12}$$

where ψ_t denotes the current heading angle of the UAV, $\text{atan2}(\cdot)$ is used to calculate the direction angle from the current position to the target point, and $\text{wrap}(\cdot)$ limits the angle to the interval $(-\pi, \pi]$.

For the 2D SimpleAvoid scenario, altitude control is not considered. Therefore, the state vector can be simplified as

$$s_t^{2D} = [p_{1,t}, p_{2,t}, p_{3,t}, p_{4,t}, p_{5,t}, d_{xy,t}, \Delta\psi_t, v_{xy,t}]. \tag{13}$$

Thus, the 2D SimpleAvoid state representation can be regarded as a reduced form of the 3D state representation, where the altitude-related state Δz_t and vertical velocity state $v_{z,t}$ are removed. This setting is mainly used to verify the influence of the proposed modules on basic obstacle avoidance and target-reaching ability.

3.4. Action Space Design

In the 3D navigation setting, a continuous action space is adopted. At time step t, the action vector is defined as

$$a_t = [v_{xy,t}, v_{z,t}, \omega_{\psi,t}], \tag{14}$$

where $v_{xy,t}$ denotes the horizontal velocity control command, $v_{z,t}$ denotes the vertical velocity control command, and $\omega_{\psi,t}$ denotes the yaw-rate control command. Therefore, the action space in the 3D quadrotor navigation setting consists of three continuous control variables.

To describe the action constraints, the action vector is bounded by predefined lower and upper limits:

$$a_t \in \mathcal{A}, \quad a_{\min} \le a_t \le a_{\max}, \tag{15}$$

where $a_{\min}$ and $a_{\max}$ denote the lower and upper bounds of the action output, respectively. These constraints are used to ensure that the generated control commands remain within the feasible flight-control range of the UAV. Specifically, the action constraints are set as

$$0 \le v_{xy,t} \le v_{xy}^{\max}, \quad -v_z^{\max} \le v_{z,t} \le v_z^{\max}, \quad -\omega_\psi^{\max} \le \omega_{\psi,t} \le \omega_\psi^{\max}. \tag{16}$$

In the experiments, $v_{xy}^{\max}$, $v_z^{\max}$, and $\omega_\psi^{\max}$ are set according to the UAV control constraints listed in Table 1. Specifically, the maximum horizontal velocity is 5 m/s, the maximum vertical velocity is 2 m/s, and the maximum yaw rate is 30 deg/s.

Table 1. Main experimental parameters.

Parameter	Value
Learning rate	3×10^{-4}
Batch size	128
Discount factor γ	0.99
Replay buffer size	50,000
Learning starts	2000
Train frequency	1
Gradient steps	1
Network architecture	[64, 32, 16]
Activation function	Tanh
Total timesteps	200,000
Maximum episode steps	600
Accept radius	2 m
Crash distance	1 m
Maximum horizontal velocity	5 m/s
Maximum vertical velocity	2 m/s
Maximum yaw rate	30 deg/s
Time step Δt	0.1 s

Under a simplified kinematic model, the position and heading of the UAV can be approximately updated as

$$\begin{cases} x_{t+1} = x_t + v_{xy,t} \cos \psi_t \Delta t, \\ y_{t+1} = y_t + v_{xy,t} \sin \psi_t \Delta t, \\ z_{t+1} = z_t + v_{z,t} \Delta t, \\ \psi_{t+1} = \psi_t + \omega_{\psi,t} \Delta t, \end{cases} \tag{17}$$

where Δt denotes the control time step. Equation (17) indicates that the three-dimensional continuous action output by the actor network directly determines the spatial displacement and heading change of the UAV at the next time step. Therefore, learning reasonable control actions from the compact state representation is one of the key objectives of the proposed method.The above motion model is a simplified discrete-time kinematic model for high-level navigation policy learning. It describes the approximate position and yaw update of the UAV according to the velocity and yaw-rate commands. This model is suitable for evaluating the navigation decision-making ability of the proposed reinforcement learning method in simulation.

However, the current model does not explicitly consider inertia, acceleration limits, minimum turning radius, actuator delay, wind disturbance, or detailed aerodynamic effects. Therefore, the simulation results should be interpreted as the validation of the proposed policy-learning framework under a simplified kinematic setting. In future work, a more complete UAV dynamics model, hardware-in-the-loop simulation, and real-world flight tests will be introduced to further evaluate the physical feasibility and transferability of the proposed method.

For the 2D SimpleAvoid scenario, altitude control is not considered and $v_{z,t} = 0$. Thus, only the horizontal velocity and yaw-rate control are used, and the action vector can be simplified as

$$a_t^{2D} = [v_{xy,t}, \omega_{\psi,t}]. \tag{18}$$

This simplified action setting is used to verify the obstacle avoidance and target-reaching capability of the proposed method in the two-dimensional ablation scenario.

3.5. Reward Function

To guide the UAV to approach the target while avoiding obstacles and satisfying flight constraints, the immediate reward is designed as a weighted combination of multiple task-driven and safety-related reward terms:

$$r_t = \lambda_g(\rho_t) r_t^g + \lambda_o(\rho_t) r_t^o + \lambda_z(\rho_t) r_t^z + \lambda_\psi(\rho_t) r_t^\psi + r_t^{term}, \tag{19}$$

where r_t^g denotes the target-approaching reward, r_t^o denotes the obstacle-avoidance reward, r_t^z denotes the altitude-constraint reward, r_t^ψ denotes the heading-error reward, and r_t^{term} denotes the terminal reward. The terms $\lambda_g(\rho_t)$, $\lambda_o(\rho_t)$, $\lambda_z(\rho_t)$, and $\lambda_\psi(\rho_t)$ are the corresponding adaptive weights, where ρ_t denotes the normalized training progress. The detailed scheduling strategy of these weights is described in the adaptive reward feedback module.

To improve the interpretability of the reward function, the target-approaching reward is defined as

$$r_t^g = c_g \left(d_{xy,t-1} - d_{xy,t} \right), \tag{20}$$

where c_g is a proportional coefficient. This term encourages the UAV to move closer to the target point. A positive reward is obtained when the horizontal distance to the target decreases.

The obstacle-avoidance reward is defined as

$$r_t^o = -c_o \max_{k \in \{1,\ldots,5\}} p_{k,t}, \tag{21}$$

where c_o is the obstacle-penalty coefficient, and $\max_{k \in \{1,\ldots,5\}} p_{k,t}$ represents the strongest obstacle-risk response among the five local perception regions. This term penalizes high-risk obstacle responses and encourages the UAV to actively move away from potentially dangerous regions during flight.

The altitude-constraint reward is defined as

$$r_t^z = -c_z |\Delta z_t|, \tag{22}$$

where c_z is the altitude-constraint coefficient. This term is used to reduce the altitude deviation between the UAV and the target point.

The heading-error reward is defined as

$$r_t^\psi = -c_\psi |\Delta \psi_t|, \tag{23}$$

where c_ψ is the heading-constraint coefficient. This term encourages the UAV to align its heading direction with the target direction.

The terminal reward is used to explicitly distinguish successful arrival and failure cases. It is defined as

$$r_t^{term} = \begin{cases} R_{goal}, & d_{xy,t} \le \varepsilon_g, \\ -R_{col}, & \text{collision occurs}, \\ -R_{out}, & \text{out of workspace}, \\ 0, & \text{otherwise}, \end{cases} \tag{24}$$

where R_{goal} denotes the positive reward for reaching the target, R_{col} and R_{out} denote the collision penalty and boundary-violation penalty, respectively, and ε_g is the target-reaching threshold.

In the 2D SimpleAvoid scenario, altitude control is not involved. Therefore, the altitude-constraint reward r_t^z is removed, while the other reward terms remain unchanged.

3.6. Maximum Entropy Optimization Objective

This study adopts Soft Actor–Critic (SAC) as the basic reinforcement learning framework. Unlike traditional reinforcement learning methods that mainly maximize the expected cumulative reward, SAC introduces an entropy regularization term to encourage policy exploration. The maximum entropy objective can be expressed as

$$J(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{T}\gamma^t(r_t - \alpha\log\pi(a_t|s_t))\right], \tag{25}$$

where τ denotes a trajectory generated by policy π, r_t is the immediate reward at time step t, γ is the discount factor, and α is the temperature coefficient. The term $-\log\pi(a_t|s_t)$ corresponds to the stochastic policy entropy and is used to encourage the agent to maintain sufficient exploration during training.

In the proposed method, the dual-path channel attention module, adaptive reward feedback, and prioritized experience replay do not change the basic maximum entropy optimization objective of SAC. Instead, they enhance the SAC framework from three complementary perspectives: state representation, reward-weight adjustment, and critical-sample utilization.

4. Proposed DPCA-ARF-PER-SAC Method

After presenting the autonomous navigation task for unmanned aerial vehicles in Section 3, this section introduces an improved Soft Actor–Critic (SAC) method, which integrates dual-pathway channel attention, adaptive reward feedback, and prioritized experience replay. The proposed method does not modify the logarithmic entropy optimization framework of SAC, but enhances the standard SAC algorithm from three aspects: feature representation in the actor network, reward weight scheduling, and the utilization of experience samples.

For clarity, the proposed method in this paper is denoted as DPCA-ARF-PER-SAC. Here, DPCA represents the dual-pathway channel attention module, ARF stands for the adaptive reward feedback mechanism, and PER refers to the prioritized experience replay mechanism. These three components correspond to the three key stages of state feature representation, reward function construction, and experience sample sampling. Through this design, the proposed method aims to enhance the training stability, sample utilization efficiency, and overall navigation performance of SAC in complex autonomous navigation tasks for unmanned aerial vehicles.

4.1. Overall Framework

The overall framework of the proposed DPCA-ARF-PER-SAC method is shown in Figure 2. The method is built upon the actor–critic architecture of SAC and introduces three enhancement mechanisms: dual-path channel attention, adaptive reward feedback, and prioritized experience replay. In this figure, solid arrows denote the main data and computation flow. Dashed arrows denote auxiliary feedback or update flows, and dashed boxes indicate related submodules or sampled data.

During training, the UAV interacts with the AirSim-based environment and obtains a compact one-dimensional state vector. The actor network uses the dual-path channel attention module to recalibrate different state channels and enhance key navigation features, and then outputs continuous actions through a squashed Gaussian policy. The critic network evaluates the sampled state–action pairs and estimates the corresponding Q-values.

The adaptive reward feedback mechanism dynamically adjusts the reward weights according to the training progress, enabling the agent to focus more on target approaching

in the early stage and gradually strengthen obstacle avoidance and stable control in the later stage. Meanwhile, prioritized experience replay improves the utilization of critical samples, such as collision, near-obstacle, and successful-reaching transitions.

The proposed framework does not change the maximum entropy optimization objective of SAC. Instead, it enhances SAC from three complementary aspects: state feature representation, reward guidance, and critical-sample utilization, thereby improving training stability and navigation performance in complex UAV autonomous navigation tasks.

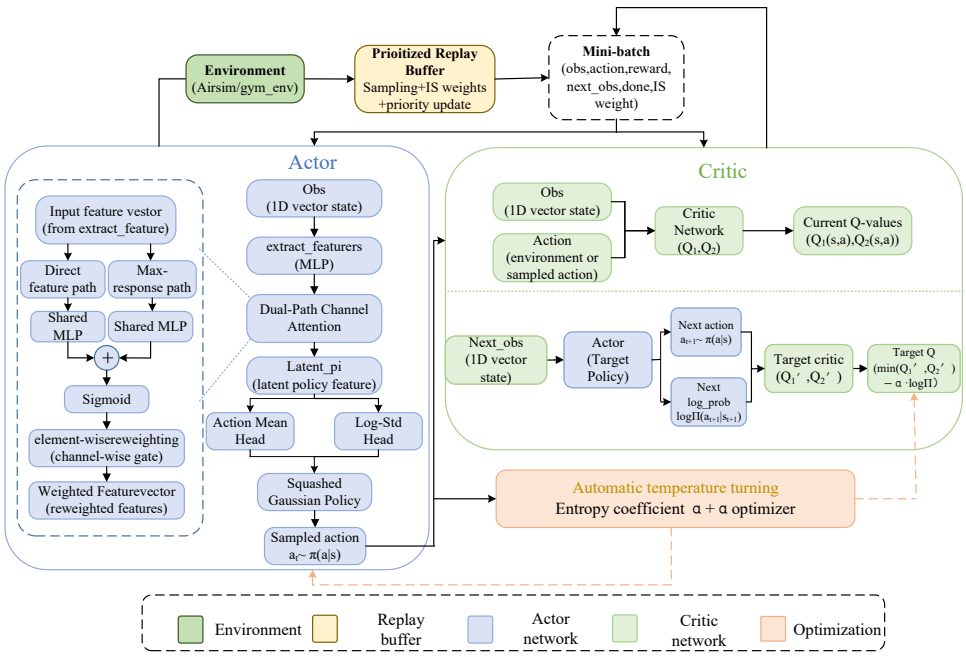


Figure 2. Overall framework of the proposed DPCA-ARF-PER-SAC method.

4.2. Dual-Path Channel Attention Module

4.2.1. Design Motivation for Dual-Path Channel Attention

In the autonomous navigation tasks of unmanned aerial vehicles in complex and unknown environments, the state inputs usually include information such as local obstacle risks, target geometric relationships, height deviations, and yaw errors. The importance of different features varies at different decision-making stages. If all state features are directly input equally into the actor network, it is likely to result in insufficient expression of key navigation information, thereby affecting the judgment ability of the policy network for high-value actions.

To address this issue, this paper introduces a dual-path channel attention (DPCA) module into the actor network. This module is specifically designed for the compact one-dimensional state vectors used in this study. Unlike the original convolutional block attention module (CBAM), which mainly targets two-dimensional convolutional feature maps and includes both channel attention and spatial attention, the proposed DPCA module focuses on the allocation of channel-level importance between the state features. Therefore, it does not directly adopt the complete CBAM structure but uses a lightweight channel recalibration mechanism suitable for one-dimensional unmanned aerial vehicle navigation states.

Specifically, the proposed DPCA module constructs two complementary paths: a direct feature path and a maximum response path. The direct feature path retains the complete state feature distribution, while the maximum response path highlights the most significant responses in the current state. By combining these two paths, the actor network can adaptively enhance the key channels related to navigation and suppress relatively

redundant feature responses. Figure 3 shows the structure of the proposed dual-path channel attention module.

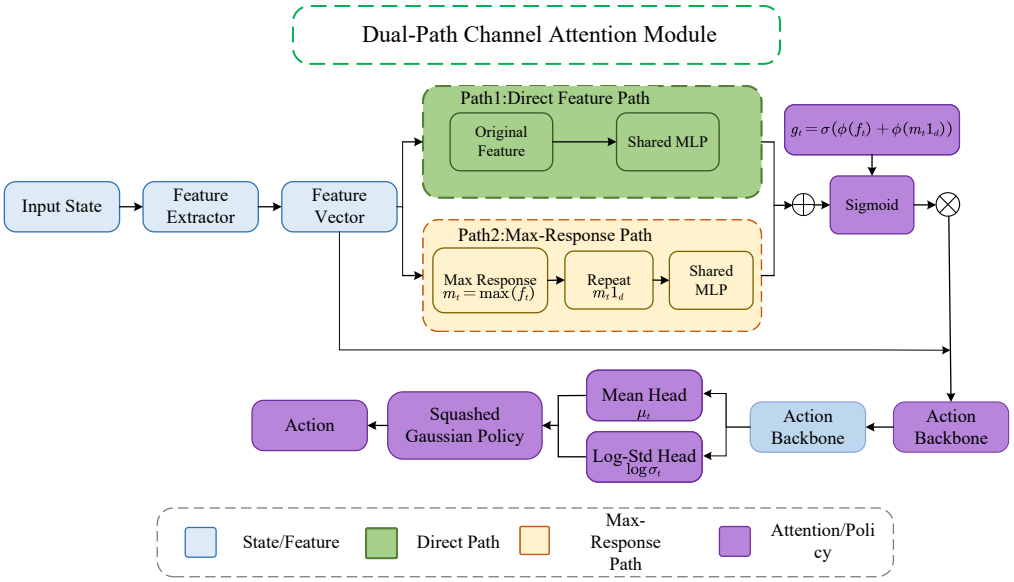


Figure 3. Structure of the proposed dual-path channel attention module.

4.2.2. Dual-Path Channel Attention

Let the state feature extracted by the actor feature extraction layer be denoted as

$$\mathbf{f}_t = E_{\phi_e}(s_t), \tag{26}$$

where $E_{\phi_e}(\cdot)$ denotes the front-end feature extractor of the actor network, ϕ_e represents its parameters, and $\mathbf{f}_t$ denotes the feature representation at time step t. This feature vector contains information related to local obstacle risk, target distance, altitude deviation, and heading error, and serves as the basis for subsequent continuous action generation.

To utilize both the complete state feature distribution and the most salient response in the current state, two parallel paths are constructed. The first path is the direct feature path, which preserves the original feature vector $\mathbf{f}_t$ to maintain the complete state feature distribution. The second path is the max-response path, which highlights the most significant feature response in the current state.

First, the maximum response of $\mathbf{f}_t$ is extracted as

$$m_t = \max_{j=1,\ldots,d_f} f_{t,j}, \tag{27}$$

where d_f denotes the dimension of the feature vector, and $f_{t,j}$ denotes the j-th element of $\mathbf{f}_t$. Then, the scalar maximum response is expanded to the same dimension as the input feature vector:

$$\hat{\mathbf{f}}_t = m_t \mathbf{1}_{d_f}, \tag{28}$$

where $\mathbf{1}_{d_f}$ denotes a d_f-dimensional vector whose elements are all equal to one, and $\hat{\mathbf{f}}_t$ denotes the expanded feature vector of the max-response path. This operation broadcasts the most salient response to all feature channels, which helps the actor network identify the overall importance distribution of the current state.

After that, the direct feature path and the max-response path are transformed by a shared mapping function $\Phi(\cdot)$, and the channel attention weight is obtained through a sigmoid activation function:

$$\mathbf{g}_t = \sigma\Big(\Phi(\mathbf{f}_t) + \Phi(\hat{\mathbf{f}}_t)\Big), \tag{29}$$

where $\sigma(\cdot)$ denotes the sigmoid activation function, and $\mathbf{g}_t$ is the channel attention weight vector. Each element of $\mathbf{g}_t$ lies in the interval $(0, 1)$, indicating the importance of the corresponding feature channel at the current decision-making step. A larger weight means that the corresponding feature should be retained and enhanced, while a smaller weight indicates that the feature is relatively less important at the current time step.

Finally, the generated channel attention weights are applied to the original input feature vector, and the enhanced feature representation is obtained as

$$\tilde{\mathbf{f}}_t = \mathbf{f}_t \odot \mathbf{g}_t, \tag{30}$$

where $\odot$ denotes the Hadamard product, and $\tilde{\mathbf{f}}_t$ denotes the feature vector enhanced by the dual-path channel attention module. Equation (30) shows that the proposed module does not change the feature dimension. Instead, it applies different gating coefficients to different feature channels, thereby adaptively enhancing key navigation information and moderately suppressing redundant features.

4.2.3. Action Generation

After being processed by the DPCA module, the enhanced feature vector $\tilde{\mathbf{f}}_t$ is fed into the main branch of the actor network to further generate the latent policy feature:

$$\mathbf{h}_t = F_\phi(\tilde{\mathbf{f}}_t), \tag{31}$$

where $F_\phi(\cdot)$ denotes the main multilayer perceptron of the actor network, and $\mathbf{h}_t$ denotes the latent feature used to generate the action distribution.

Then, two independent output heads are used to generate the mean and logarithmic standard deviation of the Gaussian action distribution:

$$\boldsymbol{\mu}_t = W_\mu \mathbf{h}_t + \mathbf{b}_\mu, \tag{32}$$

$$\log \boldsymbol{\sigma}_t = W_\sigma \mathbf{h}_t + \mathbf{b}_\sigma, \tag{33}$$

where $\boldsymbol{\mu}_t$ and $\boldsymbol{\sigma}_t$ denote the mean and standard deviation of the action distribution, respectively. Based on the reparameterization trick, the pre-squashed action is sampled as

$$\mathbf{u}_t = \boldsymbol{\mu}_t + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{34}$$

where $\odot$ denotes element-wise multiplication, and $\boldsymbol{\epsilon}_t$ denotes a noise vector sampled from a standard multivariate Gaussian distribution.

Finally, the sampled action is squashed by the hyperbolic tangent function to ensure that the normalized action remains within a bounded range:

$$\bar{\mathbf{a}}_t = \tanh(\mathbf{u}_t). \tag{35}$$

If the environment uses normalized actions, $\bar{\mathbf{a}}_t$ is directly used as the final action. Otherwise, it is further rescaled to the actual action range:

$$\mathbf{a}_t = \mathbf{a}_{\min} + \frac{\bar{\mathbf{a}}_t + \mathbf{1}}{2} \odot (\mathbf{a}_{\max} - \mathbf{a}_{\min}). \tag{36}$$

Therefore, the DPCA module only enhances the state feature representation in the actor network. It does not change the Gaussian policy construction, the reparameterized sampling process, or the maximum entropy optimization objective of SAC.

4.3. Adaptive Reward Feedback Mechanism

4.3.1. Design Motivation for Adaptive Reward Feedback

In the autonomous navigation task of unmanned aerial vehicles (UAVs), the reward function needs to take into account multiple objectives, such as target approach, obstacle avoidance, altitude control, and yaw correction simultaneously. However, the importance of each objective varies across different training stages. In the early training stage, the agent has not yet developed basic navigation capabilities, and should place more emphasis on target approach to guide the UAV to learn to fly towards the target; in the later training stage, the agent has already acquired certain target search capabilities, at which point the importance of obstacle avoidance safety, altitude constraints, and yaw correction should be gradually increased to further transform the strategy towards safer, more stable, and more efficient flight behaviors.

Therefore, this paper designs an adaptive reward feedback mechanism, representing the reward weights as a function of the training progress, so that the reward guidance can be dynamically adjusted according to the training stage.

4.3.2. Training Progress Modeling

To dynamically adjust the reward weights during training, the training progress is first normalized. Let n_t denote the current training step and N_{train} denote the total number of training steps. The normalized training progress is defined as

$$\rho_t = \min\left(\frac{n_t}{N_{train}}, 1\right), \tag{37}$$

where $\rho_t \in [0, 1]$ indicates the current training stage. When ρ_t is close to 0, the agent is in the early training stage. When ρ_t approaches 1, the training process has entered the later stage. This variable is used to characterize the current training phase and serves as the basis for the subsequent adaptive reward-weight scheduling.

4.3.3. Dynamic Weight Scheduling

Based on the reward decomposition defined in Section 3, the weights of different reward components are further designed as functions of the normalized training progress. Specifically, the adaptive weight of each reward term is defined as

$$\lambda_i(\rho_t) = \lambda_i^0 + k_i \rho_t, \quad i \in \{g, o, z, \psi\}, \tag{38}$$

where λ_i^0 denotes the initial weight of the i-th reward component, and k_i is the scheduling coefficient that controls the variation in the corresponding weight with the training progress.

To reflect the training principle of emphasizing target reaching in the early stage and safety and control stability in the later stage, the scheduling coefficients are set as $k_g < 0$, $k_o > 0$, $k_z > 0$, and $k_\psi > 0$. As the training progresses, the target-approaching weight $\lambda_g(\rho_t)$ gradually decreases, while the obstacle-avoidance weight $\lambda_o(\rho_t)$, altitude-constraint weight $\lambda_z(\rho_t)$, and heading-correction weight $\lambda_\psi(\rho_t)$ gradually increase. In this way, the agent is encouraged to first acquire basic target-reaching ability in the early training stage, and then further strengthen obstacle avoidance, vertical control stability, and heading correction in the middle and later training stages.

After applying the adaptive reward feedback mechanism, the immediate reward can be written as

$$r_t = \lambda_g(\rho_t) r_t^g + \lambda_o(\rho_t) r_t^o + \lambda_z(\rho_t) r_t^z + \lambda_\psi(\rho_t) r_t^\psi + r_t^{term}. \tag{39}$$

The definitions of the reward components are consistent with those in Section 3. In the 2D SimpleAvoid scenario, altitude control is not involved, and thus the altitude-constraint reward r_t^z is removed. The remaining reward terms are kept unchanged.

4.4. Prioritized Experience Replay

In the autonomous navigation task of unmanned aerial vehicles, different experience samples have varying values for strategy learning. Samples that involve approaching obstacles, colliding, crossing boundaries, or approaching the target area typically contain more crucial decision-making information, while ordinary flight samples contribute relatively less to strategy updates. If uniform random sampling is adopted, the critical samples may be diluted by a large number of ordinary samples, thereby reducing the efficiency of sample utilization. To address this issue, this paper introduces a priority experience replay mechanism, which assigns priorities to samples based on TD errors, enabling the critic network to pay more attention to the experience samples with larger estimation errors or higher decision values.

For the j-th transition in the replay buffer, its TD error is defined as

$$\delta_j = y_j - Q_\theta(s_j, a_j), \tag{40}$$

where y_j denotes the soft Bellman target, and $Q_\theta(s_j, a_j)$ denotes the action-value estimate of the current critic network for the sampled state–action pair. The priority of the sample is then defined as

$$p_j = |\delta_j| + \varepsilon, \tag{41}$$

where ε is a small positive constant used to avoid zero priority.

Based on the sample priority, the probability of sampling the j-th transition is defined as

$$P(j) = \frac{p_j^\kappa}{\sum_l p_l^\kappa}, \tag{42}$$

where κ is the priority exponent that controls the influence of priority on the sampling probability. When $\kappa = 0$, PER degenerates into uniform random sampling. As κ increases, samples with higher priorities are more likely to be selected.

Since non-uniform sampling changes the original experience distribution, importance-sampling weights are introduced to correct the sampling bias:

$$w_j = \frac{(NP(j))^{-\beta}}{\max_l (NP(l))^{-\beta}}, \tag{43}$$

where N denotes the size of the replay buffer, and β denotes the importance-sampling correction coefficient. The normalization term is used to prevent excessively large weights and improve training stability.

After the critic network is updated, the priority of the sampled transition is recalculated according to the updated TD error:

$$p_j \leftarrow |\delta_j| + \varepsilon. \tag{44}$$

Through the closed-loop process of prioritized sampling, network updating, and priority updating, the replay buffer can dynamically adjust the sampling probability of critical experiences during training. This mechanism improves the utilization of high-value samples and further enhances the training stability of the proposed DPCA-ARF-PER-SAC method.

4.5. SAC-Based Parameter Update Procedure

After introducing DPCA, ARF, and PER, the overall training process of the proposed method is still based on the actor–critic framework of SAC. During interaction with the environment, each transition is stored in the replay buffer $\mathcal{D}$ in the form of $(s_t, a_t, r_t, s_{t+1}, d_t)$. In the parameter update stage, a mini-batch of B transitions is sampled from $\mathcal{D}$. Since the sampled transitions do not necessarily preserve their original temporal order, the subscript b is used to denote the b-th sample in the mini-batch.

For the b-th transition $(s_b, a_b, r_b, s'_b, d_b)$, the next action is sampled from the current policy as

$$a'_b \sim \pi_\phi(\cdot|s'_b). \tag{45}$$

The corresponding soft Bellman target is defined as

$$y_b = r_b + \gamma(1 - d_b)\left[\min_{i=1,2} Q_{\bar{\theta}_i}(s'_b, a'_b) - \alpha \log \pi_\phi(a'_b|s'_b)\right], \tag{46}$$

where $Q_{\bar{\theta}_i}$ denotes the i-th target critic network, α denotes the temperature coefficient, and $(1 - d_b)$ is used to mask the future value estimation when the transition reaches a terminal state. This target considers both the estimated future action value and the entropy term, thereby encouraging the policy to maintain sufficient exploration during training.

After obtaining the target value y_b, the standard SAC critic loss for the i-th critic network is written as

$$\mathcal{L}_{Q_i}(\theta_i) = \frac{1}{B}\sum_{b=1}^{B}\left(Q_{\theta_i}(s_b, a_b) - y_b\right)^2, \quad i = 1, 2, \tag{47}$$

where $Q_{\theta_i}(s_b, a_b)$ denotes the action-value estimate of the current critic network for the sampled state–action pair.

When PER is introduced, the importance-sampling weight w_b is incorporated into the critic loss to correct the sampling bias caused by non-uniform sampling. The weighted critic loss is defined as

$$\mathcal{L}_{Q_i}^{PER}(\theta_i) = \frac{1}{B}\sum_{b=1}^{B} w_b\left(Q_{\theta_i}(s_b, a_b) - y_b\right)^2, \quad i = 1, 2. \tag{48}$$

When $w_b = 1$, Equation (48) degenerates into the standard SAC critic loss. After the critic networks are updated, the TD error used for priority updating is calculated as

$$\delta_b = y_b - \min_{i=1,2} Q_{\theta_i}(s_b, a_b), \tag{49}$$

and the corresponding sample priority is updated by

$$p_b \leftarrow |\delta_b| + \varepsilon, \tag{50}$$

where ε is a small positive constant used to avoid zero priority.

After the critic update, the actor network is updated using the current states in the mini-batch. For each state s_b, a new action is sampled from the current actor policy:

$$a_b^\pi \sim \pi_\phi(\cdot|s_b). \tag{51}$$

The actor loss is defined as

$$\mathcal{L}_\pi(\phi) = \frac{1}{B}\sum_{b=1}^{B}\left[\alpha \log \pi_\phi(a_b^\pi|s_b) - \min_{i=1,2} Q_{\theta_i}(s_b, a_b^\pi)\right]. \tag{52}$$

Here, a_b^π denotes the action newly sampled by the current actor, rather than the stored action a_b in the replay buffer. In the proposed method, the actor takes the state features enhanced by the DPCA module as input. Therefore, the policy update is performed on the basis of a more discriminative state representation.

To adaptively adjust the exploration strength, the automatic temperature tuning mechanism of SAC is retained. The temperature loss is defined as

$$\mathcal{L}_\alpha = -\frac{1}{B}\sum_{b=1}^{B}\alpha\left(\log \pi_\phi(a_b^\pi|s_b) + \bar{\mathcal{H}}\right), \tag{53}$$

where $\bar{\mathcal{H}}$ denotes the target entropy. This mechanism encourages the policy entropy to approach the expected level, thereby adaptively balancing exploration and exploitation during training.

Finally, the target critic networks are updated by soft updating:

$$\bar{\theta}_i \leftarrow \tau\theta_i + (1-\tau)\bar{\theta}_i, \quad i = 1, 2, \tag{54}$$

where τ denotes the soft update coefficient. Through the above update process, DPCA, ARF, and PER enhance SAC from the aspects of state feature representation, reward guidance, and critical-sample utilization, respectively, while the overall optimization objective still follows the maximum entropy reinforcement learning framework of SAC.

5. Experiments

To verify the effectiveness of the proposed DPCA-ARF-PER-SAC method in the autonomous navigation task of unmanned aerial vehicles, we conducted experiments in the 2D SimpleAvoid and 3D NH_center scenarios. The 2D SimpleAvoid scenario was used for module-level ablation analysis, focusing on the effects of PER, ARF, and their combined enhancements on training stability, obstacle avoidance safety, and path efficiency. The methods compared in this scenario include SAC, PER-SAC, ARF-SAC, ARF-PER-SAC, and the proposed DPCA-ARF-PER-SAC method in this paper.

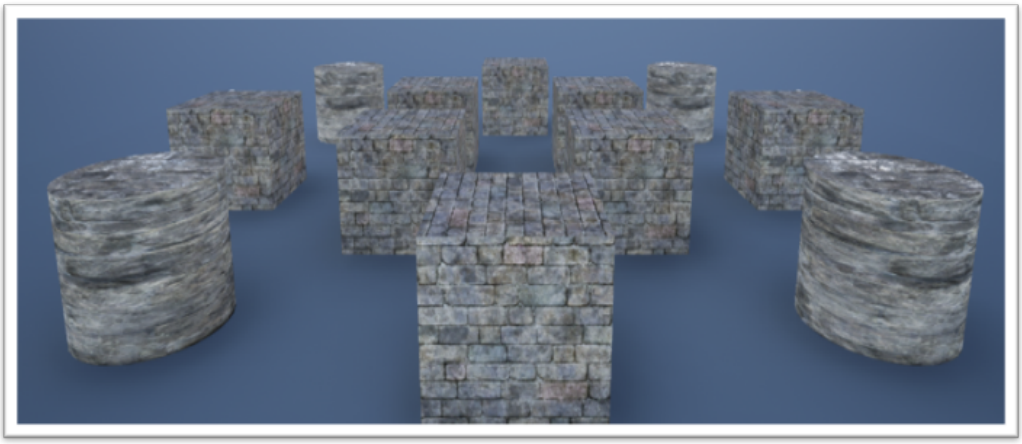
The 3D NH_center scenario was used to conduct a main performance comparison in a more complex three-dimensional environment. In this scenario, TD3, SAC, the soft actor–critic enhanced by the attention mechanism (AM-SAC), and the proposed DPCA-ARF-PER-SAC method were compared to evaluate the overall navigation performance of the proposed method. The experiments were mainly analyzed from three aspects: the training process, 50 independent test results, and typical flight trajectories.

5.1. Experimental Settings

The experiments are conducted based on the AirSim simulation platform and a deep reinforcement learning framework. The UAV agent outputs continuous control actions according to the state vector returned by the environment and learns an autonomous navigation policy through interaction with the environment. The state input consists of local obstacle-risk features and target-related navigation features, while the action output includes continuous control variables such as horizontal velocity, vertical velocity, and yaw rate.

Two experimental scenarios are used in this study, namely the 2D SimpleAvoid scenario and the 3D NH_center scenario, as shown in Figure 4. The 2D SimpleAvoid scenario is used for module-level ablation experiments to analyze the effects of PER, ARF, their com-

bined enhancement, and the complete DPCA-ARF-PER-SAC framework on basic obstacle avoidance capability and path efficiency. The 3D NH_center scenario is used as the main experimental environment to further evaluate the comprehensive navigation performance of the proposed method in a complex three-dimensional environment.



(**a**)

(**b**)

Figure 4. Experimental scenarios used in this study. (**a**) 2D SimpleAvoid scenario. (**b**) 3D NH_center scenario.

In the 2D SimpleAvoid ablation experiments, the compared methods include SAC, PER-SAC, ARF-SAC, ARF-PER-SAC, and the proposed DPCA-ARF-PER-SAC method. In the main experiment conducted in the 3D NH_center scenario, Deep Deterministic Policy Gradient (DDPG), Twin Delayed Deep Deterministic Policy Gradient (TD3), SAC, attention-mechanism-enhanced Soft Actor–Critic (AM-SAC), and the proposed DPCA-ARF-PER-SAC method are compared. DDPG and TD3 are introduced as representative deterministic continuous-control algorithms, SAC is used as the baseline maximum-entropy reinforcement learning method, and AM-SAC is used as a representative attention-based SAC variant. The main experimental parameters are listed in Table 1.

5.2. Evaluation Metrics

To comprehensively evaluate the performance of different methods in the autonomous navigation task of unmanned aerial vehicles (UAVs), this paper adopts success rate, collision rate, normalized return, and average round length as evaluation indicators. Among them,

success rate represents the proportion of rounds in which the UAV successfully reaches the target area, which is used to measure the task completion capability; collision rate represents the proportion of rounds in which collisions occur, which is used to evaluate the navigation safety; normalized return reflects the comprehensive performance of the intelligent agent in terms of approaching the target, obstacle avoidance, and control constraints; average round length represents the average number of flight steps in a single round. In cases where the success rates are similar, a shorter average round length usually indicates higher path efficiency.

In the subsequent experimental analysis, the 2D SimpleAvoid scenario was mainly used to observe the impact of each improvement module on the stability of training and the efficiency of the path. The 3D NH_center scenario focused on verifying the comprehensive navigation performance of the proposed method in complex three-dimensional environments.

The success rate is defined as

$$SR = \frac{N_{\text{success}}}{N_{\text{test}}}, \tag{55}$$

where N_{success} denotes the number of successful episodes, and N_{test} denotes the total number of test episodes.

The collision rate is defined as

$$CR = \frac{N_{\text{collision}}}{N_{\text{test}}}, \tag{56}$$

where $N_{\text{collision}}$ denotes the number of episodes in which a collision occurs.

The normalized return is calculated as

$$R_{\text{norm}} = \frac{1}{N} \sum_{i=1}^{N} R_i, \tag{57}$$

where R_i denotes the normalized cumulative return of the i-th episode, and N denotes the number of episodes used for calculating the normalized return.

The average episode length is defined as

$$L_{\text{avg}} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} L_i, \tag{58}$$

where L_i denotes the number of steps in the i-th test episode. In the subsequent experimental analysis, the 2D SimpleAvoid scenario is mainly used to evaluate the influence of different improved modules on training stability and path efficiency, while the 3D NH_center scenario is used to further verify the comprehensive navigation performance of the proposed method in a complex three-dimensional environment.

5.3. Ablation Study in the 2D SimpleAvoid Scenario

To verify the influence of the proposed enhancement mechanisms on policy learning, an ablation study is first conducted in the 2D SimpleAvoid scenario. This scenario is mainly used to evaluate the effects of prioritized experience replay, adaptive reward feedback, their combined enhancement, and the complete DPCA-ARF-PER-SAC framework on convergence speed, obstacle avoidance safety, and path efficiency. The compared methods include SAC, PER-SAC, ARF-SAC, ARF-PER-SAC, and the complete DPCA-ARF-PER-SAC method. The training performance curves of different methods are shown in Figure 5.

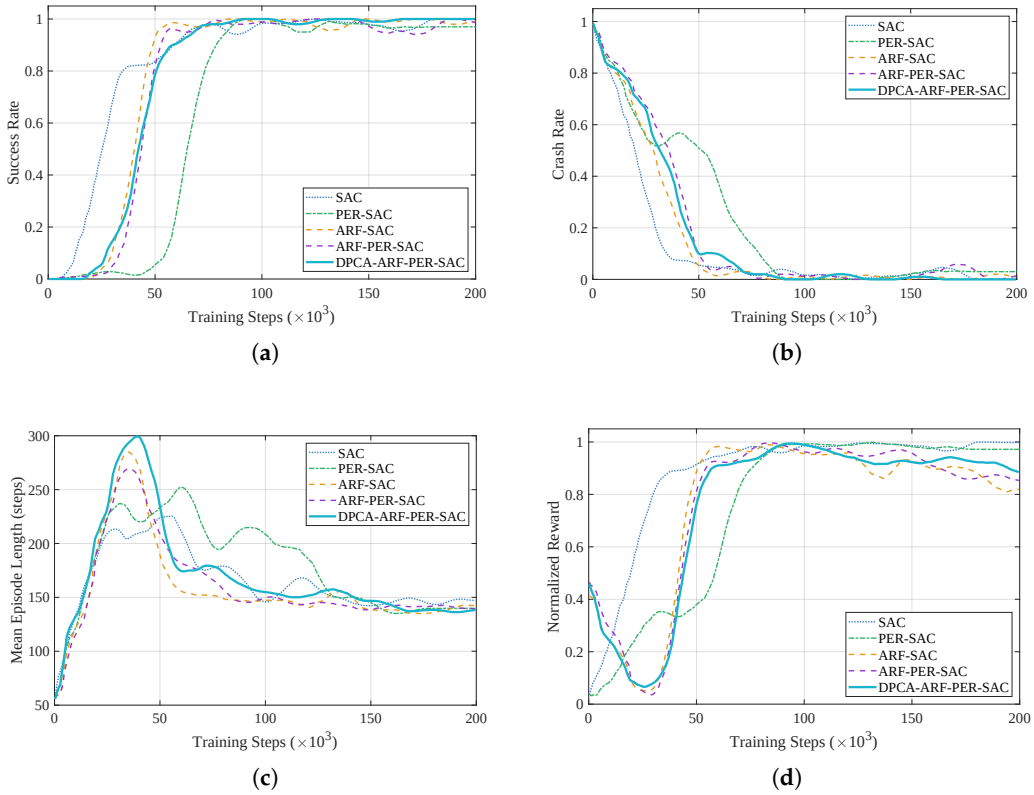


Figure 5. Training performance comparison of different methods in the 2D SimpleAvoid scenario. (**a**) Success rate. (**b**) Collision rate. (**c**) Average episode length. (**d**) Normalized return.

As shown in Figure 5a,b, all methods gradually improve the success rate and reduce the collision rate during training. The baseline SAC converges relatively quickly in the 2D SimpleAvoid scenario, indicating that this scenario has moderate difficulty and can be used as a suitable environment for module-level ablation analysis. PER-SAC improves the utilization of critical transition samples, while ARF-SAC provides adaptive reward guidance at different training stages. ARF-PER-SAC further combines reward guidance and prioritized sampling. The complete DPCA-ARF-PER-SAC method finally achieves a success rate close to 1.00 and reduces the collision rate to nearly 0, indicating that the proposed mechanisms can jointly improve training stability and obstacle avoidance safety.

Figure 5c shows the variation in average episode length. At the early training stage, the episode length of some methods increases because the UAV gradually shifts from immediate failure caused by frequent collisions to longer exploration. As training proceeds, the episode length decreases, indicating that the agent learns a more efficient target-reaching strategy. The complete DPCA-ARF-PER-SAC method maintains a relatively low episode length in the later training stage, suggesting that it can improve path efficiency while maintaining safe navigation.

Figure 5d shows the normalized return curves. It should be noted that the normalized return is mainly used to analyze the training tendency rather than as the only criterion for final performance comparison, because the ARF mechanism changes the reward composition during training. In the early stage, ARF assigns a larger weight to the target-reaching reward, which helps the agent quickly learn effective target-approaching behavior. In the later stage, the weights of obstacle avoidance, heading correction, and other safety-related terms are gradually increased. Therefore, the normalized return may fluctuate or become less dominant, but this does not necessarily indicate policy degradation. The final perfor-

mance should be evaluated together with the success rate, collision rate, average episode length, and trajectory behavior.

To further evaluate the final performance after training, each method is tested independently for 50 episodes. The statistical results are shown in Figure 6.

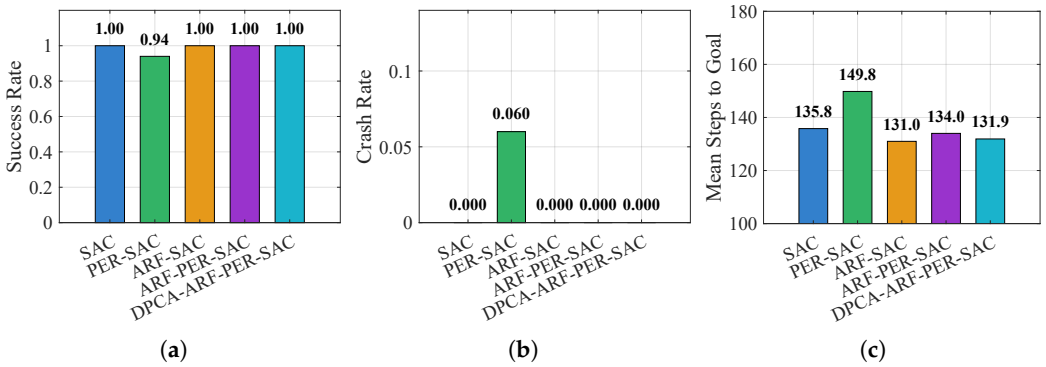


Figure 6. Statistical results of 50 independent tests in the 2D SimpleAvoid scenario. (**a**) Success rate. (**b**) Collision rate. (**c**) Average episode length.

As shown in Figure 6, SAC, ARF-SAC, ARF-PER-SAC, and the complete DPCA-ARF-PER-SAC method all achieve a success rate of 1.00 and a collision rate of 0.00, indicating that these methods can complete the basic navigation task after sufficient training in the relatively simple 2D scenario. In contrast, PER-SAC obtains a success rate of 0.94 and a collision rate of 0.06, suggesting that prioritized replay alone may be less effective in this simple environment and may introduce additional sampling fluctuations during training.

In terms of navigation efficiency, ARF-SAC achieves the shortest average episode length of 131.0 steps, while the complete DPCA-ARF-PER-SAC method obtains an average episode length of 131.9 steps, which is close to ARF-SAC and lower than the baseline SAC with 135.8 steps. ARF-PER-SAC obtains an average episode length of 134.0 steps, also showing improved path efficiency compared with SAC. PER-SAC has the longest average episode length of 149.8 steps, which further indicates that PER alone does not necessarily improve the final navigation efficiency in the simple 2D scenario.

Overall, the 50-episode test results show that ARF contributes to improving path efficiency, while the complete DPCA-ARF-PER-SAC method maintains a high success rate, a zero collision rate, and competitive path efficiency. Although the complete method does not achieve the shortest average episode length, it shows more balanced performance when considering task completion, obstacle avoidance safety, training-curve stability, and final navigation efficiency.

To further observe the actual flight behavior of different methods in the 2D SimpleAvoid scenario, typical flight trajectories are shown in Figure 7.

As shown in Figure 7, all methods can generate obstacle-avoidance trajectories from the start point to the target region after training. However, some methods still show local oscillations, detours, or collision trajectories in certain episodes. In comparison, the complete DPCA-ARF-PER-SAC method produces more balanced successful trajectories with fewer collision trajectories, indicating better flight stability in actual navigation behavior.

Overall, the 2D SimpleAvoid ablation results show that the baseline SAC can already achieve relatively good performance in the simple two-dimensional scenario. Therefore, the improvement brought by a single mechanism is not always obvious. However, the complete DPCA-ARF-PER-SAC method can maintain a high success rate and a low collision rate while improving path efficiency, suggesting that the joint use of DPCA, ARF, and PER contributes to training stability and navigation efficiency.

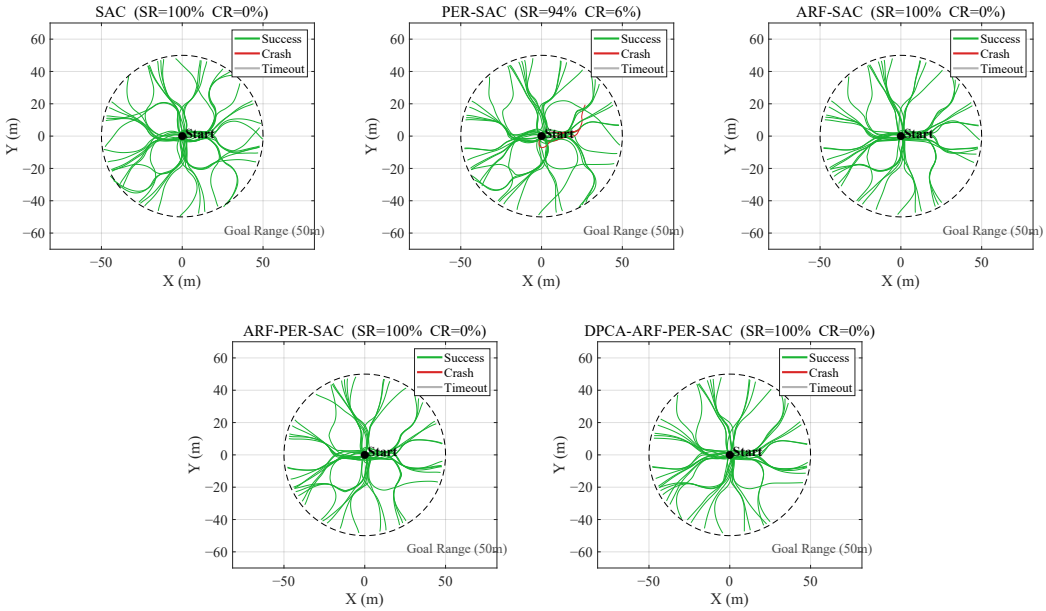


Figure 7. Typical flight trajectory comparison of different methods in the 2D SimpleAvoid scenario.

5.4. Main Results in the 3D NH_center Scenario

After completing the ablation study in the 2D SimpleAvoid scenario, the main experiments are further conducted in the 3D NH_center scenario. Compared with the 2D scenario, the 3D NH_center environment has a more complex spatial structure and higher control difficulty. The UAV needs not only to approach the target and avoid obstacles, but also to consider altitude control, heading correction, and continuous three-dimensional action outputs. Therefore, this scenario is more suitable for evaluating the comprehensive navigation performance of different methods in complex environments. The compared methods include TD3, SAC, AM-SAC, and the proposed DPCA-ARF-PER-SAC method. The training performance curves of different methods are shown in Figure 8.

As shown in Figure 8a,b, the improvement of success rate and the reduction of collision rate in the 3D NH_center scenario are more difficult and more fluctuating than those in the 2D SimpleAvoid scenario. This indicates that the complex three-dimensional environment imposes higher requirements on policy learning and safety control. TD3 can learn certain target-approaching behaviors, but its training stability may be limited by insufficient exploration in complex environments. The baseline SAC benefits from the maximum entropy framework and shows better exploration ability than deterministic policy methods. AM-SAC introduces an attention mechanism into SAC and is used as a representative attention-based SAC variant. In comparison, the proposed DPCA-ARF-PER-SAC method achieves a more balanced improvement in success rate and collision reduction during training, indicating that the joint enhancement of state representation, reward guidance, and prioritized sample utilization can improve learning efficiency and obstacle avoidance behavior in complex three-dimensional environments.

Figure 8c shows the average episode length during training. In the early stage, some methods have relatively short episode lengths because the UAV may terminate early due to collisions, boundary violations, or failed exploration. As training progresses, the episode length increases with longer exploration and then decreases when the agent learns more efficient target-reaching behaviors. The proposed DPCA-ARF-PER-SAC method shows a clear decrease in average episode length in the middle and later training stages, indicating that the proposed framework can improve path efficiency while maintaining navigation safety.

Figure 8d shows the normalized return curves. It should be noted that the normalized return is mainly used to observe the training tendency rather than as the only criterion for final performance comparison. Since the ARF mechanism dynamically adjusts the reward weights during training, the reward objective gradually shifts from rapid target approaching in the early stage to safer and more stable navigation in the later stage. Specifically, the target-reaching reward has a larger influence in the early training stage, which helps the UAV quickly learn target-approaching behavior. In the later training stage, the weights of obstacle avoidance, heading correction, altitude control, and action smoothness are gradually increased. Therefore, the normalized return of DPCA-ARF-PER-SAC may decrease or become less dominant in the later stage. This phenomenon does not necessarily indicate policy failure, but reflects the change in reward guidance from target approaching to safety and stability.

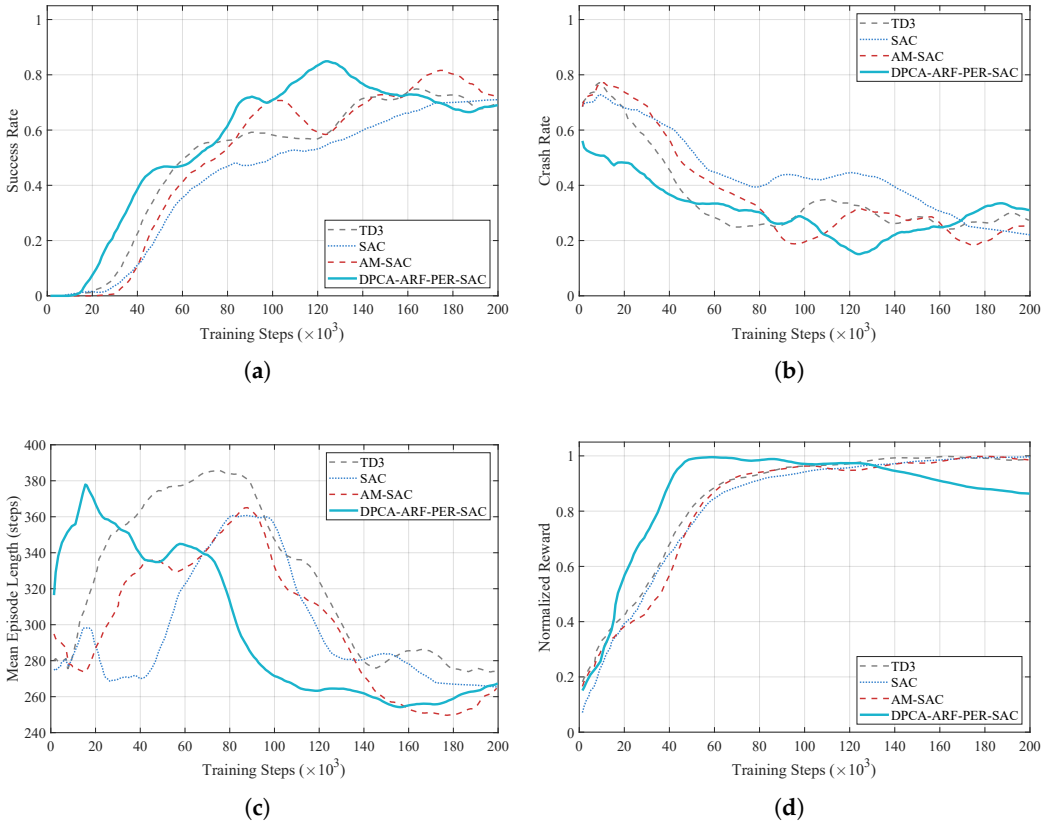


Figure 8. Training performance comparison of different methods in the 3D NH_center scenario. (**a**) Success rate. (**b**) Collision rate. (**c**) Average episode length. (**d**) Normalized return.

It is also observed that AM-SAC does not consistently outperform the baseline SAC in the 3D scenario. This indicates that introducing an attention mechanism alone cannot guarantee stable improvement in complex three-dimensional navigation tasks. A possible reason is that the state input dimension in this work is relatively compact, and the baseline multilayer perceptron already has a certain feature representation capability. Additional attention weighting may introduce fluctuations during training if it is not supported by appropriate reward guidance and sample utilization. Therefore, attention enhancement alone is not regarded as an independent source of stable performance improvement in this work. Instead, the proposed method combines DPCA with ARF and PER to improve feature representation, reward guidance, and critical-sample utilization simultaneously.

To further observe the actual flight behavior of different methods in three-dimensional space, typical flight trajectories are shown in Figure 9.

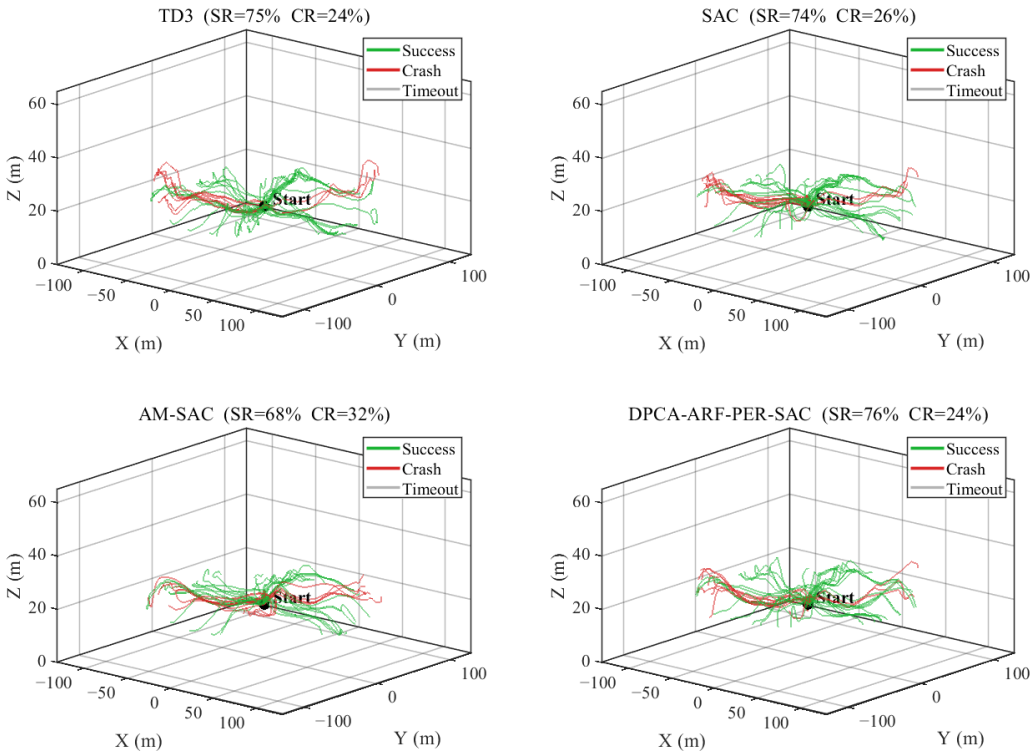


Figure 9. Typical flight trajectory comparison of different methods in the 3D NH_center scenario.

As shown in Figure 9, TD3 and the baseline SAC can generate basic target-approaching trajectories, but some trajectories still show detours, local oscillations, or premature termination. AM-SAC also shows certain instability in trajectory behavior, indicating that attention enhancement alone is insufficient to ensure stable navigation in the complex 3D environment. In comparison, the proposed DPCA-ARF-PER-SAC method produces more concentrated successful trajectories and fewer failed trajectories, indicating that it can achieve more stable actual flight behavior in the complex three-dimensional environment.

To quantitatively evaluate the final model performance, each trained method is independently tested for 50 episodes. The statistical results are shown in Figure 10.

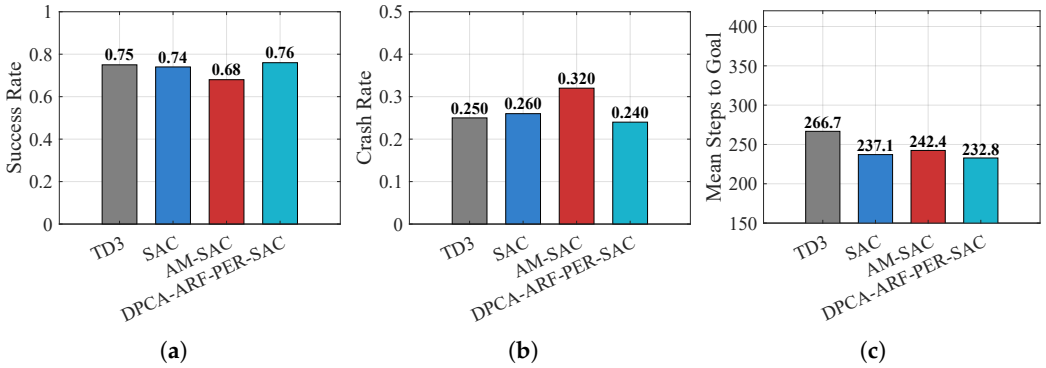


Figure 10. Statistical results of 50 independent tests in the 3D NH_center scenario. (**a**) Success rate. (**b**) Collision rate. (**c**) Average episode length.

As shown in Figure 10, the proposed DPCA-ARF-PER-SAC method achieves a success rate of 0.76, a collision rate of 0.24, and an average episode length of 232.8 steps. Compared with the baseline SAC, which obtains a success rate of 0.74, a collision rate of 0.26, and an average episode length of 237.1 steps, the proposed method improves the success rate by 2 percentage points, reduces the collision rate by 2 percentage points, and decreases the average episode length by 4.3 steps. Compared with TD3, the proposed method achieves a

slightly higher success rate and lower collision rate, while reducing the average episode length from 266.7 to 232.8 steps.

In contrast, AM-SAC achieves a success rate of 0.68, a collision rate of 0.32, and an average episode length of 242.4 steps, which does not outperform the baseline SAC. This further indicates that attention enhancement alone has limited effect in the complex 3D scenario. Therefore, the advantage of the proposed DPCA-ARF-PER-SAC method does not come from simply introducing an attention mechanism, but from the joint effect of DPCA, ARF, and PER.

These results indicate that the proposed method achieves a moderate improvement over the baseline SAC rather than a dramatic performance gain. This is reasonable because SAC is already a strong maximum-entropy reinforcement learning baseline for continuous control tasks. Under the same state space, action space, and training setting, the proposed method aims to further improve the overall navigation behavior on the basis of SAC. Therefore, the advantage of DPCA-ARF-PER-SAC is mainly reflected in the more balanced performance among success rate, collision rate, average episode length, training stability, and trajectory behavior. Overall, the 3D NH_center experiments show that complex three-dimensional navigation requires a balance among task completion, flight safety, and path efficiency, and the proposed method achieves the best overall balance among the compared methods in the final test results.

6. Discussion and Conclusions

6.1. Discussion on Statistical Reliability, Computational Cost, Transferability, and Limitations

Although the proposed DPCA-ARF-PER-SAC method achieves the best overall balance among success rate, collision rate, and average episode length in the 3D NH_center scenario, the performance improvement over the baseline SAC is moderate rather than dramatic. Compared with SAC, the proposed method improves the success rate by 2 percentage points, reduces the collision rate by 2 percentage points, and decreases the average episode length by 4.3 steps. Therefore, the advantage of the proposed method should not be interpreted as a large improvement in a single metric, but as a more balanced improvement in task completion, navigation safety, path efficiency, and training stability.

The statistical results reported in this study are based on 50 independent test episodes for each trained model. Since the success rate of the proposed method is 0.76 and that of the baseline SAC is 0.74, the difference between the two methods is relatively small. Therefore, this paper does not claim a strict statistically significant improvement based only on the current 50 test episodes. Instead, the results are interpreted as a moderate improvement under the current experimental setting. When success rate, collision rate, average episode length, training stability, and trajectory behavior are considered together, the proposed method shows a more balanced overall performance. Due to the high computational cost of AirSim-based training, the current work has not conducted multiple repeated training runs under different random seeds or systematic statistical significance tests. In future work, more random seeds, standard deviations, confidence intervals, and statistical significance tests will be introduced to further verify the robustness of the proposed method.

The proposed method also introduces additional computational costs compared with the baseline SAC. The DPCA module adds a lightweight feature recalibration operation based on two feature-mapping paths, and therefore only introduces a small number of additional parameters compared with the main actor and critic networks. The ARF mechanism mainly adjusts several scalar reward weights according to the normalized training progress, so its computational overhead is very low. PER introduces additional costs for priority calculation, importance-sampling weight computation, and priority updating. However, this additional cost mainly occurs during training and does not increase the inference

cost of the trained policy. Therefore, the proposed method represents a trade-off between moderate performance improvement and acceptable additional training cost.

There are several limitations to the proposed method. First, when the navigation scenario is relatively simple or the state vector is already sufficiently compact, the performance gain brought by the DPCA module may be limited. Second, the effectiveness of the ARF mechanism depends on the design of reward components and scheduling parameters. Improper reward-weight settings may cause the agent to overemphasize one objective and weaken the balance between target reaching and flight safety. Third, PER can improve the utilization of critical samples, but it may also introduce additional sampling fluctuations and training cost. Fourth, the current method mainly focuses on single-UAV navigation with low-dimensional state vectors, and its applicability to visual navigation, dynamic obstacle environments, and multi-UAV cooperative scenarios still requires further investigation.

In addition, all experiments in this study are conducted in AirSim-based simulation environments. When transferring the trained policy to real UAV platforms, sensor noise, wind disturbance, actuator delay, communication latency, inaccurate dynamics models, and the sim-to-real gap may affect the navigation performance. Future research will consider domain randomization, noise injection, dynamics-parameter perturbation, hardware-in-the-loop simulation, and real-world flight tests to further improve and verify the robustness and transferability of the proposed method.

6.2. Conclusions

This paper focuses on the autonomous navigation and obstacle avoidance of unmanned aerial vehicles (UAVs) in complex environments, and proposes an improved SAC method called DPCA-ARF-PER-SAC. This method integrates a dual-path channel attention module, adaptive reward feedback, and prioritized experience replay mechanism. Without changing the maximum entropy learning framework of SAC, DPCA-ARF-PER-SAC enhances the baseline algorithm from three aspects: state feature representation, reward weight scheduling, and key sample utilization.

Experiments were conducted in the 2D SimpleAvoid scenario and the 3D NH_center scenario. The 2D SimpleAvoid scenario was used for module-level ablation analysis, comparing five methods: SAC, PER-SAC, ARF-SAC, ARF-PER-SAC, and DPCA-ARF-PER-SAC. The results show that the proposed method can maintain a high success rate and low collision rate in relatively simple obstacle avoidance tasks, while improving path efficiency. These results indicate that the joint use of DPCA, ARF, and PER helps to improve training stability and navigation efficiency.

In the 3D NH_center scenario, the proposed method was further compared with TD3, SAC, and AM-SAC. In 50 independent tests, the proposed DPCA-ARF-PER-SAC method achieved a success rate of 0.76, a collision rate of 0.24, and an average episode length of 232.8 steps. Compared with the baseline SAC, the success rate increased by 2 percentage points, the collision rate decreased by 2 percentage points, and the average episode length decreased by 4.3 steps. Compared with TD3, this method also demonstrated higher path efficiency and more balanced navigation performance. Moreover, in complex 3D scenarios, AM-SAC did not always outperform the baseline SAC, indicating that attention enhancement alone may not be sufficient to achieve stable UAV navigation in complex environments.

Overall, the proposed DPCA-ARF-PER-SAC method achieves a better balance among task completion, flight safety, and path efficiency through the synergistic effect of state representation enhancement, adaptive reward guidance, and prioritized sample utilization. Future research will further explore visual-input-based navigation, multi-scenario transfer

training, domain randomization, hardware-in-the-loop simulation, and real-world flight validation to enhance the generalization ability, physical feasibility, and robustness of the proposed method in real complex environments.

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

Funding: This research was funded by the Xingliao Talents Plan under Grant XLYC2202013, the Natural Science Foundation of Liaoning Province under Grant Nos. 2024-MS-113 and 2025-MSLH-597, the Liaoning Provincial Department of Education Science and Technology Innovation Team Project under Grant LJ222510144001, and the High-Level Talent Introduction Scientific Research Support Program of Shenyang Ligong University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Meng, W.; Zhang, X.; Zhou, L.; Guo, H.; Hu, X. Advances in UAV Path Planning: A Comprehensive Review of Methods, Challenges, and Future Directions. *Drones* **2025**, *9*, 376. [CrossRef]
2. Rahman, M.; Sarkar, N.I.; Lutui, R. A Survey on Multi-UAV Path Planning: Classification, Algorithms, Open Research Problems, and Future Directions. *Drones* **2025**, *9*, 263. [CrossRef]
3. Xiong, H.; Yu, B.; Zhang, Y. Research on Autonomous Navigation and Obstacle Avoidance Methods for High-Speed Large-Inertia Rotor UAV. *Drones* **2026**, *10*, 259. [CrossRef]
4. Sheng, Y.; Liu, H.; Li, J.; Han, Q. UAV Autonomous Navigation Based on Deep Reinforcement Learning in Highly Dynamic and High-Density Environments. *Drones* **2024**, *8*, 516. [CrossRef]
5. Lei, B.; Hu, W.; Ren, Z.; Ji, S. DRL-Based UAV Autonomous Navigation and Obstacle Avoidance with LiDAR and Depth Camera Fusion. *Aerospace* **2025**, *12*, 848. [CrossRef]
6. Wang, F.; Zhu, X.; Zhou, Z.; Tang, Y. Deep-Reinforcement-Learning-Based UAV Autonomous Navigation and Collision Avoidance in Unknown Environments. *Chin. J. Aeronaut.* **2024**, *37*, 237–257. [CrossRef]
7. Guo, T.; Jiang, N.; Li, B.; Zhu, X.; Wang, Y.; Du, W. UAV Navigation in High Dynamic Environments: A Deep Reinforcement Learning Approach. *Chin. J. Aeronaut.* **2021**, *34*, 479–489. [CrossRef]
8. He, L.; Aouf, N.; Song, B. Explainable Deep Reinforcement Learning for UAV Autonomous Path Planning. *Aerosp. Sci. Technol.* **2021**, *118*, 107052. [CrossRef]
9. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 1861–1870.
10. Zhou, Y.; Shu, J.; Hao, H.; Song, H.; Lai, X. UAV 3D Online Track Planning Based on Improved SAC Algorithm. *J. Braz. Soc. Mech. Sci. Eng.* **2024**, *46*, 12. [CrossRef]
11. Song, C.; Cui, Y.; Luo, S.; Zhang, X.; She, Y.; Li, B. UAV Tracking Moving Target Mission Planning Based on TW-SAC Algorithm. In Proceedings of the 2024 IEEE 4th International Conference on Human-Machine Systems (ICHMS), Toronto, ON, Canada, 15 May 2024; pp. 1–6.
12. Huang, J.; Cui, Y.; Xi, G.; Bai, S.; Li, B.; Wang, G.; Neretin, E. GTrXL-SAC-Based Path Planning and Obstacle-Aware Control Decision-Making for UAV Autonomous Control. *Drones* **2025**, *9*, 275. [CrossRef]
13. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized Experience Replay. *arXiv* **2015**, arXiv:1511.05952. [CrossRef]
14. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In *Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 3–19.

15. Bo, L.; Zhang, T.; Zhang, H.; Hong, J.; Liu, M.; Zhang, C.; Liu, B. 3D UAV Path Planning in Unknown Environment: A Transfer Reinforcement Learning Method Based on Low-Rank Adaption. *Adv. Eng. Inform.* **2024**, *62*, 102920. [CrossRef]
16. Huang, H.; Yang, Y.; Wang, H.; Ding, Z.; Sari, H.; Adachi, F. Deep Reinforcement Learning for UAV Navigation Through Massive MIMO Technique. *IEEE Trans. Veh. Technol.* **2020**, *69*, 1117–1121. [CrossRef]
17. Liu, J.; Luo, W.; Zhang, G.; Li, R. Unmanned Aerial Vehicle Path Planning in Complex Dynamic Environments Based on Deep Reinforcement Learning. *Machines* **2025**, *13*, 162. [CrossRef]
18. Yin, Y.; Wang, Z.; Zheng, L.; Su, Q.; Guo, Y. Autonomous UAV Navigation with Adaptive Control Based on Deep Reinforcement Learning. *Electronics* **2024**, *13*, 2432. [CrossRef]
19. Wu, H.; Wang, W.; Wang, T.; Suzuki, S. Model-Free UAV Navigation in Unknown Complex Environments Using Vision-Based Reinforcement Learning. *Drones* **2025**, *9*, 566. [CrossRef]
20. McEnroe, P.; Wang, S.; Liyanage, M. FERO: Efficient Deep Reinforcement Learning Based UAV Obstacle Avoidance at the Edge. *IEEE Open J. Comput. Soc.* **2025**, *6*, 1378–1389. [CrossRef]
21. Sun, T.; Gu, J.; Mou, J. UAV Autonomous Obstacle Avoidance via Causal Reinforcement Learning. *Displays* **2025**, *87*, 102966. [CrossRef]
22. Li, B.; Bai, S.; Liang, S.; Ma, R.; Neretin, E.; Huang, J. Manoeuvre Decision-Making of Unmanned Aerial Vehicles in Air Combat Based on an Expert Actor-Based Soft Actor Critic Algorithm. *CAAI Trans. Intell. Technol.* **2023**, *8*, 1608–1619. [CrossRef]
23. Jiang, W.; Cai, T.; Xu, G.; Wang, Y. Autonomous Obstacle Avoidance and Target Tracking of UAV: Transformer for Observation Sequence in Reinforcement Learning. *Knowl.-Based Syst.* **2024**, *290*, 111604. [CrossRef]
24. Dong, W.; Liu, Y.; Guo, X.; Wang, C.; Ding, Z. Transformer-Based Data-Driven Reinforcement Learning for Collision-Free Integrated Planning and Control of Multiple UAVs. *Chin. J. Aeronaut.* **2025**, 104022. [CrossRef]
25. Zhao, X.; Yang, R.; Zhong, L.; Hou, Z. Multi-UAV Path Planning and Following Based on Multi-Agent Reinforcement Learning. *Drones* **2024**, *8*, 18. [CrossRef]
26. Tang, J.; Liang, Y.; Li, K. Dynamic Scene Path Planning of UAVs Based on Deep Reinforcement Learning. *Drones* **2024**, *8*, 60. [CrossRef]
27. Xu, Y.; Wei, Y.; Jiang, K.; Chen, L.; Wang, D.; Deng, H. Action Decoupled SAC Reinforcement Learning with Discrete-Continuous Hybrid Action Spaces. *Neurocomputing* **2023**, *537*, 141–151. [CrossRef]
28. Xie, Y.; Wang, L.; Chang, Z.; Xu, L.; Bi, S.; Han, Z. GeoAgg-HSAC: An RL-Based Framework for Trajectory and Resource Optimization in Mountainous UAV Integrated Localization and Communication Networks. *IEEE Trans. Wirel. Commun.* **2026**, *25*, 6507–6522. [CrossRef]
29. Goudarzi, S.; Soleymani, S.A.; Anisi, M.H.; Jindal, A.; Xiao, P. Optimizing UAV-Assisted Vehicular Edge Computing with Age of Information: An SAC-Based Solution. *IEEE Internet Things J.* **2025**, *12*, 4555–4569. [CrossRef]
30. Yang, X.-X.; Yao, W.-Q.; Zhang, Y.; Yu, H.; Wang, C. Multi-UAV Cooperative Search in Partially Observable Low-Altitude Environments Based on Deep Reinforcement Learning. *Drones* **2025**, *9*, 825. [CrossRef]
31. Zhang, T.; Lei, J.; Liu, Y.; Feng, C.; Nallanathan, A. Trajectory Optimization for UAV Emergency Communication with Limited User Equipment Energy: A Safe-DQN Approach. *IEEE Trans. Green Commun. Netw.* **2021**, *5*, 1236–1247. [CrossRef]
32. Chen, S.; Peng, C.; Jiang, L.; Nai, K.; Liang, W.; Wang, J.; Li, K.; Khan Pathan, A.-S. A Robust Deep Q-Network (DQN) for Heterogeneous Tasks and QoS-Aware UAV Relay Communication Optimization. *IEEE Internet Things J.* **2025**, *12*, 33980–33994. [CrossRef]
33. Chen, J.; Lv, S.; Zhang, T.; Wang, Y.; Wang, Y. The Semidouble DQN Resource Optimization Strategy for UAV-Aided Networks: A Case Study. *IEEE Trans. Aerosp. Electron. Syst.* **2025**, *61*, 7852–7862. [CrossRef]
34. Liu, Y.; Li, X.; Wang, J.; Wei, F.; Yang, J. Reinforcement-Learning-Based Multi-UAV Cooperative Search for Moving Targets in 3D Scenarios. *Drones* **2024**, *8*, 378. [CrossRef]
35. Yu, R.; Li, Q.; Ji, J.; Wu, T.; Mao, J.; Liu, S.; Sun, Z. Improved Double DQN with Deep Reinforcement Learning for UAV Indoor Autonomous Obstacle Avoidance. *Sci. Rep.* **2025**, *15*, 28133. [CrossRef] [PubMed]