electronics

Article

Utility Maximization Integrating Secrecy, Energy Consumption, and Latency for UAV-Assisted MEC Systems via Dual-Replay TD3

Yishan Zang [1], Ying Su [1,*], Jing Zhang [1] and Zhutao Dai [2]

[1] College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai 200234, China; 1000548858@smail.shnu.edu.cn (Y.Z.); jannety@shnu.edu.cn (J.Z.)
[2] College of Information and Communication Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; daizhutao@nuaa.edu.cn
* Correspondence: yingsu@shnu.edu.cn

Abstract

Unmanned aerial vehicle (UAV)-assisted mobile edge computing (MEC) has emerged as a practical means of providing computation and communication support for geographically dispersed Internet of Things (IoT) terminals. However, the broadcast nature of wireless links makes offloading data vulnerable to cooperative eavesdropping. Moreover, the limited onboard energy of UAVs and the latency-sensitive characteristics of MEC services lead to a challenging trade-off between the secrecy rate, energy consumption, and latency. To address this issue, we formulate a utility maximization problem to jointly optimize UAV trajectory and task-offloading decisions in UAV-assisted MEC systems against multiple eavesdroppers. Due to the strong coupling among optimization variables and the non-convexity of the problem, an enhanced twin-delayed deep deterministic policy gradient (TD3) framework integrating Hindsight Experience Replay (HER) and Prioritized Experience Replay (PER) is proposed to improve convergence efficiency and learning stability. Furthermore, a system utility-driven reward function is designed to balance the secrecy rate, energy consumption, and processing latency under different application requirements. The simulation results demonstrate that the proposed approach consistently outperforms DDQN, DDPG, and conventional TD3 in terms of system utility, secrecy performance, convergence speed, and adaptability to different scenarios.

Keywords: unmanned aerial vehicle (UAV); mobile edge computing (MEC); deep reinforcement learning (DRL); system utility; physical layer security

Check for updates

Academic Editor: Ashwin Ashok

Received: 2 July 2026
Revised: 10 August 2026
Accepted: 13 August 2026
Published: 1 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Driven by the vision of 6G, the exponential growth of delay-sensitive and computation-intensive applications, including the Internet of Things (IoT), AR/VR, and autonomous driving, has posed unprecedented challenges for traditional terrestrial base stations and cloud infrastructures in terms of coverage and latency [1]. As a key enabler, Mobile Edge Computing (MEC) mitigates these issues by provisioning computing resources closer to the end-users. Benefiting from their inherent high mobility, flexible on-demand deployment, and favorable Line-of-Sight (LoS) propagation conditions, Unmanned Aerial Vehicles (UAVs) have been regarded as promising MEC platforms capable of serving as aerial computing servers. UAV-assisted MEC systems represent a transformative paradigm that integrates aerial platforms with edge intelligence, enabling pervasive and on-demand

computation for applications such as emergency communications, low-altitude IoTs, and remote sensing [2].

However, in practice, UAV-assisted MEC systems remain vulnerable to potential eavesdropping threats, thus posing a significant challenge to protecting data integrity and system reliability, especially for secrecy-critical demanding scenarios [3]. Meanwhile, low latency has also become a paramount objective for next-generation wireless networks, driving the future of real-time connectivity. Both latency and transmission security are simultaneously required in some security-sensitive Ultra-Reliable and Low-Latency Communications (URLLCs) applications, such as UAV-assisted surveillance, autonomous driving, and industrial Internet of Things (IIoT) [4]. Additionally, limited onboard energy, as an major intrinsic limitation of UAVs, constrains their long-term endurance, computational efficiency, and communication performance in MEC systems [5]. Therefore, UAV-MEC systems entail significant trade-offs among secrecy rate, energy consumption, and delay, as these metrics are inherently conflicting. This renders the joint optimization of UAV trajectory and task offloading decisions challenging. The strong coupling among optimization variables and conflicting objectives makes the resulting problem non-convex and hard to solve.

Traditional optimization algorithms, including convex optimization and heuristic algorithms, have been widely adopted for the joint optimization of wireless communication resources and UAV fight path planning, their effectiveness is often limited by high computational complexity and the tendency to converge to suboptimal local solutions [6]. Recently, deep reinforcement learning (DRL) has emerged as a powerful solution to complex optimization problems that involve high-dimensional state spaces and sequential decision-making. More importantly, DRL enables agents to continuously interact with and adapt to dynamic environments, making it particularly suitable for UAV-MEC systems characterized by time-varying channel conditions and terminal demands. Various DRL algorithms, such as DQN, DDPG, PPO, SAC, and TD3, have been investigated for UAV-MEC optimization [7–9]. Owing to its superior stability and effectiveness in continuous action spaces, TD3 is adopted as the baseline framework in this work. However, its learning efficiency and convergence performance remain limited in complex UAV-MEC environments. Motivated by the above challenges, we propose an enhanced TD3-based framework to solve the formulated utility maximization problem in UAV-MEC with cooperative eavesdroppers.

1.1. Related Works

Extensive research has been devoted to jointly optimizing UAV trajectories and resource allocation for low-altitude MEC deployments. More recently, DRL has attracted increasing attention owing to its ability to efficiently handle complex resource allocation problems. Driven by the multi-dimensional and often conflicting metrics requirements of emerging applications, research focus has progressively shifted from conventional single-objective optimization (e.g., solely minimizing latency [10–12] or energy consumption [5,13]) toward the joint optimization of multiple performance metrics.

Achieving a balance between latency and energy consumption is a fundamental challenge in UAV-MEC systems. For instance, to minimize overall energy consumption, the authors of [14] proposed a two-tier iterative algorithm for joint optimization of computation offloading and resource allocation under latency limitations in MIMO-NOMA-assisted multi-user MEC networks. The authors of [15] minimized an energy- and latency-based cost function based on the block descent method in a multi-UAV-assisted MEC network. The authors of [16] developed a joint UAV scheduling and task-offloading scheme to balance latency and energy consumption, thereby improving user satisfaction. Ref. [17] investigated a UAV-aided video transmission system and maximized a system utility

function incorporating energy efficiency and quality of experience (QoE) into a unified performance metric.

Since wireless transmissions are inherently vulnerable to eavesdropping, growing attention has been devoted to enhancing communication security in UAV-MEC systems. In [7], DDQN-based trajectory and resource optimization for UAV-aided MEC secure communications was explored. In [18], energy-efficient computation offloading for secure UAV-edge-computing systems was investigated. The authors of [19] leveraged the Deep Deterministic Policy Gradient (DDPG) algorithm for iterative policy optimization to minimize task-processing delays while ensuring secure communication. In [20], security energy efficiency was optimized based on the block coordinate descent (BCD) algorithm in an aerial-IRS-assisted UAV-MEC system.

However, most existing works are restricted to optimizing the trade-off between two of the three key performance metrics, namely secrecy, latency, and energy consumption. Jointly optimizing all three objectives has received limited attention in UAV-MEC systems. The authors of [21] maximized users' requirement satisfactions in terrestrial MEC networks with multi-eavesdroppers. The satisfaction model integrated secrecy performance, latency, and energy consumption into a unified optimization objective. A successive convex approximation (SCA)-based algorithm was proposed to obtain suboptimal solutions. Inspired by this, this paper investigates a DRL-based joint optimization framework to address a multi-objective optimization problem and dynamically balance the competing performance metrics of secure rate, latency, and energy consumption.

To distinguish the proposed work from existing studies, a comparative analysis is provided in Table 1 from the perspectives of system modeling and optimization methodology.

Table 1. Comparison of existing UAV-assisted MEC optimization methods.

MEC Systems		Optimization Objectives			Solution			Reference
UAV MEC	**Terrestrial MEC**	**Secrecy**	**Latency**	**Energy**	**Convex Optimization**	**Heuristic Algorithms**	**DRL Framework**	
✓				✓			✓	[13]
✓			✓				✓	[11,12]
✓		✓					✓	[7,8,10]
✓			✓	✓	✓			[14,15]
✓		✓	✓				✓	[19]
✓			✓	✓		✓		[16]
✓			✓	✓			✓	[17]
✓		✓		✓	✓			[20]
	✓	✓	✓	✓	✓			[21]
✓		✓	✓	✓			✓	Ours

1.2. Motivations and Contributions

Motivated by the above observations, this paper investigates a UAV-assisted MEC system facing cooperative eavesdropping threats. We define a system utility metric that jointly captures security, delay, and energy consumption, which are critical in security- and latency-sensitive applications such as emergency communications and industrial IoT applications. To maximize the utility metric, we formulate a joint optimization problem of UAV trajectory and task offloading under multi-eavesdroppers. To solve the non-convex problem, we propose an enhanced Twin Delayed Deep Deterministic Policy

Gradient (TD3) algorithm with a dual-replay mechanism to solve the formulated problem. The proposed algorithm framework integrates Prioritized Experience Replay (PER) and Hindsight Experience Replay (HER) buffers to address sparse-reward and sample-efficiency challenges in UAV-assisted secure MEC networks. The main contributions of this work are summarized as follows:

- Unlike most existing works, which focus solely on latency or energy efficiency while overlooking communication security, we formulate a fractional system utility model that integrates secure rate, latency, and energy consumption into a unified utility metric, thereby enabling a unified optimization framework for UAV-assisted MEC systems under cooperative eavesdropping attacks. To maximize the proposed utility, we formulate a joint optimization problem over the UAV trajectory and task data offloading policy.
- Since the resulting problem is highly coupled and non-convex, we propose an enhanced TD3-based DRL framework to obtain efficient online solutions. A utility-driven reward mechanism that combines the penalty terms for constraint violations is designed. A dual-replay mechanism with PER and HER buffers is developed to alleviate the sparse reward problem and improve convergence stability.
- Extensive simulations demonstrate that the proposed algorithm achieves a higher system utility metric and secure rate, faster convergence, and greater stability compared with DDQN, DDPG, and baseline TD3 schemes, under different eavesdropper numbers, transmission powers, and preference weights.

The remainder of this paper is organized as follows. In Section 2, the system model of the UAV-assisted MEC network and the optimization problem are presented. Section 3 introduces the proposed TD3-enhanced approach for solving the optimization problem. In Section 4, numerical results are presented to evaluate the performance of the proposed algorithm, including its convergence behavior and robustness under different system parameters. Finally, Section 5 concludes the article and outlines possible directions for future work.

2. System Model and Problem Formulation

We consider a single-UAV-assisted MEC system, where the onboard server handles computation offloading from IoT devices within its serving cluster. Figure 1 depicts a multi-UAV scenario, where IoT devices are grouped into multiple clusters (e.g., via K-means), each served by one UAV. The single-UAV case studied in this paper corresponds to one representative cluster within this extensive architecture. The cooperative offloading-policy optimization in multi-UAV networks is left for future work. In one cluster, a UAV is equipped with an onboard MEC server, which provides computation offloading for the IoT devices. Let $\mathcal{K} = \{1, 2, \ldots, K\}$ denotes the index sets of IoT devices in a cluster. Each IoT device generates local computation tasks, which can either be executed locally or offloaded to the UAV via secure air-to-ground (A2G) links. The UAV flies from an initial location u_{uav}^{I} to a final destination u_{uav}^{F} along an optimized trajectory. To better reflect practical scenarios, in the proposed system model, cooperative eavesdroppers and corresponding anti-eavesdropping countermeasures are jointly considered. In addition to legitimate IoT devices, S ground eavesdroppers and one friendly jammer are deployed, where the jammer transmits artificial interference to degrade the eavesdroppers' reception of task-offloading transmissions. The UAV's flight period T is divided into N equal time slots, each with duration $\delta = T/N$. The spatial layout of all entities is represented in a 3D Cartesian coordinate system. Specifically, the UAV, the kth IoT device, the sth eavesdropper, and the jammer at slot n are located at $\boldsymbol{u}_{uav}[n] = (x_{uav}[n], y_{uav}[n], h)^T$, $\boldsymbol{u}_k[n] = (x_k[n], y_k[n], 0)^T$, $\boldsymbol{u}_s[n] = (x_s, y_s, 0)^T$, and $\boldsymbol{u}_j[n] = (x_j[n], y_j[n], 0)^T$, respectively. The UAV maintains a fixed alti-

tude h, with velocity $v[n] \in [0, v_{\max}]$ and heading angle $\theta[n] \in [0, 2\pi]$. The position of the UAV is updated by

$$\boldsymbol{u}_{uav}[n+1] = (x_{uav}[n] + v[n]\delta\cos(\theta[n]), y_{uav}[n] + v[n]\delta\sin(\theta[n]), h)^T. \tag{1}$$

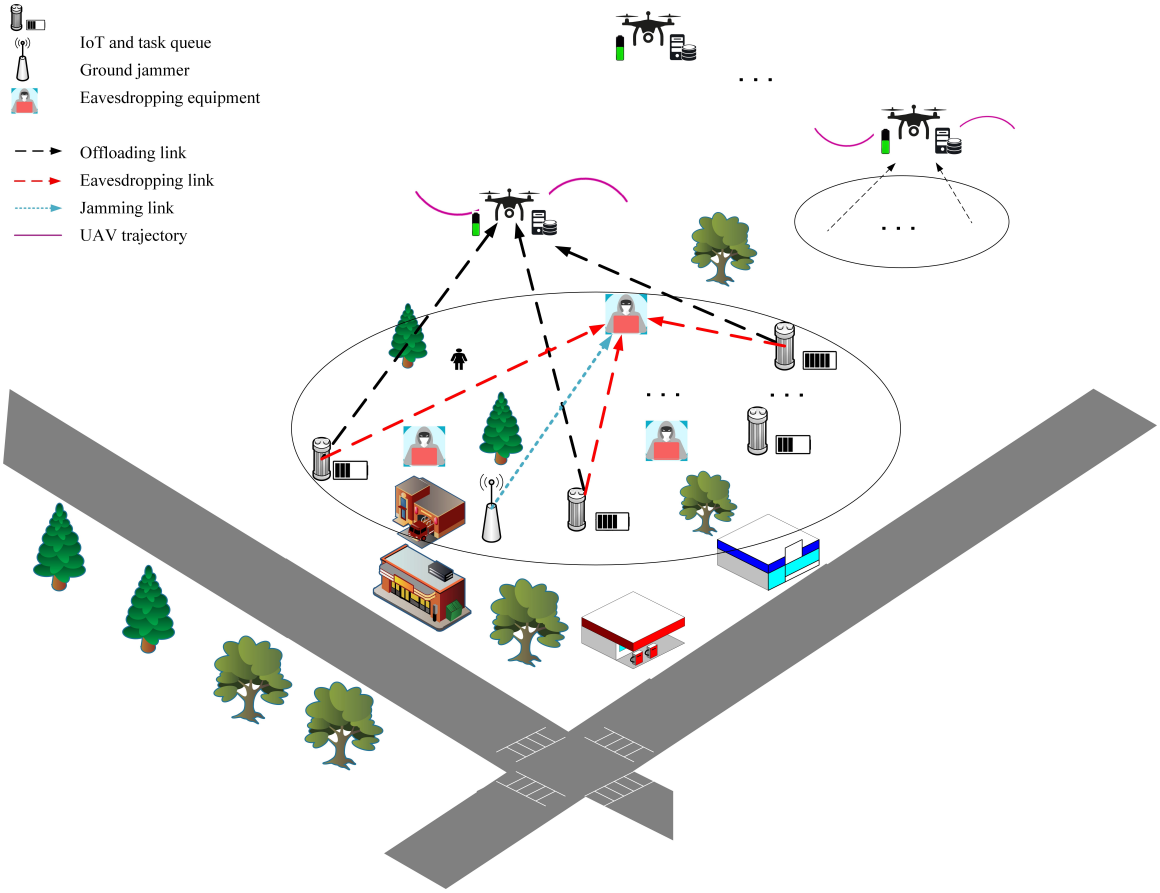


Figure 1. System model of the UAV-assisted secure MEC systems under cooperative eavesdroppers.

To represent practical UAV-MEC communication environments, we adopt an A2G channel model incorporating both large-scale path loss and small-scale fading [22]. The large-scale fading related to the kth IoT device is given by:

$$PL_k[n] = \begin{cases} 20\lg d^k_{uav}[n] + 20\lg f_c + 20\lg(\frac{4\pi}{v_c}) + \eta LoS \\ 20\lg d^k_{uav}[n] + 20\lg f_c + 20\lg(\frac{4\pi}{v_c}) + \eta NLoS \end{cases}, \tag{2}$$

where $d^k_{uav}[n] = ||\boldsymbol{u}_{uav}[n] - \boldsymbol{u}_k||$ indicates the distance between the UAV and the kth IoT device at time slot n, f_c refers to the carrier frequency, v_c denotes the speed of electromagnetic waves, and ηLoS and $\eta NLoS$ represent the propagation losses corresponding to LoS and NLoS links, respectively. The small-scale fading coefficient $\tilde{g}_k[n]$ follows Rician fading under LoS conditions and Rayleigh fading under NLoS conditions. Consequently, the overall channel gain between the UAV and the kth IoT device is expressed as

$$g_k[n] = 10^{-PL_k[n]/20}\tilde{g}_k[n]. \tag{3}$$

The channel gain between the kth IoT device and the sth eavesdropper, as well as that between the sth eavesdropper and the ground jammer, are respectively given by

$$g_{k,s}[n] = \frac{\rho}{||\boldsymbol{u}_k - \boldsymbol{u}_s||^{\tilde{a}}}\tilde{b}, \qquad g_{s,j}[n] = \frac{\rho}{||\boldsymbol{u}_j - \boldsymbol{u}_s||^{\tilde{a}}}\tilde{b}. \tag{4}$$

where ρ denotes the channel gain at a reference distance. $\tilde{a} > 2$ denotes the path-loss exponent and $\tilde{b}$ follows an exponential distribution with unit mean.

2.1. Transmission Model

Following a TDMA protocol, each time slot n is shared by all K IoT devices. For scheduling flexibility, we adopt a randomized selection mechanism during the training phase. Specifically, at the beginning of each time slot, the IoT device samples an allocation vector from a softmax distribution over the feasible set of $\delta[n]$, where the probability of selecting device k is proportional to its current allocation ratio $\hat{\delta}_k[n]$. This randomized scheduling should satisfy

$$\sum_{k=1}^{K} \hat{\delta}_k[n] \leq 1, \qquad \hat{\delta}_k[n] \in [0,1]. \tag{5}$$

The eavesdroppers attempt to intercept the offloading data from IoT devices to the UAV. To enhance communication security, the UAV employs a ground-based jammer to send jamming signals to counter eavesdropping. It is assumed that the UAV-MEC receiver knows the jamming information in advance and can thus cancel the friendly interference via pre-shared synchronization, while the eavesdroppers, lacking such prior knowledge, suffer from artificial interference. This asymmetry effectively degrades the eavesdropping rate. However, in practice, residual interference still remains due to imperfect elimination caused by hardware non-ideality. The residual interference is approximately treated as part of thermal noise P_{Ni}. During slot n, the achievable offloading rate at the UAV from the kth IoT device is

$$R_k[n] = \hat{\delta}_k[n] \log_2\left(1 + \frac{P_k[n]|g_k[n]|^2}{P_{Ni}}\right), \tag{6}$$

where $P_k[n]$ is the transmit power of IoT device k e in slot n and P_{Ni} is the receiver noise power at the UAV.

The eavesdropping rate from the kth IoT device at the sth eavesdropper is

$$R_{k,s}[n] = \hat{\delta}_k[n] \log_2\left(1 + \frac{P_k[n]|g_{k,s}[n]|^2}{|g_{s,j}[n]|^2 P_j + P_{Ns}}\right), \tag{7}$$

where P_j and P_{Ns} denote the jammer's transmit power and the receiver noise power at the eavesdropper, respectively.

To better reflect practical security threats, we consider a collaborative eavesdropping model [23], where multiple eavesdroppers share their intercepted signals and jointly decode the confidential information. Such a setting captures worst-case adversarial capabilities and provides a lower bound on the achievable secrecy performance. Under this assumption, all eavesdroppers are equivalently modeled as a single super eavesdropper $\tilde{u}_s$. Accordingly, the signal-to-noise ratio (SNR) at the super eavesdropper for the kth IoT device in slot n is expressed as

$$SNR_{k,\tilde{u}_s}[n] = \sum_{s \in S} \left(\frac{P_k[n]|g_{k,s}[n]|^2}{|g_{s,j}[n]|^2 P_j + P_{Ns}}\right), \tag{8}$$

and its eavesdropping rate becomes

$$R_{k,\tilde{u}_s}[n] = \hat{\delta}_k[n] \log_2(1 + SNR_{k,\tilde{u}_s}[n]). \tag{9}$$

Therefore, the achievable secure transmission rate from the kth IoT device to the UAV is expressed as

$$R_{k,\text{sec}}[n] = [R_k[n] - R_{k,\tilde{u}_s}[n]]^+. \tag{10}$$

where $[c]^+ \triangleq \max(c, 0)$ ensures non-negativity.

2.2. Computational Model

We adopt a binary computation offloading model, where each IoT task is either fully offloaded to the UAV or executed locally within each time slot, depending on whether the instantaneous secrecy rate between the UAV and the device exceeds a predefined threshold. This assumption is motivated by practical IoT scenarios involving indivisible sensing, monitoring, or control tasks, where partial execution may violate task integrity or security requirements. Therefore, the secure computation bits of the kth IoT device in slot n is given by

$$F_k[n] = \psi_k[n] W R_{k,\text{sec}}[n] c_{uav} \delta + (1 - \psi_k[n]) L_{loc,k}[n], \tag{11}$$

where W denotes the system bandwidth, $R_{k,sec}$ represents the secure achievable rate for offloading transmission, and δ denotes the duration of each time slot. $\psi_k[n] \in \{0, 1\}$ is the binary off-loading indicator. $L_{loc,k}[n]$ stands for the data size processed locally on the IoT device. Specifically, $\psi_k[n] = 1$ means that the kth IoT device offloads its computation task to the UAV for edge processing. Otherwise, $\psi_k[n] = 0$ means local execution. We adopt the safety-rate threshold-based offloading rule, i.e.,

$$\psi_k[n] = \begin{cases} 1, & \text{if} \quad R_{k,\text{sec}}[n] \geqslant \varepsilon_{\text{sec}}^{th} \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

Considering the downlink feedback data size is relatively small, the corresponding delay and energy consumption are neglected.

For local computation, let $f_{loc,k}$ denote the local CPU frequency and c_k denote the number of CPU cycles needed to process one bit of data. The local computation time is

$$t_{loc,k}[n] = \frac{c_k L_{loc,k}[n]}{f_{loc,k}}. \tag{13}$$

When the task is offloaded to the UAV, i.e., $\phi_k[n] = 1$, the total task completion time includes both transmission and computation time. Let f_{uav} and c_{uav} denote the UAV's CPU frequency and the number of cycles required per bit. $\hat{\delta}_k[n]$ denote the normalized transmission-time fraction within one time slot for device k. The uplink transmission and UAV-side computing time are expressed as

$$t_{trans,k}[n] = \delta \hat{\delta}_k[n], \qquad t_{com,k}[n] = \frac{F_k[n] c_{uav}}{f_{uav}}. \tag{14}$$

Accordingly, combining the two cases of local and offloaded execution, the total task completion time for the kth IoT device is

$$t_k[n] = \psi_k[n](t_{trans,k}[n] + t_{com,k}[n]) + (1 - \psi_k[n]) t_{loc,k}[n]. \tag{15}$$

2.3. Energy Model

The total system energy comprises three components: computation, UAV propulsion, and offloading transmission. The computation energy of the UAV-MEC server and the kth IoT device are respectively given by [24]

$$P_{uav}^{com} = \xi_{uav} f_{uav}^3, \qquad P_{loc}^{com} = \xi_k f_{loc,k}^3. \tag{16}$$

where ξ_{uav} and ξ_k are corresponding effective capacitance coefficients.

Based on [25], the UAV's propulsion power can be modeled as

$$P(v) = P_B\left(1 + \frac{3v^2}{U_{tip}^2}\right) + P_I\left(\sqrt{1 + \frac{v^4}{4v_0^4}} - \frac{v^2}{2v_0^2}\right)^{1/2} + \frac{1}{2} d_0 \iota \mu A v^3, \tag{17}$$

where P_B and P_I are constant parameters, U_{tip} represents the rotor blade tip speed, and v_0 represents the average induced velocity of the rotor during hovering. d_0, ι, μ, and A correspond to the fuselage drag coefficient, air density, rotor solidity, and rotor disc area, respectively.

For each IoT device at time slot n, the energy for UAV computation, local processing, transmission, and flight are respectively expressed as

$$E_{com,k}[n] = P_{uav}^{com} t_{com,k}[n], \quad E_{loc,k}[n] = P_{loc}^{com} t_{loc,k}[n], \tag{18}$$

$$E_{trans,k}[n] = P_k[n]\, t_{trans,k}[n], \quad E_{fly,k}[n] = P(v)\delta\hat{\delta}_k[n]. \tag{19}$$

Therefore, the total energy consumption to serve IoT device k in slot n is given by

$$E_k[n] = E_{fly,k}[n] + \psi_k[n](E_{trans,k}[n] + E_{com,k}[n]) + (1 - \psi_k[n])E_{loc,k}[n]. \tag{20}$$

2.4. System Utility Model

To balance multiple critical yet conflicting performance metrics, we formulate a system utility model to encapsulate security, delay and energy consumption via a unified metric [21]. The system utility is formulated in a fractional form, where the numerator represents the achievable secrecy rate, while the denominator is constructed as a weighted combination of the total energy consumption and processing latency. Since secure rate, delay, and energy consumption differ in both physical units and disparate numerical scales, a direct combination of these raw quantities would cause the metric to be dominated by the component with the largest magnitude. Relying solely on weight coefficients is insufficient to balance these heterogeneous metrics effectively. To address this, the secure computation balance factor γ_k^F and energy balance factor γ_k^E are introduced to harmonize the differences among these metrics. Specifically, the energy balance factor is defined as $\gamma_k^E = \frac{1}{\sqrt{E_{ref}^2 + \epsilon^2}}$, where E_{ref} denotes the reference maximum available system energy and ϵ is a small constant to avoid division-by-zero for numerical stability. Likewise, the computation balance factor is defined as $\gamma_k^F = \frac{1}{\sqrt{F_{ref}^2 + \epsilon^2}}$, where F_{ref} denotes the the predefined secure computation benchmark. It is worth noting that the reference parameters E_{ref}, F_{ref} and ϵ are carefully selected according to the system operating range, and their concrete numerical values are summarized in the simulation parameter table in Section 4. Accordingly, for the kth IoT device during the nth time slot, the system utility is formulated as

$$RS_k[n] = \frac{\gamma_k^F F_k[n]}{\alpha_k \gamma_k^E[n] E_k[n] + (1 - \alpha_k) t_k[n]}, \tag{21}$$

where $\alpha_k \in [0,1]$ is a tunable weight coefficient, reflecting the IoT-device's service preferences between energy consumption and processing delay. Applications with elevated α_k values prioritize energy conservation, whereas delay-sensitive applications adopt smaller α_k values.

2.5. Problem Description

The objective is to maximize the average system utility by jointly optimizing the UAV heading $\theta[n]$ and velocity $v[n]$. The UAV position $u_{uav}[n]$ is determined by $\theta[n]$ and $v[n]$. The optimization problem is formulated as

$$\begin{aligned}
&\max_{v[n],\theta[n]} \frac{1}{KN}\sum_{k=1}^{K}\sum_{n=1}^{N} \mathrm{RS}_k[n] \\
&\text{s.t.} \quad \text{C1: } R_k[n] - R_{k,\tilde{u}_s}[n] \geq \varepsilon_{\text{sec}}^{\text{th}},\ \forall k,n, \\
&\qquad \text{C2: } \textstyle\sum_{k=1}^{K} \hat{\delta}_k[n] \leq 1,\ 0 \leq \hat{\delta}_k[n] \leq 1,\ \forall k,n, \\
&\qquad \text{C3: } \textstyle\sum_{n=1}^{N} \left(\psi_k[n]E_{tr,k}[n] + (1-\psi_k[n])E_{loc,k}[n]\right) \leq P_k^{\max},\ \forall k,n, \\
&\qquad \text{C4: } \textstyle\sum_{n=1}^{N} \left(P(v)\delta + \sum_{k=1}^{K} \psi_k[n]E_{com,k}[n]\right) \leq P_{uav}^{\max},\ \forall k,n, \\
&\qquad \text{C5: } \boldsymbol{u}_{uav}[0] = \boldsymbol{u}_{uav}^{I}, \boldsymbol{u}_{uav}[N] = \boldsymbol{u}_{uav}^{F}, \\
&\qquad \text{C6: } 0 \leq v[n] \leq v_{\max};\ 0 < \theta[n] \leq 2\pi,\ \forall n, \\
&\qquad \text{C7: } x_{\min} \leq x_{uav}[n] \leq x_{\max};\ y_{\min} \leq y_{uav}[n] \leq y_{\max},\ \forall n.
\end{aligned} \tag{22}$$

where C1 enforces a minimum secrecy rate constraint during the offloading process, ensuring that the transmission from each IoT device meets the threshold. C2 regulates the time allocation among computation tasks. C3 and C4 impose energy limits on the IoT devices and UAV, respectively. C5 specifies the initial and final positions of the UAV. C6 restricts the UAV's maximum speed and yaw angle, while C7 confines its flight area within predefined spatial boundaries. Since the formulated optimization problem is highly non-convex and hard to solve using conventional optimization techniques, we resort to deep reinforcement learning. Specifically, the TD3-enhanced algorithm is employed to learn the joint optimization policy for UAV trajectory, resource allocation, and task offloading.

3. A TD3-Enhanced Approach Solution Framework

As illustrated in Figure 2, the proposed algorithm framework integrates secure communication, computation offloading, and UAV trajectory planning into a unified learning process. The UAV acts as the agent that interacts with an environment composed of IoT devices, eavesdroppers, and the MEC server. At each decision step, the UAV observes its current state, selects an action through the actor network, and receives a reward determined by the system utility metrics. The resulting transition tuples are stored in a replay buffer. To further improve training efficiency, we adopt HER and PER to process these tuples. Specifically, PER highlights high-value experiences by assigning a priority score to each tuple based on its temporal-difference (TD) error, while HER enriches training data by relabeling pseudo-goals based on spatial proximity. Mini-batches drawn from this enhanced buffer are then used to update the actor–critic networks and their target networks, promoting stable and efficient policy learning.

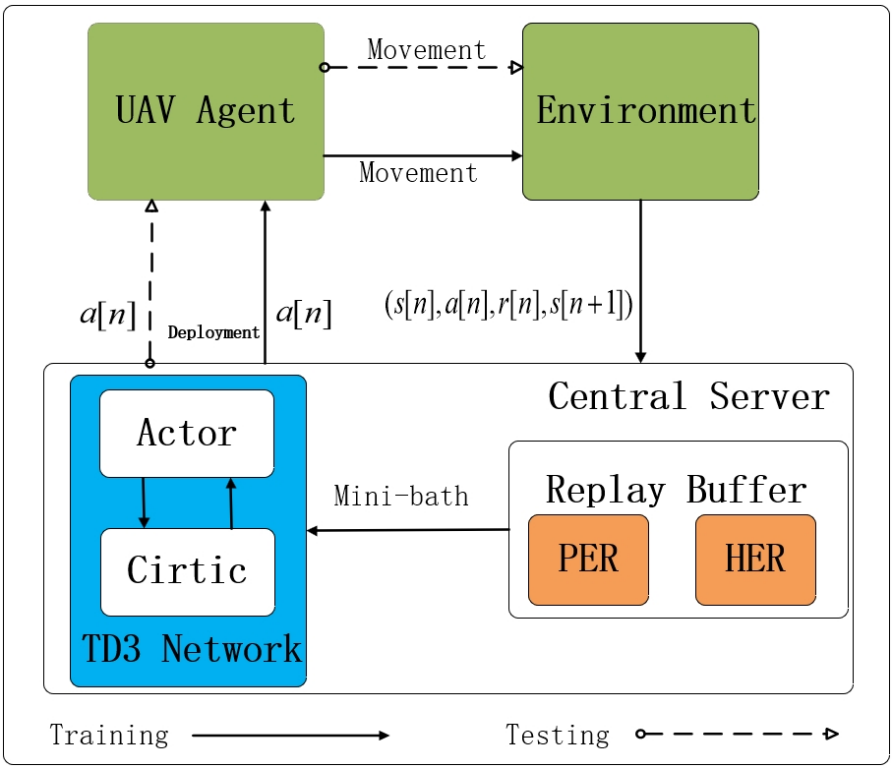


Figure 2. Block diagram of the proposed learning framework for UAV-assisted secure MEC systems.

3.1. MDP Model

The overall process can be formulated as a MDP, characterized by its state space s, actions space a, and rewards r [17]. The following description defines these components in detail.

State space: To facilitate the joint optimization of secure energy efficiency and latency, the state space must reflect the time-varying nature of both the physical environment, energy consumption, latency and secrecy performance. Thus, $s[n]$ incorporates the UAV's location ($u_{uav}[n]$, $E_t[n]$) and remaining energy ($E_t[n]$), as well as $\{\bar{R}_{s,t}, \bar{\tau}_t, \bar{E}_t\}$, which denote the historical average of the secrecy rate, latency, and energy consumption, respectively, to collectively characterize the current system status. This representation enables the UAV to capture both computational and spatial dynamics for decision-making. It is worth noting that certain static environmental parameters remain invariant throughout the entire training and execution process. These static parameters encompass the fixed locations of the eavesdroppers and the jammers, along with other invariant system settings, which serve as the contextual background for the UAV-MEC system. Although they implicitly influence the channel state transitions, computation and task states, remaining energy and reward calculations, they are deliberately excluded from the DRL state space to avoid excessive dimensionality.

Action space: During each time slot n, the action comprises the flight angle $\theta[n]$, flight speed $v[n]$, i.e., $a[n] = [\theta[n], v[n]]$. Since both $\theta[n]$ and $v[n]$ are continuous variables, the trajectory-planning problem naturally forms a continuous control task within the DRL framework.

Rewards: To maximize the system utility in UAV-assisted MEC systems, a composite reward function is designed. It jointly accounts for terminal convergence, energy limits, boundary violations, and overall system utility. This formulation converts the constrained optimization goal into an MDP objective of maximizing long-term returns.

- Terminal incentive (destination convergence): The UAV receives a terminal reward N_a at the final time slot N, which is expressed as

$$N_a = \begin{cases} \zeta_a r_a + (1-\zeta_a) r_{na}, & n = N, \\ 0, & n < N. \end{cases} \tag{23}$$

  where $\zeta_a = 1$ indicates that the UAV reaches its destination. A positive reward r_a is assigned upon success, while a negative reward r_{na} penalizes failure. This term encourages timely mission completion.
- Energy constraint: The UAV's remaining energy at time slot n is

$$E_{re}[n] = E_{uav}^{\max} - \delta \sum_{j=1}^{n} P(v[j]) - \sum_{j=1}^{n} \sum_{k=1}^{K} \psi_k[j] E_{com,k}[j], \tag{24}$$

  where $E_{uav}^{\max}$ is the maximum battery capacity. When $E_{re}[n]$ drops below the minimum required return energy $E_{\min}$, the UAV initiates its return. This constraint ensures safe operation under limited energy.
- utility reward: If the UAV's remaining energy exceeds $E_{\min}$, the utility factor is ω_e; otherwise, it is ω_{ne}, with $\omega_e > \omega_{ne}$. Thus,

$$r_{sa}[n] = \begin{cases} \omega_e \sum_{k=1}^{K} RS_k[n], & if \quad E_{re}[n] > E_{\min} \\ \omega_{ne} \sum_{k=1}^{K} RS_k[n], & if \quad E_{re}[n] \le E_{\min}. \end{cases} \tag{25}$$

  This term links secure computing performance with energy status.
- Boundary constraint: A penalty $r_{bou}[n]$ is imposed if the UAV flies beyond the predefined area:

$$r_{bou}[n] = \psi_{bou}[n] \kappa_{fa}, \tag{26}$$

  where $\psi_{bou}[n] = 1$ indicates boundary violation. This discourages unsafe or infeasible trajectories.

  Finally, the total reward at each time slot combines all components:

$$r[n] = r_{sa}[n] + r_{bou}[n] + N_a. \tag{27}$$

This unified reward encourages secure data collection, energy-aware control, and timely mission completion within the UAV-assisted MEC systems.

3.2. Optimization Framework for the UAV Trajectory

The TD3 algorithm, as a variant of the Actor-Critic (AC) method, is well suited for optimization problems in continuous action spaces. It is particularly effective for UAV trajectory control and time allocation in dynamic MEC environments [26]. Unlike DDPG, TD3 employs two critic networks to alleviate the Q-value overestimation problem. To effectively address sparse rewards and unstable learning in UAV-assisted MEC systems, an enhanced TD3 framework integrating PER and spatial HER is developed.

Figure 3 shows the architecture of the proposed TD3+PER+HER framework customized for a UAV-assisted secure MEC system. In this framework, the UAV acts as the learning agent, while the environment consists of ground IoT devices, cooperative eavesdroppers, and the MEC server. At each decision step n, the UAV observes the system state $s[n]$ and selects a action $a[n]$ through the Actor network $\pi_\varphi(s)$. The interaction produces a transition tuple $(s[n], a[n], r[n], s[n+1])$, where the reward $r[n]$ reflects the system utility that balances secure computation capacity, delay, and energy. These transitions are stored in the replay buffer.

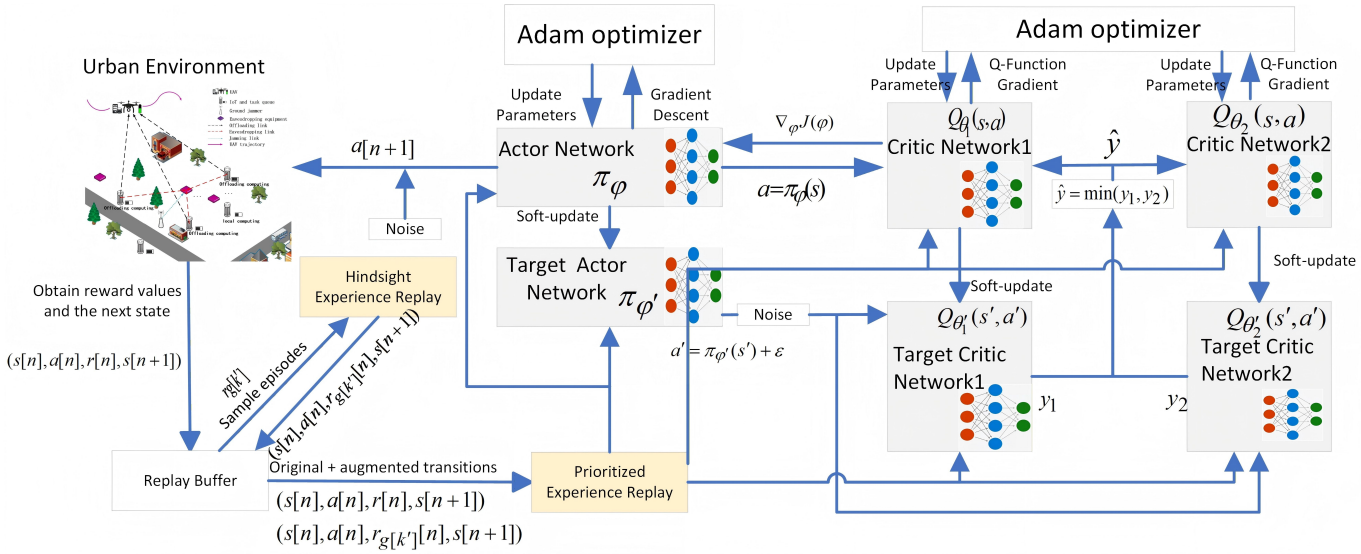


Figure 3. Overall architecture of the proposed enhanced TD3 algorithm interacting with the UAV-assisted secure MEC environment.

To improve sample efficiency and convergence stability, two replay strategies are jointly employed. First, Prioritized Experience Replay (PER) [27] samples transitions according to priorities computed from temporal-difference (TD) error, so that experiences with higher informational value are replayed more frequently. Second, to mitigate the challenges posed by sparse rewards due to rare destination arrivals, different from classical Hindsight Experience Replay (HER) [28], we introduce a spatial HER mechanism tailored for UAV trajectory learning. Rather than being triggered only when reaching the destination, our HER constructs pseudo-goals dynamically from all trajectories, regardless of whether the mission succeeds or fails. Its core function is to relabel failed incomplete trajectories to generate effective training samples. Meanwhile, successful episodes are also relabeled for data augmentation. The relabeled reward explicitly depends on the spatial distance reduction toward these reference states, providing dense intermediate feedback during trajectory execution.

During training, mini-batches of transitions are drawn from the buffer to update the six TD3 networks: the actor network $\pi_\varphi(s)$, critic networks $Q_{\theta_1}(s,a)$ and $Q_{\theta_2}(s,a)$, and their corresponding target networks $\pi_{\varphi'}(s')$, $Q_{\theta'_1}(s',a')$, and $Q_{\theta'_2}(s',a')$. The critic networks are optimized by minimizing loss function. The actor network is updated via policy gradients from the critics, while soft updates ensure stable parameter transfer to the target networks. In this way, the UAV gradually learns robust control policies that improve both secure communication reliability and MEC efficiency in the presence of malicious eavesdroppers. To further accelerate training, this paper adopts PER, assigning different weights to samples based on the TD-error, defined as

$$\hat{\vartheta}_t = r_{t+1} + \gamma Q_{\theta_i'}(s',a') - Q_{\theta_i}(s,a). \tag{28}$$

By combining uniform sampling with greedy selection, the probability of each sample j, denoted as $P(j)$, is defined as:

$$P(j) = \frac{D_j^{\hat{\alpha}}}{\sum_B D_B^{\hat{\alpha}}}, \tag{29}$$

where $D_j = \frac{1}{rank(j)} > 0$ denotes the priority of sample j, and $rank(j)$ denotes its position in the replay buffer sorted by $|\hat{\vartheta}_t|$. The hyperparameter $\hat{\alpha}$ controls the sampling degree:

$\hat{\alpha} = 0$ corresponds to uniform sampling, $\hat{\alpha} = 1$ to purely greedy sampling, and $0 < \hat{\alpha} < 1$ achieves a balance between them.

To correct the bias introduced by PER during model updates, importance sampling weights are applied to ensure unbiased learning. The weight for each sample j is defined as follows:

$$W_j = \left[\frac{1}{N} \cdot \frac{1}{P(j)}\right]^{\hat{\beta}}, \tag{30}$$

where N denotes the size of the replay buffer and $\hat{\beta}$ regulates the strength of bias correction.

In addition to the PER, HER is incorporated to mitigate the issue of sparse rewards caused by the terminal reward N_a. Specifically, at the end of each episode, HER relabels the original goals of the collected transitions with alternative goals drawn from the achieved states in the same episode. For each original transition $(s[n], a[n], r[n], s[n+1])$, a set of K relabeled goals $\{rg[k'] = s[n_{k'}]\}_{k'=1}^{K'}$ is selected, and the corresponding hindsight reward is defined as

$$r_{g[k']}[n] = \begin{cases} \dfrac{\beta_1}{1 + e^{\beta_2 l[n]}}, & \text{if } \mathrm{dist}(s[n+1], rg[k']) < \mathrm{dist}(s[n], rg[k']), \\ 0, & \text{otherwise}, \end{cases} \tag{31}$$

where β_1 and β_2 are tunable shaping parameters that control reward sensitivity. Each relabeled transition $(s[n], a[n], r_{g[k']}[n], s[n+1])$ is stored together with the original samples in the prioritized replay buffer. This hybrid PER–HER integration allows the agent to reinterpret failed trajectories as partial successes, significantly improving sample efficiency and accelerating TD3 policy convergence in sparse reward environments. During testing, the trained agent observes the system state and directly outputs flight actions through the Actor network until the task is completed. The proposed solution to system utility maximization for UAV-assisted secure MEC systems is presented in Algorithm 1.

3.3. Complexity Analysis

The computational complexity of the proposed TD3-based algorithm with PER and HER primarily stems from the parameter updates of the actor and critic networks, along with the additional cost of the replay mechanisms. Specifically, the actor update requires $\mathcal{O}\big(B \cdot \sum_{l=0}^{L_a-1} N_l^a N_{l+1}^a\big)$, where L_a and N_l^a denote the number of actor layers and the neurons in the l-th layer. The twin critic networks require $\mathcal{O}\big(2B \cdot \sum_{l=0}^{L_c-1} N_l^c N_{l+1}^c\big)$, where L_c and N_l^c denote the critic layers and neurons, respectively. The PER introduces an additional $\mathcal{O}(B \log \tilde{\omega})$ cost, where $\tilde{\omega}$ is the buffer size, while the hindsight experience replay expands the effective batch size to $B(1 + k_{\mathrm{HER}})$ with k_{HER} being the HER ratio. Therefore, the overall complexity per iteration is expressed as

$$\mathcal{O}\Big(MN\big((1 + k_{\mathrm{HER}})B\big(\sum_{l=0}^{L_a-1} N_l^a N_{l+1}^a + 2\sum_{l=0}^{L_c-1} N_l^c N_{l+1}^c\big) + B \log \tilde{\omega}\big)\Big). \tag{32}$$

This indicates that the complexity scales linearly with the batch size and HER expansion factor, and proportionally to the product of adjacent layer widths in the neural networks, while remaining computationally tractable for large-scale UAV-assisted secure MEC networks.

Algorithm 1 The proposed utility-based optimization framework

Initialize policy network π_φ with parameters φ and dual Q-networks Q_{θ_1}, Q_{θ_2} with parameters θ_1, θ_2
Set up the target networks by copying parameters: $\pi_{\varphi'} \leftarrow \pi_\varphi$, $Q_{\theta_1{}'} \leftarrow Q_{\theta_1}$, and $Q_{\theta_2{}'} \leftarrow Q_{\theta_2}$
Set up the experience replay buffer $\tilde{\omega}$
for each episode $m \in \{1, \ldots, M\}$ **do**
  Reset environment to initial state $s[0]$, set time step $n = 0$ and episode termination indicator $d \leftarrow$ False
  repeat
    Select exploration action $a[n] = \pi_\varphi(s[n]) + \varepsilon$
    Execute action, receive reward $r[n]$ and obtain $s[n+1]$
    if UAV violates operational boundaries **then**
      Augment reward $r_{bou}[n]$, Revert $a[n]$ and recompute $r[n]$ and $s[n+1]$
    end if
    if UAV arrives at target location at step N **then**
      Augment reward N_a, and set ($d \leftarrow$ True)
    end if
    Set priority $D_{\text{episode}} = \max_{i<\text{episode}} D_i$ and store $(s[n], a[n], r[n], s[n+1])$ in $\tilde{\omega}$
    Select K' future time steps $\{n_{k'}\}_{k'=1}^{K'}$ from the episode
    for each $k' = 1$ to K' **do**
      Define relabeled goal $rg[k'] = s[n_{k'}]$
      Compute hindsight reward $r_{g[k']}[n]$
      Store $(s[n], a[n], r_{g[k']}[n], s[n+1])$ in $\tilde{\omega}$
    end for
    if $size(experiencereplaybuffer) \geq \tilde{\omega}$ **then**
      for $j = 1$ to B **do**
        Calculate the TD-error from (33) and sample j from $\tilde{\omega}$ based on the priority distribution defined in (34)
        Compute the importance sampling update weights as defined in (35), and update the priority of sample j with $|\hat{\vartheta}_t|$
      end for
      Generate smoothed action: $a' = \pi_{\varphi'}(s') + \varepsilon$
      Calculate the target Q-value using: $y = r + \gamma \min_{i \in \{1,2\}} \{Q_{\theta_i'}(s', a')\}$
      Update critic networks (31), then perform gradient descent: $\theta_i \leftarrow \theta_i - \lambda \nabla_{\theta_i} L(\theta_i)$
      Update the actor network via the deterministic policy gradient as defined in (32)
      Update critic target network $Q_{\theta_1{}'}$ and $Q_{\theta_2{}'}$ accordingly:
      $\theta_1{}' \leftarrow (1-\tau)\theta_1{}' + \tau\theta_1$
      $\theta_2{}' \leftarrow (1-\tau)\theta_2{}' + \tau\theta_2$
      Update actor target network:
      $\varphi' \leftarrow \tau\varphi + (1-\tau)\varphi'$
    end if
    $n \leftarrow n + 1$
  until termination condition met ($n = N$ or UAV reaches target at step N)
end for

4. Numerical Results

Numerical simulations are conducted on a workstation equipped with an Intel Core i7-12700 CPU, 64 GB RAM, and an NVIDIA RTX 3060Ti GPU, implemented in Python 3.7. The UAV starts from the initial position $u_{uav}^I = (0, 0, 95)^T$ and flies toward the destination $u_{uav}^F = (1000, 1000, 95)^T$. It maintains a fixed altitude throughout the flight, reflecting a typical low-altitude urban operation scenario in MEC-enabled wireless networks.

The main simulation parameters, including transmission power, communication bandwidth, UAV battery capacity, and channel-related settings, are summarized in Table 2. We set simulation parameters by referring to realistic system specification and commonly used values in related literature, such as [20,22,29]. This alignment ensures both practical relevance and a fair basis for comparison with baseline algorithms.

Table 2. Simulation Parameters.

Category	Parameter	Value
Environment	Time horizon	40 s
	Number of slots	40
	Maximum velocity of UAV	20 m/s
	Maximum energy consumption of UAV	$E_{\text{uav}}^{\max} = 15{,}000$ J
Communication	Transmission power of each IoT device	20 dBm
	Transmission power of the ground jammer	20 dBm
	Minimum secure rate for each IoT device	10.0 bps/Hz
	Carrier frequency	2 GHz
	Reference channel gain	$\rho = -60$ dB
	Noise power at UAV receivers	$P_{Ni} = -100$ dBm
	Path loss for LoS links	$\eta_{\text{LoS}} = 0.1$ dB
	Path loss for NLoS links	$\eta_{\text{NLoS}} = 21$ dB
	Bandwidth	$W = 1$ MHz
Computation	Compute cycles per bit for IoT device	300 cycle/bit
	Compute cycles per bit for UAV	$c_{\text{uav}} = 100$ cycle/bit
	Computing frequency of IoT device	$f_{\text{loc}}^{\max} = 1$ GHz
	UAV computing frequency	$f_{\text{uav}}^{\max} = 5$ GHz
	Effective capacitance coefficient	$\xi = 10^{-28}$
DRL & Training	Replay buffer capacity	$\bar{\omega} = 1 \times 10^5$
	Maximum number of episodes	$M = 6000$
	Discount factor	$\gamma = 0.99$
	Learning rate of critic and actor	$l_r = 0.0001$
	Batch size	$B = 256$
	Target network soft-update rate	$\tau = 0.005$
	HER ratio	0.8

In the learning framework, all actor and critic networks are implemented as two-layer fully connected feedforward neural networks. Each layer contains 400 neurons. The actor network employs the $\tanh(\cdot)$ activation function to bound continuous actions within a normalized range. The critic networks adopt the ReLU activation function to enable efficient gradient propagation.

To comprehensively evaluate the performance of the proposed TD3+PER+HER algorithm, we consider the following benchmark schemes for comparison:

1. TD3: A standard TD3-based joint optimization algorithm without replay enhancement.
2. TD3+PER: TD3 with prioritized experience replay.
3. TD3+HER: TD3 with hindsight experience replay.
4. DDPG: A deep deterministic policy gradient algorithm for continuous control.
5. DDQN: A double deep Q-network algorithm designed for discrete action spaces.
6. Without trajectory optimization: A joint optimization of computation time allocation and offloading policy under a fixed UAV trajectory.

4.1. Convergence and Learning Performance Analysis

We consider a scenario with 10 IoT devices, each generating computational tasks of 10 Mb, and three eavesdroppers randomly distributed within the UAV coverage area. These settings represent a moderately dense IoT environment with realistic offloading demands and security threats.

Figure 4 compares the reward curves of the proposed algorithm with those of the baseline methods, while also presenting the ablation study results for PER and HER. It can be observed that the proposed TD3+PER+HER framework converges achieves the highest reward among all compared methods. It reaches a stable reward after approximately 1200 episodes, whereas TD3+HER, TD3+PER, TD3, and DDPG require considerably more training episodes to converge. Compared with the TD3 and DDPG methods, DDQN maintains a relatively low reward level and exhibits noticeable fluctuations throughout the training process, indicating that it is prone to converging to unstable or suboptimal policies. This is because DDQN is inherently designed for discrete action spaces, whereas the formulated optimization problem involves continuous control variables.

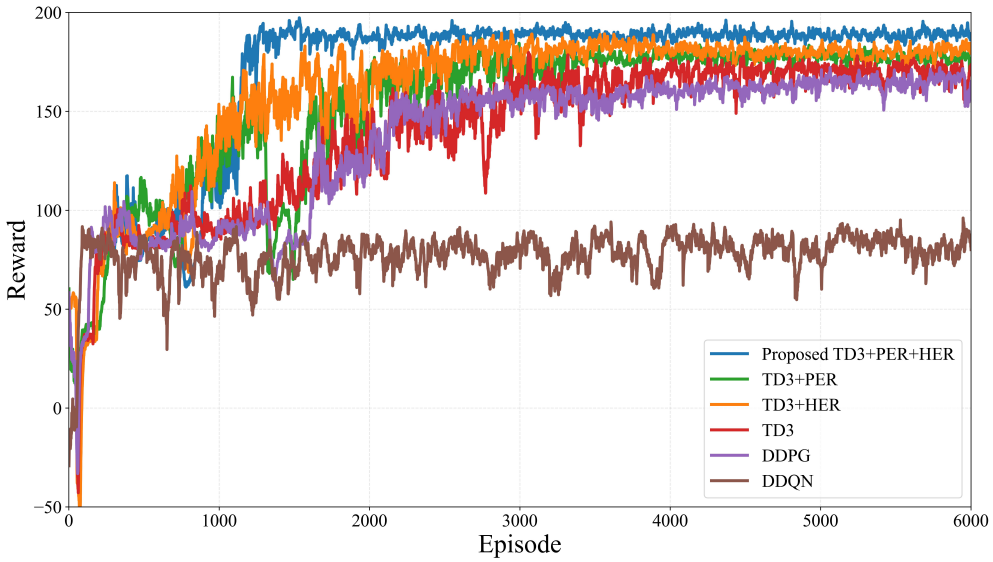


Figure 4. Convergence performance comparison of different DRL algorithms.

The ablation results further demonstrate the effectiveness of PER and HER. Compared with the original TD3 algorithm, incorporating either PER or HER individually leads to noticeable performance improvements. PER enhances sample efficiency by prioritizing informative experiences, while HER alleviates the sparse reward issue by generating additional successful experiences. Their combination further improves both convergence speed and the final reward, confirming the complementary advantages of the two replay mechanisms.

4.2. Performance Impact Analysis of Key Factors

We further investigate the effects of three key parameters, i.e., the energy preference level, the number of eavesdroppers, and the transmission power of IoT devices, on the system performance.

Figure 5 illustrates that the average system utility increases with IoT device transmit power $P_k[n]$ under different values of c_k. As $P_k[n]$ increases, the average system utility improves because IoT devices can offload more computational tasks to the UAV, thereby increasing the secure computation capability. However, for any given $P_k[n]$, the utility decreases with larger c_k. A higher c_k means that more CPU cycles are required for processing one bit of data, which leads to longer computation delays and higher energy, consequently reducing overall utility.

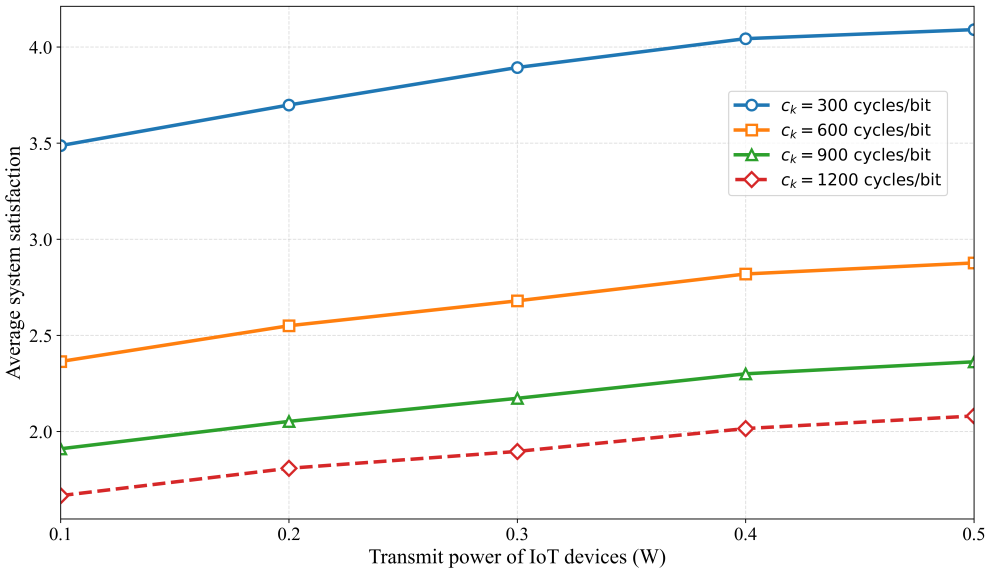


Figure 5. Average system utility versus IoT device transmit power under different (c_k).

Figure 6 shows the average system utility achieved under three representative scenarios, i.e., delay-sensitive, energy-sensitive, and delay-energy trade-off settings. By assigning different weighting coefficients, the proposed utility model can flexibly balance latency and energy consumption to satisfy diverse application requirements. Based on the utility model defined in (20), as α_k increases, the system's performance preference gradually shifts from energy consumption to delay minimization. Specifically, Scenario 1 adopts $\alpha_k = 0.1$, representing delay-sensitive applications where timely task completion is prioritized. Scenario 2 uses $\alpha_k = 0.9$, corresponding to energy-intensive tasks or devices operating under low-battery conditions. Scenario 3 assigns $\alpha_k = 0.5$, implying equal priority on delay and energy during secure offloading. The transmission power of each IoT device is set at $P_k[n] = 0.1$ W, consistent with typical low-power edge computing configurations. As indicated in Figure 4, across all scenarios, the proposed algorithm consistently achieves the highest utility compared with TD3, DDPG, and DDQN, demonstrating its flexible adaptability to diverse system requirements.

Figure 7 shows that the average secure computation capacity decreases as the number of eavesdroppers increases. According to the super eavesdropper model, a larger set of colluding eavesdroppers enhances the aggregated interception capability, thereby increasing the eavesdropping rate, and degrades link security. In this experiment, the transmission power of each IoT device is set to $P_k[n] = 0.1$ W and the weight factor $\alpha_k = 0.5$. Across all cases, the proposed algorithm consistently achieves higher secure computation capacity than DDQN, DDPG, and TD3, and its performance advantage remains stable even under severe eavesdropping conditions. Figure 8 further shows that the average secure computation capacity increases with the transmission power of IoT devices when the number of eavesdroppers is fixed ($S = 3$). A higher transmission power enables IoT devices to offload more computational tasks to the UAV for secure processing. Under the same parameter settings, the proposed algorithm again outperforms all baselines across different power levels. These results verify the robustness and adaptability of the proposed method under varying numbers of eavesdroppers and power constraints.

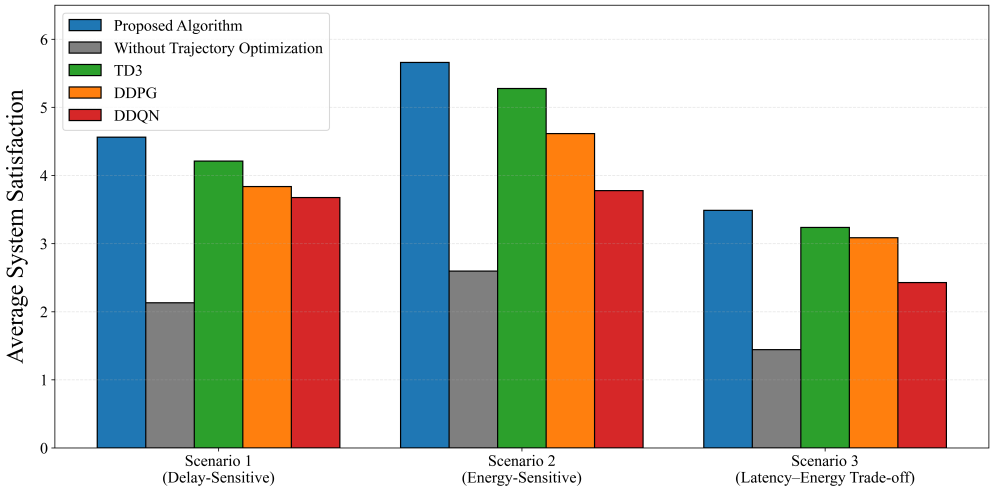


Figure 6. Comparison of average system utility achieved under diverse scenarios.

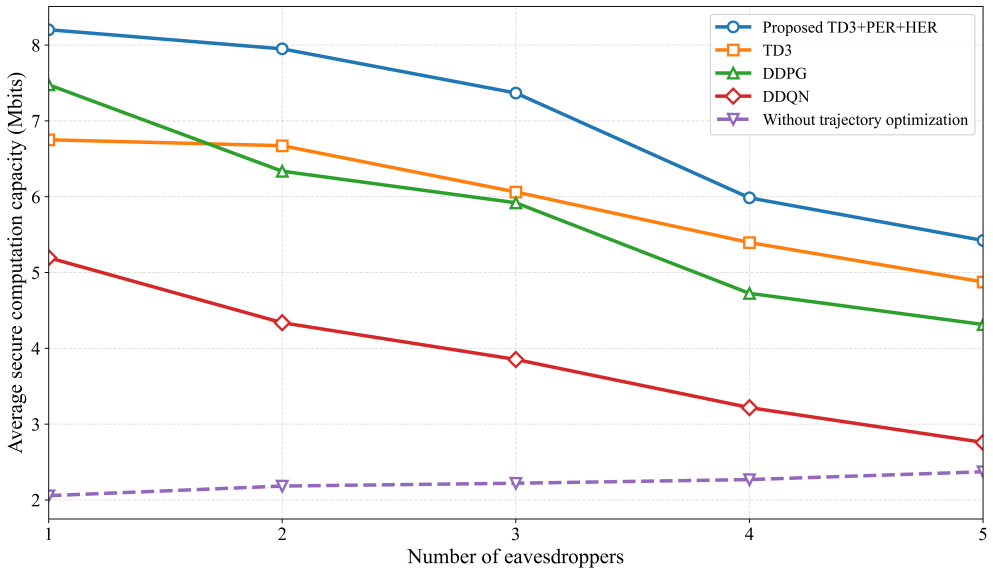


Figure 7. Average secure computation capacity versus the number of cooperative eavesdroppers.

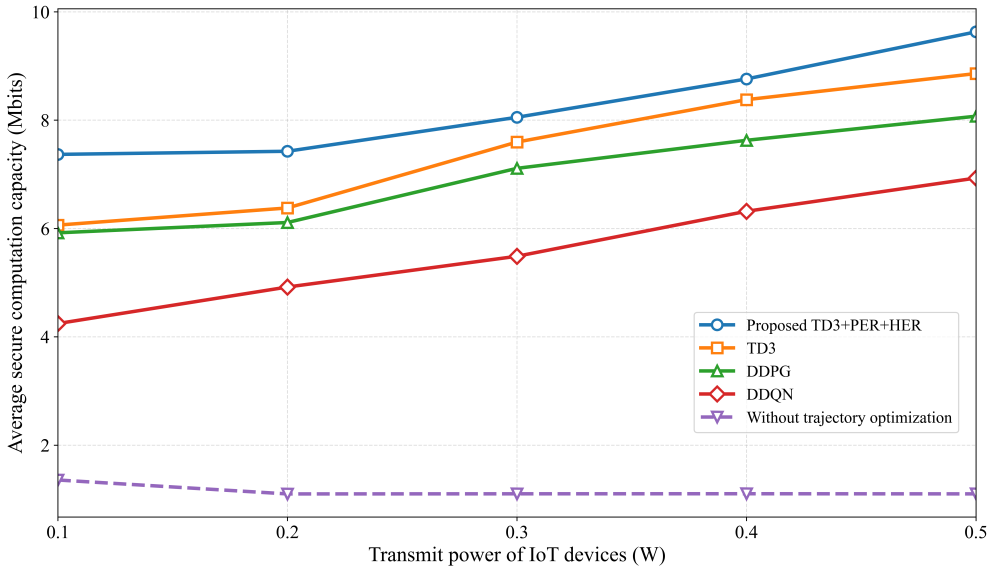


Figure 8. Average secure computation capacity versus IoT device transmit power.

5. Conclusions

This paper investigated a UAV-assisted MEC system in the presence of multiple colluding eavesdroppers. We formulated a joint optimization problem that coordinates task-offloading decisions, time allocation, and UAV trajectory design with the objective of maximizing the average system utility. The formulated non-convex problem was solved using a novel TD3-based algorithm enhanced with PER and spatial HER mechanism. The proposed hybrid PER–HER framework significantly improves training stability and sample efficiency in complex secure MEC environments. The experimental results demonstrate that the proposed algorithm achieves faster and more stable convergence than DDPG, DDQN, and the baseline TD3 algorithms. Moreover, under diverse system settings, including different user preference levels, varying numbers of eavesdroppers, and IoT device transmission powers, the proposed method consistently attains the highest average system utility and secure computation bits. These numerical results demonstrate the effectiveness of the proposed learning framework for UAV-assisted secure MEC systems facing cooperative eavesdropping threats under the considered simulation setup. Nevertheless, it should be noted that practical performance in real environments requires further validation via hardware prototypes and field experiments. Additionally, future research could extend the framework to multi-UAV coordinate networks, distributed task allocation, and multi-agent reinforcement learning strategies. Further investigations could also explore dynamic environmental adaptability, more sophisticated eavesdropper models, energy-harvesting techniques, and integration with heterogeneous network architectures. Such extensions would significantly enhance the practicality and security of UAV-assisted MEC systems in large-scale IoT deployments.

Author Contributions: Conceptualization, Y.Z. and Y.S.; methodology, Y.Z.; software, Y.Z.; writing—original draft preparation, Y.Z.; writing—review and editing, Y.Z., Y.S., Z.D. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article. Further inquiries can be directed to the corresponding author.

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

References

1. Porambage, P.; Okwuibe, J.; Liyanage, M.; Ylianttila, M.; Taleb, T. Survey on multi-access edge computing for internet of things realization. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2961–2991. [CrossRef]
2. Zhang, Q.; Luo, Y.; Jiang, H.; Zhang, K. Aerial edge computing: A survey. *IEEE Internet Things J.* **2023**, *10*, 14357–14374. [CrossRef]
3. Zhou, Y.; Pan, C.; Yeoh, P.L.; Wang, K.; Elkashlan, M.; Vucetic, B.; Li, Y. Secure communications for UAV-enabled mobile edge computing systems. *IEEE Trans. Commun.* **2019**, *68*, 376–388. [CrossRef]
4. Huda, S.A.; Moh, S. Survey on computation offloading in UAV-Enabled mobile edge computing. *J. Netw. Comput. Appl.* **2022**, *201*, 103341. [CrossRef]
5. Zhou, H.; Jiang, K.; Liu, X.; Li, X.; Leung, V.C. Deep reinforcement learning for energy-efficient computation offloading in mobile-edge computing. *IEEE Internet Things J.* **2021**, *9*, 1517–1530. [CrossRef]
6. Wang, K.; Wang, L.; Pan, C.; Ren, H. Deep reinforcement learning-based resource management for flexible mobile edge computing: Architectures, applications, and research issues. *IEEE Veh. Technol. Mag.* **2022**, *17*, 85–93. [CrossRef]
7. Ding, Y.; Han, H.; Lu, W.; Wang, Y.; Zhao, N.; Wang, X.; Yang, X. DDQN-based trajectory and resource optimization for UAV-aided MEC secure communications. *IEEE Trans. Veh. Technol.* **2023**, *73*, 6006–6011. [CrossRef]
8. Karmakar, R.; Kaddoum, G.; Akhrif, O. A novel federated learning-based smart power and 3D trajectory control for fairness optimization in secure UAV-assisted MEC services. *IEEE Trans. Mob. Comput.* **2023**, *23*, 4832–4848. [CrossRef]
9. Liu, S.; Yang, H.; Xiao, L.; Zheng, M.; Lu, H.; Xiong, Z. Learning-based resource management optimization for UAV-assisted MEC against jamming. *IEEE Trans. Commun.* **2024**, *72*, 4873–4886. [CrossRef]

10. Michailidis, E.T.; Volakaki, M.G.; Miridakis, N.I.; Vouyioukas, D. Optimization of secure computation efficiency in UAV-enabled RIS-assisted MEC-IoT networks with aerial and ground eavesdroppers. *IEEE Trans. Commun.* **2024**, *72*, 3994–4009. [CrossRef]
11. Liu, Y.; Yan, J.; Zhao, X. Deep reinforcement learning based latency minimization for mobile edge computing with virtualization in maritime UAV communication network. *IEEE Trans. Veh. Technol.* **2022**, *71*, 4225–4236. [CrossRef]
12. Xu, W.; Song, Z.; Gao, Z.; Lai, L.; Sun, Y.; Luo, W. Latency-aware MIoT service strategy in UAV-assisted dynamic MMEC environment. *IEEE Internet Things J.* **2024**, *11*, 22220–22231. [CrossRef]
13. Chen, Y.; Yang, Y.; Wu, Y.; Huang, J.; Zhao, L. Joint trajectory optimization and resource allocation in UAV-MEC systems: A Lyapunov-assisted DRL approach. *IEEE Trans. Serv. Comput.* **2025**, *18*, 854–867. [CrossRef]
14. Wang, M.; Shi, S.; Zhang, D.; Wu, C.; Wang, Y. Joint computation offloading and resource allocation for MIMO-NOMA assisted multi-user MEC systems. *IEEE Trans. Commun.* **2023**, *71*, 4360–4376. [CrossRef]
15. Pervez, F.; Sultana, A.; Yang, C.; Zhao, L. Energy and latency efficient joint communication and computation optimization in a multi-UAV-assisted MEC network. *IEEE Trans. Wirel. Commun.* **2023**, *23*, 1728–1741. [CrossRef]
16. Tian, J.; Wang, D.; Zhang, H.; Wu, D. Service satisfaction-oriented task offloading and UAV scheduling in UAV-enabled MEC networks. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 8949–8964. [CrossRef]
17. Miao, J.; Bai, S.; Mumtaz, S.; Zhang, Q.; Mu, J. Utility-oriented optimization for video streaming in UAV-aided MEC network: A DRL approach. *IEEE Trans. Green Commun. Netw.* **2024**, *8*, 878–889. [CrossRef]
18. Bai, T.; Wang, J.; Ren, Y.; Hanzo, L. Energy-efficient computation offloading for secure UAV-edge-computing systems. *IEEE Trans. Veh. Technol.* **2019**, *68*, 6074–6087. [CrossRef]
19. Dai, D.; Liang, B. Joint Optimization of UAV Trajectory, User Scheduling, and Resource Allocation for Secure MEC Systems in IoT Using Deep Reinforcement Learning. *J. Circuits Syst. Comput.* **2025**, *34*, 2550255. [CrossRef]
20. Gu, Y.; Ma, Y.; Wang, X.; Shah, A. Security energy efficiency optimization and analysis of aerial-IRS-assisted UAV-MEC system. *IEEE Access* **2024**, *12*, 118953–118967. [CrossRef]
21. Liu, S.; Yu, Y.; Guo, L.; Yeoh, P.L.; Vucetic, B.; Li, Y.; Duong, T.Q. Satisfaction-maximized secure computation offloading in multi-eavesdropper MEC networks. *IEEE Trans. Wirel. Commun.* **2021**, *21*, 4227–4241. [CrossRef]
22. Wang, Y.; Gao, Z.; Zhang, J.; Cao, X.; Zheng, D.; Gao, Y.; Ng, D.W.K.; Di Renzo, M. Trajectory design for UAV-based Internet of Things data collection: A deep reinforcement learning approach. *IEEE Internet Things J.* **2021**, *9*, 3899–3912. [CrossRef]
23. Wang, Y.; Shi, W.; Huang, M.; Shu, F.; Wang, J. Intelligent reflecting surface aided secure transmission with colluding eavesdroppers. *IEEE Trans. Veh. Technol.* **2022**, *71*, 10155–10160. [CrossRef]
24. Sun, H.; Zhou, F.; Hu, R.Q. Joint offloading and computation energy efficiency maximization in a mobile edge computing system. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3052–3056. [CrossRef]
25. Zeng, Y.; Xu, J.; Zhang, R. Energy minimization for wireless communication with rotary-wing UAV. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 2329–2345. [CrossRef]
26. Mohajer, A.; Hajipour, J.; Leung, V.C. Dynamic offloading in mobile edge computing with traffic-aware network slicing and adaptive TD3 strategy. *IEEE Commun. Lett.* **2024**, *29*, 95–99. [CrossRef]
27. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized experience replay. *arXiv* **2015**, arXiv:1511.05952.
28. Andrychowicz, M.; Wolski, F.; Ray, A.; Schneider, J.; Fong, R.; Welinder, P.; McGrew, B.; Tobin, J.; Pieter Abbeel, O.; Zaremba, W. Hindsight experience replay. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–11. [CrossRef]
29. Zhou, Y.; Ma, Z.; Liu, G.; Zhang, Z.; Yeoh, P.L.; Vucetic, B.; Li, Y. Secure multi-layer MEC systems with UAV-enabled reconfigurable intelligent surface against full-duplex eavesdropper. *IEEE Trans. Commun.* **2023**, *72*, 1565–1577. [CrossRef]

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