

Article

Federated Twin Delayed Deep Deterministic Policy Gradient for Delay and Energy Consumption Optimization in Urban Air Mobility with UAV-Assisted MEC

Chunyu Pan ^{1,*} , Zhonghao Luo ¹, Jiuchuan Zhang ¹, Lei Shi ^{2,3,4} , Jirong Yi ⁵ and Zhaohui Yang ⁶

¹ Key Laboratory of Information and Communication Systems, Ministry of Information Industry, and Key Laboratory of Modern Measurement & Control Technology, Ministry of Education, Beijing Information Science and Technology University, Beijing 102206, China; zhonghao.luo@bistu.edu.cn (Z.L.)

² State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing 100024, China; leiky_shi@bupt.cn

³ Key Laboratory of Intelligent Game, Yangtze River Delta Research Institute of NPU, Taicang 215400, China

⁴ Key Laboratory of Education Informatization for Nationalities (Yunnan Normal University), Ministry of Education, Kunming 650092, China

⁵ Artificial Intelligence in Medicine, Cedars Sinai Medical Center, Los Angeles, CA 90048, USA; jirong.yi@hotmail.com

⁶ College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China; yang_zhaohui@zju.edu.cn

* Correspondence: chunyupan@bistu.edu.cn

Abstract: With the rapid expansion of urban populations and the accelerated pace of urbanization, the concept of urban air mobility (UAM) has emerged. During flights, UAM aircraft need to transmit real-time sensing information to base stations for further processing and analysis. Large-scale real-time data require leveraging the computing capabilities of edge servers at the network edge to reduce transmission delay and energy consumption of UAM aircraft. In cases where edge servers are unable to process information, an unmanned aerial vehicle (UAV) equipped with computing capabilities and operating in low-altitude airspace can serve as a relay to assist in communication and computation. Due to the limited payloads and flight times of UAVs and UAM aircraft, delay and energy consumption within the system pose significant challenges. To tackle these challenges, two fundamental objectives have been proposed: minimizing delay and minimizing energy consumption. Furthermore, an optimization problem has been proposed to minimize the weighted sum of delay and energy consumption. Then, a UAM federated twin delayed deep deterministic policy gradient (UF-TD3) algorithm has been proposed to solve the original problems characterized by complex, non-convex, and inseparable variables. Simulation results show that the proposed UF-TD3 algorithm converges quickly and significantly outperforms four other baseline algorithms in optimizing delay and energy consumption performance. Moreover, compared to the conventional delay minimization strategy and energy minimization strategy, the proposed strategy of minimizing the weighted sum of delay and energy consumption can reduce the delay by 63.8% and reduce energy by 73.96%.

Keywords: urban air mobility; mobile edge computing; resource allocation; federated learning; reinforcement learning; emergency scenario



Academic Editor: Angelo Trotta

Received: 9 January 2025

Revised: 7 February 2025

Accepted: 10 February 2025

Published: 12 February 2025

Citation: Pan, C.; Luo, Z.; Zhang, J.; Shi, L.; Yi, J.; Yang, Z. Federated Twin Delayed Deep Deterministic Policy Gradient for Delay and Energy Consumption Optimization in Urban Air Mobility with UAV-Assisted MEC. *Drones* **2025**, *9*, 137. <https://doi.org/10.3390/drones9020137>

Copyright: © 2025 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Background and Related Works

The concept of urban air mobility (UAM) has emerged in response to the numerous challenges faced by ground transportation systems. It aims to develop aircraft that operate in urban airspace [1,2]. The introduction of UAM not only significantly reduces urban commuting time and improves overall traffic flow efficiency but also shows great potential in mitigating the environmental impact of transportation. This advancement promotes cities toward more sustainable and livable development, gradually becoming an indispensable part of future urban transportation systems [3]. Due to the high environmental sensitivity of low-altitude flight operations, real-time communication with ground systems is crucial, but this also presents challenges in communication delay. Typically, reducing delay by increasing the signal transmission frequency and boosting transmission power can be effective, but this significantly increases energy consumption [4,5]. Given the limited payload capacity of aircraft, the increase in energy consumption poses a significant constraint on safe flying. Therefore, balancing the impact of delay and energy consumption on the system while meeting performance requirements has become a crucial issue for advancing UAM development.

Mobile edge computing (MEC), as a distributed computing architecture, allows resource-constrained aircraft to offload computing tasks to nearby resource-rich MEC servers to reduce delay and energy consumption [6–8]. Many scholars have conducted research on how to utilize MEC to reduce system delay and energy consumption. In Ref. [9], the authors investigated the dynamic resource management problem of the joint transmission power control and computing resource allocation in the industrial Internet of Things. Specifically, the authors proposed a deep reinforcement learning-based dynamic resource management algorithm to minimize the long-term average delay. The task prioritization and cognitive eavesdropping in a multi-user MEC system were addressed in Ref. [10]; the authors used federated learning and deep reinforcement learning to optimize offloading ratios, bandwidth, and computational resource allocation, effectively reducing delay and energy consumption. Ref. [11] addressed industrial task co-offloading in device-to-device-assisted MEC networks, employing a learning-based co-offloading algorithm to optimize task delay and migration cost. Ref. [12] addressed the joint multi-task offloading problem for MEC systems in the satellite Internet of Things, employing a training method based on the attention mechanism and proximal policy optimization algorithm to optimize task priorities and dependencies, achieving system cost minimization. However, the aforementioned studies have not focused on the UAM scenario. Given the high dependency of UAM systems on data transmission and computation, it is crucial to conduct research on edge computing for urban air traffic networks, particularly under emergency scenarios.

In UAM systems, manned and cargo services require secure data transmission [13]. In emergency scenarios such as MEC or base station failures, it is very difficult to ensure low-delay and low-energy consumption for data processing, computation, and transmission [14,15]. With controllable mobility, low-cost, and strong line-of-sight links, unmanned aerial vehicles (UAVs) can carry MEC servers located in the same low-altitude area to assist UAM aircraft in computational offloading [16–18]. UAVs can also serve as flying base stations (BSs), relaying the data from UAM aircraft to ground base stations [19,20]. This is expected to become a significant contributor to emergency information dissemination [21]. Ref. [22] proposed a MEC system that includes several ground servers at access points and an aerial server carried by a UAV. The primary focus of this research is the computational efficiency of the MEC system, aiming to enhance processing efficiency and response speed in edge computing. Building upon similar themes, the research documented in Ref. [23] explores a collaborative mobile edge computing framework utilizing multiple UAVs and

ECs, tackling the task offloading challenge to reduce the sum of execution delays and energy consumptions through the integrated design of trajectories, computation task allocation, and communication resource management. In Ref. [24], the authors explored a UAV relay communication scenario for uplink transmissions within a 6G network MEC system, employing an adaptive genetic algorithm to minimize task completion time under resource allocation constraints. In Ref. [25], the authors used reinforcement learning algorithms to reduce the computation delay in UAV-assisted MEC networks. In our previous work, Ref. [26], the resource scheduling problem in the UAV-assisted fault-prone MEC network was addressed, ensuring the low-delay requirements of the communication network. The above studies have high delay and energy consumption in data transmission.

To address the high delay and energy consumption problems, federated learning (FL) is introduced into the MEC network [27]. As a distributed architecture, FL allows devices to use their local data to collaboratively train the machine learning model required by the FL server, and then send the model updates back to the FL server for aggregation [28–30]. This approach replaces data transmission with the exchange of model parameters between nodes, effectively reducing communication delay and energy consumption [31]. In Ref. [32], the authors studied the optimization design of FL in MEC systems. By jointly optimizing model compression, sample selection, and user selection strategies, they significantly reduced the accuracy loss and cost of FL in MEC systems. In [33], the authors investigated an FL framework within a vehicular network, aiming to maximize model accuracy while meeting latency and energy constraints. In [34], the authors proposed a joint dataset and computation management scheme for energy-efficient FL in MEC, which jointly optimizes the dataset sizes used by multiple FL terminals for local training and the computation resources to balance learning efficiency and energy consumption in FL. However, the research on utilizing FL algorithms in the UAM system is still limited.

In [35], the authors combined reinforcement learning and FL to study the issues of UAM system delay and energy consumption brought about by edge computing. For UAM systems, utilizing FL to assist edge computing offers many advantages. Firstly, FL enables model training with localized flight data, effectively avoiding centralized storage and transmission of flight data, thereby reducing the risk of privacy breaches. Secondly, through FL, multiple data sources (such as flight data, urban traffic data, and meteorological data) can share learning outcomes while maintaining data independence, thereby enhancing the model's generalization ability and prediction accuracy. Lastly, FL conducts model training through distributed computation, where each node only needs to transmit model updates instead of the data, significantly reducing communication overhead and delay, and improving system response speed.

1.2. Main Contributions and Organization

In this paper, we consider an emergency scenario in a multi-UAM mobile edge computing (MUAM-MEC) system where the edge server in a particular UAM system becomes inoperative due to a failure. By utilizing relay UAVs, UAM aircraft in the affected system will be able to choose to offload tasks, either entirely or partially, to edge servers of other systems or execute them locally. To study the computation offloading and resource allocation strategy under this scenario, we developed a joint delay-energy minimization strategy and proposed a UAM federated twin delayed deep deterministic policy gradient algorithm (UF-TD3) to solve the problem. Moreover, we also investigated the convergence performance of this algorithm as well as its joint optimization performance for system delay and energy consumption. The main contributions of this paper are summarized as follows.

1. An emergency edge computing structure for multi-UAM systems is proposed in this paper. Specifically, when an edge server fails, UAM aircraft offload part or all of their

sensory information to other edge servers via relay UAVs in the same low-altitude area for task processing. Accordingly, we developed an offloading model of the UAM network that consists of four transmission links: UAM aircraft to relay UAV, relay UAV to BS, BS to relay UAV, and relay UAV to UAM aircraft, along with a computational model that consists of local computing for UAM aircraft. Based on the offloading model and computational model, a delay model and energy model for the UAM system are proposed.

2. Two baseline strategies for computation offloading and resource allocation are proposed: A delay minimization strategy and an energy minimization strategy, based on the joint optimization of UAM aircraft transmit power, offloading rate, and computation frequency. A joint delay-energy minimization is then proposed, the goal of which is to minimize both the total UAM system delay and energy consumption in order to lower the overall UAM system cost.
3. To address the complex non-convex problem proposed in this paper, a UF-TD3 algorithm is proposed, where each TD3 algorithm runs on edge servers of UAM systems and shares model parameters with a cloud node server. The cloud server applies federated averaging and returns the averaged parameters to the edge servers, allowing multiple TD3 algorithms to jointly optimize UAM system delay and energy consumption. The simulation results verify the effectiveness of the proposed algorithm, and meanwhile, we analyze the convergence of the algorithm and determine the optimal learning rate. Additionally, the impacts of different strategies and various parameter settings on UAM system performance were examined.

The remainder of this paper is organized as follows. In Section 2, we introduce the system model of the MUAM-MEC in an emergency scenario and the task offloading computation process. In Section 3, we elaborate on the two baseline strategies and the joint delay-energy minimization strategy. In Section 4, we present a computation offloading and resource allocation algorithm, detailing the algorithm framework and process. In Section 5, we discuss the convergence of the proposed algorithm by setting simulation parameters and comparing it with the two baseline strategies to demonstrate the effectiveness of the joint delay-energy minimization strategy. In Section 6, we summarize the research findings and present our conclusions.

2. System Model

In this paper, we consider a structure comprising $N + 1$ UAM systems, each of which encompasses M UAM aircrafts. As shown in Figure 1, when the edge server of one of the UAM systems encounters an issue, we define it as the target UAM subsystem. The N neighboring subsystems will dispatch relay UAVs to assist with task computation offloading for the target subsystem.

The computation offloading process is illustrated in Figure 2. In the proposed scenario, we assume that the edge server resources of neighboring subsystems are sufficient to support the UAM aircraft in the target UAM system for task offloading. Task offloading occurs in three situations. One offloading situation involves offloading the entire task load to a single or multiple relay UAVs, which then forward the tasks to the capable edge servers. The second offloading strategy involves offloading a portion of the tasks to a single or multiple relay UAVs for forwarding to the edge server, while the remaining tasks are computed locally on the UAM aircraft. When the overall task volume is small enough to be processed on a terminal UAM aircraft, the tasks will be processed locally.

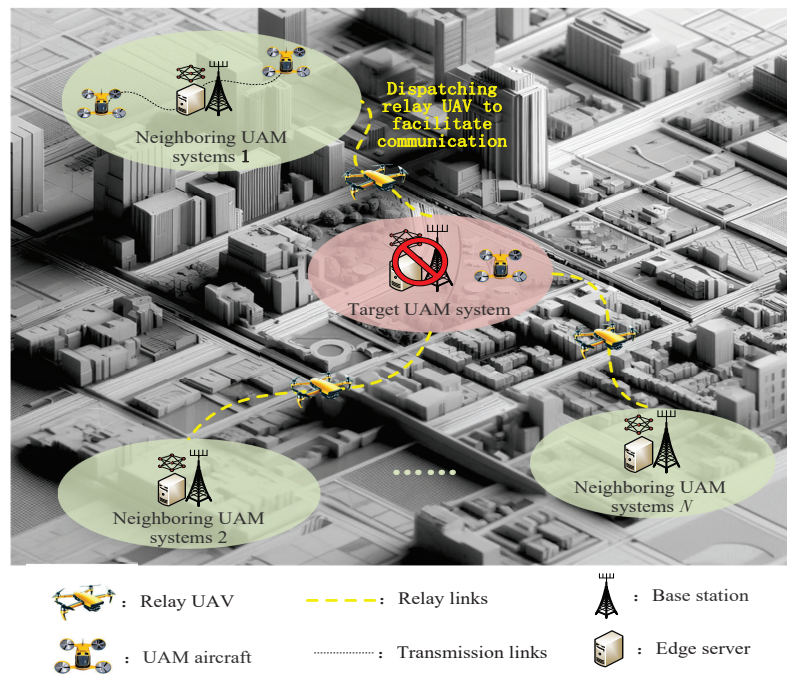


Figure 1. The MUAM-MEC system model.

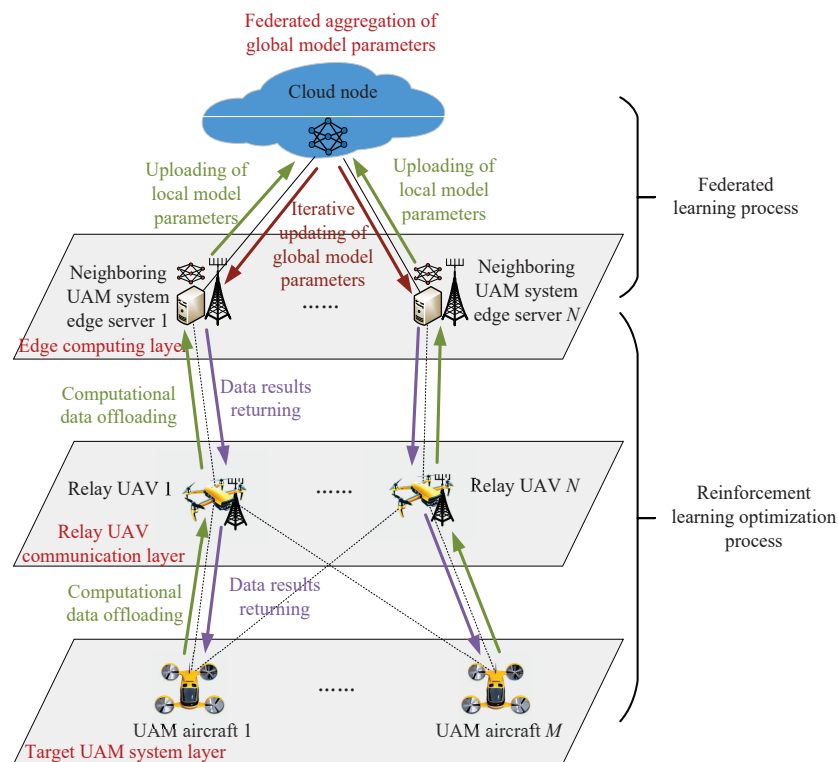


Figure 2. The computation offloading process in the MUAM-MEC system.

We assume that each task processed by the MEC system represents one processing period, denoted by $s, s \in \{1, 2, \dots, S\}$. During period s , the three-dimensional coordinates of the UAM aircraft m and the relay UAV n are denoted by $u_{m,s}^{\text{UAM}} = (x_{m,s}^{\text{UAM}}, y_{m,s}^{\text{UAM}}, z_{m,s}^{\text{UAM}})$ and $u_{n,s}^{\text{UAV}} = (x_{n,s}^{\text{UAV}}, y_{n,s}^{\text{UAV}}, z_{n,s}^{\text{UAV}})$, respectively, and the coordinate of BS n is denoted by $u_n^{\text{BS}} = (x_n^{\text{BS}}, y_n^{\text{BS}}, z_n^{\text{BS}})$. For better clarity of the system to be presented subsequently, we list the key parameters in Table 1.

Table 1. Key parameters (in order of appearance).

Notation	Description	Notation	Description
N	Number of neighboring UAM subsystems	M	Number of UAM aircraft
S	Number of periods	$I_{m,s}^{\text{UAM}}$	The task volume of the UAM aircraft m
$u_{m,s}^{\text{UAM}}$	The coordinates of the UAM aircraft m	$u_{n,s}^{\text{UAV}}$	The coordinates of the relay UAV n
u_n^{BS}	The coordinates of the BS n	N_0	The noise power
β_0	The channel gain at a reference distance of 1 m	$O_{m,n,s}$	The offloading ratio from UAM aircraft m to relay UAV n
$R_{m,n,s}^{\text{UAM}}, R_{m,n,s}^{\text{UAV}}$	Uplink rate from UAM aircraft m to relay UAV n and relay UAV n to BS n	$B^{\text{UAM}}, B^{\text{UAV}}$	The uplink bandwidth of the UAM aircraft and the relay UAV
$p_{m,n,s}^{\text{UAM}}, p_{m,n,s}^{\text{UAV}}$	Uplink power of the UAM aircraft m and relay UAV n	$T_{m,n,s}^{\text{UAM}}, T_{m,n,s}^{\text{UAV}}$	The uplink transmission delay from UAM aircraft m to relay UAV n and relay UAV n to BS
$T_{\text{total},m,s}^{\text{UAM}}, T_{\text{total},m,s}^{\text{UAV}}$	The uplink transmission delay of UAM aircraft m to all relay UAVs and all relay UAVs to all BSs	$E_{m,n,s}^{\text{UAM}}, E_{m,n,s}^{\text{UAV}}$	The uplink transmission energy of UAM aircraft m to relay UAV n and relay UAV n to BS
$E_{\text{total},m,s}^{\text{UAM}}, E_{\text{total},m,s}^{\text{UAV}}$	The uplink transmission energy of UAM aircraft m to all relay UAVs and all relay UAVs to all BSs	$\vartheta_{m,n,s}$	The data volume coefficient of the task processing results
$I_{m,n,s}^{\text{BS}}$	The volume of data resulting from task processing	$R_{m,n,s}^{\text{BS}}, R_{m,n,s}^{\text{back}}$	Downlink rate from BS to relay UAV n and relay UAV n to UAM aircraft m
$B^{\text{BS}}, B^{\text{back}}$	The downlink bandwidth of the BS and the relay UAV	$p_{m,n,s}^{\text{BS}}, p_{m,n,s}^{\text{back}}$	Downlink power of the BS and the relay UAV
$T_{m,n,s}^{\text{BS}}, T_{m,n,s}^{\text{back}}$	The downlink transmission delay from BS to relay UAV n and relay UAV n to UAM aircraft m	$T_{\text{total},m,s}^{\text{BS}}, T_{\text{total},m,s}^{\text{back}}$	The downlink transmission delay from all BSs to relay UAVs and all relay UAVs to UAM aircraft m
$E_{m,n,s}^{\text{back}}$	The downlink energy from relay UAV n to UAM aircraft m	$E_{\text{total},m,s}^{\text{back}}$	The downlink total energy of all relay UAVs
$T_{\text{total},m,s}^{\text{trans}}, E_{\text{total},m,s}^{\text{trans}}$	The total delay and energy of the uplink and downlink transmission process for UAM aircraft m	$I_{m,s}^{\text{local}}$	The volume of tasks computed locally by UAM aircraft m
$T_{m,s}^{\text{comp}}, E_{m,s}^{\text{comp}}$	The local computation delay and energy of UAM aircraft m	Q	The number of CPU cycles required to process one byte of data
$p_{m,s}^{\text{comp}}$	The power consumption during tasks computation	κ	The CPU's effective capacitance parameter

2.1. Uplink and Downlink Transmission Model

2.1.1. UAM Aircraft to Relay UAV

In this paper, we assume that the time-division multiple access (TDMA) technique is used to allow multiple UAVs or UAM aircraft to share the same frequency band without interfering with each other [36,37]. Then, we adopt the UAV channel model from Ref. [38], which typically uses free-space path loss, implying that the loss is primarily dependent on

distance. Therefore, during period s , the channel power gain from UAM aircraft m to relay UAV n follows the free-space path loss model:

$$h_{m,n,s}^{\text{UAM}} = \frac{\beta_0}{\|u_{n,s}^{\text{UAV}} - u_{m,s}^{\text{UAM}}\|^2}, \quad (1)$$

where β_0 represents the channel gain at a reference distance of 1 m. And during period s , the uplink transmission rate from UAM aircraft m to relay UAV n is given by the following:

$$R_{m,n,s}^{\text{UAM}} = B^{\text{UAM}} \log_2 \left(1 + \frac{h_{m,n,s}^{\text{UAM}} p_{m,n,s}^{\text{UAM}}}{N_0} \right), \quad (2)$$

where B^{UAM} represents the bandwidth of the carrier signal transmitted by the UAM aircraft, $p_{m,n,s}^{\text{UAM}}$ denotes the transmission power of the carrier signal emitted by the UAM aircraft, and N_0 refers to the noise power.

The offloading delay from UAM aircraft m to a single relay UAV n with the uplink transmission rate $R_{m,n,s}^{\text{UAM}}$ is given by the following:

$$T_{m,n,s}^{\text{UAM}} = O_{m,n,s} \frac{I_{m,s}^{\text{UAM}}}{R_{m,n,s}^{\text{UAM}}}, \quad (3)$$

where $O_{m,n,s}$ represents the offloading ratio from UAM aircraft m to relay UAV n , and $I_{m,s}^{\text{UAM}}$ denotes the task volume of UAM aircraft m . During period s , since the UAM aircraft m offloads computing tasks to N relay UAVs simultaneously, the total offloading delay is represented by the maximum delay among these N relay UAVs, resulting in the following:

$$T_{\text{total},m,s}^{\text{UAM}} = \max \{ T_{m,1,s}^{\text{UAM}}, T_{m,2,s}^{\text{UAM}}, \dots, T_{m,N,s}^{\text{UAM}} \}. \quad (4)$$

The energy consumed by UAM aircraft m for offloading to a single relay UAV n is given by $E_{m,n,s}^{\text{UAM}} = p_{m,n,s}^{\text{UAM}} T_{m,n,s}^{\text{UAM}}$, and the total energy consumed by UAM aircraft m for offloading to all relay UAVs is $E_{\text{total},m,s}^{\text{UAM}} = \sum_{n=1}^N E_{m,n,s}^{\text{UAM}}$.

2.1.2. Relay UAV to Edge Server

During period s , the channel power gain from relay UAV n to its corresponding edge server follows the free-space path loss model as follows:

$$h_{m,n,s}^{\text{UAV}} = \frac{\beta_0}{\|u_{n,s}^{\text{BS}} - u_{m,s}^{\text{UAV}}\|^2}, \quad (5)$$

and the uplink transmission rate from relay UAV n to its corresponding edge server is given by the following:

$$R_{m,n,s}^{\text{UAV}} = B^{\text{UAV}} \log_2 \left(1 + \frac{h_{m,n,s}^{\text{UAV}} p_{m,n,s}^{\text{UAV}}}{N_0} \right), \quad (6)$$

where B^{UAV} represents the bandwidth of the uplink carrier signal transmitted by the relay UAV, and $p_{m,n,s}^{\text{UAV}}$ denotes the transmission power of the uplink carrier signal by the relay UAV. The transmission delay for relay UAV n to the edge server can be represented as follows:

$$T_{m,n,s}^{\text{UAV}} = O_{m,n,s} \frac{I_{m,s}^{\text{UAM}}}{R_{m,n,s}^{\text{UAV}}}, \quad (7)$$

and since N aircraft simultaneously offload tasks to edge servers, the total offloading transmission delay is represented by the maximum offloading transmission delay among these N aircraft, having $T_{\text{total},m,s}^{\text{UAV}} = \max\{T_{m,1,s}^{\text{UAV}}, T_{m,2,s}^{\text{UAV}}, \dots, T_{m,N,s}^{\text{UAV}}\}$.

The energy consumed by relay UAV n for offloading to its corresponding edge server is given by $E_{m,n,s}^{\text{UAV}} = p_{m,n,s}^{\text{UAV}} T_{m,n,s}^{\text{UAV}}$, and the total energy consumed for transmission from all relay UAVs to the edge server is $E_{\text{total},m,s}^{\text{UAV}} = \sum_{n=1}^N E_{m,n,s}^{\text{UAV}}$.

After the tasks have been processed by the edge servers, the BS connected to the edge server needs to send the data processing results back to the UAM aircraft.

2.1.3. Edge Server to Relay UAV

During period s , the volume of data resulting from task processing is as follows:

$$I_{m,n,s}^{\text{BS}} = \vartheta_{m,n,s} O_{m,n,s} I_{m,s}^{\text{UAM}}, \quad (8)$$

where $\vartheta_{m,n,s}$ represents the data volume coefficient of the task processing results. The downlink transmission rate from the base station to its corresponding relay UAV for forwarding data results can be represented as follows:

$$R_{m,n,s}^{\text{BS}} = B^{\text{BS}} \log_2 \left(1 + \frac{h_{m,n,s}^{\text{UAV}} p_{m,n,s}^{\text{BS}}}{N_0} \right), \quad (9)$$

where B^{BS} represents the bandwidth of the carrier signal transmitted by the base station, and $p_{m,n,s}^{\text{BS}}$ denotes the transmission power of the carrier signal emitted by the base station. The transmission delay from the BS to its corresponding relay UAV n is given by $T_{m,n,s}^{\text{BS}} = I_{m,n,s}^{\text{BS}} / R_{m,n,s}^{\text{BS}}$ and the total transmission delay for forwarding to the relay UAVs is determined by the required maximum delay for forwarding; it can be expressed as $T_{\text{total},m,s}^{\text{BS}} = \max\{T_{m,1,s}^{\text{BS}}, T_{m,2,s}^{\text{BS}}, \dots, T_{m,N,s}^{\text{BS}}\}$.

2.1.4. Relay UAV to UAM Aircraft

During period s , the downlink transmission rate for relay UAV n to send data results back to UAM aircraft m is given by the following:

$$R_{m,n,s}^{\text{back}} = B^{\text{back}} \log_2 \left(1 + \frac{h_{m,n,s}^{\text{UAM}} p_{m,n,s}^{\text{back}}}{N_0} \right), \quad (10)$$

where B^{back} represents the bandwidth of the downlink carrier signal transmitted by the relay UAV, and $p_{m,n,s}^{\text{back}}$ denotes the transmission power of the downlink carrier signal emitted by the relay UAV. The transmission delay for relay UAV n to send data results back to UAM aircraft m is given by the following:

$$T_{m,n,s}^{\text{back}} = \frac{I_{m,n,s}^{\text{BS}}}{R_{m,n,s}^{\text{back}}}, \quad (11)$$

and the total transmission delay for sending to the UAM aircraft m is determined by the required maximum delay for sending; it can be expressed as follows:

$$T_{\text{total},m,s}^{\text{back}} = \max\{T_{m,1,s}^{\text{back}}, T_{m,2,s}^{\text{back}}, \dots, T_{m,N,s}^{\text{back}}\}. \quad (12)$$

The energy consumed by relay UAV n when forwarding data results back to UAM aircraft m is given by $E_{m,n,s}^{\text{back}} = p_{m,n,s}^{\text{back}} T_{m,n,s}^{\text{back}}$, and the total energy consumed for sending data results from all relay UAVs to UAM aircraft m is $E_{\text{total},m,s}^{\text{back}} = \sum_{n=1}^N E_{m,n,s}^{\text{back}}$. Therefore, during

period s , the delay and energy consumed in the transmission process for UAM aircraft m is as follows:

$$T_{\text{total},m,s}^{\text{trans}} = T_{\text{total},m,s}^{\text{UAM}} + T_{\text{total},m,s}^{\text{UAV}} + T_{\text{total},m,s}^{\text{BS}} + T_{\text{total},m,s}^{\text{back}} \quad (13)$$

$$E_{\text{total},m,s}^{\text{trans}} = E_{\text{total},m,s}^{\text{UAM}} + E_{\text{total},m,s}^{\text{UAV}} + E_{\text{total},m,s}^{\text{back}}. \quad (14)$$

2.2. Computational Model

In the system model of this paper, we assume that the energy consumption of the ground-edge servers is negligible. The primary reason for this assumption is as follows: With the advancement of 5G technology, edge computing nodes are often tightly integrated into the hardware architecture of base stations. In many cases, edge computing servers are embedded within the base station itself, meaning they do not significantly affect the base station's power load [39]. In contrast, the optimization focus is mainly on devices such as aircraft and drones, which need to operate in the air for extended periods, and their energy consumption has a more significant impact on system performance. Therefore, this paper assumes that the base station's power supply is adequate and the energy consumption of the edge servers is negligible. Moreover, we also assume that the CPU in UAM aircraft can continuously adjust its computing frequency $f_{m,s} \in (f_{\min}, f_{\max})$ in this paper. Based on the offloading ratio, the volume of tasks computed locally by UAM aircraft m is given by the following:

$$I_{m,s}^{\text{local}} = \left(1 - \sum_{n=1}^N O_{m,n,s}\right) I_{m,s}^{\text{UAM}}. \quad (15)$$

The computation delay for tasks computed locally by UAM aircraft m is given by $T_{m,s}^{\text{comp}} = I_{m,s}^{\text{local}} Q \backslash f_{m,s}$, where Q represents the number of CPU cycles required to process one byte of data. The power consumption during task computation is as follows: $p_{m,s}^{\text{comp}} = \kappa (f_{m,s})^3$, where κ represents the CPU's effective capacitance parameter. Therefore, the energy consumption during computation is $E_{m,s}^{\text{comp}} = p_{m,s}^{\text{comp}} T_{m,s}^{\text{comp}}$.

3. The Proposed Resource Allocation Strategies

3.1. Delay Minimization Strategy

Since tasks offloaded for computation to multiple edge servers and tasks computed locally on the UAM aircraft m occur simultaneously, the delay in executing data by UAM aircraft m during the current period, s , is determined by the greater of the total computation delay $T_{m,s}^{\text{comp}}$ and the total transmission delay $T_{m,s}^{\text{trans}}$, which can be represented as follows:

$$T_{m,s}^{\text{max}} = \max\{T_{m,s}^{\text{comp}}, T_{m,s}^{\text{trans}}\}, \quad (16)$$

and during period s , the total delay for data execution can be expressed as $T_{\text{totals}}^{\text{max}} = \max\{T_{1,s}^{\text{max}}, T_{2,s}^{\text{max}}, \dots, T_{M,s}^{\text{max}}\}$. So the total delay consumption of the system is $T = \sum_{s=1}^S T_{\text{totals}}^{\text{max}}$.

The delay optimization strategy aims to optimize the total delay of the system, which involves task offloading ratios $\mathbf{O} = \{O_{1,1,1}, \dots, O_{M,N,S}\}$, transmission powers $\mathbf{p}^{\text{UAM}} = \{p_{1,1,1}^{\text{UAM}}, \dots, p_{M,N,S}^{\text{UAM}}\}$, $\mathbf{p}^{\text{UAV}} = \{p_{1,1,1}^{\text{UAV}}, \dots, p_{M,N,S}^{\text{UAV}}\}$, $\mathbf{p}^{\text{BS}} = \{p_{1,1,1}^{\text{BS}}, \dots, p_{M,N,S}^{\text{BS}}\}$, $\mathbf{p}^{\text{back}} = \{p_{1,1,1}^{\text{back}}, \dots, p_{M,N,S}^{\text{back}}\}$, and computation resource allocation $\mathbf{f} = \{f_{1,1}, \dots, f_{M,S}\}$. It is important to note that while battery capacity is a critical limiting factor for UAVs, the main focus of this study is on optimizing overall system delay and energy consumption. As such, we assume that relay UAVs are tethered UAVs and powered by the ground station, and

therefore, we do not consider energy consumption related to flight. We treat transmission power as the primary explicit constraint. Specifically, the delay optimization problem can be formulated as follows:

$$\begin{aligned}
 & \min_{\mathbf{O}, \mathbf{p}^{\text{UAM}}, \mathbf{p}^{\text{UAV}}, \mathbf{p}^{\text{BS}}, \mathbf{p}^{\text{back}}, \mathbf{f}} T \\
 & \text{s.t.} \quad \text{C1 : } O_{m,n,s} \in [0, 1], \forall m, n, s \\
 & \quad \text{C2 : } p_{\min}^{\text{UAM}} \leq p_{m,n,s}^{\text{UAM}} \leq p_{\max}^{\text{UAM}}, \forall m, n, s \\
 & \quad \text{C3 : } p_{\min}^{\text{UAV}} \leq p_{m,n,s}^{\text{UAV}} \leq p_{\max}^{\text{UAV}}, \forall m, n, s \\
 & \quad \text{C4 : } p_{\min}^{\text{BS}} \leq p_{m,n,s}^{\text{BS}} \leq p_{\max}^{\text{BS}}, \forall m, n, s \\
 & \quad \text{C5 : } p_{\min}^{\text{back}} \leq p_{m,n,s}^{\text{back}} \leq p_{\max}^{\text{back}}, \forall m, n, s \\
 & \quad \text{C6 : } f_{\min} \leq f_{m,s} \leq f_{\max}, \forall m, s \\
 & \quad \text{C7 : } T \leq T_{\max}
 \end{aligned} \tag{17}$$

where C1 denotes the range of values for the offloading ratio; C2 to C5 denote the ranges of values for transmission power; C6 denotes the range of values for the UAM aircraft CPU's computing frequency; C7 indicates that each task must be completed within the maximum delay $T_{\max}$.

3.2. Energy Minimization Strategy

The energy optimization strategy aims to optimize the total energy of the system, which includes both the computation and transmission energy of UAM aircraft and relay UAVs. So the total energy consumption of the system can be represented as follows:

$$E = \sum_{s=1}^S \sum_{m=1}^M \left[E_{m,s}^{\text{comp}} + E_{\text{total},m,s}^{\text{trans}} \right]. \tag{18}$$

Specifically, the energy optimization problem can be formulated as follows:

$$\begin{aligned}
 & \min_{\mathbf{O}, \mathbf{p}^{\text{UAM}}, \mathbf{p}^{\text{UAV}}, \mathbf{p}^{\text{BS}}, \mathbf{p}^{\text{back}}, \mathbf{f}} E \\
 & \text{s.t.} \quad \text{C1 : } O_{m,n,s} \in [0, 1], \forall m, n, s \\
 & \quad \text{C2 : } p_{\min}^{\text{UAM}} \leq p_{m,n,s}^{\text{UAM}} \leq p_{\max}^{\text{UAM}}, \forall m, n, s \\
 & \quad \text{C3 : } p_{\min}^{\text{UAV}} \leq p_{m,n,s}^{\text{UAV}} \leq p_{\max}^{\text{UAV}}, \forall m, n, s \\
 & \quad \text{C4 : } p_{\min}^{\text{BS}} \leq p_{m,n,s}^{\text{BS}} \leq p_{\max}^{\text{BS}}, \forall m, n, s \\
 & \quad \text{C5 : } p_{\min}^{\text{back}} \leq p_{m,n,s}^{\text{back}} \leq p_{\max}^{\text{back}}, \forall m, n, s \\
 & \quad \text{C6 : } f_{\min} \leq f_{m,s} \leq f_{\max}, \forall m, s \\
 & \quad \text{C7 : } T \leq T_{\max}
 \end{aligned} \tag{19}$$

3.3. Joint Delay-Energy Minimization Strategy

Considering that the decisions of task offloading, communication, and computation resource allocation are closely related to the delay and energy consumption during data execution, the overall delay and energy consumption of the UAM aircraft, relay UAVs, and edge servers can be jointly optimized to maximize the overall system efficiency. The optimization problem can be formulated as follows:

$$\begin{aligned}
& \min_{\mathbf{O}, \mathbf{p}^{\text{UAM}}, \mathbf{p}^{\text{UAV}}, \mathbf{p}^{\text{BS}}, \mathbf{p}^{\text{back}}, \mathbf{f}} \alpha C_T T + (1 - \alpha) C_E E \\
& \text{s.t.} \quad \text{C1} : \mathbf{O}_{m,n,s} \in [0, 1], \forall m, n, s \\
& \quad \text{C2} : p_{\min}^{\text{UAM}} \leq p_{m,n,s}^{\text{UAM}} \leq p_{\max}^{\text{UAM}}, \forall m, n, s \\
& \quad \text{C3} : p_{\min}^{\text{UAV}} \leq p_{m,n,s}^{\text{UAV}} \leq p_{\max}^{\text{UAV}}, \forall m, n, s \\
& \quad \text{C4} : p_{\min}^{\text{BS}} \leq p_{m,n,s}^{\text{BS}} \leq p_{\max}^{\text{BS}}, \forall m, n, s \\
& \quad \text{C5} : p_{\min}^{\text{back}} \leq p_{m,n,s}^{\text{back}} \leq p_{\max}^{\text{back}}, \forall m, n, s \\
& \quad \text{C6} : f_{\min} \leq f_{m,s} \leq f_{\max}, \forall m, s \\
& \quad \text{C7} : T \leq T_{\max}
\end{aligned} \tag{20}$$

where α represents the weighting coefficients for delay and energy consumption, and C_T and C_E are constants greater than zero, used to normalize the units of delay and energy consumption.

4. The Proposed UAM Federated Twin Delayed Deep Deterministic Policy Gradient Algorithm

In the three proposed resource allocation strategies, the optimization problems (17), (19), and (20) involve multiple variables, including offloading decisions, communication resource allocation, and computational resource allocation. Due to the complexity and non-convex nature of the optimization problem (*Proof*. See Appendix A), we propose a UF-TD3 algorithm, aimed at optimizing the delay and energy consumption for each aircraft by exchanging model parameters through FL, replacing inter-node data transmission. By adopting a hierarchical task scheduling and resource allocation strategy, we effectively distribute the computational load, avoiding overly complex global optimization calculations. This effectively reduces delay and reduces the energy consumption of UAM aircraft, improving the efficiency of the optimization of the system.

4.1. The Structure of UF-TD3 Algorithm

The UF-TD3 algorithm employs two value networks and one policy network, incorporating noise into actions and reducing the frequency of updates to the policy and target networks. Figure 3 illustrates the specific structure and workflow of the UF-TD3 algorithm.

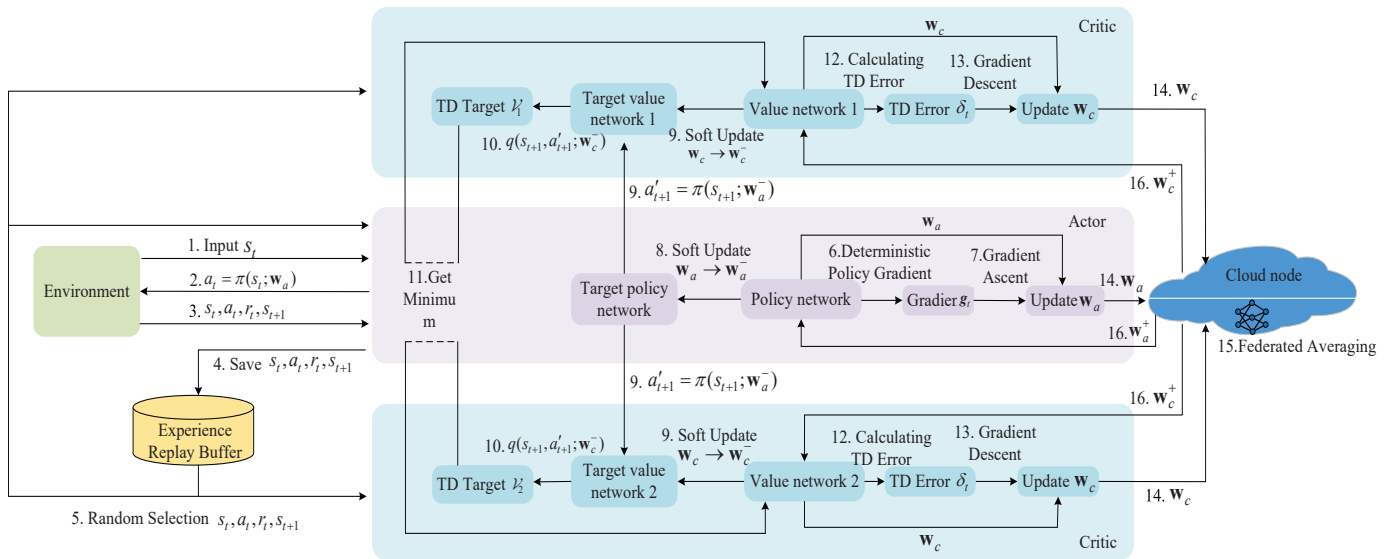


Figure 3. The UF-TD3 algorithm architecture.

- First step: At the initial run of the algorithm, it needs to initialize state s_s , then calculate action a_s with added noise ζ through the function of the policy network. In subsequent iterations, s_s and a_s are input into the environment to obtain r_s and reinitialize state s_{s+1} . Following this, a resource group s_s, a_s, r_s, s_{s+1} is obtained and stored in the experience replay buffer using the experience replay method. When the capacity of the algorithm's experience replay buffer is full, the algorithm begins to extract a resource group from the buffer for training.
- Second step: The target policy network makes a prediction using function $a'_{s+1} = \pi(s_{s+1}; \mathbf{w}_a^-) + \zeta$ to obtain a'_{s+1} , where $\mathbf{w}_a^-$ represents the parameters of the target network of the policy network, and each element of vector ζ is independently drawn from a truncated normal distribution $\mathcal{CN}(0, \sigma^2, -c, c)$.
- Third step: s_{s+1} and a'_{s+1} are input into the two target value networks, respectively, making predictions to obtain $q_{1,s+1}^- = q(s_{s+1}, a'_{s+1}; \mathbf{w}_{c1}^-)$ and $q_{2,s+1}^- = q(s_{s+1}, a'_{s+1}; \mathbf{w}_{c2}^-)$, where $\mathbf{w}_{c1}^-$ and $\mathbf{w}_{c2}^-$ represents the parameters of the target network for value networks 1 and 2, respectively.
- Fourth step: Using the two target value networks $q_{1,s+1}^-$ and $q_{2,s+1}^-$ obtained above, TD targets can be calculated by the following:

$$v_{i,s} \approx r_s + \gamma q_{i,s+1}^-, i \in \{1, 2\}, \quad (21)$$

where γ is the discount factor. Then, taking the lesser of the two as the final TD target, it can be represented as follows:

$$v_s = \min\{v_{1,s}, v_{2,s}\}. \quad (22)$$

- Fifth step: Input s_s and a_s into the two value networks to make predictions, obtaining $q_{1,s} = q(s_s, a_s; \mathbf{w}_{c1})$ and $q_{2,s} = q(s_s, a_s; \mathbf{w}_{c2})$, where $\mathbf{w}_{c1}$ and $\mathbf{w}_{c2}$ represent the parameters of value networks 1 and 2, respectively.
- Sixth step: Calculate the TD errors for value networks 1 and 2, having $\delta_{i,s} = q_{i,s} - v_s, i \in \{1, 2\}$.
- Seventh step: Update value network 1 and value network 2 using the gradient descent method; it can be expressed as follows:

$$\mathbf{w}_{ci} \leftarrow \mathbf{w}_{ci} - \theta \cdot \delta_{i,s} \cdot \frac{\partial q_{i,s}}{\partial \mathbf{w}_{ci}}, i \in \{1, 2\}, \quad (23)$$

where θ is the learning rate of the value network.

- Eighth step: The neural network parameters $\mathbf{w}_{c1,m}$ and $\mathbf{w}_{c2,m}$ from UAM aircraft m can be uploaded to a cloud node server connected to the edge server. The cloud node uses the federated averaging algorithm to process the uploaded local models to obtain a global model; it is given by the following:

$$\mathbf{w}_{ci}^+ = \frac{1}{M} \sum_{m=1}^M \mathbf{w}_{ci,m}, i \in \{1, 2\}, \quad (24)$$

and then the cloud node transmits the global model's network parameters $\mathbf{w}_{c1}^+$ and $\mathbf{w}_{c2}^+$ to the value networks of the UF-TD3 algorithms running on the edge servers, replacing the original network parameters $\mathbf{w}_{c1}^+$ and $\mathbf{w}_{c2}^+$.

- Ninth step: To reduce the frequency of updating the policy network and the target networks, the UF-TD3 algorithm updates the policy network and the three target networks (target value network 1, target value network 2, and target policy network)

every k rounds, while simultaneously federating the parameters of the policy network. First, the deterministic policy gradient is as follows:

$$\mathbf{g}_s = \frac{\partial q[s_s, \pi(s_s; \mathbf{w}_a); \mathbf{w}_{c1}]}{\partial \mathbf{w}_a} = \frac{\partial a_s}{\partial \mathbf{w}_a} \cdot \frac{\partial q(s_s, a_s; \mathbf{w}_{c1})}{\partial a_s}. \quad (25)$$

Second, update the policy network parameters $\mathbf{w}_a$ using gradient ascent; it can be denoted as $\mathbf{w}_a \leftarrow \mathbf{w}_a + \beta \cdot \mathbf{g}_s$, where β is the learning rate of the policy network. Similar to the federated averaging value network, the global model parameters of the policy network after processing through the federated averaging algorithm are as follows:

$$\mathbf{w}_a^+ = \frac{1}{M} \sum_{m=1}^M \mathbf{w}_{a,m}. \quad (26)$$

The cloud node then transmits the global model's network parameters $\mathbf{w}_a^+$ and $\mathbf{w}_c^+$ to the policy networks of the UF-TD3 algorithms running on the edge servers, replacing the original network parameters $\mathbf{w}_a$ and $\mathbf{w}_c$. Finally, the algorithm updates the parameters of the policy network's target network by calculating a weighted average, given by the following:

$$\mathbf{w}_a^- \leftarrow \varsigma \cdot \mathbf{w}_a + (1 - \varsigma) \cdot \mathbf{w}_a^-, \quad (27)$$

where $\varsigma, \varsigma \in (0, 1)$ is the hyperparameter. The algorithm updates the parameters of the target networks for value networks 1 and 2 by calculating a weighted average; it can be represented as follows:

$$\mathbf{w}_{ci}^- \leftarrow \varsigma \cdot \mathbf{w}_{ci} + (1 - \varsigma) \cdot \mathbf{w}_{ci}^-, i \in \{1, 2\}. \quad (28)$$

- Tenth step: At the end of the current iteration, the current state s_{s+1} is assigned to s_s , completing the algorithm's cycle loop.

We summarize the UF-TD3 algorithm procedure in Algorithm 1.

4.2. Status, Action, and Reward

Combined with the UF-TD3 algorithm, we set the state space, the action space, and the reward values for the two baseline strategies and the joint delay-energy minimization strategy.

- The state space: For the three mentioned strategies, the state space is the same. It consists of M UAM aircraft within the target UAM system and N edge servers from neighboring UAM systems. During period s , the system state of UAM aircraft m can be defined as follows:

$$s_{m,s} = (u_{m,s}^{\text{UAM}}, I_{m,s}^{\text{UAM}}, \vartheta_{m,s}). \quad (29)$$

- The action space: The action space is identical across the three strategies mentioned above. During the current period, s , the actions of UAM aircraft m can be represented as follows:

$$a_{m,s} = (\mathbf{O}_{m,s}, \mathbf{p}_{m,s}^{\text{UAM}}, \mathbf{p}_{m,s}^{\text{UAV}}, \mathbf{p}_{m,s}^{\text{BS}}, \mathbf{p}_{m,s}^{\text{back}}, f_{m,s}), \quad (30)$$

where $\mathbf{O}_{m,s} = \{O_{m,1,s}, \dots, O_{m,N,s}\}$ denotes the collection of offloading ratios from the UAM aircraft m to various edge servers, $\mathbf{p}_{m,s}^{\text{UAM}} = \{p_{m,1,s}^{\text{UAM}}, \dots, p_{m,N,s}^{\text{UAM}}\}$ denotes the collection of transmission powers used by UAM aircraft m when offloading to

different relay UAVs, $\mathbf{p}_{m,s}^{\text{UAV}} = \{p_{m,1,s}^{\text{UAV}}, \dots, p_{m,N,s}^{\text{UAV}}\}$ denotes the collection of transmission powers used by relay UAVs to offload tasks to different edge server wireless BSs, $\mathbf{p}_{m,s}^{\text{BS}} = \{p_{m,1,s}^{\text{BS}}, \dots, p_{m,N,s}^{\text{BS}}\}$ denotes the collection of transmission powers used by different edge server wireless BSs to send task results back to relay UAVs, and $\mathbf{p}_{m,s}^{\text{back}} = \{p_{m,1,s}^{\text{back}}, \dots, p_{m,N,s}^{\text{back}}\}$ denotes the collection of transmission powers used by different relay UAVs to send task results back to UAM aircraft m .

Algorithm 1 The UF-TD3 Algorithm

```

1: Initialize state  $s_s$ .
2: while the maximum number of iterations has not been reached do
3:   The state  $s_s$  is input into the policy network, which then generates the action  $a_s$ .
4:   The action  $a_s$  and state  $s_s$  are input into the environment, which then returns the
   reward  $r_s$ .
5:   Initialize state  $s_{s+1}$ .
6:   A resource group  $s_s, a_s, r_s, s_{s+1}$  is obtained and stored in the experience replay buffer.
7:   if the capacity of the buffer is full then
8:     Extract a resource group  $s_s, a_s, r_s, s_{s+1}$  from the buffer.
9:     The state  $s_{s+1}$  is input into the target policy network, and the next action  $a'_{s+1}$  is
     computed.
10:    The state  $s_{s+1}$  and action  $a'_{s+1}$  are input into the two target value networks,
    yielding  $q_{1,s+1}^-$  and  $q_{2,s+1}^-$ .
11:    The TD error  $\delta_{1,t}$  and  $\delta_{2,t}$  for the two target value networks is calculated.
12:    Update the parameters of value networks  $\mathbf{w}_{c1}$  and  $\mathbf{w}_{c2}$  by using gradient descent.
13:    Upload the value network parameters  $\mathbf{w}_{c1,m}$  and  $\mathbf{w}_{c2,m}$  to the cloud node.
14:    At the cloud node, federated averaging is applied to compute the values  $\mathbf{w}_{c1}^+$ 
    and  $\mathbf{w}_{c2}^+$ .
15:    The values of  $\mathbf{w}_{c1}^+$  and  $\mathbf{w}_{c2}^+$  are transmitted to the edge servers, replacing the
    previous network parameters  $\mathbf{w}_{c1}$  and  $\mathbf{w}_{c2}$ .
16:    if the iteration count is an integer multiple of  $k$  then
17:      Using the deterministic policy gradient, we calculate the gradient  $\mathbf{g}_s$  of  $q$ 
      with respect to  $\mathbf{w}_a$ .
18:      The parameter  $\mathbf{w}_a$  is updated by applying the gradient ascent method.
19:      Upload the policy network parameter  $\mathbf{w}_{a,m}$  to the cloud node.
20:      At the cloud node, federated averaging is applied to compute the values  $\mathbf{w}_a^+$ .
21:      The values of  $\mathbf{w}_a^+$  is transmitted to the edge servers, replacing the previous
      network parameters  $\mathbf{w}_a$ .
22:      Update the target network's parameters  $\mathbf{w}_a^-, \mathbf{w}_{c1}^-$  and  $\mathbf{w}_{c2}^-$ .
23:    end if
24:  end if
25:  The state  $s_{s+1}$  is assigned to  $s_s$ .
26: end while
  
```

- The reward: In the context of state $s_{m,s}$, selecting action $a_{m,s}$ results in an immediate reward $r_{m,s}$. Throughout the algorithm's training process, the value of reward $r_{m,s}$ should consistently increase until it reaches maximization. For the delay minimization strategy, the optimization problem (17) involves minimizing the overall system delay. Therefore, the reward value $r_{m,s}^{\text{delay}}$ should be negatively correlated with time consumption. During the current period s , the delay reward for UAM aircraft m can be expressed as follows:

$$r_{m,s}^{\text{delay}} = r_{m,s}^{(s_{m,s}, a_{m,s})} = -C \cdot T_{\text{totals}}^{\text{max}} \quad (31)$$

where the reward parameter C is a constant greater than zero, intended to amplify the impact of the reward, thereby more distinctly highlighting the quality of the current action.

Similar to (31), for the energy minimization strategy, the optimization objective (19) involves minimizing the overall system energy consumption. Therefore, the reward value $r_{m,s}^{energy}$ should be negatively correlated with the energy consumption; it can be represented as follows:

$$r_{m,s}^{energy} = r_{m,s}^{(s_{m,s}, a_{m,s})} = -C \cdot (E_{m,s}^{comp} + E_{totalm,s}^{trans}). \quad (32)$$

For the joint delay-energy minimization strategy, the optimization objective (20) involves minimizing the overall system cost. In addition to being negatively correlated with the consumption of delay and energy, the joint reward value also needs to consider the weighting ratio α of delay and energy consumption, given by the following:

$$\begin{aligned} r_{m,s}^{joint} &= r_{m,s}^{(s_{m,s}, a_{m,s})} \\ &= -C \cdot [\alpha T_{totals}^{max} + (1 - \alpha)(E_{m,s}^{comp} + E_{totalm,s}^{trans})]. \end{aligned} \quad (33)$$

5. Numerical Results and Discussion

5.1. Simulation Parameter Setting

We assume that four UAM aircraft are evenly distributed within a $1000 \times 1000 \text{ m}^2$ square target UAM system. The edge servers of four neighboring UAM systems are located at the four corners, and the relay UAVs are positioned at the midpoint of the line connecting the edge server locations to the center of the area, at a flying altitude of 100 m. We set the overall maximum transmission and computation delay to 100ms, which aligns with the typical delay standard in current aircraft edge computing networks. This delay threshold is designed to ensure timely task completion, thereby maintaining the system's reliability and responsiveness. We selected this value based on practical considerations from real-world application scenarios, aiming to optimize task scheduling and data processing efficiency. Table 2 shows the simulation and network parameters.

Table 2. Simulation and network parameters.

Parameter	Value
The bandwidth $B^{\text{UAM}}, B^{\text{UAV}}, B^{\text{BS}}, B^{\text{back}}$	10 MHz
The channel gain β_0	−50 dB
The noise power N_0	−100 dBm
The maximum power $p_{\max}^{\text{UAM}}, p_{\max}^{\text{UAV}}, p_{\max}^{\text{BS}}, p_{\max}^{\text{back}}$	10 w
The minimum power $p_{\min}^{\text{UAM}}, p_{\min}^{\text{UAV}}, p_{\min}^{\text{BS}}, p_{\min}^{\text{back}}$	1 w
The maximum delay $T_{\max}$	100 ms
The task volume I	2 Mbits
The clock frequency of the CPU f	0.1–0.2 GHz
The CPU computation speed Q	1000 cycles/bit
The effective capacitance parameter κ	10^{-27}
The discount factor γ	0.001
Target network hyperparameter ς	0.01
Truncated normal distribution σ, c	0.1, 1
Reward parameter C	100

5.2. Analysis of Simulation Results

We set two important hyperparameters in the algorithm for deep neural networks: the learning rate β for the policy network and the learning rate θ for the two value networks. The learning rate is a scalar that determines the magnitude of each gradient update,

influencing the convergence speed and stability of the model. Table 3 displays five learning rate allocation schemes set in the simulation.

Table 3. Learning rate allocation schemes.

Scheme	1	2	3	4	5
β	0.00005	0.0001	0.0005	0.001	0.005
θ	0.00006	0.0002	0.0006	0.002	0.006

Figure 4 illustrates the convergence of reward values for the UF-TD3 algorithm under five different learning rate schemes for delay minimization, energy consumption minimization, and joint delay-energy minimization strategies after 10,000 iterations. When using scheme 1, the reward value curves for all three strategies converge at around 5000 iterations, exhibiting the slowest convergence but achieving the highest reward values. This is due to the overly small learning rate setting, which requires many learning iterations to reach optimal convergence. When using scheme 2, the reward value curves for all three strategies converge around 3000 iterations, which is 40% faster than scheme 1. However, because the complexity of the optimization objectives varies between the three strategies, the curves in Figure 4a,b converge faster than in Figure 4c under this learning rate. When using scheme 3, all three curves converge at around 2000 iterations, displaying both a fast convergence speed and high reward values. When using scheme 5, the reward value curves for all three strategies converge after only about 200 iterations, but the curves flatten in the middle and later stages, failing to achieve optimal convergence. This is because the excessively high learning rate causes the deep neural network's update steps to be too large, leading to local convergence. When using scheme 4, the learning rate is slightly lower than that of scheme 5. For the curves in Figure 4a,c, this learning rate is at the critical point of local convergence, while the convergence in Figure 4b is worse than in scheme 3. Therefore, considering the convergence of the UF-TD3 algorithm under different optimization strategies and learning rates, the learning rate setting of scheme 3 ($\beta = 0.0005, \theta = 0.0006$) is consistently used in subsequent simulations to achieve relatively stable algorithm convergence.

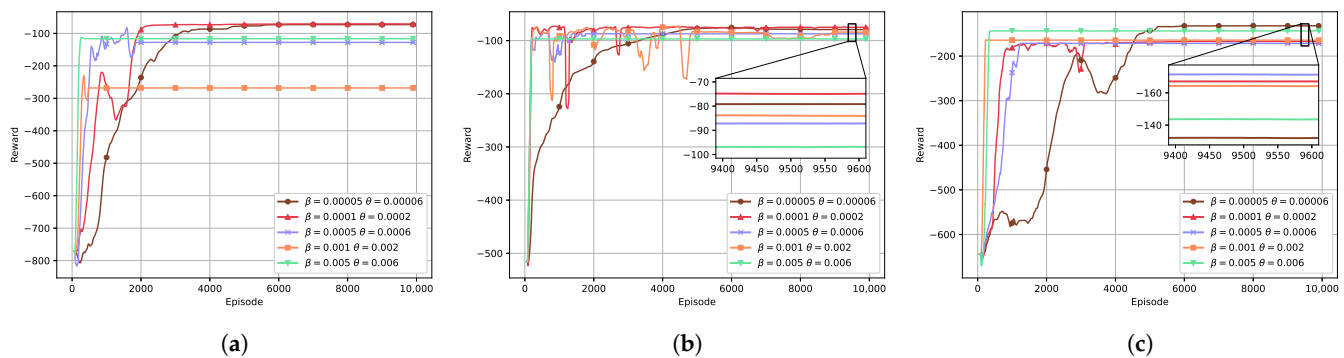


Figure 4. The convergence of reward values for the three strategies under different learning rates. (a) Delay minimization strategy. (b) Energy minimization strategy. (c) Joint delay-energy minimization strategy.

To validate the performance of the proposed UF-TD3 algorithm, simulations were conducted to compare the reward value convergence of five algorithms.

- The proposed UF-TD3 algorithm: In this comparison, the UF-TD3 algorithm is used to jointly optimize the system's delay and energy consumption.
- TD3-DE algorithm: In this comparison, the TD3 algorithm is used to jointly optimize the system's delay and energy consumption.

- DDPG-DE algorithm: In this comparison, the DDPG algorithm is used to jointly optimize the system's delay and energy consumption.
- DQN-DE algorithm: In this comparison, the DQN algorithm is used to jointly optimize the system's delay and energy consumption.
- The IF-DDPG-DE algorithm from our previous research (see Ref. [40]): In this comparison, the IF-DDPG algorithm is used to jointly optimize the system's delay and energy consumption.

As shown in Figure 5a, the UF-TD3 algorithm exhibits a significantly higher convergence speed and average reward value compared to the TD3-DE algorithm without FL. This improvement is attributed to the UF-TD3 algorithm's use of FL, which enables multiple TD3 algorithms to learn from each other and share their deep learning network parameters, resulting in enhanced convergence performance and learning efficiency. In our previous work, we proposed the IF-DDPG-DE algorithm, which also leverages the FL method. The IF-DDPG-DE algorithm involves multiple DDPG algorithms, each corresponding to a UAM aircraft. The IF-DDPG-DE algorithm not only improves the accuracy of each individual DDPG algorithm but also enhances the overall accuracy when combining multiple DDPG algorithms. This is achieved by exchanging model parameters between the DDPG algorithms, enabling the simultaneous optimization of both the delay and energy consumption of each UAM aircraft and the entire system. In contrast, the UF-TD3 algorithm integrates multiple TD3 networks, uses a delayed update strategy, and adjusts the target network update frequency to reduce interference between the policy network and the value function, thereby improving training stability. Additionally, by introducing noise during Q-value calculation, UF-TD3 mitigates overestimation, leading to enhanced accuracy. This enhancement results in a 20.94% improvement in reward values compared to the IF-DDPG-DE algorithm. The DQN-DE algorithm is designed to handle discrete actions, and in the continuous action space, the precision and accuracy of action selection are insufficient, resulting in suboptimal convergence performance. Figure 5b,c illustrate the energy and delay consumption optimization performance of UF-TD3 after 10,000 iterations in comparison to four other algorithms. The UF-TD3 algorithm begins to converge at approximately 1500 iterations, while the other algorithms require over 2000 iterations to converge. In addition to its faster convergence rate, the UF-TD3 algorithm demonstrates efficient energy and delay optimization performance.

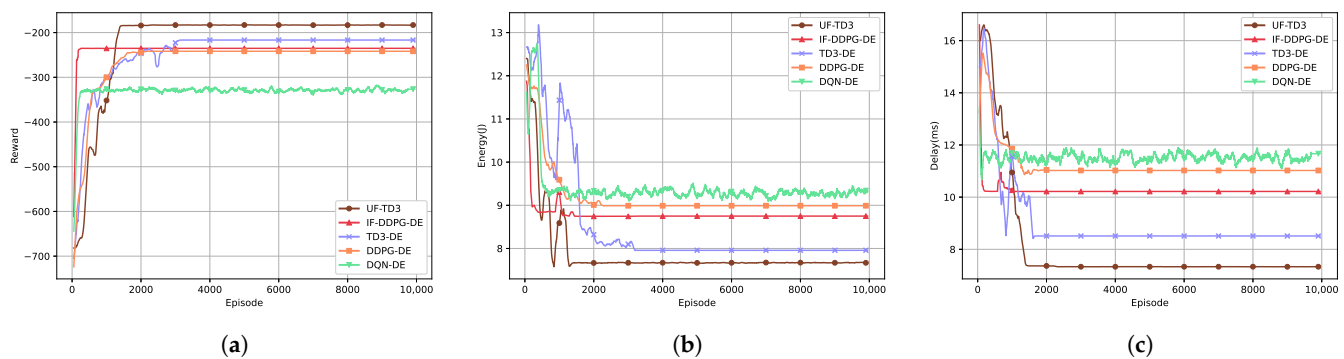


Figure 5. Comparison of the optimization performance for reward, energy, and delay consumption across five algorithms under a joint delay-energy minimization strategy. (a) The convergence of reward values. (b) The energy optimization performance. (c) The delay optimization performance.

To facilitate a clear comparison of the performances of the five algorithms, Table 4 summarizes key metrics, including convergence speed, delay reduction, and energy savings. As shown in the Table 4, energy consumption is reduced by 3.63% compared to the TD3-DE algorithm. It is also reduced by 12.24% compared to the IF-DDPG-DE algorithm, by 14.51%

compared to the DDPG-DE algorithm without federated learning, and by 17.97% compared to the DQN-DE algorithm. The delay is reduced by 13.93% compared to the TD3-DE algorithm. It is also reduced by 28.22% compared to the IF-DDPG-DE algorithm, by 33.24% compared to the DDPG-DE algorithm without federated learning, and by 37.12% compared to the DQN-DE algorithm.

Table 4. The key metrics of the performances of the five algorithms.

Algorithm	Iterations	Delay (ms)	Energy (J)
UF-TD3	1500	7.34	7.67
IF-DDPG-DE	250	10.20	8.76
TD3-DE	3200	8.5	7.96
DDPG-DE	2000	11.03	8.98
DQN-DE	400	11.68	9.33

Next, we compare the impacts of the UF-TD3 algorithm and other baseline algorithms on system delay and energy consumption under different numbers of UAM aircraft. As shown in Figures 6 and 7, with an increase in the number of UAM aircraft, the total system delay and energy consumption also increase. However, since the information transmission between different UAM aircraft adopts parallel transmission, the change in delay in Figure 6 is relatively smooth. In contrast, system energy consumption is the sum of the energy consumption of different transmission links and is directly related to the number of UAM aircraft, so the energy consumption trend in Figure 7 changes more rapidly. The proposed UF-TD3 algorithm involves multiple TD3 algorithms, each corresponding to a UAM aircraft. It not only improves the accuracy of each individual TD3 algorithm but also enhances the overall accuracy when combining multiple TD3 algorithms. This is achieved by exchanging model parameters between the TD3 algorithms, which enables the simultaneous optimization of both the delay and energy consumption for each UAM aircraft as well as the entire system. Therefore, as network complexity increases, the distributed computation approach helps to minimize the overhead caused by the growing network load.

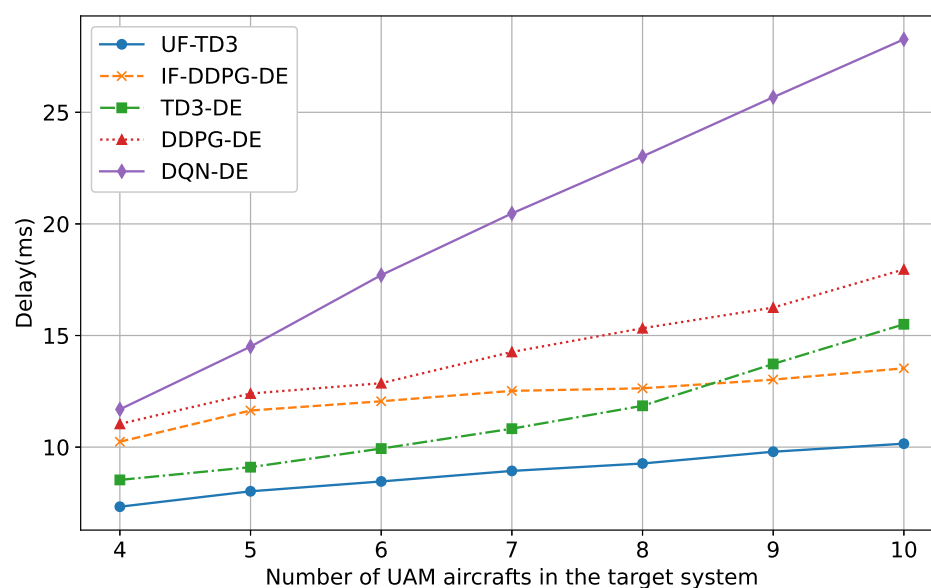


Figure 6. The effect of the number of UAM aircraft on delay performance.

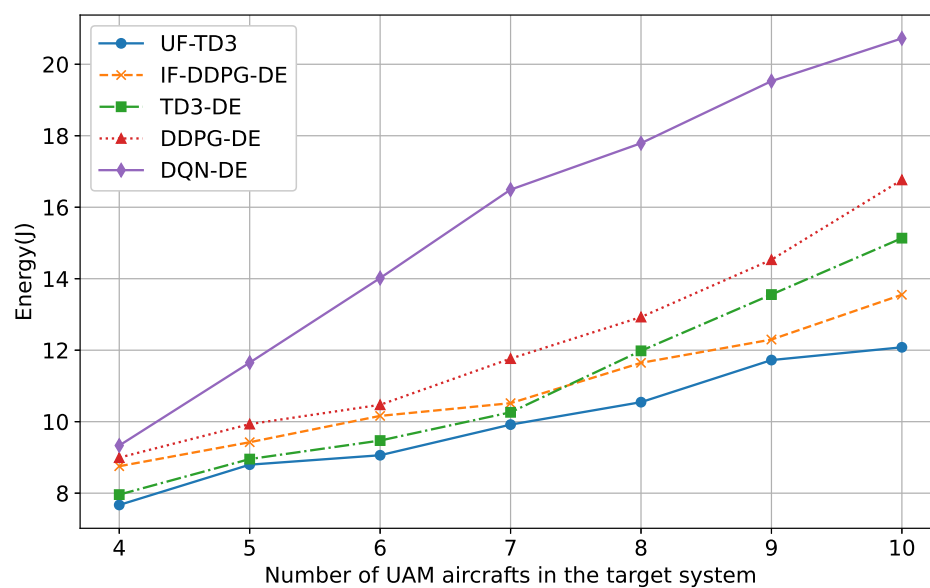


Figure 7. The effect of the number of UAM aircraft on energy performance.

Overall, the growth rate of the UF-TD3 algorithm's curve is smaller and more gradual compared to the baseline algorithms, resulting in relatively lower delay and energy consumption and being less affected by the number of terminal devices. In the mid-to-late stages of the curve, the delay and energy consumption curves of the UF-TD3 and IF-DDPG-DE algorithms grow more slowly compared to TD3-DE and DDPG-DE, due to the use of federated learning, which allows each aircraft's corresponding TD3-DE and DDPG-DE algorithms to learn from each other and share their deep learning network parameters, leading to improved learning outcomes.

Notably, as the number of UAM aircraft increases, the optimization performance of the IF-DDPG-DE algorithm surpasses that of the TD3-DE algorithm. For the DQN-DE algorithm, the problems caused by its insufficient precision and accuracy in action selection in the continuous action space become more severe as the number of users increases, leading to a much faster growth rate in delay and energy consumption compared to other algorithms.

As shown in Figure 8, the system performance optimization results for three strategies are presented: the delay minimization strategy, the energy consumption minimization strategy, and the joint delay-energy minimization strategy. The delay minimization strategy focuses solely on reducing system delay, without considering energy consumption, leading to improved responsiveness but potentially higher energy use. In contrast, the energy consumption minimization strategy prioritizes minimizing energy consumption while disregarding delay, which can result in longer delays but better energy efficiency. The joint delay-energy minimization strategy optimizes both delay and energy consumption simultaneously, offering a balanced solution that minimizes both factors. The results show that compared to the delay minimization strategy, the joint delay-energy minimization strategy reduces energy consumption by 63.86%, while still achieving significant improvements in delay. Similarly, compared to the energy consumption minimization strategy, the joint strategy reduces delay by 73.96%. While the two baseline strategies are effective in optimizing delay or energy consumption individually, the UAM scenario requires considering both factors together. The joint delay-energy minimization strategy provides a more holistic optimization, making it more suitable for practical applications where both delay and energy efficiency are critical.

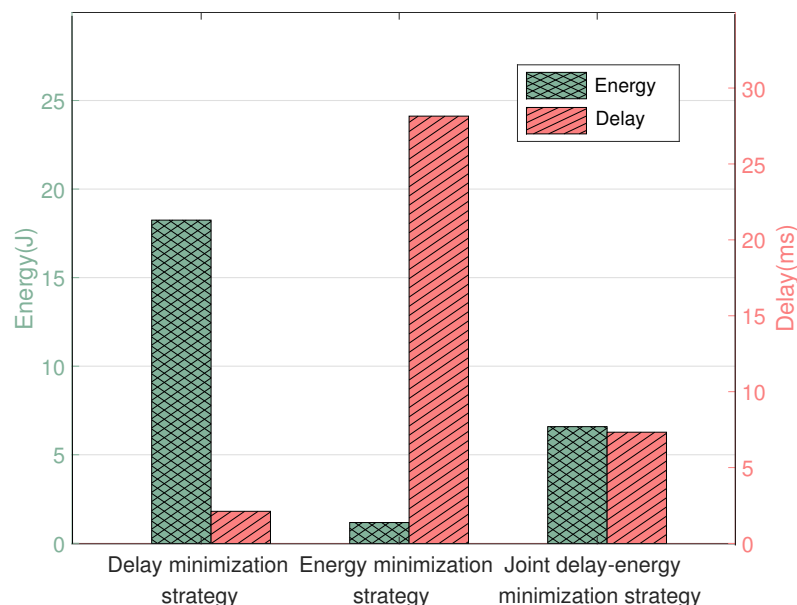


Figure 8. The results of system performance optimization under three different strategies.

Figure 9 demonstrates the impacts of different weight coefficients α on system performance optimization. The weighted coefficients are fixed values set at the beginning of the algorithm and are not adjusted during the optimization process. When $\alpha = 0$, optimization problem (20) transforms into an energy consumption optimization problem. Under these conditions, the system achieves low energy consumption without considering delay effects, aligning with the goal of an energy minimization strategy (19). This scenario is suitable for transmitting information insensitive to delays, such as the aircraft's own status information. As α increases, system energy consumption gradually increases while delay decreases. This indicates that the algorithm optimizes system delay and energy consumption based on the assigned weight α . When $\alpha = 1$, optimization problem (20) transforms into a delay optimization problem. In this scenario, the system achieves low delay, even at the cost of higher energy consumption, aligning with the goal of a delay minimization strategy (17). This is suitable for transmitting delay-sensitive information, such as surrounding environmental data. In the UAM context, the weight coefficient α can be adjusted according to different operational requirements to balance delay and energy consumption.

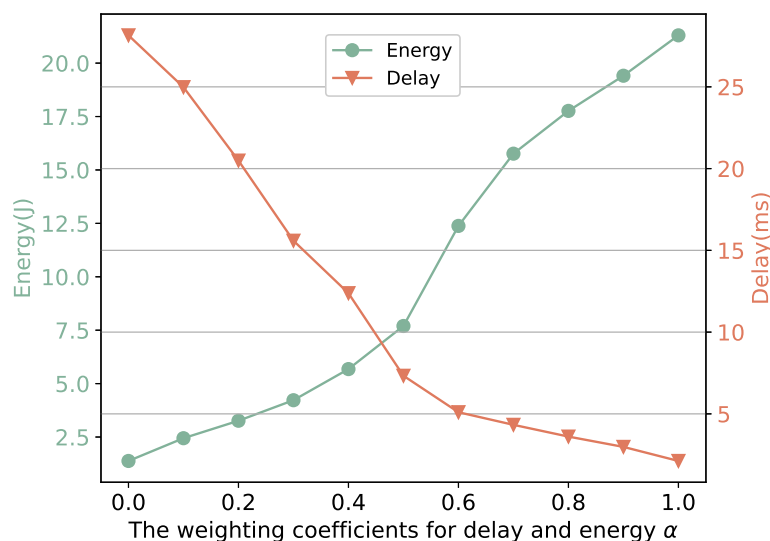


Figure 9. The impact of weighting coefficients for delay and energy on the system performance optimization of the UF-TD3 algorithm.

Figures 10 and 11 illustrate the relationship between the relay UAV deployment position and system delay and energy consumption. When deployed along the diagonal of the area (coordinates (x, x) , where $0 < x < 1000$), the information transmission distance via the relay UAV remains constant. However, because the energy consumed by the base station to transmit processed information to the relay UAV is ignored in this study, the closer the deployment position is to (1000, 1000), the higher the energy consumption. Due to this increased energy consumption, the joint optimization strategy for delay and energy consumption results in higher delay to balance the trade-off, even though the transmission distance remains the same. Similarly, when the relay UAV deployment position shifts to either side (notably, deploying relay UAVs on either side of the diagonal exhibits symmetry), it follows the same pattern. That is, for the same transmission distance, the closer the deployment position is to (1000, 1000), the greater the delay and energy consumption.

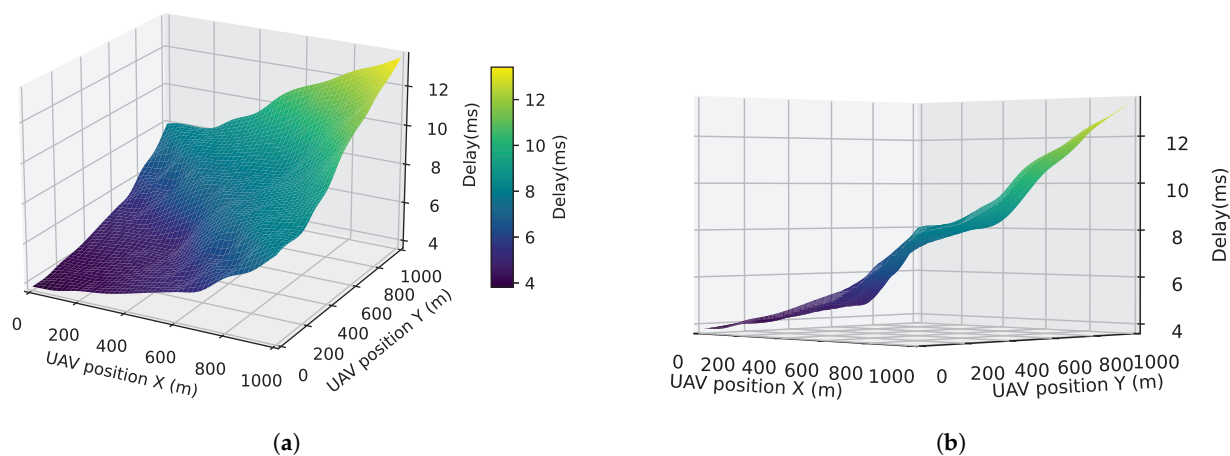


Figure 10. The impacts of different relay UAV positions on system delay. (a) The main view. (b) The side view.

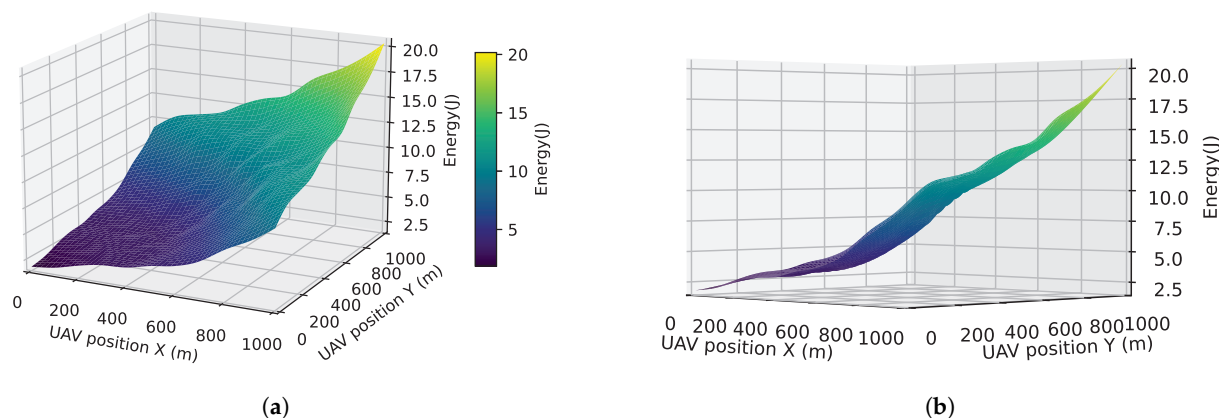


Figure 11. The impacts of different relay UAV positions on system energy. (a) The main view. (b) The side view.

6. Conclusions

This paper focuses on the UAM in an emergency scenario, proposing three resource allocation strategies and introducing the UF-TD3 algorithm to jointly optimize data offloading rate, subcarrier allocation, transmission power, and computation frequency. We validate the optimization performance of the proposed algorithm under different resource allocation strategies. Simulation results show that the UF-TD3 algorithm outperforms other baseline algorithms in terms of both delay and energy consumption optimization. Finally, we discuss the impacts of varying weight coefficients, different numbers of UAM aircraft, and the

deployment positions of relay UAVs on system delay and energy consumption. However, the current study is based on an idealized communication environment, and future research could consider factors such as interference and environmental conditions in the real world to more comprehensively evaluate the system's performance and optimization strategies.

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

Funding: This work is partially supported by the Beijing Natural Science Foundation Haidian Original Innovation Joint Fund (no. L212026); R&D Program of Beijing Municipal Education Commission (KM202211232011); The open research subject of state key laboratory of intelligent game (No. ZBKF-24-12); The Foundation of Key Laboratory of Education Informatization for Nationalities (Yunnan Normal University), the Ministry of Education (No. EIN2024C006).

Data Availability Statement: The data presented in this study are available upon request from the corresponding author due to the privacy of the data.

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

Abbreviations

The following abbreviations are used in this manuscript:

UAM	urban air mobility
UAV	unmanned aerial vehicle
MEC	mobile edge computing
BS	base station
FL	federated learning
TD3	twin delayed deep deterministic policy gradient
UF-TD3	UAM federated twin delayed deep deterministic policy gradient algorithm
DDPG	deep deterministic policy gradient
DQN	deep Q-network

Appendix A

Calculating the Hessian matrix of the objective function is a common method to verify its convexity or concavity. Since T contains the max function, we simplify the problem as follows:

$$T \approx T_{\text{total},s}^{\text{trans}} = T_{\text{total},s}^{\text{UAM}} + T_{\text{total},s}^{\text{UAV}} + T_{\text{total},s}^{\text{BS}} + T_{\text{total},s}^{\text{back}}$$

$$= \frac{O_{m,n,s} I_{m,s}^{\text{UAM}}}{R_{m,n,s}^{\text{UAM}}} + \frac{O_{m,n,s} I_{m,s}^{\text{UAM}}}{R_{m,n,s}^{\text{UAV}}} + \frac{\vartheta_{m,s} O_{m,n,s} I_{m,s}^{\text{UAM}}}{R_{m,n,s}^{\text{BS}}} + \frac{\vartheta_{m,s} O_{m,n,s} I_{m,s}^{\text{UAM}}}{R_{m,n,s}^{\text{back}}}. \quad (\text{A1})$$

Clearly, the four terms in (A1) share a similar form, and we analyze each one individually. First, we calculate the Hessian matrix of the first term.

The first-order partial derivative is expressed as follows:

$$\frac{\partial g}{\partial O_{m,n,s}} = \frac{I_{m,s}^{\text{UAM}}}{B^{\text{UAM}} \log_2 \left(1 + \frac{\beta_0 p_{m,n,s}^{\text{UAM}}}{N_0 \|u_{n,s}^{\text{UAV}} - u_{m,s}^{\text{UAM}}\|^2} \right)}, \quad (\text{A2})$$

$$\frac{\partial g}{\partial p_{m,n,s}^{\text{UAM}}} = \frac{-O_{m,n,s} I_{m,s}^{\text{UAM}} \frac{B^{\text{UAM}} \beta_0 \ln 2}{(N_0 \|u_{n,s}^{\text{UAV}} - u_{m,s}^{\text{UAM}}\|^2 + \beta_0 p_{m,n,s}^{\text{UAM}})}}{\left[B^{\text{UAM}} \log_2 \left(1 + \frac{\beta_0 p_{m,n,s}^{\text{UAM}}}{N_0 \|u_{n,s}^{\text{UAV}} - u_{m,s}^{\text{UAM}}\|^2} \right) \right]^2}. \quad (\text{A3})$$

The second-order partial derivative is expressed as $\frac{\partial^2 g}{\partial O_{m,n,s}^2} = 0$, and we have the following:

$$\frac{\partial^2 g}{\partial p_{m,n,s}^{\text{UAM},2}} = \frac{\ln 2 O_{m,n,s} I_{m,s}^{\text{UAM}} \beta_0^2 B^{\text{UAM}} \log_2 \left(1 + \frac{\beta_0 p_{m,n,s}^{\text{UAM}}}{N_0 \|u_{n,s}^{\text{UAV}} - u_{m,s}^{\text{UAM}}\|^2} \right) - 2\beta_0 \ln 2 (N_0 \|u_{n,s}^{\text{UAV}} - u_{m,s}^{\text{UAM}}\|^2 + \beta_0 p_{m,n,s}^{\text{UAM}})}{(N_0 \|u_{n,s}^{\text{UAV}} - u_{m,s}^{\text{UAM}}\|^2 + \beta_0 p_{m,n,s}^{\text{UAM}})^2 B^{\text{UAM},2} \log_2 \left(1 + \frac{\beta_0 p_{m,n,s}^{\text{UAM}}}{N_0 \|u_{n,s}^{\text{UAV}} - u_{m,s}^{\text{UAM}}\|^2} \right)^3}. \quad (\text{A4})$$

The mixed partial derivative is calculated as follows:

$$\frac{\partial^2 g}{\partial O_{m,n,s} \partial p_{m,n,s}} = \frac{-I_{m,s}^{\text{UAM}} \frac{B^{\text{UAM}} \beta_0 \ln 2}{(N_0 \|u_{n,s}^{\text{UAV}} - u_{m,s}^{\text{UAM}}\|^2 + \beta_0 p_{m,n,s}^{\text{UAM}})}}{\left[B^{\text{UAM}} \log_2 \left(1 + \frac{\beta_0 p_{m,n,s}^{\text{UAM}}}{N_0 \|u_{n,s}^{\text{UAV}} - u_{m,s}^{\text{UAM}}\|^2} \right) \right]^2}. \quad (\text{A5})$$

Using the second-order partial derivative and the mixed partial derivative, we construct the Hessian matrix as follows:

$$H = \begin{bmatrix} 0 & \frac{-I_{m,s}^{\text{UAM}} \frac{B^{\text{UAM}} \beta_0 \ln 2}{(N_0 \|u_{n,s}^{\text{UAV}} - u_{m,s}^{\text{UAM}}\|^2 + \beta_0 p_{m,n,s}^{\text{UAM}})}}{\left[B^{\text{UAM}} \log_2 \left(1 + \frac{\beta_0 p_{m,n,s}^{\text{UAM}}}{N_0 \|u_{n,s}^{\text{UAV}} - u_{m,s}^{\text{UAM}}\|^2} \right) \right]^2} \\ \frac{-I_{m,s}^{\text{UAM}} \frac{B^{\text{UAM}} \beta_0 \ln 2}{(N_0 \|u_{n,s}^{\text{UAV}} - u_{m,s}^{\text{UAM}}\|^2 + \beta_0 p_{m,n,s}^{\text{UAM}})}}{\left[B^{\text{UAM}} \log_2 \left(1 + \frac{\beta_0 p_{m,n,s}^{\text{UAM}}}{N_0 \|u_{n,s}^{\text{UAV}} - u_{m,s}^{\text{UAM}}\|^2} \right) \right]^2} & \frac{\ln 2 O_{m,n,s} I_{m,s}^{\text{UAM}} \beta_0^2 B^{\text{UAM}} \log_2 \left(1 + \frac{\beta_0 p_{m,n,s}^{\text{UAM}}}{N_0 \|u_{n,s}^{\text{UAV}} - u_{m,s}^{\text{UAM}}\|^2} \right) - 2\beta_0 \ln 2 (N_0 \|u_{n,s}^{\text{UAV}} - u_{m,s}^{\text{UAM}}\|^2 + \beta_0 p_{m,n,s}^{\text{UAM}})}{(N_0 \|u_{n,s}^{\text{UAV}} - u_{m,s}^{\text{UAM}}\|^2 + \beta_0 p_{m,n,s}^{\text{UAM}})^2 B^{\text{UAM},2} \log_2 \left(1 + \frac{\beta_0 p_{m,n,s}^{\text{UAM}}}{N_0 \|u_{n,s}^{\text{UAV}} - u_{m,s}^{\text{UAM}}\|^2} \right)^3} \end{bmatrix} \quad (\text{A6})$$

We set the constant parameters to the values in Table 2. Therefore, the Hessian matrix can be computed as follows:

$$H = \begin{bmatrix} 0 & \frac{-\frac{2 \ln 2}{5 + p_{m,n,s}^{\text{UAM}}}}{10 \log_2 \left(1 + \frac{p_{m,n,s}^{\text{UAM}}}{5} \right)^2} \\ \frac{-\frac{2 \ln 2}{5 + p_{m,n,s}^{\text{UAM}}}}{10 \log_2 \left(1 + \frac{p_{m,n,s}^{\text{UAM}}}{5} \right)^2} & \frac{2 \times 10^{-1} \log_2 \left(1 + \frac{p_{m,n,s}^{\text{UAM}}}{5} \right) O_{m,n,s} \ln 2 - 2(5 \times 10^{-14} + 10^{-14} p_{m,n,s}^{\text{UAM}}) \ln 2}{(5 + p_{m,n,s}^{\text{UAM}})^2 \log_2 \left(1 + \frac{p_{m,n,s}^{\text{UAM}}}{5} \right)^3} \end{bmatrix}. \quad (\text{A7})$$

At this point, for the variables $O_{m,n,s}$ and $p_{m,n,s}$, we assign values within their respective domains, setting $O_{m,n,s} = 0.3$ and $p_{m,n,s} = 1$. The eigenvalues and eigenvectors of the Hessian matrix can then be calculated as follows:

$$\lambda_1 = -0.3257, \lambda_2 = 0.3424. \quad (\text{A8})$$

Since the eigenvalues of the Hessian matrix include $\lambda_1 < 0$ and $\lambda_2 > 0$, the Hessian matrix is not semi-positive definite. Consequently, the first term of the optimization problem is non-convex. Similarly, it can be demonstrated that the second, third, and fourth terms are also non-convex.

We initially simplified the problem and derived each term independently. When reverting to the full problem, the Hessian matrix becomes a 5×5 matrix, which is inevitably

not semi-positive definite due to the complexity of the variables involved. Therefore, the original problem is inherently a complex, non-convex problem.

References

1. Zaid, A.A.; Belmekki, B.E.Y.; Alouini, M.-S. eVTOL communications and networking in UAM: Requirements, key enablers, and challenges. *IEEE Commun. Mag.* **2023**, *61*, 154–160. [\[CrossRef\]](#)
2. Zeng, T.; Semiari, O.; Saad, W.; Bennis, M. Wireless-enabled asynchronous federated fourier neural network for turbulence prediction in urban air mobility (UAM). *IEEE Trans. Wirel. Commun.* **2023**, *22*, 7902–7916. [\[CrossRef\]](#)
3. Park, J.; Kim, I.; Suk, J.; Kim, S. Trajectory optimization for takeoff and landing phase of UAM considering energy and safety. *Aerosp. Sci. Technol.* **2023**, *140*, 108489. [\[CrossRef\]](#)
4. Tan, F.; Pang, S.; Cao, Y.; Chen, H.; Lv, T. Energy-Efficient Device-to-Device Communications for Green Internet of Things Using Unmanned Aerial Vehicle-Mounted Intelligent Reflecting Surface. *Drones* **2024**, *8*, 122. [\[CrossRef\]](#)
5. Cheng, K.; Fang, X.; Wang, X. Energy Efficient Edge Computing and Data Compression Collaboration Scheme for UAV-Assisted Network. *IEEE Trans. Veh. Technol.* **2023**, *72*, 16395–16408. [\[CrossRef\]](#)
6. Feng, C.; Han, P.; Zhang, C.; Yang, B.; Liu, Y.; Guo, L. Computation offloading in mobile edge computing networks: A survey. *J. Netw. Comput. Appl.* **2022**, *202*, 103366. [\[CrossRef\]](#)
7. Dai, X.; Xiao, Z.; Jiang, H.; Lui, J.C.S. UAV-assisted task offloading in vehicular edge computing networks. *IEEE Trans. Mob. Comput.* **2024**, *23*, 2520–2534. [\[CrossRef\]](#)
8. Li, F.; Luo, J.; Qiao, Y.; Li, Y. Joint UAV Deployment and Task Offloading Scheme for Multi-UAV-Assisted Edge Computing. *Drones* **2023**, *7*, 284. [\[CrossRef\]](#)
9. Chen, Y.; Liu, Z.; Zhang, Y.; Wu, Y.; Chen, X.; Zhao, L. Deep reinforcement learning-based dynamic resource management for mobile edge computing in industrial internet of things. *IEEE Trans. Ind. Inform.* **2021**, *17*, 4925–4934. [\[CrossRef\]](#)
10. Guo, Y.; Zhao, R.; Lai, S.; Fan, L.; Lei, X.; Karagiannis, G.K. Distributed machine learning for multiuser mobile edge computing systems. *IEEE J. Sel. Top. Signal Process.* **2022**, *16*, 460–473. [\[CrossRef\]](#)
11. Dai, X.; Xiao, Z.; Jiang, H.; Alazab, M.; Lui, J.C.; Dustdar, S.; Liu, J. Task co-offloading for D2D-assisted mobile edge computing in industrial internet of things. *IEEE Trans. Ind. Inform.* **2022**, *19*, 480–490. [\[CrossRef\]](#)
12. Chai, F.; Zhang, Q.; Yao, H.; Xin, X.; Gao, R.; Guizani, M. Joint multi-task offloading and resource allocation for mobile edge computing systems in satellite IoT. *IEEE Trans. Veh. Technol.* **2023**, *72*, 7783–7795. [\[CrossRef\]](#)
13. Cohen, A.P.; Shaheen, S.A.; Farrar, E.M. Urban air mobility: History, ecosystem, market potential, and challenges. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 6074–6087. [\[CrossRef\]](#)
14. Deng, C.; Fang, X.; Wang, X. UAV-enabled mobile-edge computing for AI applications: Joint model decision, resource allocation, and trajectory optimization. *IEEE Internet Things J.* **2023**, *10*, 5662–5675. [\[CrossRef\]](#)
15. Chen, F.; Zhou, J.; Xia, X.; Xiang, Y.; Tao, X.; He, Q. Joint Optimization of Coverage and Reliability for Application Placement in Mobile Edge Computing. *IEEE Trans. Serv. Comput.* **2023**, *16*, 3946–3957. [\[CrossRef\]](#)
16. Lu, W.; Mo, Y.; Feng, Y.; Gao, Y.; Zhao, N.; Wu, Y.; Nallanathan, A. Secure transmission for multi-UAV-assisted mobile edge computing based on reinforcement learning. *IEEE Trans. Netw. Sci. Eng.* **2023**, *10*, 1270–1282. [\[CrossRef\]](#)
17. Sun, C.; Ni, W.; Wang, X. Joint computation offloading and trajectory planning for UAV-assisted edge computing. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 5343–5358. [\[CrossRef\]](#)
18. Savkin, A.V.; Huang, C.; Ni, W. Joint multi-UAV path planning and LoS communication for mobile-edge computing in IoT networks with RISs. *IEEE Internet Things J.* **2023**, *10*, 2720–2727. [\[CrossRef\]](#)
19. Sabuj, S.R.; Asiedu, D.K.P.; Lee, K.-J.; Jo, H.-S. Delay optimization in mobile edge computing: Cognitive UAV-assisted eMBB and mMTC services. *IEEE Trans. Cogn. Commun. Netw.* **2022**, *8*, 1019–1033. [\[CrossRef\]](#)
20. Song, F.; Xing, H.; Wang, X.; Luo, S.; Dai, P.; Xiao, Z.; Zhao, B. Evolutionary multi-objective reinforcement learning based trajectory control and task offloading in UAV-assisted mobile edge computing. *IEEE Trans. Mob. Comput.* **2023**, *22*, 7387–7405. [\[CrossRef\]](#)
21. Al-Rubaye, S.; Tsourdos, A.; Namuduri, K. Advanced Air Mobility Operation and Infrastructure for Sustainable Connected eVTOL Vehicle. *Drones* **2023**, *7*, 319. [\[CrossRef\]](#)
22. Xu, Y.; Zhang, T.; Liu, Y.; Yang, D.; Xiao, L.; Tao, M. UAV-assisted MEC networks with aerial and ground cooperation. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 7712–7727. [\[CrossRef\]](#)
23. Zhao, N.; Ye, Z.; Pei, Y.; Liang, Y.-C.; Niyato, D. Multi-agent deep reinforcement learning for task offloading in UAV-assisted mobile edge computing. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 6949–6960. [\[CrossRef\]](#)
24. Ranjha, A.; Naboulsi, D.; Emary, M.E.; Gagnon, F. Facilitating URLLC vis-à-vis UAV-enabled relaying for MEC systems in 6-G networks. *IEEE Trans. Reliab.* **2024**, *1*–15. [\[CrossRef\]](#)
25. Lu, Y.; Xu, C.; Wang, Y. Joint Computation Offloading and Trajectory Optimization for Edge Computing UAV: A KNN-DDPG Algorithm. *Drones* **2024**, *8*, 564. [\[CrossRef\]](#)

26. Li, X.; Fang, Y.; Pan, C.; Cai, Y.; Zhou, M. Resource Scheduling for UAV-Assisted Failure-Prone MEC in Industrial Internet. *Drones* **2023**, *7*, 259. [\[CrossRef\]](#)
27. Liu, J.; Xu, H.; Wang, L.; Xu, Y.; Qian, C.; Huang, J.; Huang, H. Adaptive asynchronous federated learning in resource-constrained edge computing. *IEEE Trans. Mob. Comput.* **2023**, *22*, 674–690. [\[CrossRef\]](#)
28. Chen, R.; Li, L.; Xue, K.; Zhang, C.; Pan, M.; Fang, Y. Energy Efficient Federated Learning Over Heterogeneous Mobile Devices via Joint Design of Weight Quantization and Wireless Transmission. *IEEE Trans. Mob. Comput.* **2023**, *22*, 7451–7465. [\[CrossRef\]](#)
29. Li, H.; Pan, Y.; Zhu, H.; Gong, P.; Wang, J. Resource Management for MEC Assisted Multi-Layer Federated Learning Framework. *IEEE Trans. Wirel. Commun.* **2024**, *23*, 5680–5693. [\[CrossRef\]](#)
30. Yu, H.; Liu, Y.; Chen, M.; Complex-Valued Neural-Network-Based Federated Learning for Multiuser Indoor Positioning Performance Optimization. *IEEE Internet Things J.* **2024**, *11*, 34065–34077. [\[CrossRef\]](#)
31. Chen, M.; Gündüz, D.; Huang, K.; Saad, W.; Bennis, M.; Feljan, A.V.; Poor, H.V.. Distributed Learning in Wireless Networks: Recent Progress and Future Challenges. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 3579–3605. [\[CrossRef\]](#)
32. Feng, C.; Zhao, Z.; Wang, Y.; Quek, T.Q.S.; Peng, M. On the design of federated learning in the mobile edge computing systems. *IEEE Trans. Commun.* **2021**, *69*, 5902–5916. [\[CrossRef\]](#)
33. Zhang, X.; Chang, Z.; Hu, T.; Chen, W.; Zhang, X.; Min, G. Vehicle Selection and Resource Allocation for Federated Learning-Assisted Vehicular Network. *IEEE Trans. Mob. Comput.* **2024**, *23*, 3817–3829. [\[CrossRef\]](#)
34. Kim, J.; Kim, D.; Lee, J.; Hwang, J. A novel joint dataset and computation management scheme for energy-efficient federated learning in mobile edge computing. *IEEE Commun. Lett.* **2022**, *11*, 898–902. [\[CrossRef\]](#)
35. Luo, Z.; Pan, C.; Wang, S.; Guo, L. Federated-td3:joint computing offloading and resource allocation in urban air mobility. In Proceedings of the 2024 IEEE/CIC International Conference on Communications in China (ICCC Workshops), Hangzhou, China, 7–9 August 2024; pp. 627–632.
36. Maeng, S.J.; Park, J.; Guvenç, İ. Analysis of UAV Radar and Communication Network Coexistence With Different Multiple Access Protocols. *IEEE Trans. Commun.* **2023**, *71*, 6578–6592. [\[CrossRef\]](#)
37. Wu, W.; Zeng, Y.; Zhang, R. Joint Trajectory and Communication Design for Multi-UAV Enabled Wireless Networks. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 2109–2121. [\[CrossRef\]](#)
38. Duo, B.; Wu, Q.; Yuan, X.; Zhang, R. Energy Efficiency Maximization for Full-Duplex UAV Secrecy Communication. *IEEE Trans. Veh. Technol.* **2020**, *69*, 4590–4595. [\[CrossRef\]](#)
39. Li, H.; Xu, H.; Zhou, C.; Lü, X.; Han, Z. Joint Optimization Strategy of Computation Offloading and Resource Allocation in Multi-Access Edge Computing Environment. *IEEE Trans. Veh. Technol.* **2020**, *69*, 10214–10226. [\[CrossRef\]](#)
40. Li, X.; Zhang, J.; Pan, C. Federated deep reinforcement learning for energy-efficient edge computing offloading and resource allocation in industrial internet. *Appl. Sci.* **2023**, *13*, 6708. [\[CrossRef\]](#)

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.