applied sciences

MDPI

*Article*

# UAV Cluster-Assisted Task Offloading for Emergent Disaster Scenarios

**Minglin Shi , Xiaoqi Zhang, Jia Chen and Hongju Cheng ***

College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China
* Correspondence: cscheng@fzu.edu.cn

**Abstract:** Natural disasters often have an unpredictable impact on human society and can even cause significant problems, such as damage to communication equipment in disaster areas. In such post-disaster emergency rescue situations, unmanned aerial vehicles (UAVs) are considered an effective tool by virtue of high mobility, easy deployment, and flexible communication. However, the limited size of UAVs leads to bottlenecks in battery capacity and computational power, making it challenging to perform overly complex computational tasks. In this paper, we propose a UAV cluster-assisted task-offloading model for disaster areas, by adopting UAV clusters as aerial mobile edge servers to provide task-offloading services for ground users. In addition, we also propose a deep reinforcement learning-based UAV cluster-assisted task-offloading algorithm (DRL-UCTO). By modeling the energy efficiency optimization problem of the system model as a Markov decision process and jointly optimizing the UAV flight trajectory and task-offloading policy to maximize the reward value, DRL-UCTO can effectively improve the energy use efficiency of UAVs under limited-resource conditions. The simulation results show that the DRL-UCTO algorithm improves the UAV energy efficiency by about 79.6% and 301.1% compared with the DQN and Greedy algorithms, respectively.

**Keywords:** UAV cluster; task offloading; trajectory optimization; deep reinforcement learning; emergent disaster scenarios

check for updates

**Citation:** Shi, M.; Zhang, X.; Chen, J.; Cheng, H. UAV Cluster-Assisted Task Offloading for Emergent Disaster Scenarios. *Appl. Sci.* **2023**, *13*, 4724. https://doi.org/10.3390/app13084724

Academic Editor: Juan A. Gómez-Pulido

Received: 21 March 2023
Revised: 3 April 2023
Accepted: 6 April 2023
Published: 9 April 2023

## 1. Introduction

Natural disasters such as earthquakes, typhoons, and mudslides often cause incalculable damage to human life and property. They often result in the breakdown or damage of traditional terrestrial wireless infrastructure, thus disrupting communication in the disaster area, which poses a considerable challenge for post-disaster rescue and restoration efforts. The frequency of recorded natural disasters worldwide has increased nearly five-fold over the past 50 years [1]. Although prevention before a disaster is vital, rescue and restoration efforts after a disaster are essential. The emergence of UAVs and intelligent technology has completely changed post-disaster rescue and repair [2]. With the advantages of high mobility, low cost, and flexible deployment, UAVs can easily reach disaster-stricken areas, so UAVs are considered an important tool for post-disaster regional communication restoration. In particular, features such as the airborne flight of UAVs without regard to obstacles and the existence of line-of-sight (LoS) links between UAVs and ground users have caused a research boom in using UAVs as aerial base stations [3]. In addition, the flexible communication mode of UAVs allows them to act as communication relays, thus effectively transmitting the situation and dispatch orders of post-disaster rescue to ground rescue equipment [4–6]. Based on the above advantages, UAVs are expected to play a more significant role in future rescue and disaster relief work.

However, there are two crucial challenges to UAV-assisted post-disaster communications efforts. First, although UAVs have more computing power relative to ground users, they are still insufficient to meet the demands of computationally intensive tasks with large amounts of data [7,8]. This means that if the ground users have a high complexity of computational tasks or many image-processing needs, UAVs may not be able to complete the computational tasks within a tolerable time range of the users, which will make it difficult to ensure the quality of service for the users. Second, due to the limitation of the volume and load-bearing capacity of the UAV, the battery capacity of the UAV is limited [9–11], which makes it unable to work continuously for a long time, and can only perform post-disaster auxiliary communication within a limited time and area. Furthermore, if the UAV is under high-load computing for a long time, it will shorten its battery life faster [12]. Therefore, it is necessary to make reasonable use of the flexible communication methods of UAVs and find a suitable solution for UAV-assisted post-disaster communication work to achieve effective and timely post-disaster rescue and significantly increase the sustainable operation time of UAVs.

There are many research attempts to solve the task-offloading problem of UAVs in disaster areas, partly mainly through the cooperation of UAVs and edge computing servers, allowing UAVs to act as communication relays to forward computational tasks to adjacent mobile edge computing servers, and completing computationally intensive tasks through the more substantial computing power of the servers [13,14]. Another part of the research focuses on offloading computational tasks to a more distant cloud server by UAVs, which completes the processing of complex tasks through the mighty computing power of the cloud server, and later transmits the computational results of the tasks back to the ground users via UAVs [15,16]. However, the current work does not take into account the lack of computing power of a single UAV, which may not be able to carry out a large number of tasks for computation and forwarding. Moreover, due to the impact of natural disasters, edge computing servers in disaster-stricken areas are often unavailable or even damaged, making it impossible to provide computing offloading services for UAVs. In addition, the long transmission distance and the time delay between UAVs and remote cloud servers, and offloading tasks to cloud servers via UAVs will make it difficult to guarantee the quality of service for ground users. Considering the sudden and unpredictable unknowns of natural disasters, hardly any current works find an effective solution to the problem of limited UAV resources and user survival cycle limitations. Therefore, it is urgent to design a reasonable solution for the UAV-assisted post-disaster communication problem.

Based on the above, this paper proposes a real-time and efficient UAV cluster-assisted task-offloading solution for the user task-offloading problem under the damaged ground communication infrastructure scenario in disaster areas. This solution enables UAV clusters to form an aerial mobile edge server by introducing clusters of UAVs with intelligent characteristics [17,18] to provide assisted communication and computational offloading services to ground users. The state transfer function is simplified by modeling the ground user task-offloading policy selection and UAV cluster flight trajectory formulation as Markov decision processes. Then, deep reinforcement learning algorithms are combined to jointly optimize user task-offloading policies and UAV cluster flight trajectories to maximize system energy efficiency, therefore further extending the system lifecycle. The solution not only overcomes the shortage of computing power of a single UAV, but also solves the problem of dynamic changes in the environment. The main contributions of this paper are summarized as follows:

- This paper proposes a UAV cluster-assisted task-offloading model in the disaster scenario. To ensure that the communications of ground users in the disaster area are repaired promptly, we adopt an aerial mobile edge server composed of UAV clusters to provide assisted communications and computational offloading services for ground users. The ground users first offload the computational tasks to the UAVs with free resources, and then the UAVs forward the task data within the cluster to collaboratively complete the tasks. The model significantly reduces the transmission energy consumption between ground users and traditional edge servers and overcomes the problem of the limited computing power of UAVs.
- This paper proposes a deep reinforcement learning-based UAV cluster-assisted task-offloading algorithm (DRL-UCTO) for jointly optimizing UAV flight trajectory and ground user task-offloading policy, taking full advantage of the high mobility and

flexible communication of UAVs. This paper simplifies the deep reinforcement learning state transfer model through the Markov decision process to make the modeling process feasible, and optimizes the UAV flight trajectory through deep reinforcement learning, so that the DRL-UCTO algorithm quickly locates the location of ground users and makes optimal task offloading and forwarding decisions. The problem of limited resources in the disaster area is solved by maximizing the energy efficiency of the system while guaranteeing the quality of service for users.

- In this paper, we verify the feasibility and effectiveness of the DRL-UCTO algorithm through extensive simulation experiments. Because of the stable action selection policy of the DRL-UCTO algorithm, the flight energy overhead of the UAV cluster during the search for ground users is reduced. The numerical results show that compared with other baseline algorithms, the DRL-UCTO algorithm significantly improves system transmission rate and throughput, and further increases system energy efficiency.

The rest of this paper is organized as follows. Section 2 describes the current work related to UAV applications. In Section 3, a system model using UAV clusters as aerial mobile edge servers is described, and optimization problems are presented. In Section 4, we propose the deep reinforcement learning-based UAV cluster-assisted task-offloading algorithm (DRL-UCTO) to jointly optimize the UAV cluster flight trajectory and task-offloading policy. Section 5 gives the simulation experimental results and performance evaluation. Finally, Section 6 describes the conclusions and future work.

## 2. Related Work

With the advantages of high mobility, low cost, and flexible deployment of UAVs, the applicability of UAVs in various fields is increasing. Therefore, much research has been conducted on the application of UAVs in the field of wireless communication. Because of the flexible communication and unique line-of-sight (LoS) links, most research uses UAVs as airborne wireless relays. On the one hand, UAVs are used to enhance the user service quality in areas with weak communication links, and provide stable and secure wireless communication links for these areas [19–21]. On the other hand, UAVs are used to expand the network coverage to provide communication coverage within a specified range for areas where no network infrastructure is deployed. Wang et al. [22] proposed a novel UAV-assisted IoT communication network using low-altitude UAVs as aerial anchoring nodes and mobile data collectors, thus assisting ground base stations for device localization and data collection. Huang et al. [23] jointly optimized the UAV flight trajectory and transmission power to minimize the network overhead under the premise of ensuring the quality of service of IoT devices. Zhu et al. [24] proposed a deep reinforcement learning-based UAV-assisted communication algorithm to minimize the average channel path loss by optimizing the UAV flight path. In addition, UAVs are used as effective IoT aids, not only as aerial base stations but also for aerial communication, sensing, and data analysis [25].

Recently, UAV-assisted communication networks have been considered for application in post-disaster scenarios where terrestrial wireless infrastructure has been damaged to provide temporary emergency communications for ground users. Because of the advantages of their own hardware, rotary-wing UAVs can be easily deployed in such post-disaster scenarios to provide service coverage for users in specific areas. Lin et al. [26] investigated how to deploy a rotary-wing UAV as a wireless base station in disaster scenarios to provide emergency communication to ground users with unknown distribution, and proposed two UAV trajectory planning schemes based on multi-armed slot machine algorithms to maximize the throughput of the system under UAV resource constraints. Lyu et al. [27] proposed a novel hybrid network architecture that uses UAVs to fly along the cell edges to provide computational offloading services to the users around the cell, and effectively improves the minimum throughput for all users by jointly optimizing bandwidth allocation, user partitioning, and UAV trajectories. However, the above work does not take into account the critical energy constraints, especially the limited-resource constraints in the disaster area that will be a crucial factor in extending the lifetime of the UAV-assisted communication network.

Optimizing the energy consumption of UAV-assisted communication networks has been of great interest, and several studies have attempted to improve the lifetime and energy consumption of networks by clustering [28–31]. In [32], the authors describe a comparison of clustering techniques for implementing energy-aware routing in device-to-device (D2D) communication in public safety networks, comparing several different clustering schemes in terms of energy consumption, throughput, and the number of dead nodes in the system through extensive simulation experiments, and the results indicate that the clustering scheme proposed by the authors outperforms the other compared schemes in all aspects. Although extended communication applications for UAVs in disaster scenarios have been extensively researched, there is still a need for continuous exploration to find a UAV-assisted task-offloading solution that can provide reliable services in unknown disaster scenarios. Unlike previous research that mainly focuses on the communication optimization and resource allocation of UAVs, this paper proposes a deep reinforcement learning-based UAV cluster-assisted task-offloading solution, which provides stable and efficient computational offloading services for ground users in the disaster area by jointly optimizing UAV flight trajectories and task-offloading policies.

## 3. System Model

In this paper, consider a typical UAV-assisted task-offloading system, as shown in Figure 1, which contains $G$ ground users and $U$ UAVs, used, respectively, the sets $\mathcal{G} = \{1, 2, \ldots, G\}$ and $\mathcal{U} = \{1, 2, \ldots, U\}$ are denoted. In the scenario described in this paper, it is assumed that the ground users' IoT devices have a certain amount of computing power to meet their own small computing task needs, but not enough to afford computationally intensive tasks with large data volumes. Therefore, adopting a UAV cluster as an aerial mobile edge server provides more substantial computing power for these ground users, which in turn helps them to complete the computation of large tasks. In addition, Table 1 summarizes the main notations used in this paper.

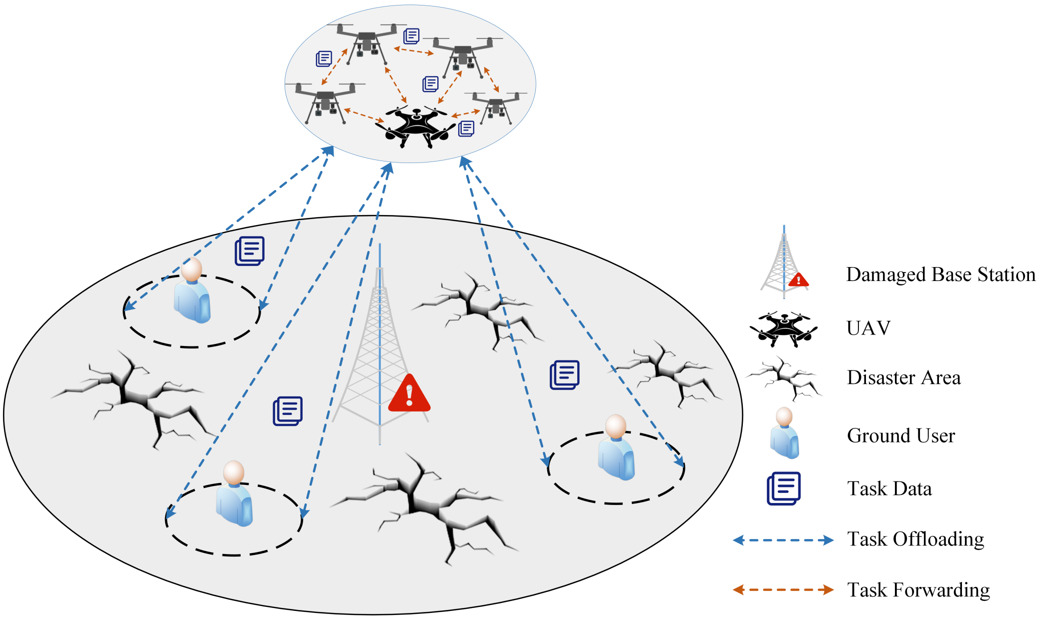


**Figure 1.** UAV-assisted task-offloading model for disaster scenarios.

**Table 1.** Table of notations.

| Notations | Description |
|---|---|
| $G$ | Number of ground users |
| $U$ | Number of UAVs |
| $T$ | Number of time slices |
| $\xi$ | The length of time slice |
| $L_g$ | The coordinates of ground user $g$ |
| $h$ | The UAV flight height |
| $L_u$ | The projected coordinates of UAV $u$ |
| $v_u$ | The horizontal flight speed of UAV $u$ |
| $E_u^{fly}$ | The propulsion energy consumption of UAV $u$ |
| $m_{uav}$ | The mass and load of UAV |
| $J_g$ | The computational task of ground user $g$ |
| $M_g$ | The amount of task data for user $g$ |
| $T_g$ | The task deadline of user $g$ |
| $s_g$ | Number of CPU cycles per bit of data to be executed for the task of user $g$ |
| $\lambda_{g1}$ | The proportion of task offloads for ground users $g$ |
| $\lambda_{g2}$ | The proportion of ground user $g$ tasks forwarded via UAVs |
| $T_g^{loc}$ | The local computation delay of ground user $g$ |
| $f_g$ | Number of CPU cycles per second that can be processed by ground user $g$ |
| $E_g^{loc}$ | The local computing energy consumption of ground user $g$ |
| $\mu$ | The effective switching capacitance |
| $h_{u,g}$ | The channel gain between UAV $u$ and ground user $g$ |
| $h_0$ | The average path gain at a reference distance of 1 meter |
| $P$ | The transmit power |
| $r_{u,g}$ | The communication rate between UAV $u$ and ground user $g$ |
| $B$ | The available bandwidth |
| $\sigma$ | The thermal noise power |
| $h_{u',u}$ | The channel gain between UAV $u'$ and UAV $u$ |
| $r_{u',u}$ | The communication rate between UAV $u'$ and UAV $u$ |
| $\alpha_{g,u}$ | The task offload state of ground user $g$ |
| $\beta_{g,u,u'}$ | The state of ground user $g$ forwarding tasks via UAV |
| $T_{g,u,u'}^{trans}$ | The transmission delay of UAV-assisted task offloading |
| $T_{g,u,u'}^{com}$ | The computational delay of UAV-assisted task offloading |
| $f_{g,u}$ | The computational resources obtained by ground user $g$ from UAV $u$ |
| $f_{u,u'}$ | The computational resources obtained by UAV $u$ from UAV $u'$ |
| $T_{g,u,u'}^{uav}$ | The total delay of UAV-assisted task offloading |
| $E_{g,u,u'}^{uav}$ | The total energy consumption of UAV-assisted task offloading |
| $T_{g,u,u'}^{total}$ | The total task processing delay of ground user $g$ |
| $E_{g,u,u'}^{total}$ | The total task processing energy consumption of ground user $g$ |
| $d_{\min}$ | The minimum safe distance between UAVs |
| $V_{\max}$ | The maximum flight speed of UAVs |

### *3.1. UAV Mobile Model*

This paper constructs the system region as a three-dimensional right-angle coordinate system, and for computational convenience divides the time equally into $T$ time slices, i.e., $t \in \mathcal{T} = \{1, 2, \ldots, T\}$, where the length of each time slice $t$ is $\xi$. Using $L_g(t) = [x_g(t), y_g(t)]$ to denote the coordinates of the ground user, assuming that the UAV cluster is flying in the air at a fixed height $h$, then the projected coordinates of the

UAV on the ground at moment $t$ can be expressed as $L_u(t) = [x_u(t), y_u(t)]$. Therefore, the horizontal flight speed of the UAV in time slice $t$ can be expressed as:

$$v_u(t) = \|L_u(t+1) - L_u(t)\| / \xi. \tag{1}$$

The primary energy overhead of the UAV in flight is the flight propulsion energy consumption, and by calculating the flight speed, the propulsion energy of the UAV in each time slice can be expressed as follows:

$$E_u^{fly}(t) = k(v_u(t))^2, \tag{2}$$

where $k = 0.5 m_{uav} \xi$ and $m_{uav}$ denotes the mass and load of the UAV.

### 3.2. Computational Process Model

First, the computational task of ground user $g$ is described as $J_g(t) = (M_g(t), T_g(t))$, where $M_g(t)$ denotes the data size of the task and $T_g(t)$ denotes the deadline of the task. If the current task cannot be completed within the specified deadline, it will be considered a failed execution. Define $s_g$ to denote the number of CPU cycles required to execute each bit of data for the task of user $g$. It is used to describe the complexity of the task.

Next, the variable $\{0 \le \lambda_{g2}(t) \le \lambda_{g1}(t) \le 1\}$ is defined to denote the proportion of data offloaded by the task, where $\lambda_{g1}(t)$ denotes the proportion of ground user $g$ that offloads part of the task data $M_g(t)$ to UAV $u$, and $\lambda_{g2}(t)$ denotes the proportion of data that UAV $u$ forwards to other UAVs $u'$ for collaborative computation from the part of the task data offloaded by ground user $g$. Then the proportion of task data performed locally by the ground user device is $1 - \lambda_{g1}(t)$, the proportion of tasks offloaded to UAV $u$ is $\lambda_{g1}(t) - \lambda_{g2}(t)$, and the proportion of tasks forwarded from UAV $u$ to UAV $u'$ is $\lambda_{g2}(t)$. The calculation process of task offloading is divided into two parts, which are local calculation by ground user devices and UAV cluster-assisted offloading calculation.

(1) Local calculation: The computation of the task is conducted by the ground user device itself, and the size of the task data it needs to compute is $(1 - \lambda_{g1}(t))M_g(t)$, then the local computation delay of the ground user $g$ can be expressed as:

$$T_g^{loc}(t) = \frac{(1 - \lambda_{g1}(t))}{f_g} s_g M_g(t), \tag{3}$$

where $f_g$ denotes the number of CPU cycles per second that the ground user device can process. Therefore, the local computing energy consumption of the ground user is:

$$E_g^{loc}(t) = \mu_1 (f_g)^3 T_g^{loc}(t), \tag{4}$$

where $\mu_1$ is the effective switching capacitance associated with the chip architecture of the device itself [33].

(2) UAV cluster-assisted offloading calculation: Usually, the communication channels between UAV clusters and ground user devices are modeled as large-scale and small-scale fading. However, in the scenario considered in this paper, the channel between the UAV cluster and ground users is usually dominated by the line-of-sight component, and the communication link has a strong line-of-sight range, so the effect of small-scale fading can be negligible. Therefore, the channel gain between the UAV $u$ and ground user $g$ at moment $t$ can be expressed as:

$$h_{u,g}(t) = h_0 \left( \sqrt{\|L_u(t) - L_g(t)\|^2 + h^2} \right)^{-2}, \tag{5}$$

where $h_0$ is the average path gain at a reference distance of 1 meter and $h$ is the UAV flight height. Assuming that the communication between the UAV and the ground user always uses $P$ as the transmit power, the instantaneous communication rate between the UAV $u$ and the ground user $g$ is expressed as:

$$r_{u,g}(t) = B\log_2\left(1 + \frac{Ph_{u,g}(t)}{\sigma^2}\right), \tag{6}$$

where $B$ denotes the available bandwidth. $\sigma$ denotes the thermal noise power, which is linearly proportional to the assigned bandwidth. Similarly, the channel gain between the UAVs in the cluster can be obtained as:

$$h_{u',u}(t) = h_0\left(\sqrt{\|L_{u'}(t) - L_u(t)\|^2 + h^2}\right)^{-2}. \tag{7}$$

The communication rate between UAVs is:

$$r_{u',u}(t) = B\log_2\left(1 + \frac{Ph_{u',u}(t)}{\sigma^2}\right). \tag{8}$$

We assume that the offloading behavior of ground user devices follows an offloading policy at each moment, where ground users perform task offloading at the beginning of each time slice. Therefore, it is assumed that each ground user can select at most one UAV for data transmission in each time slice, so the transmission scheduling should satisfy the following constraints:

$$\sum_{u=1}^{U} \alpha_{g,u}(t) \le 1, \quad \forall t, g, \tag{9}$$

where $\alpha_{g,u}(t)$ is a binary variable indicating the task offload state of ground user $g$ at moment $t$. If $\alpha_{g,u}(t) = 1$, it means that at moment $t$ ground user $g$ offloads the task data to UAV $u$ for assisted computation, otherwise it is 0. In other words, each ground user can offload the task to at most one UAV.

Due to the large amount of computationally intensive task data, a single UAV may not be able to store it entirely and needs to forward part of the task data to other UAVs in the cluster for collaborative computation. Therefore, this paper assumes that in each time slice, UAV $u$ can only forward the computational tasks offloaded by ground user $g$ to one other UAV $u'$, so the UAV forwarding policy needs to satisfy the following constraints:

$$\sum_{u'=1}^{U} \beta_{g,u,u'}(t) \le 1, \quad \forall t, g, u, \tag{10}$$

where $\beta_{g,u,u'}(t)$ are binary variables, $\beta_{g,u,u'}(t) = 1$ if the UAV $u$ forwards its own task data, and $\beta_{g,u,u'}(t) = 0$ if the computation of the task is conducted by it independently.

Since the amount of uploaded data for computationally intensive tasks is much larger than that for the return transmission of computation results, the delay and energy consumed for the return transmission of computation results through the downlink are ignorable. Therefore, the total transmission delay of introducing the UAV cluster for assisted task-offloading consists of two parts: the transmission delay of the ground user $g$ to offload the task data to the UAV $u$ on the one hand, and the transmission delay of the UAV $u$ to forward the task data to the UAV $u'$ on the other hand. Thus, the total transmission delay of UAV-assisted task offloading can be expressed as follows:

$$T^{trans}_{g,u,u'}(t) = \alpha_{g,u}(t)\left[\frac{\big(\lambda_{g1}(t) - \lambda_{g2}(t)\big)M_g(t)}{r_{u,g}(t)} + \beta_{g,u,u'}(t)\frac{\lambda_{g2}(t)M_g(t)}{r_{u',u}(t)}\right]. \tag{11}$$

The computational delay is similarly composed of the processing delay of the UAV $u$ and the processing delay of the UAV $u'$, denoted as follows:

$$T^{com}_{g,u,u'}(t) = \frac{\left(\lambda_{g1}(t) - \lambda_{g2}(t)\right)}{f_{g,u}} s_g M_g(t) + \frac{\lambda_{g2}(t)}{f_{u,u'}} s_g M_g(t), \tag{12}$$

where $f_{g,u}$ denotes the computational resources acquired by ground user $g$ from UAV $u$ and $f_{u,u'}$ denotes the computational resources acquired by UAV $u$ from UAV $u'$. Therefore, the total delay of the UAV cluster-assisted task offload is the sum of transmission delay and computation delay, expressed as follows:

$$T^{uav}_{g,u,u'}(t) = T^{trans}_{g,u,u'}(t) + T^{com}_{g,u,u'}(t). \tag{13}$$

The total energy consumption generated is:

$$E^{uav}_{g,u,u'}(t) = \mu_2 (f_u)^3 T^{uav}_{g,u,u'}(t), \tag{14}$$

where $\mu_2$ denotes the effective capacitance switch of the UAV CPU. Therefore, the total task processing delay of the ground user $g$ can be expressed as:

$$T^{total}_{g,u,u'}(t) = T^{loc}_{g}(t) + T^{uav}_{g,u,u'}(t). \tag{15}$$

The total task processing energy consumption consists of three components: ground user local computation energy consumption, UAV cluster-assisted task-offloading energy consumption, and flight energy consumption, denoted as:

$$E^{total}_{g,u,u'}(t) = E^{loc}_{g}(t) + E^{uav}_{g,u,u'}(t) + E^{fly}_{u}(t). \tag{16}$$

### *3.3. Optimization Problem Definition*

To extend the lifetime of the UAV-assisted task-offloading system while ensuring user service requirements, it is required to complete the computational tasks generated by ground users within a specified time frame and to enable each device to operate continuously with limited resources. In this paper, we propose to jointly optimize the UAV flight trajectory and task-offloading policy to maximize the energy efficiency of the system under the condition of limited resources of ground user devices and UAV clusters. Therefore, the optimization problem and constraints in this paper are formulated as:

$$\max \sum_{t=1}^{T} \sum_{g=1}^{G} M_g(t) / \sum_{u=1}^{U} \sum_{u'=1}^{U} E^{total}_{g,u,u'}(t) \tag{17}$$

$$\text{s.t.} \quad \mathcal{C}1 : \sum_{u=1}^{U} \alpha_{g,u}(t) \leq 1, \forall t, g, \tag{17a}$$

$$\mathcal{C}2 : \sum_{u'=1}^{U} \beta_{g,u,u'}(t) \leq 1, \forall t, g, u, \tag{17b}$$

$$\mathcal{C}3 : 0 \leq \lambda_{g2}(t) \leq \lambda_{g1}(t) \leq 1, \forall t, g, \tag{17c}$$

$$\mathcal{C}4 : T^{total}_{g,u,u'}(t) \leq T_g(t), \forall t, g, \tag{17d}$$

$$\mathcal{C}5 : \|L_u(t) - L_{u'}(t)\| \geq d_{\min}, \forall u, u' \in U, u \neq u', \tag{17e}$$

$$\mathcal{C}6 : \|L_u(t+1) - L_u(t)\| / \xi \leq V_{\max}, t = 1, \ldots, T-1. \tag{17f}$$

Constraint $\mathcal{C}1$ ensures that each ground user can select at most one UAV to upload task data. Constraint $\mathcal{C}2$ ensures that the task data uploaded by the ground user can be forwarded at most once. $\mathcal{C}3$ indicates the proportion of task offload and the proportion of UAV task forwarding for each ground user device. Constraint $\mathcal{C}4$ ensures that user-generated computational tasks are completed within a tolerable time frame. $\mathcal{C}5$ guarantees a minimum safe distance $d_{\min}$ between deployed UAVs to avoid collisions. $\mathcal{C}6$ limits the maximum flight speed of UAVs.

It can be seen that (17) is a mixed integer nonlinear programming problem (MINLP), which is usually difficult to solve by conventional methods due to the presence of multivariate variables $\alpha_{g,u}(t)$ and $\beta_{g,u,u'}(t)$, and nonconvex objective functions. In addition, the trajectory of the UAV is usually optimized by offline processing before scheduling the UAV, but the system does not have access to the latest changes in the terminal at the next moment before scheduling. Although theoretically, some heuristics may be able to alleviate this problem, it is impractical to try to explore all possible changes in the system environment. Given the high complexity and uncertainty of the problem, efficient and low-complexity alternatives are needed to solve the situation. Therefore, this paper introduces deep reinforcement learning algorithms to observe and learn the state changes in the environment and to solve the proposed problem. The next section specifies the proposed algorithm.

## 4. Deep Reinforcement Learning-Based UAV Cluster-Assisted Task Offloading Solution

Deep reinforcement learning is the product of combining deep learning and reinforcement learning, which not only integrates the powerful understanding ability of deep learning in visual perception problems, but also possesses the decision-making ability of reinforcement learning, which makes the reinforcement learning technology move toward practical applications and enable the solving of complex problems in realistic scenarios. To this end, this paper first models the optimization problem of the system as a Markov decision process, so that the goal of optimization is converted from maximizing the energy efficiency of the system to maximizing the reward, and then combines the deep reinforcement learning algorithm to optimize the problem.

### *4.1. Markov Decision Process*

The environmental states of the UAV cluster-assisted task-offloading model are divided according to time slices, and its state space is continuous and countable, and the state at the next moment is only related to the current state, so the system model satisfies Markov properties. Therefore, this paper transforms the UAV cluster-assisted task-offloading problem into a Markov decision process. Markov decision processes represent sequential decision problems with Markov transfer models and additional rewards in fully observable stochastic environments. The Markov decision process consists of a quaternion $(\mathcal{S}, \mathcal{A}, \gamma, \mathcal{R})$, where $\mathcal{S}$ is defined as the set of states, representing the observed UAV and ground user state information at each moment. The set of UAV actions executed on the observed states is defined as $\mathcal{A}$. After the UAV executes an action based on the current state, the environment will enter a new state, the reward return for the current action is generated, and the set of reward returns is then denoted by $\mathcal{R}$. $\gamma$ denotes the discount factor determining the present value of future rewards. These state changes and action choices constitute the policy $\pi$ of the intelligent agent, and the cumulative reward of the system is maximized by continuously updating the policy $\pi$ during long-term interaction learning with the environment. In this paper, the UAV is considered an intelligent agent, and the motion trajectory and resource information of the UAV and ground users are considered to be the environment. Next, the state space, action space, and reward functions are defined in detail:

1. $\mathcal{S}$ denotes the state space, and each state $s(t) \in \mathcal{S}$ is a multi-dimensional vector containing multiple parameters that represent the current state of the ground user and the UAV, including their current location and resource information, as well as the amount of task data and the task deadline for the ground user. Overall, the state $s(t)$ of each time slice is defined as:

$$s(t) = \big(L_g(t), L_u(t), M_g(t), T_g(t)\big), g \in \{1, 2, \ldots, G\}, u \in \{1, 2, \ldots, U\}. \tag{18}$$

2. $\mathcal{A}$ denotes the action space, which contains all feasible actions that the UAV can take. The main points are the flight direction and distance of the UAV, the task offloading of the ground user, the UAV task forwarding, and the proportion of data offloaded and forwarded. Therefore, the action $a(t)$ for each time slice can be defined as follows:

$$a(t) = \Big(d_u(t), \alpha_{g,u}(t), \beta_{g,u,u'}(t), \lambda_{g1}(t), \lambda_{g2}(t)\Big), g \in \{1, 2, \ldots, G\}, u, u' \in \{1, 2, \ldots, U\}, \tag{19}$$

where $d_u(t) \in \{(x_d, 0), (-x_d, 0), (0, y_d), (0, -y_d), (0, 0)\}$ denotes the UAV horizontal displacement vector, which is used to control the UAV flight direction or hovering. Thus, the UAV position for the next time slice can be expressed as $L_u(t+1) = L_u(t) + d_u(t)$.

3. $\gamma$ denotes the effect of the policy adopted by the current UAV on the reward returns at future moments, where $0 \le \gamma \le 1$.

4. $\mathcal{R}$ denotes the reward set, which consists of the reward returns for the actions performed by the UAV in the current environmental state. The reward function at a certain moment mainly indicates the energy efficiency of the system and gives a certain penalty when the UAV flies out of the service area. Thus, the reward function at moment *t* is defined as follows:

$$r(s(t), a(t)) = \sum_{g=1}^{G} M_g(t) / \sum_{u=1}^{U} \sum_{u'=1}^{U} E_{g,u,u'}^{total}(t). \tag{20}$$

### *4.2. Deep Reinforcement Learning-Based UAV Cluster-Assisted Task-Offloading Algorithm (DRL-UCTO) Idea*

Deep reinforcement learning has a strong dependence on the before and after states of the environment, and often the state at the next moment is not only related to the current state, but even has a close connection with the earlier state or the initial state. Due to many user devices and UAVs as well as multiple decision variables in the UAV cluster-assisted task offload model, such a high-dimensional and highly correlated state sequence will lead to a complex state transfer model of the system environment, which is difficult to model. In addition, multilayer neural networks in deep reinforcement learning indicate the existence of computations with higher order polynomials, which will lead to the gradient vanishing problem [34]. Therefore, this paper simplifies the environmental state transfer model for deep reinforcement learning by transforming the system problem into a Markov decision process so that the environmental state at the next moment is only related to the current state, thus making the modeling feasible. Based on this, the deep reinforcement learning-based UAV cluster-assisted task-offloading algorithm (DRL-UCTO) is proposed.

The DRL-UCTO algorithm is mainly based on the DDQN algorithm, whose core idea is to model the UAV cluster flight trajectory and the scheduling offload policy to ground user devices as discrete actions $a(t)$. To evaluate the action quality, each state-action pair $(s(t), a(t))$ is measured using a $Q$-value, where $Q$-value is defined as:

$$Q(s(t), a(t)) = \mathbb{E}\left[\sum_{t'=t}^{T} \gamma r\big(s(t'), a(t')\big) \mid s(t), a(t)\right]. \tag{21}$$

The meaning of $Q$-value indicates the cumulative discount reward expected under the current policy. During the learning process of interaction with the environment, the policy of the intelligent agent is continuously adjusted by evaluating the $Q$-value, and the adjusted policy is used to find the best action. In general, the policy uses a greedy algorithm to select the action that produces the highest $Q$-value in state $s(t)$, which obtains the optimal UAV trajectory and the optimal offloading policy, with the following expression:

$$\pi(s(t)) = \arg\max_{a(t)} Q(s(t), a(t)). \tag{22}$$

DRL-UCTO uses two deep neural networks with weights to replace the Q-table in the original Q-learning algorithm, thus solving the problem that the Q-learning algorithm cannot be used in high-dimensional action spaces, which are, respectively, called the main network and the target network. The main network generates predictions $Q(s(t), a(t) \mid \varphi^Q)$ based on the input states and actions, where $\varphi^Q$ denotes the weight parameter of the main network, and the main network also selects the behavioral action for the next state according to (22). In the deep neural network learning process, the weight parameter $\varphi^Q$ of the network is trained using the loss function.

$$L\left(\varphi^Q\right) = \mathbb{E}\left[y(t) - Q\left(s(t), a(t) \mid \varphi^Q\right)\right], \tag{23}$$

where $y(t)$ is the target value, which is evaluated by the target network $Q'$ while considering the predicted results of the main network $Q$, the calculation process is expressed as follows:

$$y(t) = r(s(t), a(t)) + \gamma Q'\left(s(t+1), \arg \max_{a(t+1)} Q\left(s(t+1), a(t+1) \mid \varphi^Q\right) \mid \varphi^{Q'}\right), \tag{24}$$

where $\varphi^{Q'}$ denotes the weight parameter of the target network. Since the traditional DQN uses the greedy policy in (22) to select the action with the largest $Q$ value, the obtained target value $Q'(s(t), a(t) \mid \varphi^{Q'})$ is often larger than the actual value, which leads to the overestimation problem. Therefore, the advantage of using DRL-UCTO in this paper is that it uses two different $Q$ networks for selecting and evaluating behavioral actions, and periodically updates the weight parameter $\varphi^{Q'}$ of the target network using the main network weight parameter $\varphi^Q$ instead, thus avoiding the overestimation problem.

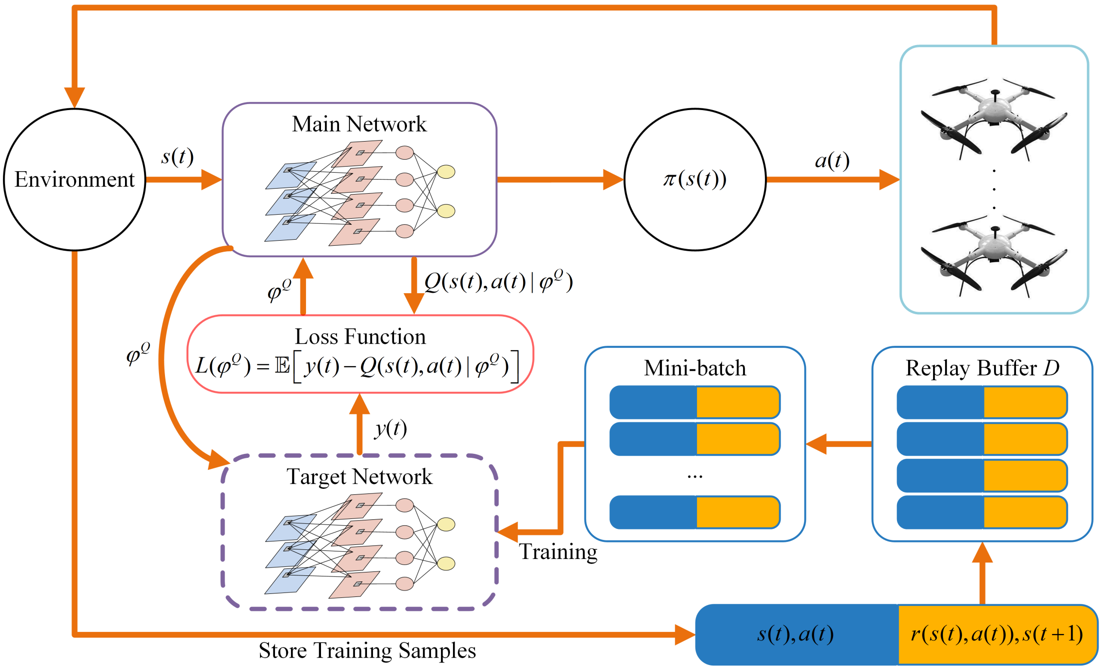


**Figure 2.** Deep reinforcement learning-based UAV cluster-assisted task-offloading algorithm (DRL-UCTO) flowchart.

Based on the above Markov decision process and deep neural networks, the process of the DRL-UCTO algorithm to solve the objective problem (17) is shown in Figure 2, where the target network and the experience replay technique are used. The target value $y(t)$ is calculated according to the DRL-UCTO algorithm, considering the results of the main network $Q$ and the target network $Q'$.

### *4.3. DRL-UCTO Algorithm Implementation*

The DRL-UCTO algorithm process is defined as shown in Algorithm 1, which consists of two phases for each training episode, the exploration phase (steps 4–15) and the training phase (steps 16–19). During the exploration phase, the UAV flight trajectory and the assisted offloading policy will be obtained by either the random or greedy methods defined above. If the current behavior causes the UAV to leave the service area, then the reward $r(s(t), a(t))$ in the current state is penalized, the current behavior is canceled, and the flight policy of the UAV is set to hover to control the UAV to stay in the current position. The environment will then move to the next state $s(t+1)$ and receive the corresponding current reward $r(s(t), a(t))$. Meanwhile, the sample data $(s(t), a(t), r(s(t), a(t)), s(t+1))$ will be stored in the experience buffer $D$.

In the training phase, this paper will use a random sampling method to randomly select $n$ training samples from the experience buffer for updating the main network $Q$ and the target network $Q'$, thus breaking the correlation between the data. First, the weight parameter $\varphi^Q$ of the main network is updated by minimizing the loss in step 18 through $y(t)$ generated by the target network. Next, the weight parameter $\varphi^{Q'}$ of the target network is periodically replaced by the weight parameter $\varphi^Q$ of the main network, thus completing the update of the target network, but the target network is updated less frequently than the main network, to avoid the overestimation problem. The algorithm execution is terminated until the end of the assigned training episodes, or the computational tasks of each ground user have been completed.

**Algorithm 1** Deep Reinforcement Learning-Based UAV Cluster-Assisted Task-Offloading Algorithm (DRL-UCTO).

```
1:  Initialize the replay buffer D;
2:  Randomly initialize main network $Q$ and target network $Q'$ with weights $\varphi^Q$ and $\varphi^{Q'}$,
    and set $\varphi^{Q'} = \varphi^Q$;
3:  for episode $= 1, \ldots, E$ do
4:    Initialize the environment to state s(1);
5:    for $t = 1, \ldots, T$ do
6:      Select action $a(t) \in A(s(t))$ based on policy $\pi(s(t))$;
7:      Execute $a(t)$ obtain $r(s(t), a(t))$ and $s(t+1)$;
8:      for $u = 1, \ldots, U$ do
9:        if UAV $u$ flies out of service scope then
10:         Apply the penalty to $r(s(t), a(t))$;
11:         Cancel the movement of UAV $u$ and update $s(t+1)$;
12:       end if
13:     end for
14:     Store training sample $(s(t), a(t), r(s(t), a(t)), s(t+1))$ in $D$;
15:   end for
16:   Select a random mini-batch of training samples $(s(i), a(i), r(s(i), a(i)), s(i+1))$ of
      size $n$ from $D$;
17:   $y(i) = r(s(i), a(i)) + \gamma Q'\Big(s(i+1), \arg\max_{a(i+1)} Q\big(s(i+1), a(i+1) \mid \varphi^Q\big) \mid \varphi^{Q'}\Big)$;
18:   $L(\varphi^Q) = \mathbb{E}\big[y(i) - Q\big(s(i), a(i) \mid \varphi^Q\big)\big]$;
19:   Update weights $\varphi^{Q'}$ of $Q'$: $\varphi^{Q'} = \varphi^Q$;
20: end for
```

## 5. Performance Evaluation and Analysis

In this section, this paper evaluates the performance of the proposed deep reinforcement learning-based UAV cluster-assisted task-offloading algorithm (DRL-UCTO) through simulation experiments. In the first subsection, we describe the simulation environment setup for the experiments and the compared baseline algorithms. In the second subsection, we analyze the optimization effect of the DRL-UCTO algorithm on the system performance.

### *5.1. Experimental Environment Settings*

In this paper, the DRL-UCTO algorithm is compared with the baseline algorithm to verify the effectiveness of the algorithm. To explore the differences between the DRL-UCTO algorithm and the basic deep reinforcement learning algorithm and non-reinforcement learning algorithm, the baseline algorithm uses the Deep Q Network (DQN) and Greedy algorithms. The DQN algorithm is a deep learning-based Q-learning algorithm that combines neural network and value function approximation techniques and uses experience replay to train the network. Greedy is an algorithm that considers locally optimal solutions and always chooses the behavior most favorable to the reward in the current situation. Good greedy policies often result in higher returns with lower overhead.

This paper considers a rectangular ground area of 500 m × 500 m in which 3 UAVs are deployed, while there are 30 ground users randomly distributed in the area. At the initial moment, all UAVs fly at a height of 200 m from an initial location with horizontal coordinates [0 m, 0 m] to search for ground users within the area and provide them with task-offloading services. This paper simulates the joint optimization solution of UAV flight trajectory and task-offloading policy using Python and TensorFlow. The hardware environment for the experiments is a 32-core AMD ROME high-performance computer with 64 GB RAM. The operating system of the computer is Ubuntu 18.04. We refer to the scenarios in the literature [35–37], and the relevant parameters are set in Table 2.

**Table 2.** Simulation parameter.

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $m_{uav}$ | 9.65 kg | $B$ | 2 MHz |
| $f_g$ | 1 GHz | $P$ | 0.1 W |
| $f_u$ | 8 GHz | $\sigma^2$ | −110 dBm |
| $s_g$ | 1000∼2000 | $D$ | 10,000 |
| $\mu_1$ | $10^{-24}$ | $n$ | 128 |
| $\mu_2$ | $10^{-24}$ | $M_g$ | 1∼3 Mbit |
| $h_0$ | −30 dB | $T_g$ | 10∼20 s |

### *5.2. Result Analysis*

First, the convergence of the algorithm is verified, as shown in Figure 3a. In the initial stage, the policy is not yet stable due to the lack of interaction between the intelligent agent and the environment, so the UAV keeps exploring and searching for ground users in the system area, and the reward value grows faster. After 700 iterations, the DRL-UCTO algorithm policy gradually stabilizes, and the reward value starts to converge and reach a high level. Second, Figure 3b compares the effect of different learning rates on the training efficiency of the algorithms by plotting the loss of Equation (23). The two pairs of curves in the figure show that the DRL-UCTO algorithm and the DQN algorithm have comparable training effects when the learning rate is 0.005. Furthermore, when the learning rate is 0.001, the DRL-UCTO algorithm possesses a higher training efficiency and can make the loss lower for the same number of training steps. It can be observed that the DRL-UCTO algorithm can quickly reach the convergence state and obtain a better reward value in the disaster area rescue scenario, resulting in an effective improvement of the system energy efficiency. In addition, if more training samples are provided for the DRL-UCTO algorithm, it may further improve the performance of the algorithm.

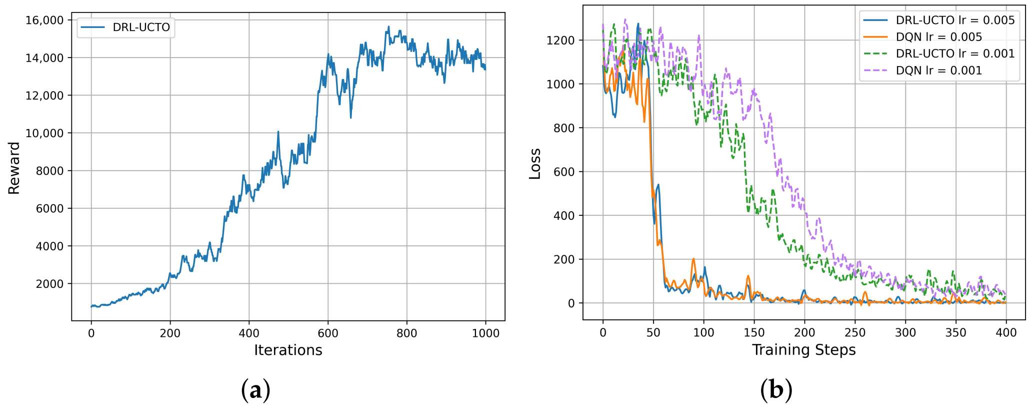


**Figure 3.** Convergence verification of the algorithm. (**a**) Reward value versus number of iterations for the DRL-UCTO algorithm. (**b**) Loss versus training steps for different algorithms.

The speed of the transmission rate will affect the overall latency of ground users offloading computational tasks to the UAV for processing, and a faster transmission rate can make the UAV-assisted task-offloading solution more efficient. To compare the effects of different algorithms on the transmission rate between the UAV and the ground user, simulation experiments were conducted in this paper on the changes in their transmission rates with time growth for DRL-UCTO and the other two baseline algorithms, and the results are shown in Figure 4a. It can be seen that all three algorithms have a low transmission rate between the UAV and the ground user in the first 50 training episodes. After several training episodes, their transmission rates are improved accordingly. Especially the DRL-UCTO algorithm, whose transmission rate proliferates and gains a significant increase over time, followed by the DQN algorithm, while the Greedy algorithm causes the UAVs to blindly approach a certain ground user, which leads to an increase in the distance between the UAV cluster and other ground users and a slow increase in the transmission rate. The DRL-UCTO algorithm proposed in this paper can enable the UAV to find the location of ground users and establish communication faster in the disaster area, and can further improve the transmission rate between the UAV and ground users by optimizing the UAV flight trajectory. In Figure 4b, this paper further compares the optimization of the three algorithms on the system throughput. It can be seen that the DRL-UCTO algorithm still has better performance in terms of optimizing the system throughput. In contrast, the DQN and Greedy algorithms have a more limited effect on the system throughput.

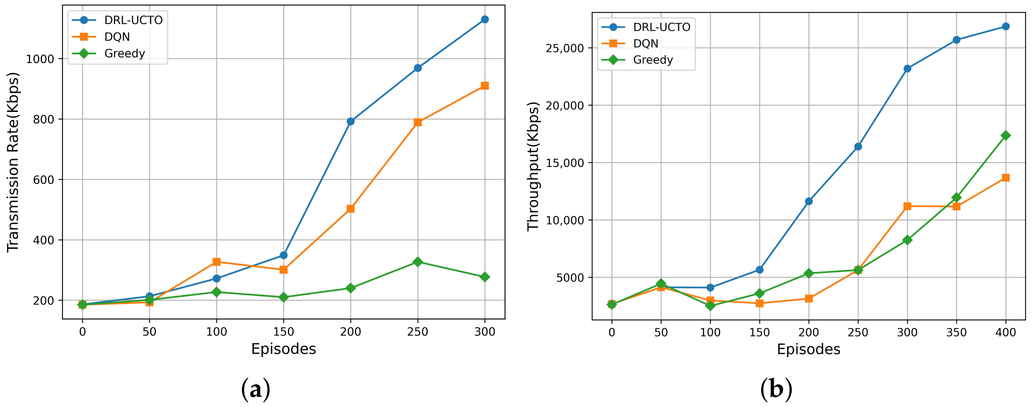
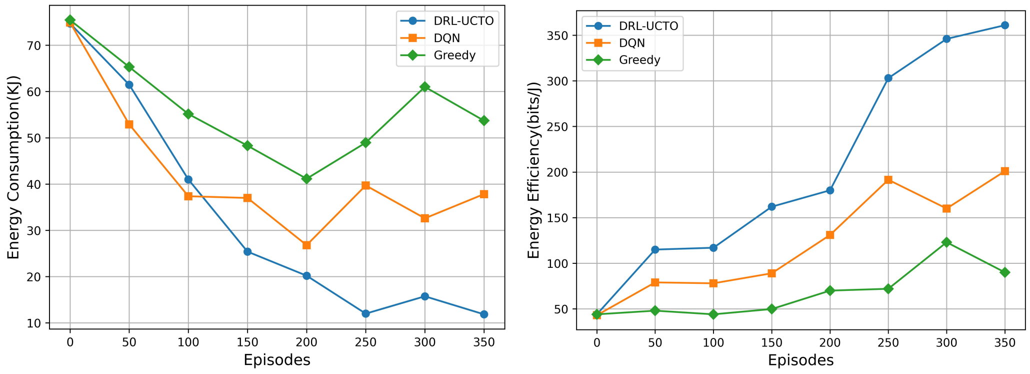


**Figure 4.** The performance of different algorithms in terms of transmission rate and throughput. (**a**) Transmission rate of UAVs versus training episodes. (**b**) System throughput versus training episodes.

Figure 5a compares the performance of the three algorithms in terms of the energy consumption overhead of the computational tasks. As the number of training episodes increases, the UAV flight trajectory is optimized, the distance between the UAV cluster and the ground user is shrinking, and the energy consumed for transmission is decreasing. All three algorithms showed some improvement in the energy consumption generated by the system processing tasks, especially the DRL-UCTO algorithm, which kept the energy consumption of the system at a low level after 200 training episodes.

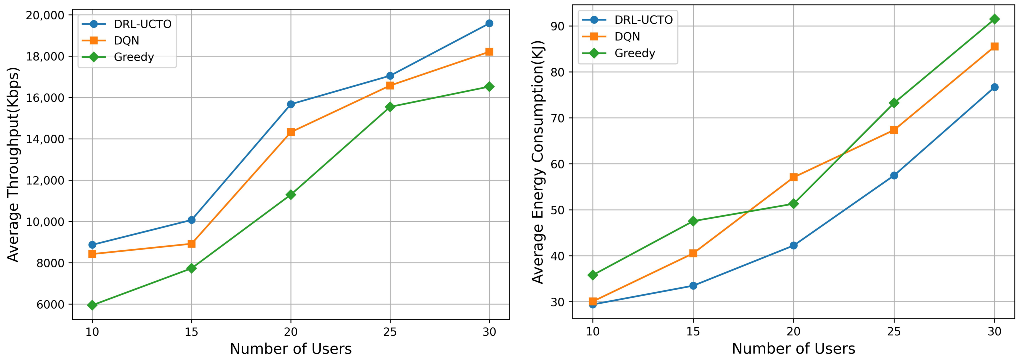
In resource-limited disaster areas, the battery capacity limitation of the UAV will be a crucial factor in extending the life cycle of the system, so the energy efficiency of the UAV is an important indicator to verify the effectiveness of the algorithm. In this paper, the three algorithms are trained for 350 episodes under the same environmental parameters, and the energy efficiency of the UAVs changes as shown in Figure 5b. As the UAV moves, the distance between it and the ground user keeps growing closer, the energy consumption generated in data transmission keeps decreasing, and the energy efficiency is improved accordingly. Thus, the energy efficiency of the UAVs under the three algorithms continues to improve as the number of training episodes increases. As can be seen from the figure, the energy efficiency of the UAV with the DRL-UCTO algorithm is always higher than the other two baseline algorithms under the same training episodes, and has a stable trend after 300 training episodes, while the optimization of the energy efficiency of the UAV with the Greedy algorithm is less satisfactory and remains at a lower level.



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

**Figure 5.** Impact of different algorithms on system energy indicators. (**a**) System energy consumption versus training episodes. (**b**) System energy efficiency versus training episodes.

Figure 6 compares the behavioral policies obtained by the DRL-UCTO algorithm with other baseline algorithms for different numbers of ground user scenarios. As shown in the figure, DRL-UCTO shows better performance than other baseline algorithms for different numbers of user scenarios. With the increase in the number of users, the average throughput of the system shows an upward trend. However, the number of users becomes larger, making it impossible for the UAV to approach every user simultaneously, so the throughput growth is relatively slow. However, the DRL-UCTO algorithm still obtained higher throughput, and the DQN and Greedy algorithms only obtained suboptimal behavioral policies for the same number of training episodes, which could not be well adapted to different numbers of user scenarios, proving that the DRL-UCTO algorithm is more robust. As for the average energy consumption of the system, the DRL-UCTO algorithm shows a performance closer to the optimal global solution. As the number of ground users increases, the state space to be considered by the three algorithms further increases, and the difficulty of making behavioral decisions increases accordingly, leading to a rise in the average energy consumption of the system. However, the DRL-UCTO algorithm shows a slower growth trend than the other baseline algorithms, while the DQN and Greedy algorithms show comparable performance results. Because both the DQN and Greedy algorithms are suboptimal policies leading to significant variations in the results.



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

**Figure 6.** Comparison of system average throughput and average energy consumption under a different number of user scenarios. (**a**) System average throughput versus number of users. (**b**) System average energy consumption versus number of users.

In addition, we compare the DRL-UCTO algorithm with the DQN algorithm and the Q-learning algorithm described in [38] under a scenario with 30 users to explore the effect of deep reinforcement learning versus traditional reinforcement learning on the average energy consumption of the system. As shown in Table 3, $S$ and $A$ denote the state and action space size, respectively, and $T$ denotes the number of time slices for each training episode. It can be seen that the complexity of the deep reinforcement learning algorithm is higher than that of the traditional reinforcement learning algorithm, because deep reinforcement learning uses deep neural networks instead of Q-tables in Q-learning. It overcomes the difficulty for traditional reinforcement learning to adapt to high-dimensional state spaces, but it also leads to an increase in the complexity of the algorithm. Deep reinforcement learning avoids the problem of dimensional explosion and has stable performance over a large state space. Therefore, the average energy consumption of the DRL-UCTO and DQN algorithms is lower than that of the Q-learning algorithm in scenarios with a higher number of users.

**Table 3.** Comparison of DRL-UCTO with other reinforcement learning algorithms.

| **Algorithm** | **Complexity** | **Average Energy Consumption (KJ)** |
|---|---|---|
| DRL-UCTO | $O(SAT)$ | 76.68 |
| DQN | $O(SAT)$ | 85.51 |
| Q-learning | $O(SA)$ | 89.47 |

This paper next explores the impact of deploying different numbers of UAVs in the same environment on the energy consumption of the system, as shown in Figure 7. It can be simply seen that as the number of UAVs deployed in the system increases, the energy consumption generated by the system also grows, because deploying more UAVs in the system requires correspondingly more energy to be consumed for the propulsion energy overhead generated by maintaining the UAVs in flight. The energy consumption of DQN and Greedy algorithms grows faster as the number of UAVs increases, especially the DQN algorithm, because it is prone to overestimation problems that lead to excessive UAV exploration in areas without ground users, thus causing the UAV to spend more energy on flight overheads. On the contrary, the DRL-UCTO algorithm can quickly move the UAV to the area of the ground user to complete the task-offloading requirements of the user due to its stable action selection policy. As a result, the DRL-UCTO algorithm generates relatively low energy consumption and grows more slowly, which means that the DRL-UCTO algorithm is more adaptable and stable than the other two baseline algorithms. Deploying more UAVs in the system will lead to an increase in energy consumption, but it will likewise result in an effective increase in the computing power of the system, which can provide stronger task-offloading services to ground users.

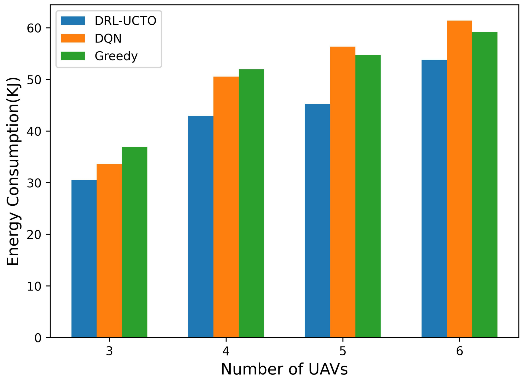


**Figure 7.** Impact of deploying different numbers of UAVs on system energy consumption.

## 6. Conclusions

In this paper, we propose a UAV cluster-assisted task-offloading model for disaster scenarios, by adopting UAV clusters as aerial mobile edge servers to provide task-offloading services to ground users. Moreover, we model the optimization problem of system energy efficiency as a Markov decision process, and convert the optimization objective from maximizing energy efficiency to maximizing reward value. Finally, we propose a deep reinforcement learning-based UAV cluster-assisted task-offloading algorithm (DRL-UCTO) to jointly optimize UAV flight trajectories and task-offloading policies. The simulation experimental results show that the DRL-UCTO algorithm improves the energy efficiency of the UAV-assisted task-offloading system by 79.6% and 301.1% compared to the DQN and Greedy algorithms, respectively.

In future work, we consider reducing the number of task passes by optimizing the routing between multiple UAVs to further improve the transmission efficiency of UAVs. Meanwhile, we will explore the extended applications of UAVs in more scenarios.

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

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data used to support the findings of this study are available on request from the corresponding author.

**Acknowledgments:** The authors would like to thank all the anonymous reviewers for their insightful comments and constructive suggestions.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Zheng, Y.J.; Chen, Q.Z.; Ling, H.F.; Xue, J.Y. Rescue wings: Mobile computing and active services support for disaster rescue. *IEEE Trans. Serv. Comput.* **2015**, *9*, 594–607. [CrossRef] [CrossRef]
2. Erdelj, M.; Król, M.; Natalizio, E. Wireless sensor networks and multi-UAV systems for natural disaster management. *Comput. Netw.* **2017**, *124*, 72–86. [CrossRef] [CrossRef]
3. Alzenad, M.; El-Keyi, A.; Lagum, F.; Yanikomeroglu, H. 3-D placement of an unmanned aerial vehicle base station (UAV-BS) for energy-efficient maximal coverage. *IEEE Wirel. Commun. Lett.* **2017**, *6*, 434–437. [CrossRef] [CrossRef]

4. Zhang, S.; Zhang, H.; He, Q.; Bian, K.; Song, L. Joint trajectory and power optimization for UAV relay networks. *IEEE Commun. Lett.* **2017**, 22, 161–164. [CrossRef] [CrossRef]
5. Fu, H.; Hu, P.; Zheng, Z.; Das, A.K.; Pathak, P.H.; Gu, T.; Zhu, S.; Mohapatra, P. Towards automatic detection of nonfunctional sensitive transmissions in mobile applications. *IEEE. Trans. Mob. Comput.* **2020**, *20*, 3066–3080. [CrossRef] [CrossRef]
6. Zhu, J.; Zhao, H.; Wei, Y.; Ma, C.; Lv, Q. Unmanned aerial vehicle computation task scheduling based on parking resources in post-disaster rescue. *Appl. Sci.* **2023**, *13*, 289. [CrossRef] [CrossRef]
7. 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] [CrossRef]
8. He, X.; Jin, R.; Dai, H. Multi-hop task offloading with on-the-fly computation for multi-UAV remote edge computing. *IEEE Trans. Commun.* **2021**, *70*, 1332–1344. [CrossRef] [CrossRef]
9. Qin, Y.; Kishk, M.A.; Alouini, M.S. Performance evaluation of UAV-enabled cellular networks with battery-limited drones. *IEEE Commun. Lett.* **2020**, *24*, 2664–2668. [CrossRef] [CrossRef]
10. Zhang, X.; Duan, L. Optimal patrolling trajectory design for multi-uav wireless servicing and battery swapping. In *2019 IEEE Globecom Workshops, Proceedings of the IEEE Communications Society's Flagship Global Communications Conference, Waikoloa, HI, USA, 9–13 December 2019*; IEEE: Piscataway, NY, USA, 2019; pp. 1–6.
11. Abeywickrama, H.V.; Jayawickrama, B.A.; He, Y.; Dutkiewicz, E. Comprehensive energy consumption model for unmanned aerial vehicles, based on empirical studies of battery performance. *IEEE Access* **2018**, *6*, 58383–58394. [CrossRef] [CrossRef]
12. Mir, T.; Waqas, M.; Tu, S.; Fang, C.; Ni, W.; MacKenzie, R.; Xue, X.; Han, Z. Relay hybrid precoding in uav-assisted wideband millimeter-wave massive mimo system. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 7040–7054. [CrossRef] [CrossRef]
13. Zou, Y.; Yu, D.; Hu, P.; Yu, J.; Cheng, X.; Mohapatra, P. Jamming-Resilient Message Dissemination in Wireless Networks. *IEEE. Trans. Mob. Comput.* **2021**, *22*, 1536–1550. [CrossRef] [CrossRef]
14. Zhang, T.; Xu, Y.; Loo, J.; Yang, D.; Xiao, L. Joint computation and communication design for UAV-assisted mobile edge computing in IoT. *IEEE Trans. Ind. Inform.* **2019**, *16*, 5505–5516. [CrossRef] [CrossRef]
15. Sacco, A.; Esposito, F.; Marchetto, G.; Montuschi, P. Sustainable task offloading in UAV networks via multi-agent reinforcement learning. *IEEE Trans. Veh. Technol.* **2021**, *70*, 5003–5015. [CrossRef] [CrossRef]
16. Chen, W.; Liu, B.; Huang, H.; Guo, S.; Zheng, Z. When UAV swarm meets edge-cloud computing: The QoS perspective. *IEEE Netw.* **2019**, *33*, 36–43. [CrossRef] [CrossRef]
17. Yin, R.; Li, W.; Wang, Z.Q.; Xu, X.X. The application of artificial intelligence technology in UAV. In Proceedings of the 2020 5th International Conference on Information Science, Computer Technology and Transportation (ISCTT), Shenyang, China, 13–15 November 2020; pp. 238–241.
18. Li, P.J.; Mao, P.J.; Geng, Q.; Huang, C.P.; Fang, Q.; Zhang, J.R. Research status and trend of UAV swarm technology. *Aero Weaponry* **2020**, *27*, 25–32. [CrossRef]
19. Ahmed, S.; Chowdhury, M.Z.; Sabuj, S.R.; Alam, M.I.; Jang, Y.M. Energy-efficient UAV relaying robust resource allocation in uncertain adversarial networks. *IEEE Access* **2021**, *9*, 59920–59934. [CrossRef] [CrossRef]
20. Jiang, X.; Wu, Z.; Yin, Z.; Yang, Z. Power and trajectory optimization for UAV-enabled amplify-and-forward relay networks. *IEEE Access* **2018**, *6*, 48688–48696. [CrossRef] [CrossRef]
21. Pathak, P.H.; Feng, X.; Hu, P.; Mohapatra, P. Visible light communication, networking, and sensing: A survey, potential and challenges. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2047–2077. [CrossRef] [CrossRef]
22. Wang, Z.; Liu, R.; Liu, Q.; Thompson, J.S.; Kadoch, M. Energy-efficient data collection and device positioning in UAV-assisted IoT. *IEEE Internet Things J.* **2019**, *7*, 1122–1139. [CrossRef] [CrossRef]
23. Huang, X.; Yang, X.; Chen, Q.; Zhang, J. Task offloading optimization for UAV-assisted fog-enabled Internet of Things networks. *IEEE Internet Things J.* **2021**, *9*, 1082–1094. [CrossRef] [CrossRef]
24. Zhu, S.; Gui, L.; Cheng, N.; Sun, F.; Zhang, Q. Joint design of access point selection and path planning for UAV-assisted cellular networks. *IEEE Internet Things J.* **2019**, *7*, 220–233. [CrossRef] [CrossRef]
25. Yuan, Z.; Jin, J.; Sun, L.; Chin, K.W.; Muntean, G.M. Ultra-reliable IoT communications with UAVs: A swarm use case. *IEEE Commun. Mag.* **2018**, *56*, 90–96. [CrossRef] [CrossRef]
26. Lin, Y.; Wang, T.; Wang, S. UAV-assisted emergency communications: An extended multi-armed bandit perspective. *IEEE Commun. Lett.* **2019**, *23*, 938–941. [CrossRef] [CrossRef]
27. Lyu, J.; Zeng, Y.; Zhang, R. UAV-aided offloading for cellular hotspot. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 3988–4001. [CrossRef] [CrossRef]
28. Özdamar, L.; Demir, O. A hierarchical clustering and routing procedure for large scale disaster relief logistics planning. *Transp. Res. Pt. e-Logist. Transp. Rev.* **2012**, *48*, 591–602. [CrossRef] [CrossRef]
29. Arafat, M.Y.; Moh, S. Localization and clustering based on swarm intelligence in UAV networks for emergency communications. *IEEE Internet Things J.* **2019**, *6*, 8958–8976. [CrossRef] [CrossRef]
30. Masood, A.; Scazzoli, D.; Sharma, N.; Le Moullec, Y.; Ahmad, R.; Reggiani, L.; Magarini, M.; Alam, M.M. Surveying pervasive public safety communication technologies in the context of terrorist attacks. *Phys. Commun.* **2020**, *41*, 101109. [CrossRef] [CrossRef]
31. Yang, J.; Huang, X. A distributed algorithm for UAV cluster task assignment based on sensor network and mobile information. *Appl. Sci.* **2023**, *13*, 3705. [CrossRef] [CrossRef]

32. Minhas, H.I.; Ahmad, R.; Ahmed, W.; Alam, M.M.; Magarani, M. On the impact of clustering for Energy critical Public Safety Networks. In Proceedings of the 2019 International Symposium on Recent Advances in Electrical Engineering (RAEE), Islamabad, Pakistan, 28–29 August 2019; pp. 1–5.
33. Mao, Y.; Zhang, J.; Letaief, K. B. Dynamic computation offloading for mobile-edge computing with energy harvesting devices. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 3590–3605. [CrossRef] [CrossRef]
34. Abuqaddom, I.; Mahafzah, B.; Faris, H. Oriented stochastic loss descent algorithm to train very deep multi-layer neural networks without vanishing gradients. *Knowl.-Based Syst.* **2021**, *230*, 107391. [CrossRef] [CrossRef]
35. Jeong, S.; Simeone, O.; Kang, J. Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning. *IEEE Trans. Veh. Technol.* **2017**, *67*, 2049–2063. [CrossRef] [CrossRef]
36. Du, Y.; Yang, K.; Wang, K.; Zhang, G.; Zhao, Y.; Chen, D. Joint resources and workflow scheduling in UAV-enabled wirelessly-powered MEC for IoT systems. *IEEE Trans. Veh. Technol.* **2019**, *68*, 10187–10200. [CrossRef] [CrossRef]
37. Yu, Z.; Gong, Y.; Gong, S.; Guo, Y. Joint task offloading and resource allocation in UAV-enabled mobile edge computing. *IEEE Internet Things J.* **2020**, *7*, 3147–3159. [CrossRef] [CrossRef]
38. Tu, S.; Waqas, M.; Rehman, S.U.; Mir, T.; Abbas, G.; Abbas, Z.H.; Halim, Z.; Ahmad, I. Reinforcement learning assisted impersonation attack detection in device-to-device communications. *IEEE Trans. Veh. Technol.* **2021**, *70*, 1474–1479. [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.