electronics

Article

The Dynamic Response of Dual Cellular-Connected UAVs for Random Real-Time Communication Requests from Multiple Hotspots: A Deep Reinforcement Learning Approach

Shengzhi Yang *, Jianming Zhou and Xiao Meng

School of Integrated Circuits and Electronics, Beijing Institute of Technology, Beijing 100081, China
* Correspondence: yang_554819835@126.com

Abstract: It is gradually becoming popular to use multiple cellular-connected UAVs as inspectors to fulfill automatic surveillance and monitoring. However, in actual situations, UAVs should respond to several service requests from different hotspots, whilst the requests usually present randomness in the arrival time, data amount, and the concurrency. This paper proposes a dynamic dual-UAV response policy for multi-hotspot services based on single-agent deep Q-learning, where the UAVs controlled by a ground base station can be dispatched automatically to hotspots and then send videos back. First, this issue is formulated as an optimization problem, whose goal is to maximize the number of successfully served requests with the constraints of both the UAV's energy limit and request waiting time. Second, a reward function based on service completion is designed to overcome the potential challenges posed by the delay reward. Finally, a simulation was conducted, comparing the conventional time priority algorithm and distance priority algorithm, respectively, to the proposed algorithm. The results illustrate that the proposed algorithm can achieve one more response than the others under different service densities, with the lowest failure number and appropriate average waiting time. This method can give a technical solution for the joint communication-and-control problem of multiple UAVs within complex situations.

Keywords: multi-UAV; dynamic response; communication service; energy constraints; deep Q-learning; UAV surveillance

check for updates

Citation: Yang, S.; Zhou, J.; Meng, X. The Dynamic Response of Dual Cellular-Connected UAVs for Random Real-Time Communication Requests from Multiple Hotspots: A Deep Reinforcement Learning Approach. *Electronics* **2024**, *13*, 4181. https://doi.org/10.3390/electronics13214181

Academic Editor: Zhengguo Sheng

Received: 5 May 2024
Revised: 2 July 2024
Accepted: 4 July 2024
Published: 25 October 2024

Copyright: © 2024 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

Unmanned Aerial Vehicles (UAVs), also known as drones, are gaining wide popularity and have tremendous potential in many civilian and commercial applications [1,2], including smart cities [1,2], precision agriculture [3], border patrolling [4], and public safety [5]. Meanwhile, providing UAVs with cellular connectivity will widen their usage cases, especially as user equipment (UE) to perform tasks [6,7]. In particular, the development of high-speed wireless communication techniques, such as Sixth Generation Mobile Communication Technology (6G), will enable UAVs to transmit data at higher rates [8] and over longer distances [7] compared with present conventional short-range communication techniques, such as WiFi and Bluetooth. In these actual scenarios, one typical use of UAV-UEs is real-time automatic inspection [6,9,10], where the UAV-UEs are controlled by control centers and dispatched to several specific hotspots and then send video streams recorded by cameras mounted on the UAV-UEs to the base stations (BSs) [11].

However, this mode is confronted by several challenges [12]. First, each UAV can only operate for a limited time due to its battery capacity. This requires energy-efficient path planning and wireless communications [13]. Second, multiple UAVs may be present at multiple hotspots, while the requests from these hotspots are usually reported with uncertainty, comprising a random data amount and a limited waiting time. Under these constraints, the response control and the communication between UAV-UEs and BSs mutually interact. Therefore, such communication and control (C&C) problems under a

dynamic environment should be addressed so as to fulfill the real-time service response of UAV-UEs.

Some existing studies have handled joint C&C problems with different mathematical models and theoretical analyses, including optimization theory [14–18], game theory [19], graph theory [20–22], and queue theory [23]. Such research generally focuses on optimizing the service order, developing a strategy for UAV collaboration in a fixed condition and the scheme of UAV dispatching based on divided sub-regions. However, in most studies, the dynamics of the environment are not learnt and the real-time action of UAVs cannot be realized. Furthermore, most contributions also do not consider the limits of energy, although some authors have considered these energy constraints [21].

One of the effective approaches to handle the joint C&C problem is reinforcement learning (RL) [24], in particular deep reinforcement learning (DRL) based on (partially observed) Markov decision problems ((PO)MDPs) [25]. However, the achievements using RL approaches to solve the joint C&C problems are still not enough. Some researchers [26] have utilized double Q-learning to find the optimal serving order, where a UAV as the BS serves several users in 3D space so that the number of successfully serviced requests could be maximized. However, the energy limitation of the UAV and the dynamics of the environment were not considered, and the real-time motion of the UAV also could not be realized. Another work presented a DRL approach to control a group of UAVs working as BSs to achieve the maximum communication coverage with the minimum total energy cost [27]. However, the limitation of the total energy for each UAV was not considered, and the communications between the UAVs and users also were not modelled. Nonetheless, there have already been some studies proving that using the DRL-based method can allow the UAVs to accomplish navigation tasks in complex resource-limited environments [28,29].

This paper proposes a DRL-based method to tackle the joint C&C problem, considering two cellular-connected UAV-UEs in an autonomous inspection system with the constraints of both UAV energy and service waiting time. As the feedback of each single service is just commonly obtained when the service is completed, a proper reward function was designed to avoid the delay reward issue, and the goal was to maximize the number of successfully served requests. The major contributions are as follows:

(1) Both the limitation of the UAV-UE's energy and the service waiting time are considered with respect to the actual conditions. The UAV-UEs can fly back to the BS for battery charging if the residual energy is low and can continue to respond to service requests after being fully charged. Meanwhile, a service request must become invalid if there is a long wait time (i.e., longer than a given threshold).
(2) By modeling the dual-UAV joint C&C problem of the continuous random services from multiple hotspots, a deep Q-Learning approach with a single agent at the BS is proposed. This method can address both the environmental complexity and infinite time that conventional mathematical methods cannot deal with.
(3) The reward based on each completion of a service request is designed to overcome the delay reward difficulty, which can result in the final real-time decision for the UAVs in dynamic conditions. The performance of the proposed method is compared with the shortest distance priority algorithm and the shortest waiting time algorithm, demonstrating that the proposed algorithm can outperform these two counterparts.

The rest of this paper is organized as follows. In Section 2, the cellular-connected inspection system model with a single BS and multiple UAV-UEs is demonstrated in detail. In Section 3, the deep Q-learning network with the observations, actions, and reward is developed. In Section 4, the training results and advantages are illustrated. Section 5 concludes the paper.

2. System Model and Problem Formulation

2.1. System Model

As illustrated in Figure 1, a cellular-connected inspection system comprising a group of N identical UAV-UEs controlled by a BS is considered. The BS is situated at the centroid

of a circular inspection region with radius R_s, within which M emergency hotspots are uniformly distributed. In the event of an emergency, such as a public security incident or fire alarm, an inspection request with the maximum waiting deadline w_k originating from a hotspot is transmitted to the BS. It is assumed that the number of requests arriving at the BS follows the Poisson distribution at each time slot. Upon receipt of a request, a UAV-UE is dispatched by the BS to the hotspots where the request originates. Aerial on-site real-time videos will be sent from the UAV to the BS as long as the UAV enters the hotspot's perimeter with a radius of r. Furthermore, it is also postulated that the BS maintains visibility over the status of the UAVs and the environment. Initially, all UAVs commence their operations from the BS. During each time slot, each UAV either provides on-site video services, or navigates to the subsequent location at a predefined flight altitude h and a constant velocity v. The overarching goal is to acquire the optimal dynamic response policy for the UAVs, aimed at maximizing the total number of serviced requests.

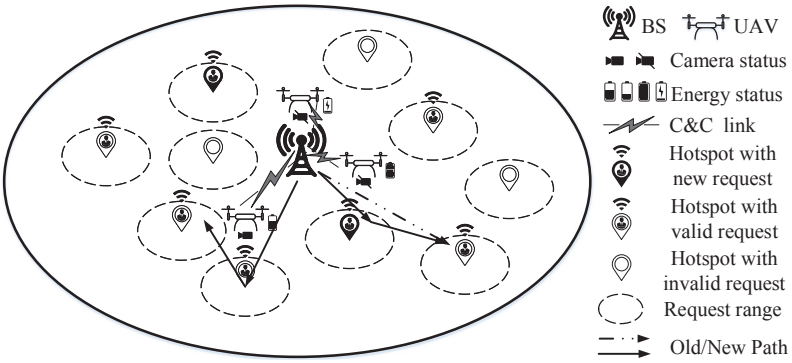


Figure 1. Illustration of system model.

2.1.1. Request Arrival

As shown in Figure 2, the requests emanating from the hotspots arrive at the BS at the end of each time slot. They are mutually independent, and share an equal priority [30]. It is supposed that the number of requests arriving in the inspection region during each time slot follows a Poisson distribution with a density of λ. Once a request is reported from a hotspot, the request remains valid until it is fully serviced by a UAV-UE, or until it expires due to the absence of any UAV service. Generally, these requests may emerge randomly with equal likelihood at any hotspot within each time slot.

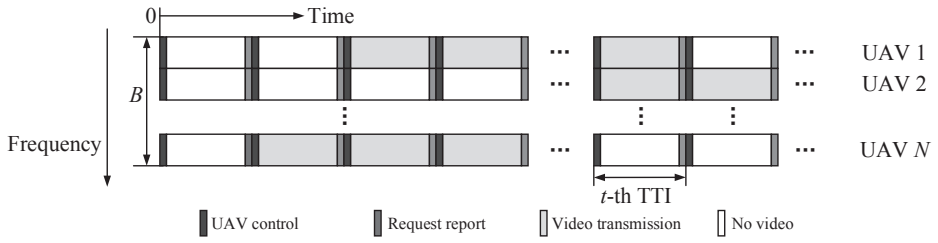


Figure 2. Time slot.

Each request k reported to the BS consists of three elements: the coordinates (x_k, y_k), the current size of video data D_k^t with $0 < D_k^t \leq D_k^0$ bits, where D_k^0 is the initial value following uniform distribution $[0, 2\Psi]$ [31] , and given by the probability density function:

$$p_{D_k^0} = \begin{cases} \frac{1}{2\Psi}, & 0 \leq D_k^0 \leq 2\Psi \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Additionally, there is the current waiting deadline w_k^t with $0 \leq w_k^t \leq w_k$, where w_k is the initially given value. If a UAV provides on-site service to a request before the expiration of

its waiting deadline and successfully delivers all the data, the request is deemed successfully served. Conversely, if the request exceeds its waiting deadline without completion, it is considered overdue and subsequently dropped. Consequently, the corresponding hotspot awaits the next valid request.

2.1.2. The UAV Flight Time and the Corresponding Energy Consumption

The energy consumption model, as outlined in [21,32], is adopted for analysis. For each UAV candidate i, the total energy expenditure for surveillance tasks generally comprises three distinct components: the energy consumed during flight, the energy utilized for video transmission, and the energy expended while hovering.

The minimum power with forward motion for a UAV is given by

$$P_{min} = G(\tilde{v} + v \cdot \sin\phi), \tag{2}$$

where v is the ground speed of the UAV, ϕ is the pitch angle, and $\tilde{v}$ is the induced velocity required for a given thrust G. The required thrust G of the UAV with mass m is

$$G = m \cdot g + f_d, \tag{3}$$

where g is the gravitational constant and f_d is the drag force depending on the air speed, density of air ξ, and the drag coefficient. Given thrust G, the induced velocity is given by the following nonlinear equation:

$$\tilde{v} = \frac{2 \cdot G}{qr_u^2 \cdot \pi\xi\sqrt{(v\cos\phi)^2 + (v\sin\phi + \tilde{v})^2}}, \tag{4}$$

where r_u is the diameter of each rotor and q is the number of UAV rotors. Thus, the flying power can be calculated by

$$P_f = \frac{P_{min}}{\eta_u}, \tag{5}$$

where η_u is the power efficiency of the UAV. The time for UAV i to traverse from the current location (x_i, y_i) to the task k at (x_k, y_k) is

$$t_{f,i}^k = \frac{\sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}}{v}. \tag{6}$$

Therefore, the corresponding energy consumption for UAV i to traverse from the current location (x_i, y_i) to the task k at (x_k, y_k) is

$$\begin{aligned} E_{f,i}^k &= P_f \cdot \frac{\sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}}{v} \\ &= \frac{P_{min} \cdot \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}}{v \cdot \eta_u}. \end{aligned} \tag{7}$$

2.1.3. The On-Site UAV Video Service Time and the Corresponding Energy Consumption

As the communication links between the UAVs and the BS are typically dominated by line-of-sight (LoS) connections, the channel quality is solely dependent on the UAV-BS distance [15]. Furthermore, it is assumed that each UAV employs a frequency division multiple access (FDMA) technique during communication with the terrestrial base station [33,34]. Thus, the channel power gain from the UAV candidate i ($1 \le i \le N$) positioned at (x_i, y_i) to the BS located at $(0, 0)$ adheres to the free-space path loss model, expressed as:

$$g_i = \beta_0 \cdot d_i^{-2} = \frac{\beta_0}{h^2 + x_i^2 + y_i^2} \tag{8}$$

where β_0 denotes the channel power gain at a reference distance $d_0 = 1$ m, and (x_i, y_i) denotes the current coordinate of UAV candidate i. Therefore, if UAV candidate i is scheduled for communication with the BS, the maximum achievable rate R_i can be formulated as:

$$R_i = \frac{B}{N} \log_2 \left(1 + \frac{P_v \cdot g_i}{\sigma^2}\right) = \frac{B}{N} \log_2 \left(1 + \frac{\gamma_0}{h^2 + x_i^2 + y_i^2}\right) \tag{9}$$

where σ^2 is the additive noise power at the receiver, assumed to be identical for all the UAVs, and $\gamma_0 = \frac{P\beta_0}{\sigma^2}$ signifies the reference received signal-to-noise ratio (SNR) at $d_0 = 1$ m. Hence, if UAV candidate i transmits D_k bits of video data to the BS for task k, the on-site service time $t_{v,i}^k$ can be calculated as:

$$t_{v,i}^k = \frac{D_k}{R_i} = \frac{N \cdot D_k}{B \cdot \log_2 \left(1 + \frac{\gamma_0}{h^2 + x_k^2 + y_k^2}\right)}. \tag{10}$$

The transmission energy for task k can be expressed as:

$$E_{v,i}^k = P_v \cdot t_{v,i}^k = \frac{N \cdot P_v D_k}{B \cdot \log_2 \left(1 + \frac{\gamma_0}{h^2 + x_k^2 + y_k^2}\right)}, \tag{11}$$

where P_v denotes the video transmit power, and η_u is the power efficiency of the UAV.

In addition, the hovering energy required by UAV candidate i to complete task k is given by [21,32]:

$$E_{h,i}^k = P_h \cdot t_{v,i}^k = \frac{G^{3/2}}{\sqrt{\frac{1}{2}\pi \cdot q r_u^2 \xi}} \cdot t_{v,i}^k, \tag{12}$$

where P_h represents the hovering power, defined as:

$$P_h = \frac{G^{3/2}}{\sqrt{\frac{1}{2}\pi \cdot q r_u^2 \xi}}. \tag{13}$$

At the beginning of each time slot, a control link is established between the BS and UAV i for the transmission of control signals. However, given the brevity of the transmission duration (typically measured in milliseconds), the energy consumption associated with the transmission of these control signals is considered negligible, and can therefore be disregarded in the overall energy consumption analysis.

2.1.4. Energy Consumption Model and Flight Safety

Therefore, the overall energy consumption for UAV i located at (x_i, y_i) to complete task k at (x_k, y_k) can be expressed as:

$$E_i^k = E_i^{k'} + E_{f,i}^k + E_{h,i}^k + E_{v,i}^k, \tag{14}$$

where k' denotes the task visited immediately prior to k. Assuming an initial energy of $e_i^0 = 0$, Equation (14) enables the recursive calculation of the overall task energy consumption for any given task.

To dispatch UAV candidate i at (x_i, y_i) to task k, the UAV's current energy E_i must be sufficient to fly to the task, complete it, and potentially return to the BS if not allocated to another task afterward. This constitutes the safety requirement for UAV dispatch. Specifically, the current energy E_i of UAV i must satisfy:

$$E_i = E - E_i^k \geq E_{f,i}^k + E_{h,i}^k + E_{v,i}^k + E_{f,k}^0. \tag{15}$$

where $E^0_{f,k}$ represents the flight energy consumption from task k at (x_k, y_k) to the base station at $(0, 0)$, and E denotes the full energy capacity of each UAV.

2.2. Problem Formulation

As outlined in [26], an indicator function for task k is defined as:

$$\delta_k = \mathbb{1}_{\{w_k \geq 0\}}, \tag{16}$$

where $\mathbb{1}_{\{x\}} = 1$ if x is true, and 0 otherwise. $\delta_k = 1$ indicates that the request k is successfully served within its waiting time.

The objective of this research is to determine the optimal paths for the UAVs to maximize the number of successfully served requests, subject to the energy constraints of each UAV. The problem can be formulated as:

$$\begin{aligned} \max \quad & \sum_k \delta_k \\ s.t. \quad & E_i \geq 0, \end{aligned} \tag{17}$$

where $\sum_k \delta_k$ represents the total number of successfully served requests, and E_i is the residual energy as defined in Equation (15).

3. The Solution Approach of the Dual-UAV Dynamic Response Based on Deep Q-Learning

The reinforcement learning approaches are highly effective in tackling the dynamic control problem within complex Partially Observed Markov Decision Problem (POMDP). When considering a Q-agent deployed at the BS to optimize the energy efficiency, the objective is to maximum the number of successfully served requests while minimizing the total energy cost, subject to UAV energy constraints. To achieve this, the Q-agent must explore the environment to progressively select appropriate actions that lead towards the optimization goal. In a standard reinforcement learning framework, at the commencement of the t-th time slot ($t \in \{0, 1, 2, \dots\}$), the agent (the base station) observes current state $S[t]$, executes action $A[t] \in \mathcal{A}(S[t])$ belonging to the set of available actions conditioned on that state, transitions to a new state $S[t+1]$, and receives a reward $R[t]$ as a result of its action.

3.1. State

First, the state $S[t]$ (at time slot t) is defined, which consists of seven parts, given in Table 1. Formally, state $S[t]$ is represented as an open set to accommodate the dynamic nature of the requests. The agent's decisions are based on the current state of both the UAVs and the requests, taking into account factors such as the energy consumption and the waiting time.

Table 1. State parameters and notations.

Role	Parameter	Notation	Instruction
UAV i	Current location	$(x_i[t], y_i[t])$	Initial at $(0, 0)$, $0 \leq i \leq N$
	Current energy	$E_i[t]$	$0 \leq E_i[t] \leq E$
Request k	Location	(x_k, y_k)	Uniform distribution
	Workload	D^t_k	Initial with uniform distribution $[0, 2\Psi]$
	Waiting deadline	w^t_k	Declining with time
	Association mark	b_k	$b_k = \{-1, 0, 1\}$, $b_k = -1$: already served or overdue; $b_k = 0$: unserved; $b_k = 1$: being served.

For each UAV, the information on its coordinates and the current residual energy is available. Meanwhile, for each hotspot, details like the location, current workload, and

waiting deadline are necessary. Additionally, an association mark is employed to categorize the status of each hotspot as either unserved with a pending request, being served by a UAV, or already served or overdue with no pending request. This fundamental information from both the UAVs and the hotspots characterizes the current state of all involved entities. Collectively, these data constitute a vector representing the current state of the environment, which serves as input to the Deep Q-Network (DQN).

3.2. Action

With knowledge of state $S[t]$, the Q-agent selects an action $A[t]$ from the set $\mathcal{A}$. As depicted in Figure 3, each UAV has the option to choose one of nine actions at each time slot: up, upper right, right, lower right, down, lower left, left, upper left, and hover.

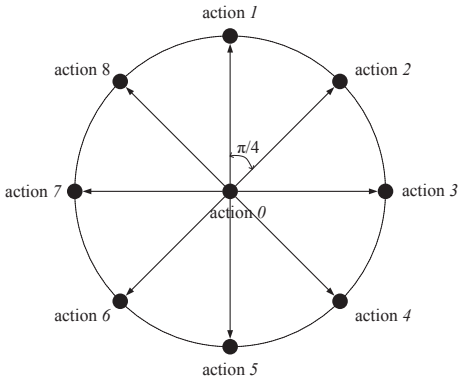


Figure 3. Nine action options for each UAV.

Formally, the action space for each UAV, $A_i[t]$, comprises nine discrete actions. This definition stems from the control policy's need to dictate the UAV's flight trajectory at each time slot. Consequently, the dimensionality of the action set can be determined by $|\mathcal{A}| = |A_i[t]|^i$, for instance, with two UAVs, the dimensionality is $9^2 = 81$.

The ϵ-greedy approach is employed to balance exploitation and exploration within the actor of the agent, where ϵ is a positive real number constrained by $\epsilon \leq 1$. With a probability of ϵ, the algorithm randomly selects an action from the remaining feasible actions, aiming to refine the estimation of non-greedy action values. Conversely, with a probability of $1 - \epsilon$, the algorithm exploits the current understanding of the Q-value network to choose the action that maximizes the expected reward. Given the vast state space in this paper, ϵ is defined as a decreasing function of the time step, as shown in Equation (18).

$$\epsilon = \epsilon_{min} + (\epsilon_{max} - \epsilon_{min}) \exp(-\xi_0 \cdot step), \tag{18}$$

where $\epsilon_{max} = 1$; $\epsilon_{min} = 0.1$; $\xi_0 = 10^{-7}$. This setting offers the advantage of initially allowing the agent to explore the environment more extensively, with a higher probability, to comprehensively grasp the interaction's rules and rewards. As time progresses, the agent can then exploit its acquired knowledge with $\epsilon = \epsilon_{min}$, ensuring the convergence while maintaining the potential for further refinement.

3.3. Reward Based on Service Completion and Energy Consumption

After performing an action $A[t]$, the Q-agent receives a scalar reward $R[t+1]$ and observes a subsequent new state $S[t+1]$. The reward $R[t+1]$ quantifies the extent to which the executed action $A[t]$ contributes to achieving the optimization goal, as defined by the newly observed state $S[t+1]$. For each UAV within each time slot, the reward $R_i[t+1]$ is formulated as a function that encompasses the sum of requests being served and successfully served requests, along with the energy cost incurred during that time slot.

The component of the reward related to the energy consumption of each UAV is specified in Equation (19).

$$R_i^E[t+1] = \begin{cases} -\frac{e^{-1}}{10E_{max}}(E_i[t] - E_i[t+1]), & \text{for working} \\ -1, & \text{for energy outage} \\ +1, & \text{for charging with energy no more than 20\%} \\ \frac{e^{-1}}{2E_{max}}, & \text{for charging with energy more than 20\%} \end{cases} \tag{19}$$

where $E_{max} = E_{f,i}^k + E_{v,i}^k$. A positive reward is awarded if the UAV is charged, whereas a small negative value is assigned for working due to the associated energy cost. The factor e^{-1} ensures appropriate normalization of $R_i^E[t+1]$. However, the most negative reward is reserved for energy outage scenarios, emphasizing the importance of flight safety.

In the context of on-site video services provided by UAV i, a potential delay reward issue arises if feedback is solely provided upon service completion, which poses a challenge. To address this, a reward mechanism considering the residual unserviced data is designed to reflect the service progress at each time slot, as defined in Equation (20).

$$R_i^S[t+1] = \begin{cases} \exp\frac{-D_k^t}{D_k^0}, & \text{for being served} \\ 0, & \text{for others} \end{cases} \tag{20}$$

This reward structure incentivizes UAVs to initiate and complete services, as it grows with the amount of data serviced. The maximum reward is achieved upon full service completion, whereas no reward is given for pending or overdue requests. In actual situations, the UAVs may be engaged at other hotspots when random requests arise, therefore potentially preventing the timely arrival at the corresponding place. This is not a attributable to UAV fault, thus justifying the use of non-negative values.

Specifically, for each UAV, based on Equations (19) and (20), the combined reward of proper operation surpasses that of energy costs, ensuring that the total reward $R_i^E[t+1] + R_i^S[t+1]$ can be positive.

Therefore, the cumulative reward for each time slot is in Equation (21), which represents the summation of individual rewards achieved by each UAV.

$$R[t+1] = \sum_i^N (R_i[t+1]) = \sum_i^N (R_i^E[t+1] + R_i^S[t+1]), \tag{21}$$

where $R[t+1]$ is the sum of the reward of each UAV.

3.4. Training and Updating by Deep Q-Learning

Q-learning is a value-based reinforcement learning approach. To obtain an action $A[t]$, the highest value scalar from the numerical value vector $Q(S[t], a)$ is selected. The objective is to find an optimal Q-value table $Q^*(s, a)$ with the optimal policy π^* that can select actions to dynamically optimize the paths of UAVs.

If an initial Q-value table $Q(s, a)$ in the environment is trained using Q-learning algorithm, $Q(s, a)$ is immediately updated using the observed reward $R[t+1]$ after each action, which follows the Bellman Equation as

$$Q^{new}(S[t], A[t]) = (1-\alpha)Q(S[t], A[t]) + \alpha\left\{R[t+1] + \gamma \max_{a \in \mathcal{A}} Q(S[t+1], a)\right\}. \tag{22}$$

where α is the learning rate considering how fast the algorithm adapts to a new environment, γ is the discount rate that determines how current rewards affect the value function updating, and $\max_{a \in \mathcal{A}} Q(S[t+1], a)$ approximates the value in optimal Q-value table $Q^*(s, a)$ via the up-to-date Q-value table $Q(s, a)$ and the obtained new state $S[t+1]$.

However, in this scenario, the dimensionality of both state space and action space can be very high if the traditional tabular Q-learning is used, causing a very complicated and huge state–action Q-value table, which is very challenging to update. Since the tabular

Q-learning can be difficult with limited time and computational resources, particularly in this problem, a value function approximator rather than a Q-value table is utilized in order to find a sub-optimal approximated policy.

Deep Q-learning can be considered as the "deep" version of the conventional tabular Q-learning, where a deep neural network (DNN) is used to approximate the Q function. In this paper, the state–action value function $Q(s, a)$ is parameterized by using a function $Q(s, a; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the weight vector of the DQN. The conventional DNN with fully connected layers between arbitrary adjacent layers is exploited. The input of the DNN is the state given by $S[t]$, the intermediate activation functions are Rectifier Linear Units (ReLUs), and the units of the output layer are correspondent with all available actions in $\mathcal{A}$. In each time slot, the weight vector is updated using Stochastic Gradient Descent (SGD), given by

$$\boldsymbol{\theta}[t+1] = \boldsymbol{\theta}[t] - \gamma_{\mathrm{ADAM}} \cdot \nabla \mathcal{L}(\boldsymbol{\theta}[t]), \tag{23}$$

where γ_{ADAM} is the Adam learning rate, and $\gamma_{\mathrm{ADAM}} \cdot \nabla \mathcal{L}(\boldsymbol{\theta}[t])$ is the gradient of the loss function $\mathcal{L}(\boldsymbol{\theta}[t])$, which is given by

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{S[i],A[i],R[i+1],S[i+1]}[(Q_{tar} - Q(S[i], a[i]; \boldsymbol{\theta}[t])) \cdot \nabla Q(S[i], A[i]; \boldsymbol{\theta}[t])], \tag{24}$$

where Q_{tar} is the target Q-value that can be estimated by

$$Q_{tar} = R[i+1] + \gamma \max_{a \in \mathcal{A}} Q(S[i+1], a; \bar{\boldsymbol{\theta}}[t]). \tag{25}$$

where $\bar{\boldsymbol{\theta}}[t]$ is weight vector of the target Q-network to be used to estimate the future value of the Q-function in the update rule, which is periodically copied from $\boldsymbol{\theta}[t]$. Note that the samples $(S[i], A[i], R[i+1], S[i+1])$ are randomly selected from the replay memory $i \in \{t - M_r, t - M_r + 1, \ldots, t\}$ with size M_r.

3.5. Baseline Methods

In real applications, the principles of time priority and distance priority are frequently employed to address random response issues of this kind. The time priority principle dictates that UAVs serve requests based on the order of their arrival, disregarding the proximity to hotspots. Meanwhile, the distance priority principle stipulates that UAVs prioritize fulfilling the nearest requests. These two strategies are subsequently compared with the proposed method.

4. Results and Discussions

4.1. Simulation Results

The parameters used in the simulation are listed in Table 2. The simulation platform is a computing server, equipped with an i9-10900x CPU chip, two GPUs of RTX 2080ti, and 64 GB of memory, marked by Linkzol LZ540-GR. In addition, Python 3.6, TensorFlow 1.13, and Keras 2.2 are utilized.

The Deep Q-Learning Network utilized in this paper is a fully connected neural network architecture, featuring 56 inputs, 81 outputs, and three hidden layers with 256, 512, and 512 nodes, respectively. The Rectified Linear Unit (ReLU) serves as the activation function. Furthermore, each episode contains 100 time slots, and the batch size is set to 256. The network is updated every 1000 iterations, with a learning rate of 10^{-5}. The size of replay memory is 50,000.

Figure 4 presents the reward outcomes achieved under varying request densities, specifically $\lambda = 0.1$, 0.15, 0.2, and 0.25. It demonstrates the convergence of the proposed method. Notably, for any converged model, the value of λ can be seamlessly adjusted to simulate different scenarios or performance outcomes. For example, the learning mode of a converged model trained with $\lambda = 0.2$ can be transitioned to an application mode, and if its λ is changed to 0.1, the results equal the model with $\lambda = 0.1$. Hence, the proposed method shows universality in terms of different request densities.

Table 2. Simulation parameters.

Parameter	Value	Unit
Video transmit power P_v	30	dBm
UAV number N	2	-
Total bandwidth B	3	MHz
Noise power σ^2	-174	dBm/Hz
Reference channel gain β_0	10^{-6}	-
Flight height h	50	m
Power efficiency η_u	0.7	-
Flight speed v	10	m/s
UAV mass m	2	kg
Drag Force f_d	9.6998	N
Rotor number q	4	-
Rotor diameter r_u	0.254	m
Density of air ξ	1.225	kg/m^3
Gravitational acceleration g	9.8	m/s^2
Pitch angle ϕ	$\pi/18$	rad
Induced velocity $\tilde{v}$	4.9556	m/s
Request density in each time slot λ	0.05, 0.1, 0.15, 0.2, 0.25	$/m^2$
Max waiting time w_k	30	time slots
Workload parameter Ψ	500	Mbits
UAV moving step	50	m
Potential hotspots M	10	-
Radius of the target region R_s	250	m
Full energy capacity E	100,000	Joule
Radius of a hotspot r	50	m
Discount rate γ	0.8	-

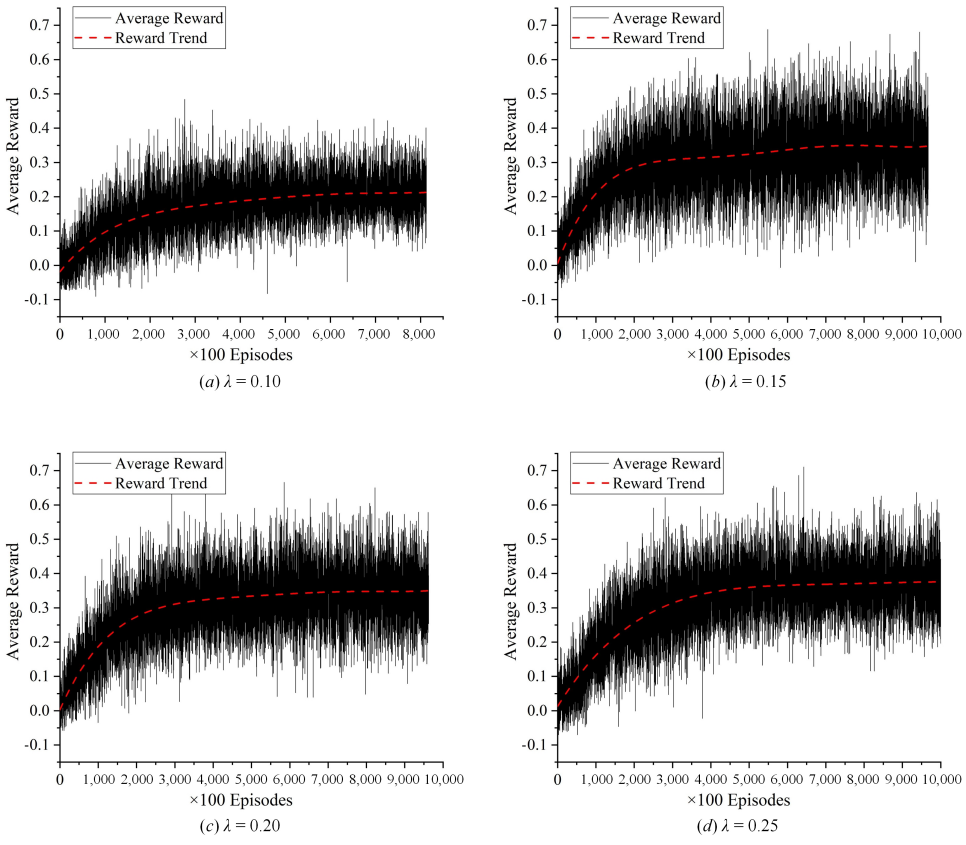


Figure 4. The rewards under different request densities λ.

4.2. Algorithm Comparison Results

Figures 5–7 illustrate the average quantities of successful service, failure, and waiting time, respectively. Based on the simulation parameters, varying request densities ($0.05 \leq \lambda \leq 0.25$) correspond to an average number of requests ranging from 5 to 25.

Figure 5 illustrates the average number of successful services across different request densities ($0.05 \leq \lambda \leq 0.25$). The DQL-based method outperforms the other two methods, demonstrating the highest successful number. Conversely, the distance-priority method exhibits the poorest performance. Figure 6 presents the average numbers of failures under different request densities, where the proposed method generally records the lowest number. Even at the highest request density of 0.25, the number of failures remains below two. In contrast, the failure counts of the other two baseline methods increase markedly. Figure 7 showcases the average waiting time, with the proposed method generally falling within a moderate range. As the density increases, the average waiting time rises smoothly from four to eight time units.

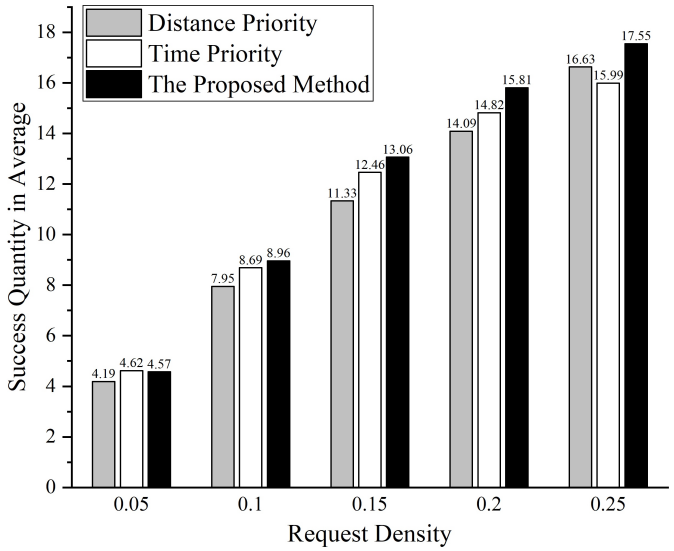


Figure 5. The average success number under different request densities λ.

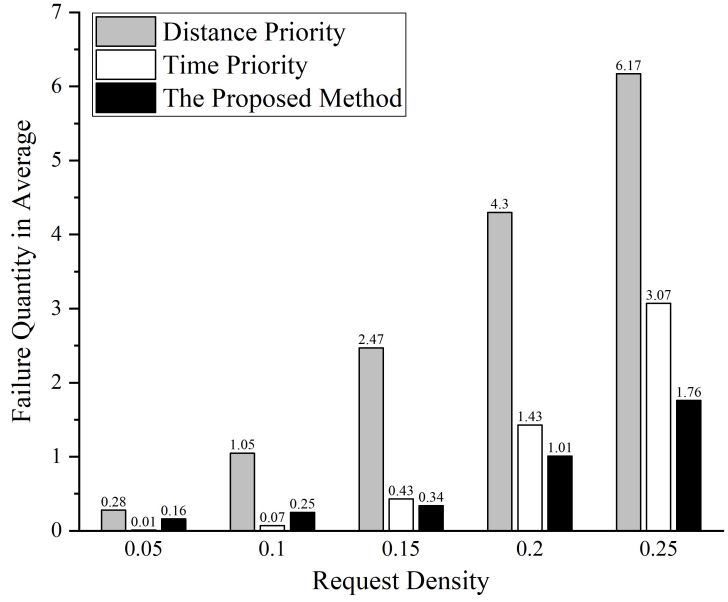


Figure 6. The average failure under different request densities λ.

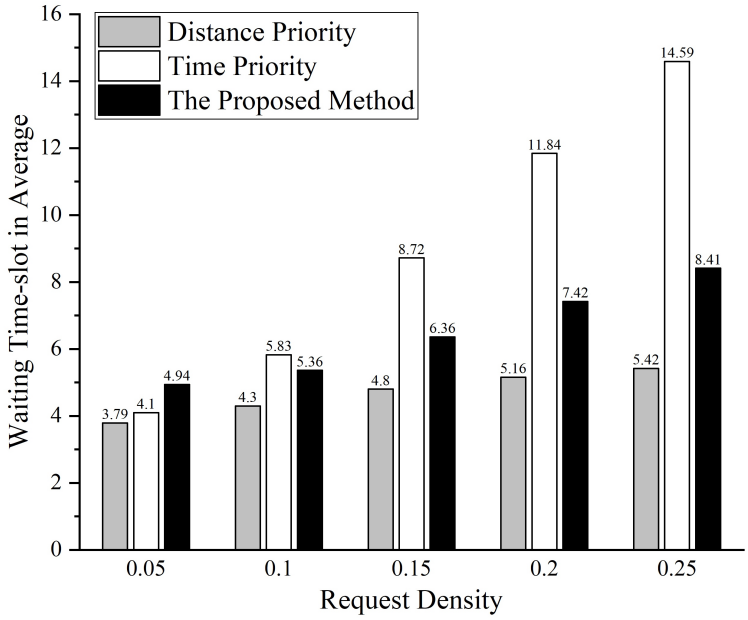


Figure 7. The average waiting time slot under different request densities λ.

Moreover, an analysis of Figures 5–7 reveals the following insights:

(1) The algorithm proposed in this paper demonstrates remarkable performance. When λ is minimal ($\lambda = 0.05$), the time-priority algorithm exhibits a slight advantage in terms of the highest number of successful services and the lowest number of failures. However, this advantage is not substantial, as the average numbers of successful and failed services across all three strategies are comparable. This is attributed to the low concurrency of random tasks, enabling all strategies to adequately meet service demand with probably short-term waiting or even fast response. As λ increases ($0.1 \leq \lambda \leq 0.2$), while all algorithms show an increasing trend in the number of successful services, the advantage of the proposed algorithm becomes more distinct. It not only leads the largest number of successful services and the smallest number of failures, but also has an average waiting time comparable to the best-performing algorithm. Specifically, the average number of successful services is approximately one more than the other two algorithms. This is because, at higher request densities, the time-priority algorithm serves requests sequentially, leading to longer waiting times for queued requests. The distance-priority algorithm, while reducing the average waiting time, may prioritize nearby requests at the expense of father ones, resulting in a "short-sighted" behavior that increases the number of unserved requests and failures.

(2) The number of UAV-UEs constitutes a pivotal factor constraining the overall service quality. As λ escalates to a significantly high range ($0.2 \leq \lambda \leq 0.25$), it becomes evident that the average count of successful services for any given strategy augments slowly, whereas the mean number of failed services escalates markedly. This phenomenon underscores the existence of performance limitations across all algorithms, which is obviously due to the limited number of UAVs. To attain a greater number of successful services, particularly under conditions of higher request densities, increasing the number of UAVs emerges as an indispensable measure, rather than relying solely on intelligent response algorithms to improve overall service quality.

4.3. Discussions

The utilization of cellular-connected UAVs to deal with dynamic and random responses for multiple hotspots is a heated discussion issue. However, current studies have not considered the environmental complexity, including UAV energy constraints, request randomness, and service-limited waiting time. The proposed DQL-based method in this paper surpasses conventional time-priority and distance-priority algorithms by achieving

not only more successful service quantities (one more on average) but also the lowest failure rate. Under different request densities ($0.05 \leq \lambda \leq 0.25$), all trained models demonstrate convergence. Furthermore, the request density of a converged model can be adjusted to the others under the execution mode, performing the same result. To tackle such C&C issues of multi-UAV systems, this paper presents two key contributions: modelling the optimization problem and designing a proper reward. The simulation results validate the effectiveness of these contributions.

The scenario abstracted in this paper is from real-world applications, especially in multi-UAV data acquisition for smart life and military solutions [1,3]. For instance, a study aimed at maximizing the achievable rate in multi-UAV-assisted industrial Internet of Things for device-to-device communications overlooked the dynamics and the energy constraints [35]. In contrast, the constraints considered in this paper accord with the real conditions, showcasing broader practical applicability.

Although such constraints are considered in this paper, the different priority of requests with multi-agent learning is not modelled, as the mode of one controller for two UAVs is often used and communication services are usually for video and call demand [36]. Therefore, this paper models common scenarios. If the problem combining request priority with multiple agents is researched, specific analysis under customized demand should be conducted.

Future works could concentrate on the verification of the algorithm by using appropriate hardware and UAVs, as well as the combination of the different request priorities with various arrival densities and frequencies under real-time multi-agent services.

5. Conclusions

Aiming at dual-UAV dynamic response to random requests from multiple hotspots, this paper introduces a DQL-based method to address the difficulties from the randomness of requests and data amount, and the limitation of UAV energy and request waiting time. Additionally, a reward mechanism according to the service completion principle is designed to tackle the potential delay reward. Simulations are conducted to prove the convergence and the effectiveness of the proposed method. The key conclusions are:

(1) Compared with the conventional time-priority and distance-priority methods, the proposed method shows one more average number of successful services and the failure is the lowest, especially under the densities from 0.1 to 0.25. Notably, even under high request densities, the average failure remains less than two, with an average request arrival of 25. The average waiting time of the proposed method is at a moderate level, which can be seen as the combination of advantages of the two baseline methods. The results indicate the effect of the proposed method.

(2) The number of UAVs emerges as a pivotal factor influencing overall service quality. When the request density exceeds 0.2, the performances of all the methods involved in this paper exhibit a similar performance trend, characterized by a deceleration in the growth of successful service counts and an acceleration in failed service counts. It is suggested that deploying additional UAVs to the region could be a viable strategy to enhance the system performance.

In conclusion, this study provides a technical solution for the dual-UAV C&C issue under complex environmental conditions.

Author Contributions: S.Y.: Conceptualization, Methodology, Software, Validation, Writing; J.Z.: Project Administration, Writing—review and editing; X.M.: Methodology, Investigation, Writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: The work station used in this paper is funded by the first author. This research received no external funding.

Data Availability Statement: Data will be made available by the corresponding authors upon request.

Acknowledgments: Thanks for the help from Yansha Deng (King's College London) and Xuetian Wang (Beijing Institute of Technology).

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

References

1. Mozaffari, M.; Saad, W.; Bennis, M.; Nam, Y.; Debbah, M. A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2334–2360. [CrossRef]
2. Qi, F.; Zhu, X.; Mang, G.; Kadoch, M.; Li, W. UAV Network and IoT in the Sky for Future Smart Cities. *IEEE Netw.* **2019**, *33*, 96–101. [CrossRef]
3. Wolfert, S.; Ge, L.; Verdouw, C.; Bogaardt, M.J. Big Data in Smart Farming—A review. *Agric. Syst.* **2017**, *153*, 69–80. [CrossRef]
4. Gupta, L.; Jain, R.; Vaszkun, G. Survey of Important Issues in UAV Communication Networks. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1123–1152. [CrossRef]
5. Motlagh, N.H.; Bagaa, M.; Taleb, T. UAV-Based IoT Platform: A Crowd Surveillance Use Case. *IEEE Commun. Mag.* **2017**, *55*, 128–134. [CrossRef]
6. Mozaffari, M.; Taleb Zadeh Kasgari, A.; Saad, W.; Bennis, M.; Debbah, M. Beyond 5G With UAVs: Foundations of a 3D Wireless Cellular Network. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 357–372. [CrossRef]
7. Challita, U.; Saad, W.; Bettstetter, C. Interference Management for Cellular-Connected UAVs: A Deep Reinforcement Learning Approach. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 2125–2140. [CrossRef]
8. Gapeyenko, M.; Petrov, V.; Moltchanov, D.; Andreev, S.; Himayat, N.; Koucheryavy, Y. Flexible and Reliable UAV-Assisted Backhaul Operation in 5G mmWave Cellular Networks. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 2486–2496. [CrossRef]
9. Kumbhar, A.; Koohifar, F.; Güvenç, I.; Mueller, B. A Survey on Legacy and Emerging Technologies for Public Safety Communications. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 97–124. [CrossRef]
10. Huang, H.; Savkin, A.V. Aerial Surveillance in Cities: When UAVs Take Public Transportation Vehicles. *IEEE Trans. Autom. Sci. Eng.* **2023**, *20*, 1069–1080. [CrossRef]
11. Wei, Z.; Zhu, M.; Zhang, N.; Wang, L.; Zou, Y.; Meng, Z.; Wu, H.; Feng, Z. UAV-Assisted Data Collection for Internet of Things: A Survey. *IEEE Internet Things J.* **2022**, *9*, 15460–15483. [CrossRef]
12. Xu, H.; Wang, L.; Han, W.; Yang, Y.; Li, J.; Lu, Y.; Li, J. A Survey on UAV Applications in Smart City Management: Challenges, Advances, and Opportunities. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2023**, *16*, 8982–9010. [CrossRef]
13. Zeng, Y.; Zhang, R.; Lim, T.J. Wireless communications with unmanned aerial vehicles: Opportunities and challenges. *IEEE Commun. Mag.* **2016**, *54*, 36–42. [CrossRef]
14. Zeng, Y.; Zhang, R. Energy-Efficient UAV Communication with Trajectory Optimization. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 3747–3760. [CrossRef]
15. Wu, Q.; Zeng, Y.; Zhang, R. Joint Trajectory and Communication Design for Multi-UAV Enabled Wireless Networks. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 2109–2121. [CrossRef]
16. Zeng, Y.; Xu, X.; Zhang, R. Trajectory Design for Completion Time Minimization in UAV-Enabled Multicasting. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 2233–2246. [CrossRef]
17. Zhang, S.; Zhang, H.; Di, B.; Song, L. Joint Trajectory and Power Optimization for UAV Sensing Over Cellular Networks. *IEEE Commun. Lett.* **2018**, *22*, 2382–2385. [CrossRef]
18. Mozaffari, M.; Saad, W.; Bennis, M.; Debbah, M. Mobile Internet of Things: Can UAVs Provide an Energy-Efficient Mobile Architecture? In Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, USA, 4–8 December 2016; pp. 1–6. [CrossRef]
19. Koulali, S.; Sabir, E.; Taleb, T.; Azizi, M. A green strategic activity scheduling for UAV networks: A sub-modular game perspective. *IEEE Commun. Mag.* **2016**, *54*, 58–64. [CrossRef]
20. Ramaithitima, R.; Whitzer, M.; Bhattacharya, S.; Kumar, V. Automated Creation of Topological Maps in Unknown Environments Using a Swarm of Resource-Constrained Robots. *IEEE Robot. Autom. Lett.* **2016**, *1*, 746–753. [CrossRef]
21. Monwar, M.; Omid Semiari, W.S. Optimized Path Planning for Inspection by Unmanned Aerial Vehicles Swarm with Energy Constraints. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018.
22. Zhang, S.; Zeng, Y.; Zhang, R. Cellular-Enabled UAV Communication: A Connectivity-Constrained Trajectory Optimization Perspective. *IEEE Trans. Commun.* **2019**, *67*, 2580–2604. [CrossRef]
23. Lin, C.; Wang, Z.; Deng, J.; Wang, L.; Ren, J.; Wu, G. mTS: Temporal-and Spatial-Collaborative Charging for Wireless Rechargeable Sensor Networks with Multiple Vehicles. In Proceedings of the IEEE INFOCOM 2018—IEEE Conference on Computer Communications, Honolulu, HI, USA, 16–19 April 2018; pp. 99–107. [CrossRef]
24. Kurunathan, H.; Huang, H.; Li, K.; Ni, W.; Hossain, E. Machine Learning-Aided Operations and Communications of Unmanned Aerial Vehicles: A Contemporary Survey. *IEEE Commun. Surv. Tutor.* **2024**, *26*, 496–533. [CrossRef]
25. Wang, C.; Wang, J.; Shen, Y.; Zhang, X. Autonomous Navigation of UAVs in Large-Scale Complex Environments: A Deep Reinforcement Learning Approach. *IEEE Trans. Veh. Technol.* **2019**, *68*, 2124–2136. [CrossRef]

26. Liu, X.; Chen, M.; Yin, C. Optimized Trajectory Design in UAV Based Cellular Networks for 3D Users: A Double Q-Learning Approach. *arXiv* **2019**, arXiv:1902.06610. [CrossRef]
27. Liu, C.H.; Chen, Z.; Tang, J.; Xu, J.; Piao, C. Energy-Efficient UAV Control for Effective and Fair Communication Coverage: A Deep Reinforcement Learning Approach. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 2059–2070. [CrossRef]
28. Sha, P.; Wang, Q. Autonomous Navigation of UAVs in Resource Limited Environment Using Deep Reinforcement Learning. In Proceedings of the 2022 37th Youth Academic Annual Conference of Chinese Association of Automation (YAC), Beijing, China, 19–20 November 2022; pp. 36–41. [CrossRef]
29. Cai, Y.; Zhang, E.; Qi, Y.; Lu, L. A Review of Research on the Application of Deep Reinforcement Learning in Unmanned Aerial Vehicle Resource Allocation and Trajectory Planning. In Proceedings of the 2022 4th International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), Shanghai, China, 28–30 October 2022; pp. 238–241. [CrossRef]
30. Osborn, D.R.; Tseloni, A. The Distribution of Household Property Crimes. *J. Quant. Criminol.* **1998**, *14*, 307–330. [CrossRef]
31. Renna, F.; Doyle, J.; Giotsas, V.; Andreopoulos, Y. Media Query Processing for the Internet-of-Things: Coupling of Device Energy Consumption and Cloud Infrastructure Billing. *IEEE Trans. Multimed.* **2016**, *18*, 2537–2552. [CrossRef]
32. Stolaroff, J.K.; Samaras, C.; O'Neill, E.R.; Lubers, A.; Mitchell, A.S.; Ceperley, D. Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery. *Nat. Commun.* **2018**, *9*, 409. [CrossRef]
33. Mozaffari, M.; Saad, W.; Bennis, M.; Debbah, M. Wireless Communication Using Unmanned Aerial Vehicles (UAVs): Optimal Transport Theory for Hover Time Optimization. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 8052–8066. [CrossRef]
34. Lysenko, O.; Valuiskyi, S.; Kirchu, P.; Romaniuk, A. Optimal control of telecommunication aeroplatform in the area of emergency. *Inf. Telecommun. Sci.* **2013**. [CrossRef]
35. Tuong, V.D.; Noh, W.; Cho, S. Sparse CNN and Deep Reinforcement Learning-Based D2D Scheduling in UAV-Assisted Industrial IoT Networks. *IEEE Trans. Ind. Inform.* **2024**, *20*, 213–223. [CrossRef]
36. 3GPP. Study on New Radio (NR) to Support Non Terrestrial Networks V15.4.0 (Release 15). 2020. Available online: https://portal.3gpp.org/Specifications.aspx?q=1&releases=190 (accessed on 4 May 2024).

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.