DOI:
10.1109/TVT.2022.3152146

Document Version
Peer reviewed version

QoE Optimization for Live Video Streaming in UAV-to-UAV Communications via Deep Reinforcement Learning

Liyana Adilla binti Burhanuddin, *Student, IEEE,* Xiaonan Liu, *Student, IEEE,* Yansha Deng, *Member, IEEE,* Ursula Challita, *Member, IEEE,* and András Zahemszky, *Member, IEEE*

***Abstract*—A challenge for rescue teams when fighting against wildfire in remote areas is the lack of information, such as the size and images of fire areas. As such, live streaming from Unmanned Aerial Vehicles (UAVs), capturing videos of dynamic fire areas, is crucial for firefighter commanders in any location to monitor the fire situation with quick response. The 5G network is a promising wireless technology to support such scenarios. In this paper, we consider a UAV-to-UAV (U2U) communication scenario, where a UAV at a high altitude acts as a mobile base station (UAV-BS) to stream videos from other flying UAV-users (UAV-UEs) through the uplink. Due to the mobility of the UAV-BS and UAV-UEs, it is important to determine the optimal movements and transmission powers for UAV-BSs and UAV-UEs in real-time, so as to maximize the data rate of video transmission with smoothness and low latency, while mitigating the interference according to the dynamics in fire areas and wireless channel conditions. In this paper, we co-design the video resolution, the movement, and the power control of UAV-BS and UAV-UEs to maximize the Quality of Experience (QoE) of real-time video streaming. We applied the Deep Q-Network (DQN) and Actor-Critic (AC) to maximize the QoE of video transmission from all UAV-UEs to a single UAV-BS to learn the dynamic fire areas and communication environment. Simulation results show the effectiveness of our proposed algorithm in terms of the QoE, delay and video smoothness compared to the Greedy algorithm.**

***Index Terms*—Quality of Experience (QoE), UAV-to-UAV (U2U) communication, video streaming, Deep Q Network (DQN), Actor Critic (AC).**

I. Introduction

Over the years, an increasing number of wildfires has inevitably created new challenges for firefighters to control and monitor fire in remote areas [1], [2]. Without new technology to monitor the incident area from the control station, the current practice of the fire station control lacks the technology to remotely visualize the dynamic fire situation in real-time for immediate action [2]. Therefore, monitoring multiple fire-fighting areas in different locations with dynamic fire heights and areas is vital. Unmanned Aerial Vehicles (UAVs) with low cost, high mobility, and the capability to capture high-definition video, can be a good solution to oversee the fire situation, and facilitate the fire commander's response for the choice of number of firefighters and firefighting machines. The use of UAVs provides the fire commander with sufficient information of the overall situation of the fire and danger, such as explosions or human requiring rescue. More importantly, it helps to reduce any imminent dangers and obstacles to firefighters. Existing wireless technologies, such as WiFi, Bluetooth, and radio wave, can only support UAVs' communication within a short transmission range, which are inefficient for multi-UAV collaboration with limited multi-UAV control [3]. In particular, by using the existing advanced cellular technologies, cellular-connected UAV has the great potential to achieve advantage of remote UAV operation with unlimited range [4]–[7]. Authors in [4] used a commercial LTE network to study the data collected during drone flights in the applicability of terrestrial networks for connected drones. Meanwhile, with advantage cellular networks also can support the real-time video streaming from UAV users (UAV-UEs) with beyond the line of sight control, low latency, real-time communication, and ubiquitous coverage from base stations (BSs) with wireless backhaul to the core networks. Despite the growing interests in cellular-connected UAVs, there are still many unsolved challenges for commercial deployment [3] [8]. Therefore, to achieve the high effectiveness among terrestrial-UAV, powerful sensing capability should be considered i.e., UAV massive MIMO [7], to complementary network-based and UAV-based solutions. Also, multi-tier UAVs could be used to assist in wireless communication and improve their efficiency [9]. To ensure the effectiveness of multi-tier UAVs, the optimal intensity, altitude of drones, and the specific network load conditions should be considered to ensure the deployment of multi-tier UAVs [10].

Therefore, to support an inadequate network, UAV has been initially proposed as a relay to help other UAVs transmit to a nearby terrestrial BS with low signal to noise ratio (SNR) [8]. In addition, the dedicated UAV could employ as aerial BS (UAV-BS), access points (APs), or relays, to assist the wireless communications of ground nodes, which we refer to as UAV-assisted wireless communication [5]. When the distance of UAV-to-UAV (U2U) communication decreases, the SNR of the transmission among the UAVs increases resulting in a better transmission performance [11].

L. A. B. Burhanuddin, X. Liu and Y. Deng are with Department of Engineering, King's College London, London, UK (e-mail: liyana.burhanuddin@kcl.ac.uk, xiaonan.liu@kcl.ac.uk, yansha.deng@kcl.ac.uk). L. A. B. Burhanuddin is also with School of Electrical Computer Engineering, Xiamen University Malaysia, Sepang, Malaysia.

U. Challita and A. Zahemszky are with Ericsson AB, Stockholm, Sweden (email: ursula.challita@ericsson.com, andras.zahemszky@ericsson.com).

This work was supported in part by Engineering and Physical Sciences Research Council (EPSRC), U.K., under Grant EP/W004348/1.
Corresponding author: Yansha Deng

The use of UAVs in disaster scenarios has been investigated in literature [12]–[19]. In [12], the UAV was introduced as an emergency BS to serve the affected ground users with limited coverage. In [13], multiple mini-UAVs were used to form flying ad-hoc network (FANET) to explore large and disjoint terrain in disaster areas while adapting their transmission power to optimize the energy usage. In [17], through optimizing the trajectory, the transmit power of the UAV and the mobile device, the outage probability of the UAV relay network in the disaster area was minimized. In [18], a UAV platform was developed to compensate the communication loss during a natural disaster, with the aim to obtain the optimal flight paths in high-rise urban and urban microcell environment. In [19], UAV-assisted networks was studied in disaster area, and the proposed power control optimization problem was solved via relaxing the non-convex problem. Nevertheless, no studies have focused on the real-time video streaming between UAV-UEs and UAV-BS.

Real-time video streaming has higher requirements in terms of data rate, latency, and smoothness compared to other data types. In a firefighting scenario, the network channel capacity fluctuates dramatically with the dynamic environment alongside the UAVs' movement, which can cause poor network performance and undesirable delays. This in turn makes it harder to learn the pattern variance of the channel capacity, thus resulting in failure to transmit with high capacity and high video quality. To capture the practical performance from testbed, authors in [20] used single UAV to conduct indoor experimental to measure the video streaming performance from one LTE base station. Therefore, to overcome the limitation of fluctuate environment, the authors in [21] applied the Additive Variation Bitrate (ABR) method with Deep Reinforcement Learning (DRL) to select proper video resolution based on previous communication rate and throughput. However, [21] only focused on a single video source ABR, which was guided by RL to make decisions based on the network observations and video playback states for selecting the optimal video resolution. While managing large firefighting areas, multiple UAVs are required, authors in [22] used multiple UAVs to stream a video and optimize the QoE to solved resource allocation using game theory technique, however, the QoE utility measurement used error statistic of PSNR and mean of sum (MOS) scale, which could lead to biased measurement. Authors in [23] used UAV relay network and considered two factors, the bit rate of the video and the freezing time to maintain the quality. However, the dynamic channel and different requests are not considered. Therefore, we improve the quality measurement by introducing three video quality factors, i.e., video resolution measurement, video smoothness, and latency penalty. Also, our long-term optimization problem is solved by DRL algorithms. The DRL algorithms can be adapt to the fluctuated channel quality in networks and ensure the long-term QoE. However, in large search and rescue firefighting scenario, a nonordinary optical camera [24] should be considered to ensure the reception of a high quality video. To deal with a more complex environment and practical scenarios, such as search and rescue firefighting scenarios, the DRL algorithm is a promising tool for solving the problem of jointly optimizing the UAVs location while maximizing the data rate [25], [26]. DRL scheme has been applied to improve the performance of Vehicular Ad doc networks [27] and interference alignment problems in wireless networks [28].

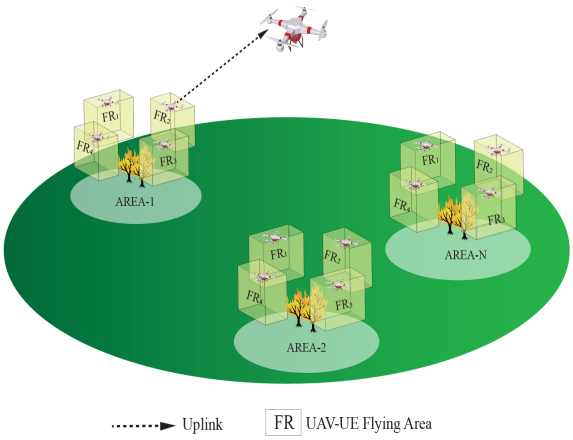

Fig. 1. Illustration of System Model

In this paper, based on [29] and [30], we consider a cellular-connected UAV-BS streaming the real-time video captured by UAV-UEs from the firefighting area for fire monitoring. The contributions of this paper are summarized as follows:

- We develop a framework for a dynamic UAV-to-UAV (U2U) communication model with a moving UAV-BS in multiple firefighting areas to capture a live-streaming panoramic view. We model the dynamic fire arrival with different heights in every fire area and UAVs' request arrival as Poisson process in each time slot, and design the UAV-UEs location spaces to capture a full panoramic view with multiple UAVs.
- To guarantee the smoothness and latency of the live video streaming among UAV-BS and UAV-UEs in this U2U network, we formulate a long-term Quality of Experience (QoE) maximization problem via optimizing the UAVs' positions, video resolution, and transmit power over each time slot.
- To solve the above problem, we propose a Deep Reinforcement Learning (DRL) approach based on the Actor-Critic (AC) and the Deep Q Network (DQN). Our results shown that our proposed AC and DQN approaches outperform the Greedy algorithm in terms of QoE.

The rest of this paper is organized as follows. The system model and problem formulation are given in Section II. The optimization problem via reinforcement learning is presented in Section III. Simulation results and conclusion are presented in Sections IV and V, respectively.

II. System Model and Problem Formulation

As illustrated in Fig. 1, we consider a single UAV-BS to provide a network coverage for multiple UAV-UEs to satisfy the network rate requirement of each UAV-UE to stream high quality video of multiple firefighting areas. The UAV-BS is located at the center of the environment, such as forest area, with the maximum coverage radius $r_{\max}$. The UAV-BS is

connected through wireless network to the fixed or mobile control station. We assume that the arriving distribution of the fire video streaming request is the same as that of the fire arrival distribution [31], which follows Poisson process distribution with density λ_a. The reason for this model is that the authors in [31] used real data of 30 years of annual areas burned data to model the distributions, where the distributions of size and arrival time in real data are proved to follow the Poisson distribution. The UAV-BS receives a request when a fire event occurs, and the kth UAV-UE automatically flies to the center of kth flying region FR$_k$ to serve the ith fire area $A_i(x_i, y_i)$.

We consider a video streaming task that lasts for T time slots with an equal duration t. The selection of the optimal location to stream the video plays an important role in ensuring the UAV-UEs capture the full firefighting area of A_i. Therefore, the kth UAV-UE needs to find the optimal position $U(x_k^*, y_k^*, h_k^*)$ to transmit the video to the UAV-BS. The size of the kth fire region FR$_k$ for the kth UAV-UE depends on the number of UAV-UEs that perform the video streaming for the ith fire area A_i. To make sure that all UAV-UEs can jointly capture the panoramic video of A_i, K UAV-UEs are distributed evenly around A_i, as shown in Fig. 1. Meanwhile, the UAV-BS also searches for the optimal location $P(x_{BS}^*, y_{BS}^*, h_{BS}^*)$ to satisfy the minimum data rate requirement for all UAV-UEs. In addition, the safety region of the A_i is considered to guarantee FR$_k$ and A_i, and A_i and A_{i+1} are not overlapping to guarantee that the UAV-BS and UAV-UEs are safe from fire.

A. Request Arrival

The request contains the ith area A_i with its centre at (x_i, y_i) with radius r_i. We assume that K UAV-UEs serve each fire area and stream real-time videos simultaneously. We assume that the height of the fire h_i follows Log-normal distribution [32], thus, the minimum flying height of all UAVs is $h_{\min}$, which satisfies $h_{\min} = \max(h_i)$. All UAV-UEs in A_i will be operated at the same altitude. The environment is divided into W square grids, thus, the length, width and height of each grid are $\frac{X}{\sqrt[3]{W}}$, $\frac{Y}{\sqrt[3]{W}}$, $\frac{Z}{\sqrt[3]{W}}$, respectively. At the tth time slot, the flying position $\vec{U}(x_{i,k}, y_{i,k}, h_{i,k})$ of the kth UAV-UE can be calculated as

$$\vec{U}^{t+1}(x_{i,k}, y_{i,k}, h_{i,k}) = \vec{U}^{t}(x_{i,k}, y_{i,k}, h_{i,k}) + \vec{a}^{t}(x, y, z), \quad (1)$$

with

$$x_i - a \leq x_{i,k} \leq x_i + a, \quad (2)$$
$$y_i - a \leq y_{i,k} \leq y_i + b, \quad (3)$$
$$h_{\min} \leq h_{i,k} \leq h_{\max}, \quad (4)$$

$$U_{(i,k=1)}^{t} = \{(x_1, y_1, h_1)| \\ x_i - a \leq x_{i,1} \leq x_i + a, \\ y_i + a \leq y_{i,1} \leq y_i + b, \\ h_i \leq h_1 \leq h_{max}\}, \quad (5a)$$

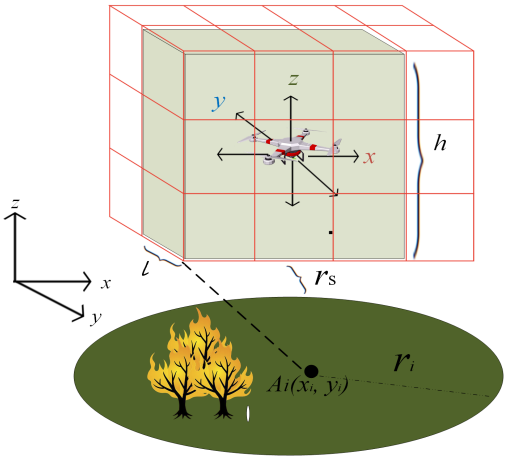

Fig. 2. Flying boundry of the kth UAV-UE.

$$U_{(i,k=2)}^{t} = \{(x_2, y_2, h_2)| \\ x_i - b \leq x_{i,2} \leq x_i - a, \\ y_i - a \leq y_{i,2} \leq y_i + a, \\ h_i \leq h_2 \leq h_{max}\}, \quad (5b)$$

$$U_{(i,k=3)}^{t} = \{(x_3, y_3, h_3)| \\ x_i - a \leq x_{i,3} \leq x_i + a, \\ y_i - b \leq y_{i,3} \leq y_i - a, \\ h_i \leq h_3 \leq h_{max}\}, \quad (5c)$$

$$U_{(i,k=4)}^{t} = \{(x_4, y_4, h_4)| \\ x_i + a \leq x_{i,4} \leq x_i + b, \\ y_i - a \leq y_{i,4} \leq y_i + a, \\ h_i \leq h_4 \leq h_{max}\}. \quad (5d)$$

where $\vec{a}^t(x, y, z)$ is the action vector to determine the flying direction of the UAV-UE. The action vector $a = r_i + r_s$ limits the horizontal boundaries of flying UAV-UE, and $b = r_i + r_s + l$ is the vertical boundaries of the UAV-UE. r_s is the safe distance between A_i and FR$_k$ to ensure the UAV cannot be affected by the fire and close enough to stream the fire area, l is the length of flying region, and $h_{\max}$ is the maximum height of UAV-UE regulated by the government (i.e. 120 m in UK [33]). The upper boundaries are introduced to ensure better uplink performance, capture a clear picture. This is because the picture frame can be clearer when the UAV-UEs are closer to the surveillance area. Furthermore, to capture full panoramic

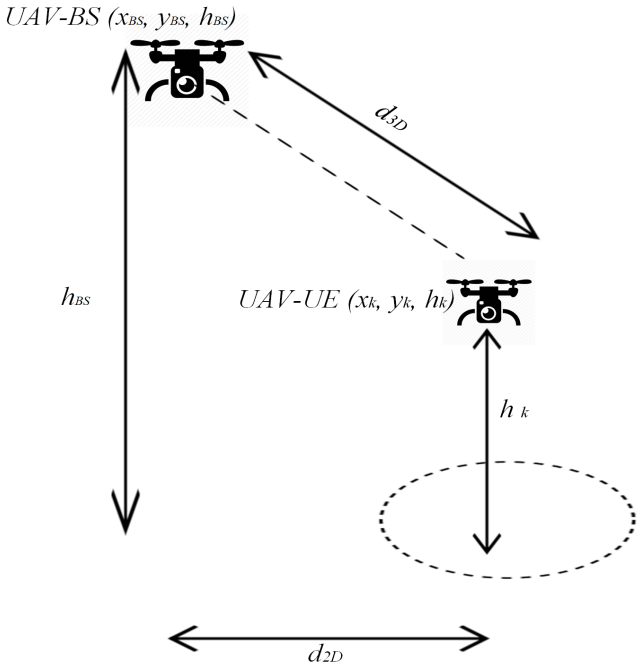

Fig. 3. UAV-to-UAV communication.

video, we propose the boundary flying area for UAV-UEs in each fire area, which can be written as Eq. (5).

B. Channel Model

In the wireless network, we assume that the channel model between the kth UAV-UE and the UAV-BS contains large-scale fading (path loss and channel gain) and small-scale fading [3]. We assume that the link between the UAVs are line-of-sight (LoS). Also, we consider that the wildfires have occurred in rural areas, and the height of the UAV should be higher than that of the fire to guarantee the UAV cannot be damaged by the fire. As all UAVs are flying in free space area, there are no blockages between the UAVs, and the UAVs can capture the videos following the Rural Macrocell Aerial Vehicular (RMa-AV) path loss model in 3GPP standard [34, Table B-2]. The pathloss from the kth UAV-UE to the UAV-BS can be written as

$$PL^t_{\mathrm{LoS,k}} = 20\log\left(\frac{4\pi f_c d^t_k}{c}\right) + \eta_{\mathrm{LoS}}, \tag{6}$$

where f_c is the carrier frequency, c is the speed of light in vacuum, η_{Los} is the additional attenuation factors due to the LoS connection, and d^t_k is distance between the kth UAV-UE and the UAV-BS, as shown in Fig. 3, which can be calculated as

$$d^t_k = \sqrt{\left(x^t_{BS} - x^t_k\right)^2 + \left(y^t_{BS} - y^t_k\right)^2 + \left(h^t_{BS} - h^t_k\right))^2}. \tag{7}$$

In our model, we use the Rician distribution [35] [36] to define small scale fading $p_\xi(d_k)$, which can be denoted as

$$p_\xi(d^t_k) = \frac{d^t_k}{\sigma_0^2}\exp\left(\frac{-{d^t_k}^2 - \rho^2}{2\sigma_0^2}\right) I_0\left(\frac{d^t_k\rho}{\sigma_0^2}\right), \tag{8}$$

with $d^t_k \geq 0$, and ρ and σ are the strength of the dominant and scattered (non-dominant) paths, respectively. The Rice factor κ can be defined as

$$\kappa = \frac{\rho^2}{2\sigma_0^2}. \tag{9}$$

It is possible that the selected position of each UAV-UE can generate more interference to the UAVs nearby, which can result in poor transmission performance and make it difficult for the UAV-UE to maintain the connection with the UAV-BS. Power control can be a solution to minimize the uplink interference among UAV-UEs at appropriate power level [37]. Through properly controlling the transmit power of each UAV-UE in the uplink transmission, the interference among UAV-UEs can be mitigated. According to the 3GPP guidelines [34], we consider fractional power control for all UAVs and the power transmitted by the kth UAV-UE while communicating with the UAV-BS can be given by

$$P^t_{U_k} = \min\left\{P^{\max}_{U_k}, \left(\ 10\log_{10}(B)) + \rho_{\mathrm{u_k}} PL^t_{\mathrm{LoS,k}}\ \right)\right\}, \tag{10}$$

where $P^{\max}_{U_k}$ is the maximum transmit power of the UAV-UE, B is the channel bandwidth, and $\rho_{\mathrm{u_k}} = \{0, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ is a fractional path loss compensation power control parameter [37].

In the proposed wireless UAV network, the received power from the kth UAV-UE to the UAV-BS at the tth time slot is presented as

$$P^t_k = P^t_{U_k} G\left(d^t_k\right)^{-\alpha} 10^{\frac{-p_\xi(d^t_k)}{10}}, \tag{11}$$

where P_{U_k} is the transmit power of the kth UAV-UE, G is the channel power gains factor introduced by the amplifier and antenna [8], $(d^t_k)^{-\alpha}$ is the pathloss, α is the path loss exponent, and $p_\xi(d^t_k)$ is the Rician small scale fading. The interference from the mth UAV-UE to the UAV-BS at the tth time slot can be written as

$$I^t_{U2U} = \sum_{\mathbf{m}\in\mathbf{K}\setminus k} \psi^t_m P^t_m, \tag{12}$$

where $\psi^t_m = 1$ indicates that the transmission between the kth UAV-UE and the UAV-BS is active, otherwise, $\psi^t_m = 0$, and P^t_m is the transmit power of mth UAV-UE. The signal to interference plus noise ratio (SINR) of the UAV-BS is given by

$$\gamma^t_k = \frac{P^t_k}{N + \sum_{\mathbf{m}\in\mathbf{K}\setminus k}\psi^t_m P^t_m}, \tag{13}$$

where N is the noise power at the UAV-BS whose elements are average of independent random Gaussian variables with the variances σ^2_n. Then, the transmission uplink rate from the kth UAV-UE to the UAV-BS can be denoted as

$$R^t_k = B\log_2\left(1 + \gamma^t_k\right). \tag{14}$$

TABLE I
TYPE OF VIDEO QUALITY [38]

Video Quality	Resolution (pixels)	Framrate (FPS)	Bitrate (average)	Data used per minute	Data used per 60 minutes
144p	256x144	30	80-100 Kbps	0.5-1.5 MB	30-90 MB
240p	426x240	30	300-700 Kbps	3-4.5 MB	180-250 MB
360p	640x360	30	400-1,000 Kbps	5-7.5 MB	300-450 MB
480p	854x480	30	500-2,000 Kbps	8-11 MB	480-660 MB
720p (HD)	1280x720	30-60	1.5-6.0 Mbps	20-45 MB	1.2-2.7 GB
1080p (FHD)	1920x1080	30-60	3.0-9.0 Mbps	50-68 MB	2.5-4.1 GB

C. Video Streaming Model

In this paper, we consider the long-term video streaming that are modelled as consecutive video segments. Each segment consists of multiple frames, and the frame is considered to be the smallest data unit. The resolution of each frame corresponds to its minimum data rate requirement. Table I presents the type of Video Quality [38]. For example, if the communication rate (bitrate) is between 300-700 kbps, the video type that we should consider to use is 240 p. Knowing that 144p corresponds to the smallest size of the video type, all UAV-UEs need to satisfy the minimum uplink bitrate, i.e., R_{min}=80 kbps.

Each UAV-UE is equipped with a non-ordinary optical camera with the resolution of $r_{px} \times r_{py}$, and the video is consisted of multiple consecutive frames [24], which is used to monitor the fire area with three main goals: 1) detect the size of fire by continuous capturing the panoramic video; 2) verify and locate fires reported; and 3) closely monitor a known fire by streams using distribution relationship around the incident. The quality of the video frame depends on its resolution of the ith video frame at the tth time slot v_i^t. Furthermore, for each video frame, we assume that it has the same playback time T_l, i.e. 2ms to 4ms, which depends on 30 FPS or 60 FPS. In addition, the delay of the video streaming via UAVs is consisted of three elements, i.e. capture time, encoding time, and transmission time. As all UAVs capture a video using the same resolution, the capture time and the encoding time are constant. Thus, we mainly focus on the uplink transmission time, which can be expressed as

$$T_{i,k}^t = \frac{D(v_i^t)}{R_k^t} = \frac{r_{px} \cdot r_{py} \cdot b}{B \log_2 (1 + \gamma_k^t)}, \tag{15}$$

where b is the number of bits per pixel, and $D(v_i^t)$ is the data size based on v_i^t. The video frames are processed in parallel in multi-core processors, and the time consumption at the tth time slot is $T^t = \max\{T_{i,k}^t\}$ [39]. To guarantee the smoothness and seamless of the video streaming, T^t must satisfy the delay constraint, namely, $T^t < T_l$.

D. Quality of Experience Model

The key parameters of video streaming are video quality, quality of variation, rebuffer time, and the startup delay [40]. Therefore, QoE is formulated by three factors, 1) the sum of video quality over K UAV users in ith area, 2) jitter between video frames (video smoothness penalty), and 3) video latency (delay penalty), where I is the maximum number of fire areas at the tth time slot. In practice, the video quality metric measures each video frame quality based on the selection of bitrate. However, the quality will decrease if the long-term video playback is not smooth, so we introduce two parameters, namely, video smoothness penalty and video latency. In long-term scenario, the drastic changes of video resolution can lead to uncomfortable of firefighters. Therefore, in our learning algorithm, we consider this element to ensure the smoothness of the playback. Finally, the latency is determined by streaming time and transmission time at the tth time slot, T^t, rebuffer time, and the startup delay [42]. According to [22], the rebuffering time and startup delay can be ignored. Thus, the video transmission may be suffered from a delay, which can be calculated as $D^t = T^t - T_l$, where T_l is the delay constraint. The QoE is denoted as

$$QoE = \frac{\kappa_{i,k}^t}{IK} \left(\sum_{i=1}^{I} \sum_{k=1}^{K} q(R_{i,k}^t) - |q(R_{i,k}^t) - q(R_{i,k}^{t-1})| \right) - \omega^t D^t, \tag{16}$$

where $q(R_{i,k}^t)$ is video quality metrics [41], which can be written as

$$q(R_{i,k}^t) = \log \left(\frac{R_{i,k}^t}{R_{\min}(v_i^t)} \right), \tag{17}$$

where $\kappa_{i,k}^t$ and ω^t are the weights of video quality and delay, respectively. As our aim is to maximize the QoE, the condition of $\kappa_{i,k}^t > \omega^t$ must be guaranteed, and $R_{\min}(v_i^t)$ is the minimum rate that should be satisfied for the selected v_i^t.

E. Problem Formulation

Our aim is to maximize the QoE that jointly exploit the optimal positions of the UAV-BS and UAV-UEs, power control, and the optimal adaptive bitrate selection. The fluctuation of the transmission link will cause unstable network performance that leads to low QoE and high delay. Thus, to minimize the delay and maintain the smoothness at each Transmission Time Interval (TTI) and maximize the quality of video streaming. We jointly consider the optimal UAV-BS location $\mathcal{P} = (x_{BS}^t, y_{BS}^t, h_{BS}^t)$, the position of the kth UAV-UE $\mathcal{U} = (x_{i,k}^t, y_{i,k}^t, h_{i,k}^t)$, the maximum power control of UAV-UE P_{U_k}, the bitrate resolution $BV = \{144, 240, 360, 480, 720, \text{and } 1080\}$p, and UAV-UE's power $P_{U_k} = \{23, 25, \text{and } 30\}$ dBM, so that the adequate throughput can be achieved.

In this work, we aim to tackle the problem of optimizing the control factors defined as $A_t = \{BP, BU, BV, P_{U_k}\}$ in an online manner for every frames. At the tth time slot, the UAV-BS aims at maximizing the total long-term QoE in continuous

time slots with respect to the policy π that maps the current state information s_t to the probabilities of selecting possible actions in A_t. Therefore, based on the QoE of each UAV-UE, the optimization problem can be formulated as

$$\max_{\pi(A_t|S_t)} \sum_{i=t}^{\infty} \sum_{k=1}^{K} \gamma^{i-t} \, \mathrm{QoE}_k(i) \tag{18}$$

$$\text{s.t.} \max h_i > h_{BS}^t > h_{\max}, \tag{19}$$

$$R_{i,k}^t > R_{(\min)}^k(v_i^t), \tag{20}$$

$$v_i^t \in \{144\text{p}, 240\text{p}, 360\text{p}, 480\text{p}, 720\text{p}, 1080\text{p}\} \tag{21}$$

$$P_{(min)} > P_{U_k}^t > P_{(\max)}, \tag{22}$$

$$\sqrt{(x_{BS}^t - x_i)^2 + (y_{BS}^t - y_i)^2} > r_i + r_s, \tag{23}$$

$$\mathcal{U} \in \text{Eq.}(1). \tag{24}$$

where the objective function in Eq. (18) captures the average QoE received at the UAV-BS and $\gamma \in [0, 1)$ is the discount factor to determine the weight accumulated in the future frames, and $\gamma = 0$ means that the agent concerns only the immediate reward. The UAV-BS's height must follow the condition in Eq. (19). The minimum requirement of data rate of UAV-UEs based on the adaptive bitrate selection guarantees R_k obtained from $\mathcal{U}_k$ as shown in Eq. (20) and follows minimum bitrate in Eq. (21) as shown in Table 1, while $P_{U_k}^t$ in Eq. (22) follows the maximum and the minimum power constraints. Then, Eq. (23) guarantees that the position of the UAV-BS will not intersect with the UAV-UE's flying region. $\mathcal{U}$ follows the requirement of the flying region FR_i presented in Eq. (1). In the experiment, the UAVs are hovering and flying at a constant speed.

In our study, there are several trafe-offs in this problem: 1) throughput-bit rate trade-off, 2) throughput-power control trade-off, 3) throughput-distance trade-off, 4) power-distance trade-off, 5) throughput-video smoothness trade-off and 6) throughput-delay trade-off. Therefore, to achieve maximum QoE in long-term time slot, it is important to solve an optimal trade-off between data rate, bit-rate resolution selection, power control, and positions, which further motivates us to use the learning algorithms to jointly optimize the total long-term QoE of all UAV-UEs. All the factors mentioned above help to measure the QoE from the selected resolution to maintain the whole performance in long-term time slots. Also, the correlation between the video smoothness and the penalty delay is to ensure the overall video performance from the beginning to the end.

F. Channel State Information Sharing

Signal exchange happens in the uplink, the UAV-UEs have to send their locations, fire areas to the UAV-BS, and the QoE information of each UAV-UE will be readily available at the UAV-BS. After the learning is performed at the UAV-BS, the outputs are the actions, including the movement of the UAVs, selected video resolution, and power. After that, the selected actions will be sent through the downlink from the UAV-BS to each UAV-UE for its control. The whole process is illustrated in Fig. 4.

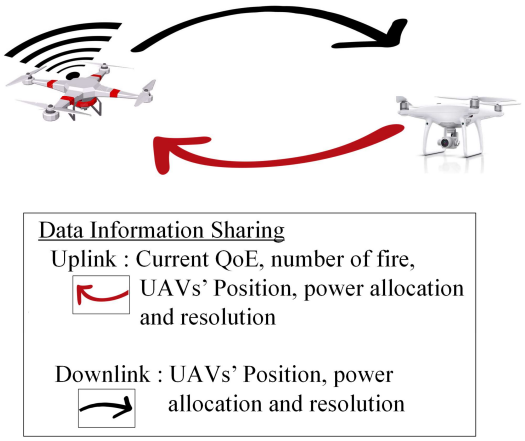

Fig. 4. UAV-BS to UAV-UE communication information sharing.

III. Optimization Problem via Reinforcement Learning

In this section, we design several DRL algorithms to maximize the long-term QoE in a UAV-to-UAV network. Since the channel and locations of fire change over time, different number of UAVs are required at each time slot. In our problem, we consider the long-term quality and smoothness of video streaming, which cannot be solved by the traditional optimization problem. Thus, we deploy deep reinforcement learning (DRL) to solve the problem. Specifically, we propose two DRL algorithms, which are Deep Q-Learning and Actor-Critic, to maximize the long-term QoE of live video streaming in U2U communication.

A. Reinforcement Learning

For our proposed RL-based method, the UAV-BS acts as centralized agent to collect video from UAV-UEs while maximizing QoE. The QoE optimization problem is influenced by the delay, UAVs' positions, and bitrate selection during each TTI, and forms a partially observable Markov decision process (POMDP). At each TTI, the channel network condition, fire arrival, and network condition are different. Therefore, through learning algorithms, the UAV-BS (agent) is able to select the positions of the UAV-BS, positions of the UAV-UEs, the adaptive resolution and the maximum power allocation in order to maximize the individual QoE at each time slot and the long-term QoE objective.

1) State Representation: The current state s^t corresponds to a set of current observed information. The state of the UAV-BS can be denoted as $s = [\mathcal{P}, \mathcal{V}, \mathcal{U}, P_{U_k}, \mathrm{QoE}]$, where $\mathcal{P}$= (x_{BS}^t, y_{BS}^t, h_{BS}^t) is the position of the UAV-BS, $\mathcal{V}$ is the bitrate selection, $\mathcal{U} = (x_k^t, y_k^t, h_k^t)$ is the positions of UAV-UEs, and P_{U_k} is k-UAV-UE's power.

2) Action Space: Q-agent will choose action $a = (BP, BU, BV, P)$ from set $\mathcal{A}$. The dimension of the action set can be calculated as $\mathcal{A} = BP \times BU^{i \times k} \times BV^i \times P$. The actions for UAVs include (i) UAV-BS's flying direction (BP), (ii) UAV-UEs' flying directions (BU), (iii) resolution of the

ith UAV-UE (BV), and (iv) UAV-UE's power (P). The action space is presented as

- BP = (up, down, left, right, ascent, descent or hover)
- BU = (up, down, left, right, or hover)
- BV= (144, 240, 360, 480, 720, or 1080) p
- P = (23, 25, 30) dBm

To ensure the balance of exploration and exploitation actions of the UAV-BS, ϵ -greedy ($0 < \epsilon \leq 1$) exploration is deployed. At the tth TTI, the UAV-BS randomly generates a probability p_ϵ^t to compare with ϵ. If the probability $p_\epsilon^t < \epsilon$, the algorithm randomly selects an action from the feasible actions to improve the value of the non-greedy action. However, if $p_\epsilon^t \geq \epsilon$, the algorithm exploits the current knowledge of the Q-value table to choose the action that maximizes the expected reward.

3) Rewards: When the a^t is performed, the corresponding reward re^t is defined as

$$re^t = \frac{\psi_{i,k}^t}{IK}\left(\sum_{i=1}^{I}\sum_{k=1}^{K} q(R_{i,k}^t) - |q(R_{i,k}^t) - q(R_{i,k}^{t-1})|\right) - \omega^t D^t, \tag{18}$$

where $q(R_{i,k}^t)$ is video quality metrics [41], which can be written as

$$q(R_{i,k}^t) = \log\left(\frac{R_{i,k}^t}{R_{\min}(v_i^t)}\right), \tag{19}$$

$\psi_{i,k}^t$ and ω^t are the weights of video quality and delay, respectively. If $R_{i,k}^t$ is unable to satisfy the minimum transmission rate for $R_{\min}^k(v_i^t)$, namely, $R_{i,k}^t < R_{\min}^k(v_i^t)$, the system will receive negative reward, which means $\mathrm{re}^t < 0$.

B. Q-learning

The learning algorithm needs to use Q-table to store the state-action values according to different states and actions. Through the policy $\pi(s,a)$, a value function $Q(s,a)$ can be obtained through performing action based on the current state. At the tth time slot, according to the observed state s^t, an action a^t is selected following ϵ -greedy approach from all actions. By obtaining a reward re^t, the agent updates its policy π of action a^t. Meanwhile, Bellman Equation is used to update the state-action value function, which can be denoted as

$$\begin{aligned} Q(s^t,a^t) =& (1-\alpha)Q(s^t,a^t) \\ &+ \alpha\left\{\mathrm{re}^{t+1} + \gamma \max_{a^t\in\mathcal{A}} Q(s^{t+1},a^t)\right\}, \end{aligned} \tag{20}$$

where α is the learning rate, $\gamma \in [0,1)$ is the discount rate that determines how current reward affects the updating value function. Particularly, α is suggested to be set to a small value (e.g., α = 0.01) to guarantee the stable convergence of training.

C. Deep Q-learning

However, the dimension of both state space and action space can be very large if we use the traditional tabular Q-learning, which will cause high computation complexity. To solve this problem, deep learning is integrated with Q-learning, namely, Deep Q-Network (DQN), where a deep neural network (DNN) is used to approximate the state-action value function [42]. $Q(s,a)$ is parameterized by using a function $Q(s,a;\boldsymbol{\theta}_{\mathrm{DQN}})$, where $\boldsymbol{\theta}_{\mathrm{DQN}}$ is the weight matrix of DNN with multiple layers. s is the state observed by the UAV and acts as an input to Neural Networks (NNs). The outputs are selected actions in $\mathcal{A}$. Furthermore, the intermediate layer contains multiple hidden layers and is connected with Rectifier Linear Units (ReLu) via using $f(x) = \max(0,x)$ function. At the tth time slot, the weight vector is updated by using Stochastic Gradient Descent (SGD) and Adam Optimizer, which can be written as

$$\boldsymbol{\theta}_{\mathrm{DQN}}^{(t+1)} = \boldsymbol{\theta}_{\mathrm{DQN}}^t - \lambda_{\mathrm{ADAM}} \cdot \nabla\mathcal{L}(\boldsymbol{\theta}_{\mathrm{DQN}}^t), \tag{21}$$

where λ_{ADAM} is the Adam learning rate, and $\lambda_{\mathrm{ADAM}} \cdot \nabla\mathcal{L}(\boldsymbol{\theta}_{\mathrm{DQN}}^t)$ is the gradient of the loss function $\mathcal{L}(\boldsymbol{\theta}_{\mathrm{DQN}}^t)$, which can be written as

$$\begin{aligned} \nabla\mathcal{L}(\boldsymbol{\theta}_{\mathrm{DQN}}^t) = \mathbb{E}_{S^i,A^i,Re^{i+1},S^{i+1}}\big[&(Q_{\mathrm{tar}} - Q(S^i,A^i; \\ &\boldsymbol{\theta}_{\mathrm{DQN}}^t)\cdot\nabla Q(S^i,A^i;\boldsymbol{\theta}_{\mathrm{DQN}}^t)], \end{aligned} \tag{22}$$

where the expectation is calculated with respect to a so-called minibatch, which are randomly selected in previous samples $(S^i, A^i, Re^{i+1}, S^{i+1})$ for some $i \in \{t-M_r, t-M_r+1, \ldots, t\}$, with M_r being the replay memory. The minibatch sampling is able to improve the convergence reliability of the updated value function [43]. In addition, the target Q-value Q_{tar} can be estimated by

$$Q_{\mathrm{tar}} = re^{i+1} + \gamma \max_{a\in\mathcal{A}} Q(S^{i+1},a;\bar{\boldsymbol{\theta}}_{\mathrm{DQN}}^t), \tag{23}$$

where $\bar{\boldsymbol{\theta}}_{\mathrm{DQN}}^t$ is the weight vector of the target Q-network to be used to estimate the future value of the Q-function in the update rule. This parameter is periodically copied from the current value $\boldsymbol{\theta}_{\mathrm{DQN}}^t$ and kept fixed for a number of episodes. The DQN algorithm is a value-based algorithm, which can obtain an optimal strategy through using experience replay and target networks. It enables the agent to sample from and be trained by the previously observed data online. This is due to the experience replay mechanism and randomly sampling in DQN, which use the training samples efficiently to smooth the training distribution over the previous behaviours. Not only does this massively reduce the amount of interactions needed with the environment, but also reduce the variance of learning updates. The DQN algorithm will create a sequence of policies whose corresponding value functions converge to the optimal value function, when both the sample size and the number of iteration go to infinity. The DQN algorithm is presented in Algorithm 1.

D. Actor-Critic

Different from the DQN algorithm, which obtains the optimal strategy indirectly by optimizing the state-action value function, while the AC algorithm directly determines the strategy that should be executed by observing the environment state. The AC algorithm combines the advantages of value-based function method and policy-based function method. In the AC algorithm, the agent is consisted of two parts, i.e., actor network and critic network, and it solves the problem through using two neural networks. Meanwhile, the AC algorithm deploys a separate memory structure to explicitly represent

Algorithm 1 : Optimization by using DQN

Input: The set of UAV-BS position $\{x_{BS}, y_{BS}, h_{BS}\}$, bitrate selection V, the position of the kth UAV-UE $U_k = (x_k^t, y_k^t, h_k^t)$, $\sum QoE$ and operation iteration I.
Algorithm hyperparameters: Learning rate $\alpha \in (0, 1]$, $\epsilon \in (0, 1]$, target network update frequency K;
Initialization of replay memory M, the primary Q-network $\boldsymbol{\theta}$, and the target Q-network $\bar{\boldsymbol{\theta}}$;
for $e \leftarrow 1$ ***to*** I **do**
  Initialization of s^1 by executing a random action a^0;
  for $t \leftarrow 1$ ***to*** T **do**
    if $p_\epsilon < \epsilon$ **then**: Randomly select action a^t from $\mathcal{A}$;
    else select $a^t = \underset{a \in \mathcal{A}}{\operatorname{argmax}} Q\left(S^t, a, \theta\right)$;
    The UAV-BS performs a^t at the tth TTI ;
    The UAV-BS observes s^{t+1}, and calculate re^{t+1} using Eq. (18);
    Store transition $(s^t; a^t; re^{t+1}; s^{t+1})$ in replay memory M;
    Sample random minibatch of transitions $(S^i; A^i; Re^{i+1}; S^{i+1})$ from replay memory M;
    Perform a gradient descent for $Q(s; a; \boldsymbol{\theta})$ using (22) ;
    Every K steps update target Q-network $\bar{\boldsymbol{\theta}} = \boldsymbol{\theta}$.
  end
end

Algorithm 2 : Actor-Critic Algorithm

Inputs: The set of UAV-BS position $\{x_{BS}, y_{BS}, h_{BS}\}$, bitrate selection V, the position of the kth UAV-UE $U_k = (x_k^t, y_k^t, h_k^t)$, $\sum QoE$ and operation iteration I.
Algorithm hyper-parameter: Learning rate $\alpha_c \in (0, 1]$, $\epsilon \in (0, 1]$, Target network update frequency K;
Initialization of policy parameter θ_{AC}, weight of the actor network **w**, value of the critic network $\boldsymbol{V}$;
for $e \leftarrow 1$ ***to*** I **do**
  Initialization of s^0 by executing a random action;
  for $t \leftarrow 1$ ***to*** T **do**
    Select action a^t according to the current policy;
    The UAV-BS observes s^{t+1}, and calculate re^{t+1} using (18);
    Store transition $(s^t; a^t; re^{t+1}; s^{t+1})$;
    Update TD-error functions;
    Update the weights **w** of critic network by minimizing the loss;
    Update the policy parameter vector θ for actor network;
    Update the policy θ_{AC} and state-value function $V(s^t, \boldsymbol{w}^t)$.
  end
end

the policy which is independent of the value function. The policy structure is known as the actor network, which is used to select actions. Meanwhile, the estimated value function is known as the critic network, which is used to criticize the actions performed by the actor. The AC algorithm is an on-policy method and temporal difference (TD) error is deployed in the critic network. To sum up, the actor network aims to improve the current policies while the critic network evaluates the current policy to improve the actor network in the learning process.

The critic network uses value-based learning to learn a value function. The state-action value function $V(s^t, \boldsymbol{w}^t)$ in the critic network can be denoted as

$$V(s, \boldsymbol{w}^t) = \boldsymbol{w}^\top \boldsymbol{\Phi}(s^t), \tag{24}$$

where $\boldsymbol{\Phi}(s^t) = s^t$ is state features vector and $\boldsymbol{w}^t$ is critic parameters, which can be updated as

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t + \alpha_c^t \delta^t \nabla_{\boldsymbol{w}} V\left(s^t, \boldsymbol{w}^t\right), \tag{25}$$

where α_c is the learning rate in the critic network. After performing the selected action, TD error δ^t is used to evaluate whether the selected action based on the current state performs well [44], which can be calculated as

$$\delta^t = \mathrm{re}^{t+1} + \gamma_{\boldsymbol{w}}(V\left(s^{t+1}, \boldsymbol{w}^t\right) - V\left(s^t, \boldsymbol{w}^t\right)). \tag{26}$$

Then, the actor network is used to search the best policy to maximize the expected reward under the given policy with parameters $\boldsymbol{\theta}_{\mathrm{AC}}$, which can be updated as

$$\boldsymbol{\theta}_{\mathrm{AC}}^{t+1} = \boldsymbol{\theta}_{\mathrm{AC}}^t + \alpha_a \nabla_{\boldsymbol{\theta}_{\mathrm{AC}}} J\left(\pi_{\boldsymbol{\theta}_{\mathrm{AC}}^t}\right), \tag{27}$$

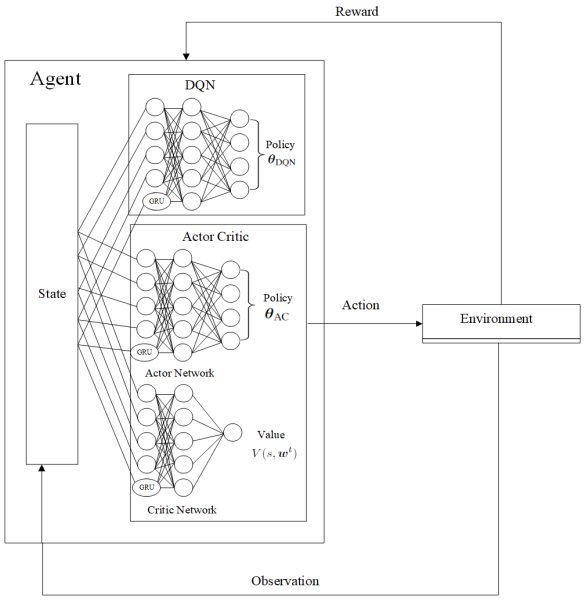

Fig. 5. The network architecture designed.

where α_a is the learning rate in the actor network, which is positive and must be small enough to avoid causing oscillatory behavior in the policy, and according to [44], $\nabla_{\boldsymbol{\theta}_{AC}} J\left(\pi_{\boldsymbol{\theta}_{AC}}\right)$ can be calculated as

$$\nabla_{\boldsymbol{\theta}_{\mathrm{AC}}} J\left(\pi_{\boldsymbol{\theta}_{\mathrm{AC}}^t}\right) = \delta^t \nabla_{\boldsymbol{\theta}_{\mathrm{AC}}} \ln\left(\pi\left(a^t | s^t, \boldsymbol{\theta}_{\mathrm{AC}}^t\right)\right). \tag{28}$$

The AC algorithm is presented in Algorithm 2.

Finally, Fig. 5 shows the network architecture design, where the current state is input to the neural network for both algorithms, DQN and Actor-Critic. Next, the RNN based GRU

network is used to approximate the value function or the policy of each DRL algorithm. The GRU helps capture the correlation among the state or action over time, which can help DRL select more optimal action and guarantee the high quality of video transmission. While in Actor-Critic, two neural networks are included in our proposed model. The actor-network is to find a strategy policy, θ_{AC}. Then, the critic network is used to make an objective assessment and make an accurate assessment for the current state. The next key step is to determine the action to be sent to the environment and the reward to measure the QoE. Then, the new states are generated from observation for the next round of updates.

E. Analysis Complexity of Reinforcement Learning Algorithms

The computational complexity of the DQN/AC algorithm, which includes DQN/AC learning architecture, the action selection of the UAVs, and the downlink transmission, are given by $O(mlogn + 2^A + N_i N_k)$, where m is the number of layers, n is the number of units per learning layer, A is number of action, N_i is number of fire area, and N_k is number of UAV for each fire area.

IV. Simulation Results

In this section, we evaluate our proposed learning algorithms in our problem setup. The area of the region is 5000 m x 5000m x 100m. In the simulation, the maximum flying height $h_{\max}$ of the UAV-BS is 100m, which is satisfied with the maximum flying height 120m that is stipulated by the UK government. We assume that the available video bitrates of the adaptive video streaming for each video frame are $(80, 300, 700, 1000, 2000, 3000)$kbps. The target area is captured by K UAV-UE(s), i.e., $K = 4$ in the ith fire area A_i $(i = 1, 2, \text{and}, 3)$. At the beginning, the UAV-BS will be deployed at the centre of the environment, i.e. (1250, 1250, $h_{\min}$), where $h_{\min}$ is the maximum height of the fire. When the fire occurs at the remote area, the UAV-UEs will immediately reach the fire location to stream and oversee the real-time situation. The height of the UAV-UEs in each fire area are fixed and follow the distribution of the fire height [31]. The network parameters for the system are shown in Table II and follow the existing approach and 3GPP specifications in [8], [34], and [45]. The performance of all results is obtained by averaging around 100 episodes, where each episode is consisted of 100 TTIs. The result is measured for the equal duration of the time slot at each time slot t and also called as TTI, where each TTI is equal to 0.5ms as follows in 3GPP [34]. Finally, the channel model parameters and grid environment parameters are set according to [8].

Fig. 6 plots the average QoE value over different grid sizes via AC and DQN algorithms. From the result, it can be seen the 25m $\times$ 25m grid sizes produced the highest average QoE of the UAV-BS for both DQN and AC algorithms, therefore in the next simulation, we use 25 $\times$ 25m grid size. From the result, the number of grids will influence the movement of the UAV, the UAV will move more frequently in small grid size with more number of grids. In this case, the performance can be improved due to that the UAV can explore and exploit

TABLE II
Parameter

Parameter	Value
Number of UAV-UEs	12
Transmission power, PUe	23 dBm [8]
Bandwidth, B	3 MHz
Noise variance σ^2	-96 dBm [8]
Center frequency, f_c	2 GHz [46, pp. 3777]
Power gains factor, G	-31.5 dB [8]
Alpha, α	2
Channel parameter, LoS	0.1 [45, pp. 572]
Channel parameter, $NLoS$	21 [45]
Channel parameter, a	4.88 [46, pp. 3777], [47, pp. 7]
Channel parameter, b	0.43 [46, pp. 3777] , [47, pp. 7]
Radius of target region	1250 m
Radius of Surveillance region, r_i	250 m
Learning Rate	0.1, 0.01
Initial, Final Exploration	1, 0.1
Discount Rate	0.8
Replay memory	1000

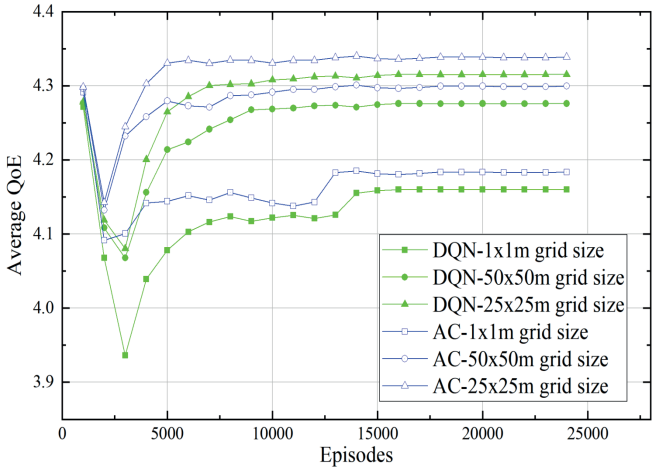

Fig. 6. Average QoE of the UAV-BS with different schemes via different learning algorithms with different grid size of each episode.

the environment more accurately. However, increasing the number of grids can lead to increased complexity in learning algorithms. It is because the action space will increase with more number of grids. In practice, the UAV operator has to decide what will be the best square size according to the movement step of each UAV. However, if we want to reduce the complexity by increasing the grid size or decreasing the number of grid, the result shows degraded performance of QoE and it takes more time to obtain the convergence results due to difficulties in finding an optimal solution in long-term QoE analysis.

In each scenario, our proposed DQN and AC algorithms are compared with the Greedy algorithm. The Greedy algorithm selects the actions based on the immediate reward and local optimum strategy. The DQN is designed with 3 hidden layers, which each layer consists of 256, 128, 128 ReLU units, respectively. For the AC method, the critic DNN consists of an input layer with 19 neurons, a fully-connected neural network with two hidden layers, each with 128 neurons, and an output layer with 1 neuron. The UAV-BS is initially set at the centre of the environment with the height $h_{\min}$. In wildfires environment

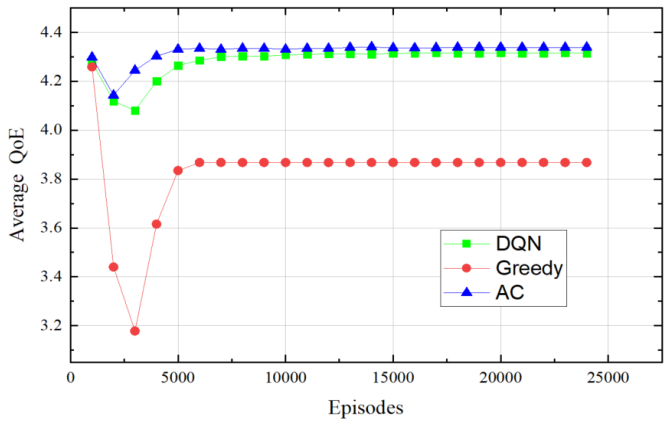

Fig. 7. Average QoE value for each frame via AC, DQN and Greedy algorithms.

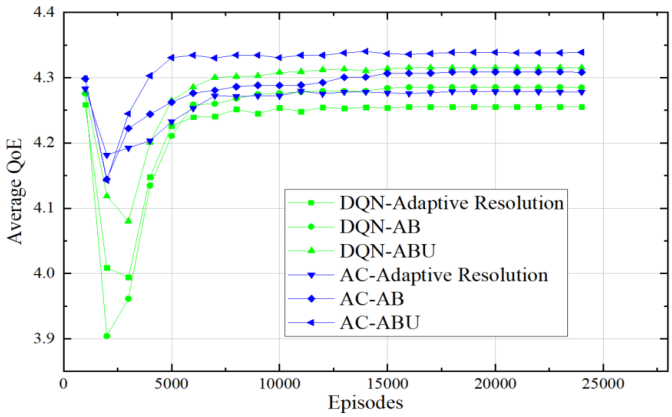

Fig. 8. Average QoE of the UAV-BS with different schemes via different learning algorithms and with different optimization schemes of each episode.

problem, the network coverage with smooth streaming needs to overview the real-time situation. To guarantee high quality of video transmission from multiple UAVs in continuous time slots, the Recurrent Neural Network (RNN) is deployed. In temporal data, RNN based GRU network can approximate the value function or the policy of each DRL algorithm, where the stateless RNN does not need to re-initialize the memory at each training step, while the training progress is more resource-hungry and less stable [27]. The learning-based predictor uses a modern RNN model with parameters θ to predict the traffic statistic at each frame. The use of RNN is due to its ability to capture the time correlation of traffic statistics over multiple frames, which can aid in learning the time-varying traffic trend and improving prediction accuracy. Thus, RNN can capture the correlation among the state or action in over time, which can help DRL select more optimal action, and guarantee the high quality of video transmission.

Fig. 7 plots the average QoE value over all frames via AC, DQN and Greedy algorithms. It can be seen that DRL algorithms outperform the non-learning based algorithm, i.e., Greedy algorithm. The convergence of the reinforcement learning algorithms has been proved in [48], [49], an agent of the Q-learning algorithm is assured to converge to the optimal Q. Fig. 7 plots the average QoE value over all frames in each episode via DQN/AC learning algorithms, which shows the convergence of the proposed two algorithms. It is observed that the total reward and the convergence speed of these two DRL learning algorithms follows: $AC > DQN$. This is due to the fact that the AC algorithm is updated in two steps, including the critic step and actor step. At each step, the critic network judges the action selected by the actor network, which can select the actions more appropriately. Moreover, it can be seen that the DRL algorithms outperform the Greedy algorithm, where the convergence speed of the DRL algorithms is faster than that of the Greedy algorithm. Specifically, in the Greedy algorithm, the UAVs only consider exploiting the current reward, rather than exploring the long-term reward. Therefore, the UAVs are not able to achieve higher expected reward compared to the DRL algorithm.

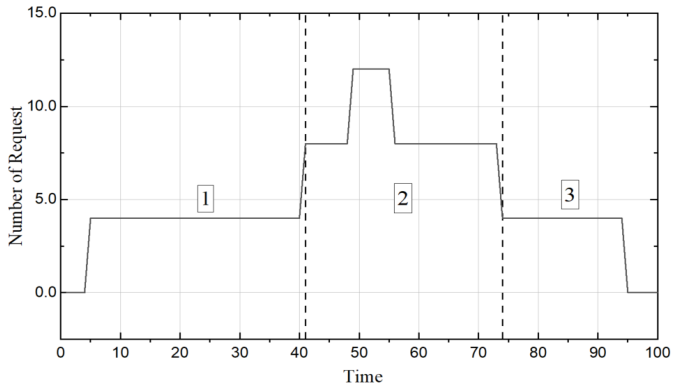

Fig. 9. The request of the UAV-UEs in continuous time slots.

Fig. 8 plots the average QoE of the UAV-BS with different video transmission schemes via different learning algorithms in each episode. For simplicity, "Adaptive Resolution" represents the scheme with adaptive resolution, "AB" is the scheme with adaptive resolution and dynamic UAV-BS, and "ABU" is the scheme with adaptive resolution, dynamic UAV-BS and UAV-UEs. It is observed that the average QoE of the AC algorithm outperforms all other algorithms, with it being able to achieve an optimal trade-off between data rate, bitrate resolution selection, power control, and positions. From the result, it is observed that with the dynamic environment and large size of the action, and the AC algorithm is able to select proper positions of UAVs and video resolution of video frames. This is mainly due to the experience replay mechanism, which efficiently utilizes the training samples, and the actor and critic functions are able to smooth the training distribution over the previous behaviours compared to DQN. In addition, we can observe that the strategies of selecting optimal positions for UAVs achieve higher performance compared to the UAVs with

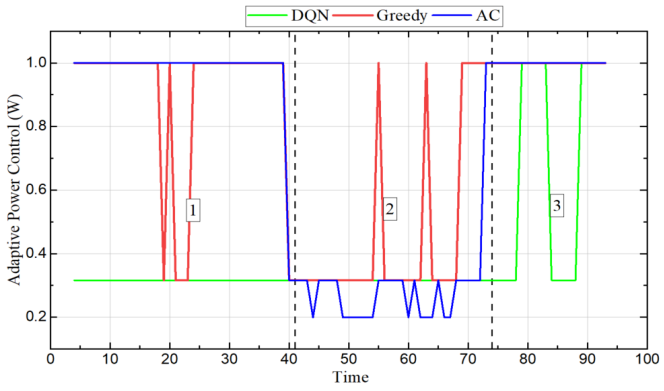

Fig. 10. The power control of the UAV-UEs in continuous time slots with different learning algorithms.

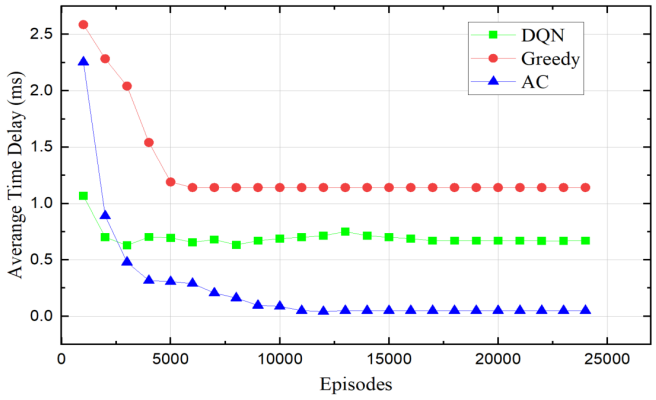

Fig. 12. Average latency of video streaming with different learning algorithms.

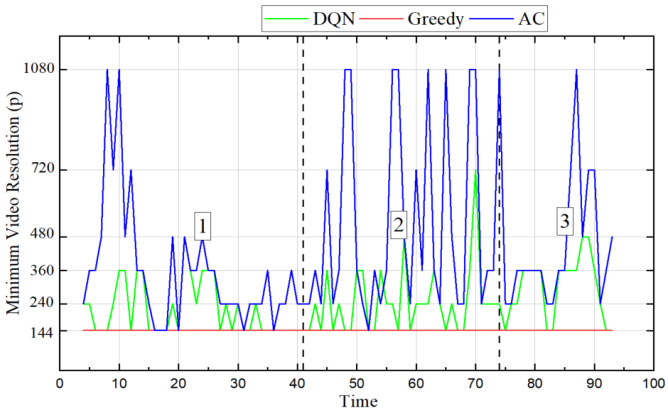

Fig. 11. The average adaptive resolution of the UAV-UEs in continuous time slots with different learning algorithms.

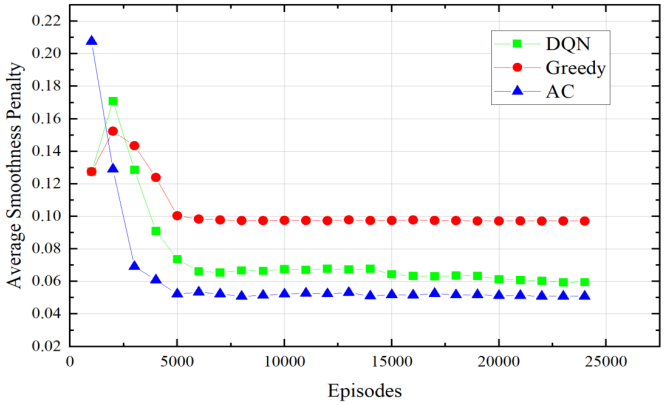

Fig. 13. Average smoothness penalty with different learning algorithms.

fixed locations. This result emphasizes the importance of the strategy with mobile UAVs. This is due to the fact that mobile UAVs can move through the network to reach the optimal positions that are able to adapt to dynamic fire scenarios.

Next, we provide more in-depth investigation of the relationship between the number of UAV request, adaptive video resolution, adaptive power control, and throughput with different learning algorithms in continuous 100 time slots. The results are also compared among the three algorithms, namely DQN, AC, and Greedy algorithms. The detailed results show how the optimization control helps UAVs to maximize the QoE at each time slot.

Fig. 9 plots the UAV's requests follow the fire arrival distribution, which follow Poisson process distribution with density λ. In phase 1, there is a small number of fire arrival which leads to low request of UAV's number. However, as the number of fire increases, more UAVs are needed, as shown in phase 2. While in phase 3, it shows that the number of request decreases, so that less UAVs are required. As the number of requests rapidly changes, we introduce power control to control the transmit power at UAV-UEs to mitigate the interference among UAV-UEs, thus maximizing the achievable rate of each UAV-UE.

Following the fire arrival requests depicted in Fig 9, Fig. 10 shows the plots of the average power control over all UAV-UEs in continuous time slots with AC, DQN and Greedy algorithms. The power control helps mitigate the interference among UAV-UEs. As shown in phase 1 and phase 3 in Fig. 9, there is a small number of fire requests with small number of UAVs to transmit the data. However, when the number of requests increases, a large number of UAVs are demanded as shown in phase 2 of Fig. 9. As can be seen from Phase 2 of Fig. 10, the DRL algorithms learn the environment and effectively reduce the transmit power of each UAV-UE, to reduce the interference from UAV-UEs. We see that the Greedy algorithm maintains the higher power, even though high power can provide high received signal, it also causes high interference at the UAV-BS and failure in transmission.

Following the fire arrival requests depicted in Fig. 9, Fig. 11 shows the plots of the minimum adaptive resolution over all UAV-UEs in continuous time slots with different learning

algorithms. It is shown that the minimum video resolution of the AC algorithm is higher than that of the DQN and the Greedy algorithm in all scenarios. The AC algorithm is able to maintain an optimal video resolution at each time slot and guarantee high quality and smooth video playback with new request. However, the Greedy algorithm exploits with a minimum video resolution to maintain high rewards, and it only uses local optimal policy and causes poor performance. For phase 1 and 3, when the number of requests is low at the tth time slot, the power is high, and the throughput increases, thus, the resolution of video is high. However, when the number of request is increasing in phase 2, the AC algorithm is able to maintain a high resolution due to helps of adaptive power, which leads to better QoE for each UAV-UE. This will help to reduces the interference and improve the quality of the video resolution.

In Fig. 12, we plot the average latency of video streaming of AC, DQN and Greedy algorithms. It can be seen that the latency performance of the AC algorithm outperforms that of the DQN algorithm. When multiple video streaming exist in the U2U communication, the interference among UAV-UEs occur and causes higher latency. Based on the observed state, the AC algorithm is able to select proper positions and transmission power of the UAV-UEs to mitigate the interference, which further decreases the latency. Thus, the AC algorithm is able to maximize the average QoE with the lowest average time latency. However, the Greedy algorithm is unable to exploit the violation of latency constraints resulting in higher latency, which leads to lower QoE.

Fig. 13 plots the average smoothness penalty of AC, DQN and Greedy algorithms. The smoothness penalty demonstrates the average video stability occupancy of UAV-UEs at each episode. When the learning algorithm is able to automatically choose the suitable resolution at the tth time slots and $(t-1)$th time slot, it will obtain lower smoothness penalty and higher QoE. Moreover, the AC algorithm is able to automatically choose the proper action based on actor and critic function, which leads to better smoothness of the AC algorithm compared to that of the DQN and Greedy algorithms. It is proves that the AC algorithm guarantees the smoothness of video transmission with high QoE. Meanwhile, the Greedy algorithm shows the worst performance as it only makes local optimal selections.

Finally, the dynamic movement of UAVs is shown in Fig. 14, and the duration time is 100s. In this simulation, we assume that all the UAVs moved at a constant speed. At each time slot, the UAV-BS selects a direction from the action space, which contains 7 directions, while the action space of the UAV-UE contains 5 directions. Then, the dynamic movement maximizes the total long-term QoE of all UAVs. To reduce the complexity, we select only one UAV-UE for each fire area to illustrate the optimized trajectory of UAV-BS and UAV-UEs, which is shown in Fig. 14.

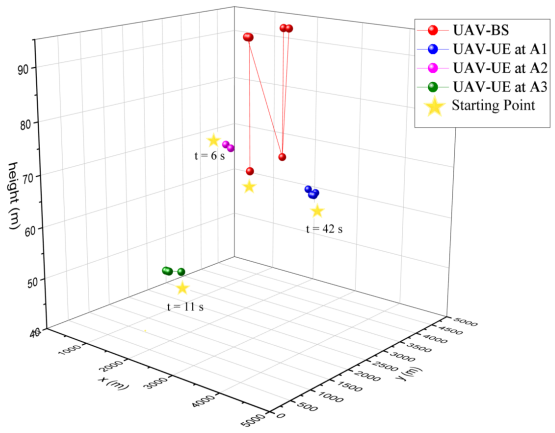

Fig. 14. Dynamic trajectory of UAVs when dynamic fire arrives from t=0 to t=100s.

V. Conclusion

In this paper, we developed a deep reinforcement learning approach for the mobile U2U communication to maximize the Quality of Experience (QoE) of UAV-UEs, through optimizing the locations for all UAVs, the additive video resolution, and the transmission power for UAV-UEs. The dynamic interference problem was resolved by utilizing the adaptive power control to achieve a higher achievable rate. Through our developed Deep Q Network and Actor-Critic methods, the optimal additive video resolution can be selected to stream real-time video frames, and optimal positions of the UAV-BS and UAV-UEs can be selected to satisfy the transmission rate requirement. Simulation results demonstrated the effectiveness of our proposed learning-based schemes compared to the Greedy algorithm in terms of higher QoE with low latency and high video smoothness. In conclusion, AC achieved a higher achievable rate and QoE in the U2U communication scenario, because of integrating the advantages of the value-based and policy-based functions. However, since AC has two neural networks and needs more parameters to update, AC is more complex in terms of computation complexity compared to that of DQN. Thus, in future research, DQN can be more preferable to use if the scenario is more complex than our current scenario.

References

[1] M. Müller, L. Vilà-Vilardell, and H. Vacik, "Forest fires in the alps–state of knowledge, future challenges and options for an integrated fire management," *EUSALP Action Group*, vol. 8, 2020.

[2] K. W. Sung *et al.*, "PriMO-5G: making firefighting smarter with immersive videos through 5G," in *Proc. 2019 IEEE 2nd 5G World Forum (5GWF)*, Sep. 2019, pp. 280–285.

[3] M. M. Azari, G. Geraci, A. Garcia-Rodriguez, and S. Pollin, "Cellular UAV-to-UAV communications," in *Proc. IEEE 30th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2019, pp. 120–127.

[4] X. Lin *et al.*, "Mobile network-connected drones: Field trials, simulations, and design insights," *IEEE Veh. Tech. Mag.*, vol. 14, no. 3, pp. 115–125, 2019.

[5] Y. Zeng, J. Lyu, and R. Zhang, "Cellular-connected UAV: Potential, challenges, and promising technologies," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 120–127, 2018.

[6] W. Mei and R. Zhang, "Aerial-ground interference mitigation for cellular-connected UAV," *IEEE Wireless Commun.*, vol. 28, no. 1, pp. 167–173, 2021.

[7] Garcia-Rodriguez *et al.*, "The essential guide to realizing 5g-connected UAVs with massive MIMO," *IEEE Commun. Mag.*, vol. 57, no. 12, pp. 84–90, 2019.

[8] S. Zhang, H. Zhang, B. Di, and L. Song, "Cellular UAV-to-X communications: Design and optimization for multi-UAV networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 1346–1359, Feb. 2019.

[9] S. Sekander, H. Tabassum, and E. Hossain, "Multi-tier drone architecture for 5g/b5g cellular networks: Challenges, trends, and prospects," *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 96–103, 2018.

[10] I. Bor-Yaliniz and H. Yanikomeroglu, "The new frontier in RAN heterogeneity: Multi-tier drone-cells," *IEEE Commun. Mag.*, vol. 54, no. 11, pp. 48–55, 2016.

[11] M. M. Azari, G. Geraci, A. Garcia-Rodriguez, and S. Pollin, "UAV-to-UAV communications in cellular networks," *IEEE Trans. on Wireless Commun.*, vol. 19, no. 9, pp. 6130–6144, Jun. 2020.

[12] X. Liu *et al.*, "Transceiver design and multihop D2D for UAV IoT coverage in disasters," *IEEE Internet of Things J.*, vol. 6, no. 2, pp. 1803–1815, Apr. 2019.

[13] A. Joshi, S. Dhongdi, S. Kumar, and K. Anupama, "Simulation of multi-UAV Ad-Hoc network for disaster monitoring applications," in *2020 Int. Conf. on Inf. Network. (ICOIN)*, Jan. 2020, pp. 690–695.

[14] A. Masaracchia *et al.*, "The concept of time sharing NOMA into UAV-Enabled communications: An energy-efficient approach," in *2020 4th Int. Conf. on Recent Advances in Signal Processing, Telecommunications & Comput. (SigTelCom)*, Aug. 2020, pp. 61–65.

[15] U. Challita, W. Saad, and C. Bettstetter, "Deep reinforcement learning for interference-aware path planning of cellular-connected UAVs," in *Proc. 2018 IEEE Int. Commun. Conf. (ICC)*. IEEE, Jul. 2018, pp. 1–7.

[16] Y. Sadi, S. C. Ergen, and P. Park, "Minimum energy data transmission for wireless networked control systems," *IEEE Trans. on Wireless Commun.*, vol. 13, no. 4, pp. 2163–2175, Feb. 2014.

[17] S. Zhang *et al.*, "Joint trajectory and power optimization for UAV relay networks," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 161–164, Oct. 2017.

[18] G. E. G. Padilla, K.-J. Kim, S.-H. Park, and K.-H. Yu, "Flight path planning of solar-powered UAV for sustainable communication relay," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6772–6779, Aug. 2020.

[19] M. M. Selim *et al.*, "On the outage probability and power control of D2D underlaying NOMA UAV-assisted networks," *IEEE Access*, vol. 7, pp. 16 525–16 536, Jan. 2019.

[20] H. Zhou *et al.*, "Real-time video streaming and control of cellular-connected UAV system: Prototype and performance evaluation," *IEEE Wireless Communications Letters*, pp. 1–1, 2021.

[21] X. Xiao *et al.*, "Sensor-augmented neural adaptive bitrate video streaming on UAVs," *IEEE Trans. on Multimedia*, vol. 22, no. 6, pp. 1567–1576, June 2020.

[22] C. He, Z. Xie, and C. Tian, "A QoE-Oriented uplink allocation for Multi-UAV video streaming," *Sensors*, vol. 19, no. 15, p. 3394, 2019.

[23] Y. Chen, H. Zhang, and Y. Hu, "Optimal power and bandwidth allocation for multiuser video streaming in UAV relay networks," *IEEE Trans. on Veh. Tech.*, vol. 69, no. 6, pp. 6644–6655, 2020.

[24] K. Govil, M. L. Welch, J. T. Ball, and C. R. Pennypacker, "Preliminary results from a wildfire detection system using deep learning on remote camera images," *Remote Sensing*, vol. 12, no. 1, p. 166, 2020.

[25] N. Jiang, Y. Deng, A. Nallanathan, and J. A. Chambers, "Reinforcement learning for real-time optimization in NB-IoT networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1424–1440, Jun. 2019.

[26] N. Jiang, Y. Deng, A. Nallanathan, and J. Yuan, "A decoupled learning strategy for massive access optimization in cellular IoT networks," *IEEE J. on Sel. Areas in Commun.*, vol. 39, no. 3, pp. 668–685, 2021.

[27] Y. He, N. Zhao, and H. Yin, "Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach," *IEEE Trans. on Veh. Tech.*, vol. 67, no. 1, pp. 44–55, 2017.

[28] Y. He *et al.*, "Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks," *IEEE Trans. on Veh. Tech.*, vol. 66, no. 11, pp. 10 433–10 445, 2017.

[29] S. Yang *et al.*, "Energy efficiency optimization for UAV-assisted backscatter communications," *IEEE Commun. Lett.*, vol. 23, no. 11, pp. 2041–2045, 2019.

[30] D. Fan *et al.*, "Channel estimation and self-positioning for UAV swarm," *IEEE Trans. on Commun.*, vol. 67, no. 11, pp. 7994–8007, 2019.

[31] J. J. Podur, D. L. Martell, and D. Stanford, "A compound poisson model for the annual area burned by forest fires in the province of ontario," *Environmetrics*, vol. 21, no. 5, pp. 457–469, 2010.

[32] M. Val Martin, R. Kahn, and M. Tosca, "A global analysis of wildfire smoke injection heights derived from space-based multi-angle imaging," *Remote Sensing*, vol. 10, no. 10, p. 1609, Oct. 2018.

[33] "Drones: how to fly them safely and legally," Sep 2017. [Online]. Available: https://www.gov.uk/government/news/drones-are-you-flying-yours-safely-and-legally

[34] "Study on enhanced lte support for aerial vehicles," 3GPP, TR 36.777, Dec. 2017, V15.0.0.

[35] G. D. Durgin, *Space-time wireless channels*. Prentice Hall Professional, 2003.

[36] N. Goddemeier and C. Wietfeld, "Investigation of air-to-air channel characteristics and a UAV specific extension to the rice model," in *2015 IEEE Globecom Workshops (GC Wkshps)*, Dec. 2015, pp. 1–5.

[37] V. Yajnanarayana *et al.*, "Interference mitigation methods for unmanned aerial vehicles served by cellular networks," in *2018 IEEE 5G World Forum (5GWF)*, Jul. 2018, pp. 118–122.

[38] "Recommended upload encoding settings - youtube help." [Online]. Available: https://support.google.com/youtube/answer/1722171?hl=en-G

[39] P. Carballeira, J. Cabrera, A. Ortega, F. Jaureguizar, and N. García, "A framework for the analysis and optimization of encoding latency for multiview video," *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 5, pp. 583–596, Sept. 2012.

[40] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over HTTP," in *Proc. 2015 ACM Conf. on Special Interest Group on Data Commun.*, Aug. 2015, p. 325–338.

[41] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proc. Conf. of the ACM Special Interest Group on Data Commun.*, Aug. 2017, pp. 197–210.

[42] Y. Zeng, X. Xu, S. Jin, and R. Zhang, "Simultaneous navigation and radio mapping for cellular-connected UAV with deep reinforcement learning," *IEEE Trans. on Wireless Commun.*, vol. 20, no. 7, pp. 4205–4220, 2021.

[43] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, Feb. 2015.

[44] Z. Zhang *et al.*, "QoE aware transcoding for live streaming in SDN-Based Cloud-Aided HetNets: An actor-critic approach," in *Proc. 2019 IEEE Int. Commun. Conf. Workshops (ICC Workshops)*, May 2019, pp. 1–6.

[45] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.

[46] C. She *et al.*, "Ultra-reliable and low-latency communications in unmanned aerial vehicle communication systems," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3768–3781, May 2019.

[47] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *2014 IEEE Global Commun. Conf.*, Dec. 2014, pp. 2898–2904.

[48] T. Jaakkola, M. I. Jordan, and S. P. Singh, "On the convergence of stochastic iterative dynamic programming algorithms," *Neural computation*, vol. 6, no. 6, pp. 1185–1201, 1994.

[49] J. Hu, H. Zhang, and L. Song, "Reinforcement learning for decentralized trajectory design in cellular UAV networks with sense-and-send protocol," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6177–6189, 2018.