Optimizing AoI in UAV-RIS Assisted IoT Networks: Off Policy vs. On Policy

Michelle Sherman, *Student Member, IEEE*, Sihua Shao, *Member, IEEE*, Xiang Sun, *Member, IEEE*, and Jun Zheng, *Member, IEEE*

***Abstract*—In urban environments, tall buildings or structures can pose limits on the direct channel link between a base station (BS) and an Internet-of-Thing device (IoTD) for wireless communication. Unmanned aerial vehicles (UAVs) with a mounted reconfigurable intelligent surface (RIS), denoted as UAV-RIS, have been introduced in recent works to enhance the system throughput capacity by acting as a relay node between the BS and the IoTDs in wireless access networks. Uncoordinated UAVs or RIS phase shift elements will make unnecessary adjustments that can significantly impact the signal transmission to IoTDs in the area. The concept of age of information (AoI) is proposed in wireless network research to categorize the freshness of the received update message. To minimize the average sum of AoI (ASoA) in the network, two model-free deep reinforcement learning (DRL) approaches – Off-Policy Deep Q-Network (DQN) and On-Policy Proximal Policy Optimization (PPO) – are developed to solve the problem by jointly optimizing the RIS phase shift, the location of the UAV-RIS, and the IoTD transmission scheduling for large-scale IoT wireless networks. Analysis of loss functions and extensive simulations is performed to compare the stability and convergence performance of the two algorithms. The results reveal the superiority of the On-Policy approach, PPO, over the Off-Policy approach, DQN, in terms of stability, convergence speed, and under diverse environment settings.**

***Index Terms*—UAV-RIS, large-scale wireless networks, AoI, On-Policy, PPO, Off-Policy, DQN.**

I. Introduction

THROUGH technology advances, Internet of Things (IoT) has become a part of our everyday lives by providing seamless connectivities to IoT devices, such as smart microgrid sensors, smart home security systems, wearable health monitors, remote car locks, smart phones, routers, smart watches, etc. Every year, there is growing traffic that is generated by IoT devices, which leads to new requirements of optimizing the IoT network to improve the network capacity, driving businesses, companies, academia, and others to evolve and re-invent current operations and processes to make them more efficient.

This work was supported by the National Science Foundation (NSF) under Grant OIA-1757207.

M. Sherman and S. Shao are with the Department of Electrical Engineering, New Mexico Tech, Socorro, NM 87801 USA (e-mail: michelle.sherman@student.nmt.edu; sihua.shao@nmt.edu).

X. Sun is with the Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87131 USA (e-mail: sunxiang@unm.edu).

J. Zheng is with the Department of Computer Science & Engineering, New Mexico Tech, Socorro, NM 87801 USA (e-mail: jun.zheng@nmt.edu).

Capabilities of IoT applications have evolved from connected devices to smart devices, and now are expanding through the use of artificial intelligence (AI) and machine learning (ML) [1], [2]. AI or ML can be used to forecast future shifts, analyze vast amounts of data using complex algorithm designs (such as identifying security threats on corporate networks and managing other IoTs or security of smart homes), smart manufacturing, precision agriculture, etc. With adaptive IoTDs, and AI and ML techniques integrated to the core of IoT architecture, the practical deployment of such a smart autonomous network becomes a vital research topic of focus in the scientific and industrial communities.

According to the November 2021 Ericsson Mobility Report [3], 5G (fifth-generation wireless mobile network) is projected to support 62% of total mobile data traffic by 2027. What does this mean for IoTs? 5G is opening up new opportunities and applications of IoTs requiring higher network capacity, higher data rates, more connectivities, and lower latency, such as smart microgrids, virtual reality, high-definition streaming, transportation, and mission-critical communications. However, 5G faces a challenge to meet the data rate and latency requirements of IoT applications. One of the potential solutions to solve this challenge is to use high frequency bands, such as mmWave. However, high frequency signals are limited by line-of-sight (LOS) since they can be easily blocked by obstacles, such as tall buildings and geological features. As such, mmWave is better suited for short-range communications to mitigate the probability of obstruction.

Massive MIMO systems have been a core component to expand the coverages of 5G and mmWave [4] using high-directional beamforming antenna arrays at a base station (BS) and at the IoTDs. However, to achieve the required antenna gains, signal directivity, and large coverage distances, several antenna elements may be required at the BS or the IoTDs to battle the imposing factor of path loss and interference. Recently, reconfigurable intelligent surface (RIS) devices have emerged as a cost-effective, energy efficient, and promising technology for the future generations of wireless networks [5], [6]. An RIS is a light weight device consisting of multiple programmable reflectors that offer the capability of passively (or actively depending on the design) tuning the phase of each on-board element to enhance and alter the propagating direction of an incident signal, without the need for several RF chains at the transmitter end or signal processing onboard the UAV-RIS unit. A feedback link between a ground station and the RIS controller (such as an FPGA or microcontroller) can be established to form the optimal beamforming vector to

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3246925

improve the quality of service (QoS) of the IoTDs. Although RIS devices require low power to operate (i.e. 1W for an 1100-element RIS in [7]), the power consumption could be optimized for low power applications.

Depending on the environment, direct channel links from the BS to the IoTDs may not be feasible. Static RIS devices placed on rooftops or the side of buildings have limitations when considering dynamic environments and the mobility of IoTDs. Recent works have shown interest in incorporating unmanned aerial vehicles (UAVs) with mounted RIS devices, denoted as a UAV-RIS station, to increase the flexibility of the system [8]. Fig. 1 demonstrates the system architecture, where an access network is established to enable a BS to transmit/receive the network traffic to/from the IoTDs. The integration of UAVs and RIS devices into wireless networks comes with many challenges, such as UAV trajectory control, scheduling, phase shift optimization, transmission or network delays, resource allocation, channel capacity, LOS and non-LOS (NLOS) cases, and dense communication scenarios. The recent research to resolve these challenges are reviewed in Section II with the objective of optimizing the UAVs 3D location, RIS phase shifts, and/or the BS transmit vector, to achieve secure beamforming or to maximize the throughput/signal capacity for the IoTDs.

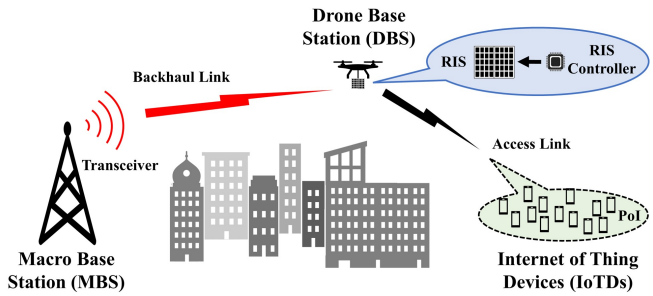

Fig. 1: UAV-RIS assisted wireless cellular architecture.

Conventional methods to optimize the problem formulation in wireless networks rely on offline optimization theories such as convex optimization, greedy algorithms, dynamical programming, and game theory. However, these offline optimization tools suffer from low robustness, high overhead [9], and typically require immense iterations to reach a global optimum. The orientation of the RIS, building obstacles, the mobility of IoTDs, and other dynamics and uncertainties further complicate the problem.

Typically, it is highly desirable for network optimization control mechanisms to not solely depend on a prior model design to develop the optimal policy. Instead, the optimal decision policy should be developed by learning the model, namely following the "model-free" decision-making process. Through interacting with the environment, deep reinforcement learning (DRL) algorithms have been developed as a promising solution to autonomously solve complex, non-convex, and model-free optimization problems, especially in communications and networking [10]. In addition, with the use of neural networks (NN), DRL has the capability of converging rapidly to optimal solutions and improving the robustness. Different from the traditional RL algorithms, which can only handle low-dimensional optimization problems, DRL can observe continuous variables from the environment and has the capability of continuous control in the decision-making.

Even though DRL has been demonstrated to have great potential in optimizing wireless networks, proper design considerations need to be taken into account when developing the NN architecture and the RL framework. To better understand the optimal learning algorithm design strategy for the next generation UAV-RIS assisted IoT infrastructure, we develop two DRL-based algorithms to minimize the average sum of age of information (AoI) of the network. AoI is used as a performance metric for the freshness of the information received at an IoTD. The average sum of AoI (ASoA) is minimized by jointly optimizing the RIS phase shifts, UAV-RIS location, and IoTD transmission scheduling. We find that the optimal RIS phase shifts for transmission to an IoTD is largely dependent on the wave vector of the device, which comprises appropriate elevation and azimuth angles of arrival and departure between the BS to the UAV-RIS and from the UAV-RIS to the scheduled IoTD. The action space framework of the DRL algorithms is designed to accommodate the discussed large-scale network through careful consideration of the decision options. We use two state-of-the-art DRL algorithms – Off-Policy Deep Q-Network (DQN) and On-Policy Proximal Policy Optimization (PPO) in order to combat the high-dimensional input data introduced by the scenario that is too difficult for conventional optimization methods to solve. Through analysis of the performance evaluation functions used to update the policy for each algorithm, and by executing extensive simulations, we compare the design framework and the performance of the Off-Policy and On-Policy based methods in terms of stability, convergence speed, and under different dynamic environments.

Contributions: In summary, the three key contributions of this work are as follows:

- A UAV-RIS-assisted 3D urban IoT network is established while considering the effect of building blockages.
- A generalized RIS phase shift model is proposed to optimize the signal-to-noise ratio (SNR) at the IoTDs based on arbitrary RIS orientation and the number of reflecting elements.
- Two model-free DRL based frameworks for UAV-RIS assisted large-scale IoT networks, and performance analysis based on loss functions and NN updating process.
- Extensive simulations to validate the analysis of the Off-Policy DQN vs. On-Policy PPO based framework in terms of stability, convergence speed, and performance under different dynamic environments.

Organization: The remainder of this work is organized as follows. Section II presents related work. Section III presents the system model. Section IV describes the designed DRL framework and a discussion of the Off-Policy DQN and On-Policy PPO algorithm frameworks. In Section V, the simulation results are presented and evaluated in terms of the mentioned performance metrics. Section VI concludes and summarizes the key observations from this work.

II. Related Work

Minimizing AoI has been widely studied in single hop and multiple access wireless networks [11]–[17]. More recently, the authors in [18]–[20] considered minimizing the AoI in UAV-assisted wireless networks through optimal UAV trajectory planning and time allocation scheduling using a DQN and incorporated artificial neural networks (ANN), optimization algorithms, dynamic programming and genetic algorithms, respectively. The authors in [21] utilized a DQN based strategy to maximize the "confidence level" of the AoI through optimal trajectory planning of vehicular networks. The work in [22] adapts a lifelong RL (LLRL) approach to enable the UAV to jointly optimize the IoTDs AoI and energy consumption, while considering the dynamic changes in the environment by accumulating history knowledge to make informed decisions in future processes. Other optimization algorithms have been proposed in [23] and [24] to jointly optimize the UAV's trajectory, it's energy allocation and scheduling for delivering status updates, as well as monitoring the number of update instances.

To fully take advantage of the benefits and communication enhancements that RIS devices can offer, researchers have investigated fundamental challenges arising in the channel models [25], such as optimal active/passive beamforming design, energy harvesting, resource allocation, channel state information (CSI) estimation at the RIS, BS, and IoTDs, reflection optimization, and deployment scheme. In [26], the power allocation at the BS and the beamforming vector at the RIS are optimized to maximize the secrecy rate rate in a RIS-assisted physical-layer service integration system using optimization algorithms. In [27], a DRL approach is developed to achieve secure beamforming at the BS and RIS against eavedroppers in a dynamic system. The work in [28] uses DRL to maximize the sum rate of multiuser downlink MISO systems by jointly optimizing the beamforming vectors at the BS and RIS. The works in [26]–[28] incorporate static RIS devices in the system model by assuming they are mounted on a wall or the facade of a building. However, static RIS systems have limited capabilities since they cannot adapt to IoT foot traffic or obstacles caused by buildings or walls.

Considering UAV-RIS assisted wireless networks, [29] maximizes the secrecy rate for secure communication by iteratively optimizing the UAV-RIS's 3D location and the phase shifts of the RIS. The work in [30] considers a MISO-NOMA two-user downlink system to maximize the data rate of the "strong" user, while meeting QoS requirements of the "weak" user, by iteratively optimizing the UAV-RIS horizontal location, the phase shifts of the RIS, and the BS beamforming vector. In [31], Q-learning with NNs is used to maximize the downlink transmission capacity for a single IoTD (whose route is given by a MDP process) by optimizing the UAV-RIS location (such that the CSI can be realized) and reflection coefficients of the RIS. The work in [32] considers the same deployment, but instead uses distributed RL to maximize long-term downlink communication in the multi-user case by optimizing the UAV-RIS location, the precoding matrix at the BS, and the reflection coefficients of the RIS. It is shown that the integration of the aerial UAV-RIS with DRL increased the communication performance by 50% and 25% compared to static RIS and non-learning UAV-RIS deployments, respectively.

Considering the aforementioned research, minimizing the AoI in large-scale (or large network capacity) UAV-RIS assisted wireless networks has not been well-explored. Recently, the authors in [33] minimizes the AoI over a two-minute period by optimizing the RIS phase shifts, the UAV-RIS altitude, and the IoTD transmission scheduling. The design frameworks of the DRL algorithms in [27] and [33] consist of action spaces whose size is linearly proportional to the number of IoTDs in the network. Dense urban areas can consist of tens, hundreds, or thousands of connected devices on a single network. According to [27] and [33], such a dense network will result in a massive action space, causing a deterioration in the convergence performance of DRL algorithms.

Motivated by developing a generalized system model to serve a large-capacity network, we investigate the problem of minimizing the ASoA in the UAV-RIS assisted network by optimizing the UAV-RIS location, while taking into account obstructions (located between the BS and the UAV-RIS and between the UAV-RIS and IoTDs), and the transmission IoTD scheduling process. Owing to the stochastic nature of the arrival process of the data packets and the IoTD locations, we develop an Off-Policy DRL algorithm and an On-Policy DRL algorithm to solve the minimization problem by learning an optimal decision-making policy. Opposed to [32], our work delivers a comprehensive comparative study using analytical and numerical methods, as well as evaluations in a variety of different environmental settings. New contributions that need to be highlighted include a low-complexity determination process of optimal phase shifts considering full phase variation, scalable action space design, comparative analysis of neural network updating and loss functions, and thorough and realistic environmental settings, such as multi-user scheduling and urban building blockages.

III. System Models

A. Scenario

This work considers a UAV-RIS assisted wireless communication system that serves up to K IoTDs distributed in an area with the size of $X \times Y$ km^2. IoTDs are clustered into K_M groups of varying size based on the K-Means Clustering algorithm. As shown in Fig. 1, we consider the scenario that the links between the IoTDs and the local BS are NLOS owing to obstacles, such as mountains and buildings, located in-between. In this case, a single UAV with an RIS mount is deployed near the edge of the area to establish a virtual link from the BS and provide wireless connectivity to the IoTDs. At every time-step t, the UAV-RIS will adjust in altitude, the phase of the RIS elements will shift, and finally, a cluster of IoTD(s) is scheduled for data packet transmission for time t. These parameters are adjusted in order to improve the collective AoI of the IoTDs and their received SNR. The total service time is given by T and is divided equally into time-steps $t \in [0, T]$ of length t_S. It is assumed that the only non-payload information exchanged in the uplink from the

TABLE I
ENVIRONMENT PARAMETER NOTATIONS

Notation	Definition	Notation	Definition
K	Number of IoTDs	N_c	Number of Clusters
$X \times Y$	Grid Size $(m \times m)$	$K_1,\ K_2$	Rician K-Factors
(x_k, y_k)	IoTD Coordinates (m)	d_H, d_V	Spacing of RIS elements in horizontal and vertical direction
(x, y)	UAV-RIS Coordinates (m)	F, M_A	Number of Antenna Elements on the RIS, BS
$d_{U,B}$	Euclidean distance between the UAV-RIS and the BS (m)	$\mathbf{H}_{BS}$	Channel Coeff. matrix between the UAV-RIS and the BS
$d_{k,U}$	Euclidean distance between IoTD k and the UAV-RIS (m)	$\mathbf{h}_{k,RIS}$	Channel Coeff. matrix between IoTD k and the RIS elements
(x_m, y_m)	BS Coordinates (m)	$\eta_{U,B}$	Pathloss in ATA Link
$h_{UAV}(t)$	UAV-RIS Height at time t (m)	$\eta_{k,U}^{LOS}$, $\eta_{k,U}^{NLOS}$	Pathloss in ATG Link to IoTD k
h_{BS}	Height of BS (m)	$\eta_{LOS},\ \eta_{NLOS}$	Mean Additional Path loss in the ATG Link (dB)
$\theta_{k,U}$, $\xi_{k,U}$	Elevation and azimuth angles between IoTD k and the UAV-RIS (rad)	$\mathbf{a}(\theta, \xi, N)$	Array response vector
$\theta_{U,B}$, $\xi_{U,B}$	Elevation and azimuth angles between the UAV-RIS and the BS (rad)	$\mathbf{k}(\theta, \xi)$	Wave vector
$[H_{U,min}, H_{U,max}]$	Altitude Range of UAV-RIS (m)	P_T	Transmission Power of IoTDs
V_{Max}	Velocity of UAV-RIS (m/s)	$\sigma^2_{N_0}$	Noise Variance
α_B, β_0	Pathloss Parameters in the ATA Link	β_i	Amplitude coefficient of RIS element $i \in [1, F]$
f_c, λ	Carrier Frequency (Hz) and wavelength (m)	ϕ_i	Phase shift of RIS element $i \in [1, F]$
c	Speed of Light (m/s)	$\gamma_k(t)$	Received SNR at IoTD k at time t
g_0	Antenna gain of RIS elements	t_S	Time-step (sec.)
t_0	Initial time t	T	Time duration of episode (sec.)

IoTD to the UAV-RIS to the BS is CSI. There is a separate node called the ground control station (GCS) that interacts with the UAV-RIS and captures spatial/activation information from the IoTDs. It is responsible for collecting and monitoring the UAV's location, location and activation of the IoTDs, scheduling the IoTD(s) for transmission in the corresponding scheduled cluster, and control of the RIS phase shifts.

The locations of the IoTDs are given by the horizontal coordinates (x_k, y_k) (m), where $k \in \{1, ..., K\}$. The horizontal location of the UAV-RIS is fixed and denoted by (x, y) (m). The location of the BS is static and given by (x_m, y_m) (m). The height of the UAV-RIS at time t is denoted as $h_{UAV}(t)$ (m) and the height of the BS is denoted by h_{BS} (m). The 3D Euclidean distance, elevation angle, and azimuth angle formed between the IoTDs and the UAV-RIS are denoted as $d_{k,U}(t)$ (m), $\theta_{k,U}(t)$ (rad), and $\xi_{k,U}(t)$ (rad), respectively. Similarly, the 3D Euclidean distance, elevation angle, and azimuth angle formed between the UAV-RIS and the BS are denoted as $d_{U,B}(t)$ (m), $\theta_{U,B}(t)$ (rad), and $\xi_{U,B}(t)$ (rad), respectively.

At any time-step t, the UAV-RIS has the capability of tuning its height, however, it must not drop below or exceed the range $h_{UAV}(t) \in [H_{U,min}, H_{U,max}]$. The UAV-RIS has a fixed maximum velocity of V_{Max} (m/s) to transition between altitudes. Thus, in each time-step, the UAV-RIS can travel at most $V_{max} \times t_S$ meters. Utilizing a small displacement allows for finer control of the UAV-RIS altitude adjustment in finding an optimal height.

B. Path loss Model

In this section, the channel coefficients for the backhaul and access links from the BS to the UAV-RIS and from the UAV-RIS to the IoTDs, respectively, are modelled according to the Rician fading assumption [34].

ATA Link: Assuming the air-to-air (ATA) link between the BS and UAV-RIS are LOS, then the average path loss between the BS and UAV-RIS is given by [35]:

$$\eta_{U,B} = \beta_0 d_{U,B}^{-\alpha_B} \text{ (dB)},$$

where $\alpha_B \geq 2$ is the path loss exponent and β_0 is the channel's power gain at reference distance $d_0 = 1$m.

ATG Links: The access link between the UAV-RIS and an IoTD follow air-to-ground (ATG) communication channel properties. Based on the free-space path loss model, i.e., $FSPL_{k,U} = 20 \log_{10}\left(\frac{4\pi f_c d_{k,U}}{c}\right)$, the path loss of an ATG link between the UAV-RIS and an IoTD in LOS and NLOS can be described as [36]:

$$\eta_{k,U}^{LOS} = FSPL_{k,U} + \eta_{LOS} \text{ (dB), and}$$
$$\eta_{k,U}^{NLOS} = FSPL_{k,U} + \eta_{NLOS} \text{ (dB)},$$

where f_c is the carrier frequency (Hz), c is the speed of light (m/s), and η_{LOS} and η_{NLOS} account for the mean additional path loss (dB) in the LOS and NLOS scenarios, respectively, which are determined by the environment.

C. Medium Access Control and Channel Model

The uplink of the wireless communication system is considered to operate in Time-Division-Multiple-Access (TDMA) mode where at each time-step, exactly one IoTD cluster is scheduled to transmit data to the UAV-RIS, i.e., one transmission per time-step. The BS is equipped with a phased array with M_A antenna elements to transmit data to the UAV-RIS, which then relays the data to the scheduled IoTD. The RIS has F reflecting elements spaced a distance of $d_H = \lambda/2$ and $d_V = \lambda/2$ in the horizontal and vertical directions, respectively, where λ (m) is the wavelength of the carrier wave.

Let $\mathbf{H}_{BS} \in \mathbb{C}^{F \times M_A}$ denote the channel coefficient matrix between the the radiating elements at the BS and the reflecting elements of the RIS. In general, the channel coefficient matrix for $\mathbf{H}_{BS}$ consists of the LOS and NLOS components defined by the following equations [35], [37], [38]:

$$\mathbf{H}_{BS}(t) = \sqrt{\eta_{U,B}} \left(\sqrt{\overline{\rho}_1} \times \overline{\mathbf{H}}_{BS}(t) + \sqrt{\widetilde{\rho}_1} \times \widetilde{\mathbf{H}}_{BS}(t) \right) \quad (1)$$

$$\overline{\mathbf{H}}_{BS}(t) = \mathbf{a}_{rx,BS}\left(\theta_{U,B}^{AoA}, \xi_{U,B}^{AoA}, F\right) \mathbf{a}_{tx,BS}^{H}\left(\theta_{U,B}^{AoD}, \xi_{U,B}^{AoD}, M_A\right)$$

$$\widetilde{\mathbf{H}}_{BS}(t) \sim \mathcal{CN}(0,1)$$

where $\overline{\rho}_1 = \frac{K_1}{1+K_1}$, $\widetilde{\rho}_1 = \frac{1}{1+K_1}$, K_1 is the Rician K-factor, and the vectors $\mathbf{a}_{rx,BS}$ and $\mathbf{a}_{tx,BS}$ are the receive and transmit array response vectors, which depend on the corresponding elevation and azimuth angles of arrival (AoA) and angles of departure (AoD), respectively, shown in Fig. 2. The elements in $\widetilde{\mathbf{H}}_{BS}(t)$ are i.i.d complex Gaussian random variables with zero mean and unit variance.

For an N-element array of dimensions $N_H \times N_V = N$, the array response vector is given by [39]:

$$\mathbf{a}(\theta, \xi, N) = \left[e^{j\mathbf{k}(\theta,\xi)\mathbf{u}_1} \quad \dots \quad e^{j\mathbf{k}(\theta,\xi)\mathbf{u}_N} \right] \quad (2)$$

where

$$\mathbf{k}(\theta, \xi) = \frac{2\pi}{\lambda} \left[\cos(\theta)\cos(\xi), \quad \cos(\theta)\sin(\xi), \quad \sin(\theta) \right]$$

is the wave vector, which indicates the phase variation of a plane wave w.r.t. the spherical coordinates θ, $\xi \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, and

$$\mathbf{u}_n = \begin{bmatrix} 0 \\ \text{mod}\left((n-1), N_H\right) d_H \\ \lfloor (n-1)/N_V \rfloor d_V \end{bmatrix}, \; n \in [1, N]$$

denotes the location of the nth antenna element. Similarly, let $\mathbf{h}_{k,RIS} \in \mathbb{C}^{1 \times F}$ denote the channel channel coefficient between the reflecting elements of the RIS and IoTD k. The channel coefficient matrix for $\mathbf{h}_{k,RIS}$ is given by the following equations [35], [37], [38]:

$$\mathbf{h}_{k,RIS}(t) = \sqrt{\overline{\rho}_2 \eta_{k,U}^{LOS}} \times \overline{\mathbf{h}}_{k,RIS}(t) + \sqrt{\widetilde{\rho}_2 \eta_{k,U}^{NLOS}} \times \widetilde{\mathbf{h}}_{k,RIS}(t) \quad (3)$$

$$\overline{\mathbf{h}}_{k,RIS}(t) = \mathbf{a}_{rx,RIS}\left(\theta_{k,U}^{AoA}, \xi_{k,U}^{AoA}, 1\right) \mathbf{a}_{tx,RIS}^{H}\left(\theta_{k,U}^{AoD}, \xi_{k,U}^{AoD}, F\right)$$
$$= \mathbf{a}_{tx,RIS}^{H}\left(\theta_{k,U}^{AoD}, \xi_{k,U}^{AoD}, F\right)$$
$$\widetilde{\mathbf{h}}_{k,RIS}(t) \sim \mathcal{CN}(0,1)$$

where $\overline{\rho}_2 = \frac{K_2}{1+K_2}$, $\widetilde{\rho}_2 = \frac{1}{1+K_2}$, K_2 is the Rician K-factor, and the vectors $\mathbf{a}_{rx,RIS}$ and $\mathbf{a}_{tx,RIS}$ are the receive and transmit array response vectors as defined in (2) with corresponding AoA and AoD angles as illustrated in Fig. 2. The elements of $\widetilde{\mathbf{h}}_{k,RIS}(t)$ are also i.i.d complex Gaussian random variables with zero mean and unit variance. The solid black arrows in Fig. 2 indicate the facing direction of the corresponding arrays.

D. Optimal Phase Shifts and SNR

The received Signal-to-Noise-Ratio (dB) is given by [37]:

$$\gamma_k(t) = \frac{P_T \times \mathbb{E}\left[\|\mathbf{h}_{k,RIS}(t)\mathbf{\Phi}\mathbf{H}_{BS}(t)\|^2 \right]}{\sigma_{N_0}^2} \quad (4)$$

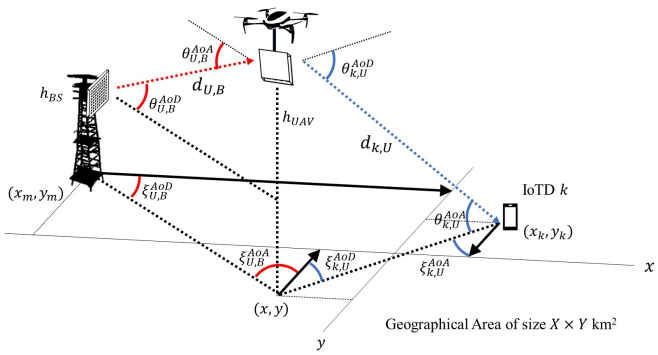

Fig. 2: Channel wave vector architecture.

where P_T (dBm) is the transmission power of the IoTDs, $\sigma_{N_0}^2$ is the noise variance (dBm), $\mathbb{E}(\cdot)$ denotes the expectation operator, which is necessary due to the stochastic behavior of NLOS channel coefficient matrices $\tilde{\boldsymbol{H}}_{BS}(t)$ and $\tilde{\boldsymbol{h}}_{k,RIS}(t)$. $\mathbf{\Phi} = g_0 \cdot \text{diag}\left(\left[\beta_1 e^{j\phi_1}, \cdots, \beta_F e^{j\phi_F}\right]\right)$ is the RIS reflection matrix, where g_0 (dB) denotes the RIS element gain, $\beta_i \in [0,1]$ denote the amplitude coefficients, and $\phi_i \in [0, 2\pi]$ denote the induced phase shifts, where $i \in [1, F]$. We can see from (4) that an IoTD is able to achieve a higher SNR by properly tuning the RIS reflection matrix. To form the phase optimization problem, we provide an upper bound for maximizing the SNR in the following proposition [37].

Proposition 1: The received SNR for the scheduled IoTD k at time t in the UAV-RIS assisted wireless network (assuming k is active at time t) has the upper bound:

$$\gamma_k(t) \leq \gamma_k^{ub}(t) = \frac{P_T \eta_{U,B}}{\sigma_{N_0}^2} \left(A^2 \|\overline{\mathbf{h}}_{k,RIS}(t)\mathbf{\Phi}\overline{\mathbf{H}}_{BS}(t)\|^2 + C \right) \quad (5)$$

which is maximized under phase shift vector:

$$\boldsymbol{\phi} = [\phi_1, \cdots, \phi_F] = \left[\mathbf{k}(\theta_{k,U}^{AoD}, \xi_{k,U}^{AoD}) - \mathbf{k}(\theta_{U,B}^{AoA}, \xi_{U,B}^{AoA}) \right] \mathbf{U} \quad (6)$$

where $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_F]$.

First, we break down $\|\mathbf{h}_{k,RIS}(t)\mathbf{\Phi}\mathbf{H}_{BS}(t)\|^2$ into

$$\|\mathbf{h}_{k,RIS}(t)\mathbf{\Phi}\mathbf{H}_{BS}(t)\|^2 = \eta_{U,B} \|\boldsymbol{z_1} + \boldsymbol{z_2} + \boldsymbol{z_3} + \boldsymbol{z_4}\|^2$$

where

$$\boldsymbol{z_1} = \sqrt{\overline{\rho}_1 \overline{\rho}_2 \eta_{k,U}^{LOS}} \left(\overline{\mathbf{h}}_{k,RIS}(t)\mathbf{\Phi}\overline{\mathbf{H}}_{BS}(t) \right),$$
$$\boldsymbol{z_2} = \sqrt{\overline{\rho}_1 \widetilde{\rho}_2 \eta_{k,U}^{NLOS}} \left(\widetilde{\mathbf{h}}_{k,RIS}(t)\mathbf{\Phi}\overline{\mathbf{H}}_{BS}(t) \right),$$
$$\boldsymbol{z_3} = \sqrt{\widetilde{\rho}_1 \overline{\rho}_2 \eta_{k,U}^{LOS}} \left(\overline{\mathbf{h}}_{k,RIS}(t)\mathbf{\Phi}\widetilde{\mathbf{H}}_{BS}(t) \right), \text{ and}$$
$$\boldsymbol{z_4} = \sqrt{\widetilde{\rho}_1 \widetilde{\rho}_2 \eta_{k,U}^{NLOS}} \left(\widetilde{\mathbf{h}}_{k,RIS}(t)\mathbf{\Phi}\widetilde{\mathbf{H}}_{BS}(t) \right).$$

Let $A = \sqrt{\overline{\rho}_1 \overline{\rho}_2 \eta_{k,U}^{LOS}}$. Given that $\widetilde{\mathbf{H}}_{BS}(t)$ and $\widetilde{\mathbf{h}}_{k,RIS}(t)$ have zero mean, unit variance, and are i.i.d., then $\mathbb{E}[\boldsymbol{z_i}] = 0$ for

$i = 2, 3, 4$ and

$$\begin{aligned}
&\mathbb{E}\left[\|\boldsymbol{z_1}+\boldsymbol{z_2}+\boldsymbol{z_3}+\boldsymbol{z_4}\|^2\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{M_A}(z_{1,i}+z_{2,i}+z_{3,i}+z_{4,i})^2\right] \\
&= \mathbb{E}\left[\|\boldsymbol{z_1}\|^2+\|\boldsymbol{z_2}\|^2+\|\boldsymbol{z_3}\|^2+\|\boldsymbol{z_4}\|^2\right] \\
&= \|\boldsymbol{z_1}\|^2+\mathbb{E}\left[\sum_{i=1}^{M_A}\left((z_{2,i})^2+(z_{3,i})^2+(z_{4,i})^2\right)\right] \\
&= \|\boldsymbol{z_1}\|^2+C,
\end{aligned}$$

where C is some constant. Based on the definition of $\boldsymbol{z_1}$ and (4), we obtain the upper bound of (5). Since $A, P_T, \eta_{U,B}, \sigma^2_{N_0}$ are all constants, in order to find the optimal phase shifts that maximize the recived SNR, we have:

$$\begin{aligned}
\boldsymbol{\Phi}_{opt} &= \max_{\boldsymbol{\Phi}}\left(\|\overline{\mathbf{h}}_{k,RIS}(t)\boldsymbol{\Phi}\overline{\mathbf{H}}_{BS}(t)\|^2\right) \\
&= \max_{\boldsymbol{\Phi}}\left(\|\mathbf{a}^H_{tx,RIS}\boldsymbol{\Phi}\mathbf{a}_{rx,BS}\mathbf{a}^H_{tx,BS}\|^2\right) \\
&= \max_{\boldsymbol{\Phi}}\left(\|\mathbf{a}^H_{tx,RIS}\boldsymbol{\Phi}\mathbf{a}_{rx,BS}\|^2\cdot\|\mathbf{a}^H_{tx,BS}\|^2\right)
\end{aligned}$$

where $\|\mathbf{a}^H_{tx,BS}\|^2 = M_A$. Expanding $\boldsymbol{\Phi}_{opt}$, we find

$$\boldsymbol{\Phi}_{opt} = \max_{\boldsymbol{\Phi}}\|\sum_{i=1}^{F} e^{j\left(-\mathbf{k}(\theta^{AoD}_{k,U},\xi^{AoD}_{k,U})\mathbf{u}_i+\phi_i+\mathbf{k}(\theta^{AoA}_{U,B},\xi^{AoA}_{U,B})\mathbf{u}_i\right)}\|^2.$$

Thus, we obtain (6). We observe that the optimal phase shifts only depend on the elevation/azimuth angles $(\theta^{AoD}_{k,U}, \xi^{AoD}_{k,U}, \theta^{AoA}_{U,B}, \xi^{AoA}_{U,B})$, and number/spacing of RIS elements, such that the GCS control of phase shifts is independent from the CSI. In order to achieve a reliable signal at the IoTDs, the SNR received by each IoTD must be above the threshold, i.e., $\gamma_k(t) \geq \gamma_{min}$.

IV. Model-Free Deep Reinforcement Learning: Off-Policy (DQN) vs. On-Policy (PPO)

A. DRL Framework

In this work, DQN and PPO based DRL algorithms are developed with the aim of learning the best coordination between the scheduling of the IoTDs for signal transmission and the altitude adjustment of the UAV-RIS. Specific definitions of the DRL elements are described as follows.

1) State Space: One of the advantages of DRL is that we can utilize continuous state spaces. In each time-step, the state $s(t) \in \mathcal{S}$, is given by the vector

$$\begin{aligned}
s(t) = \{&AoI_1(t)\ \ AoI_2(t)\ \ \cdots\ \ AoI_K(t), \\
&\gamma_1(t)\ \ \gamma_2(t)\ \ \cdots\ \ \gamma_K(t),\ \ h_{UAV}(t)\}
\end{aligned} \tag{7}$$

where $AoI_k(t) \in \mathbb{N}$, for $k \in \{1, ..., K\}$, represents the AoI of IoTD k at the beginning of time-step t. The AoI is computed for the whole uplink transmission from IoTD k to the BS. This can be computed by [12]:

$$AoI_k(t+1) = \begin{cases} 1, & \text{if } \gamma_k(t) \geq \gamma_{min},\ I_k(t) = 1, \\ & \text{and } W_k(t) = 1, \\ AoI_k(t)+1, & \text{otherwise}, \end{cases} \tag{8}$$

where $W_k(t) = 1$ means IoTD k is active at time t and $I_k(t) = 1$ implies that IoTD k is scheduled for transmission at time t. If the BS receives a packet update from IoTD k by the end of time-step t, then $AoI_k(t+1) = 1$. On the other hand, if the BS does not receive a packet update from IoTD k by the end of time-step t, then $AoI_k(t+1) = AoI_k(t)+1$ since the information at the IoTD is older by one time-step. In summary, the AoI of IoTD k will increase *iff* the transmission is unsuccessful or if there is no scheduled transmission from IoTD k. Thus, $s(t)$ is a vector of $2K+1$ elements.

2) Action Space: For any DRL algorithm, the size of the action space is a major contributing factor on the algorithm's convergence and stability performance. Here, we evaluate the outcome of the algorithms by designing a fixed action space whose size does not depend on K or K_M. At each time-step, the UAV-RIS will carry out a multi-dimensional action $a(t) \in \mathcal{A}$. One dimension of the action space is given by the UAVs altitude adjustment, that is, let $(\Delta h_{UAV})(t) \in \{V_{max}\cdot t_S,\ 0,\ -V_{max}\cdot t_S\}$ represent the altitude adjustment of the UAV-RIS, i.e., Up, Hover, and Down, respectively.

For the second dimension, to limit the number of available actions, we implement a "Stay or Switch" capability w.r.t. which IoTD cluster will be scheduled. For example, at time t, IoTD cluster k_m is scheduled. At time $t+1$, the agent can choose to schedule the current cluster again, or switch to another cluster. Let $I(t)$ be a binary variable to imply if the UAV-RIS downloaded data from an IoTD cluster that is the same IoTD cluster from the previous time-step (i.e., $I(t) = 0$) or a different IoTD cluster from the previous time-step (i.e., $I(t) = 1$). Here, $I(t) = 1$ implies the UAV-RIS stops downloading data from an IoTD cluster, which transferred data to the UAV-RIS in the previous time-step, and starts downloading data from a new IoTD cluster at current time-step t. Thus, the action, $a(t)$, is defined by $a(t) = \{I(t),\ (\Delta h_{UAV})(t)\}$ and the action space, $\mathcal{A}$, consists of six possible actions.

When $I(t) = 1$, there are two criterias used to determine which IoTD cluster to switch to. Let $k'_m \in [1, K_M]$ be the IoTD cluster the agent served in the previous time-step $t-1$. **Criteria 1** is used to find which IoTD cluster, ν, currently has the highest average AoI:

$$v = \arg\text{ where}\left(\textbf{AoI} == \max\left(\textbf{AoI'}\right)\right),$$

where $\textbf{AoI} = \{A_{k_m}(t)\}_{k_m=1}^{K_M}$, $A_{k_m}(t)$ is the average AoI of the IoTDs in cluster k_m for $k_m \in [1, K]$, and $\textbf{AoI'} = \{A_{k_m}(t)\}$ for $k_m \in [1, K_M]\backslash\{k'_m\}$. **If** *len*$(v) = 1$, then the IoTD cluster to be scheduled in the current time-step is IoTD cluster v. **Else**, i.e., multiple IoTD clusters have the same high average AoI, **Criteria 2** is used to select the IoTD cluster with the highest average SNR among the set of the IoTD clusters with the same average AoI, i.e.,

$$v' = \arg\max_{k_m \in v}\left(\gamma_{k_m}(t)\right).$$

Then the scheduled IoTD cluster will be $n_c = v[v']$ (i.e. $I_{n_c} = 1$ for $n_c \in [1, K_M]$).

For both algorithms, we eliminate invalid actions from the action space when the agent is in a certain state. For example,

TABLE II
DEEP REINFORCEMENT LEARNING PARAMETER NOTATIONS

Notation	Definition	Notation	Definition
$\mathcal{S}, \mathcal{A}$	State and Action Spaces	$s(t), a(t), r(t)$	state, action, and reward at time t
$I_{n_c}(t)$	Indicator if cluster n_c is scheduled to download data from the UAV-RIS	$W_k(t)$	Indicator if IoTD k is active at time t
$AoI_k(t)$	AoI of IoTD k at time t	$AoI_{k_m}(t)$	Total AoI for cluster k_m
$\pi_\theta(a\|s)$	Current Policy	$\pi_{\theta\,old}(a\|s)$	Latest version of the policy
$Q(s,a\|\theta)$	Evaluation Q Network (NN parameter θ)	$Q(s',a\|\hat{\theta})$	Target Q Network (NN parameter $\hat{\theta}$)
T_f	DQN Target Network update frequency	ϵ	ϵ-greedy parameter ($\epsilon \in [0,1]$)
$\mathcal{R}$	Replay Buffer	B	Minibatch size
$Min.Mem.Cap$	Minimum memory capacity of experiences	$\mathcal{D}$	PPO Memory
L	PPO Memory length	$V_\theta(s), V_{\theta_{old}}(s)$	Estimated and current state-values
$\mathbf{S}^{L\times 1}$, $\mathbf{A}^{L\times 1}$, $\mathbf{R}^{L\times 1}$	Collected states, actions, and rewards in $\mathcal{D}$	$\mathbf{S}_i^{B\times 1}$, $\mathbf{A}_i^{B\times 1}$	Collected states and actions in batch i of batch size B
$\mathcal{L}(\theta)$	Loss Function	$MaxIter$	Maximum number of episodes
Q_T	Target Q value	α	NN Learning Rate
γ	Discount Factor	c_1	PPO Loss Coefficient
N_{eps}	Number of PPO Epochs	ϵ_{clip}	PPO Policy Clip

if the UAV-RIS is currently at its minimum height, $H_{U,min}$, or at its maximum height, $H_{U,max}$, then we eliminate the actions involving down and up adjustments of the UAV-RIS's altitude, respectively. This ensures the agent does not prioritize learning the altitude range of the UAV-RIS, but rather focuses on exploring the best scheduling strategy based on the optimal RIS phase shifts.

3) Reward Function: The reward function is computed at the end of each time-step. Since the objective is to minimize the ASoA of all the IoTDs, we then define the immediate reward function as follows.

$$r(t) = -\sum_{k=1}^{K} AoI_k(t). \tag{9}$$

B. Deep Q-Network (DQN) and ϵ-greedy Method

DQN seeks to optimize its approximation to the Q-value from the Bellman equation [40] using an Evaluation Q Network (parameterized by θ and has output $Q(s,a|\theta)$) and a Target Q Network (parameterized by $\hat{\theta}$ and has output $Q(s',a|\hat{\theta})$), where s' denotes the next state, i.e., $s(t+1)$. Fig. 3 shows the basic architecture of DQN. The agent's past experiences with the environment can be stored in memory using an experience replay buffer $(\mathcal{R})$. DQN requires that a series of $Min.Mem.Cap.$ experiences must be collected in memory before learning begins. Experiences are drawn randomly from memory and stored in a minibatch of size B for Off-Policy learning. The Evaluation network is trained and optimized towards the Target network whose weights are updated with frequency T_f. This approach reduces the correlation between successive sample experiences as to not negatively impact the gradients in stochastic gradient descent (SGD). The Target Q-values are updated according to:

$$Q_T = \begin{cases} r, & \text{if } s' \text{ is the end state,} \\ r + \gamma \max\limits_{a' \in \mathcal{A}} Q(s', a'|\hat{\theta}), & \text{otherwise,} \end{cases} \tag{10}$$

where γ is the discount factor.

To perform action selection, we use the *ϵ-greedy* strategy to balance exploration and exploitation defined as:

$$a = \begin{cases} \text{Random action } a \in \mathcal{A}, & \text{if } \ r^* \leq \epsilon, \\ \arg\max\limits_{a' \in \mathcal{A}} \left(Q(s, a'|\theta)\right), & \text{otherwise,} \end{cases} \tag{11}$$

where $r^* \sim \mathcal{U}[0,1)$ and $\epsilon \in [0,1]$ is a parameter.

The pseudo-code of the proposed DQN method is summarized in Algorithm 1. The algorithm first randomly initializes the IoTD and building locations, the IoTD cluster assignments, sets the initial UAV-RIS 3D location, the BS height and

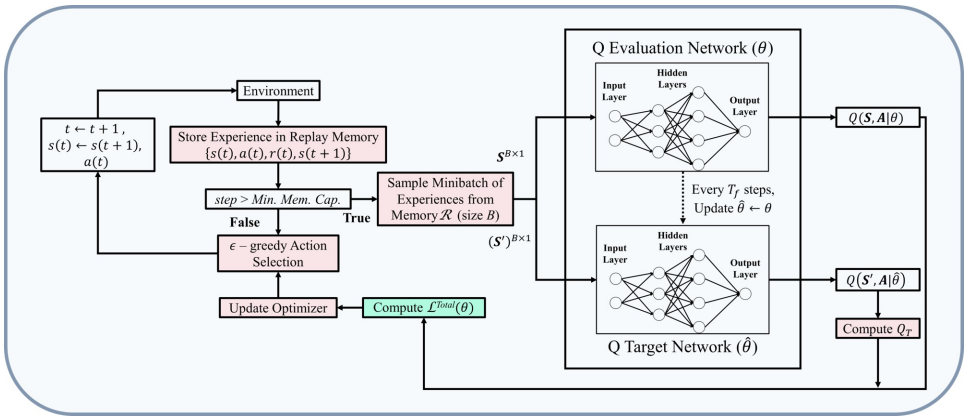

Fig. 3: DQN Architecture.

location, the number of RIS and BS antenna elements, and other environment and hyperparameters parameters listed in Table IIIa and Table IIIb. At the start of each episode, K_{ep}, the UAV-RIS's initial height is reset and the AoI of the IoTDs is reset to zero. The algorithm then starts stepping through the environment where the state $s(t_0)$ holds the initial AoI, and SNR of the IoTDs and the UAV-RIS height location. The algorithm selects a valid action from the action space via the ϵ-greedy method where the action $a(t_0)$ holds an indicator for switch/stay with the current IoTD cluster and hover/ascend/descend the UAV-RIS. Once the action is selected, we compute the current SNR of the IoTDs before the UAV-RIS adjusts its vertical location depending on the selected action. Once the SNRs are computed, the processes outlined in Section IV-A2 are followed for the fixed action space to perform the IoTD cluster scheduling. Once a cluster is selected for transmission, the UAV-RIS height is updated.

After executing the current action, the reward is then computed as the negative of the sum of AoI of all the IoTDs. The transition $\{s(t),\ a(t),\ r(t),\ s(t+1),\ done(t)\}$ is then recorded in the replay buffer memory (where $done(t)$ indicates whether or not the episode duration has been reached). Once enough samples have been collected ($step > Min.Mem.Cap$), then the algorithm enters the learning portion of the algorithm. At this point, if there have been a sufficient number of steps through the environment ($step \,\%\, T_f == 0$), then the target Q-network is updated with the current policy. Otherwise, a batch of interactions is grabbed randomly from memory and we compute the target Q-values and the loss from Sections IV-B and IV-D. The objective function is then optimized via Stochastic gradient descent with the Adam optimizer and then a new iteration begins. At the end of each episode, the average sum of AoI of the network is computed as $ASoA(K_{ep}) = -\frac{1}{TK}\mathbb{E}\left[\sum_{t=0}^{T-1} r(t)\right]$, where the expectation is taken due to the stochastic behavior in the channels.

Algorithm 1: Proposed DQN Algorithm

Input: F, M_A, K, K_M, M, W_k activation patterns, locations, thresholds, $step = 0$, other parameters.
Output: Avg. Sum of AoI ($ASoA$)
Initialization: DQN Hyper-parameters, $MaxIter$, T.
for $K_{ep} = 1 : MaxIter$ **do**
  Set: UAV-RIS initial height $H[0]$, $AoI_k(0) = 1, \forall k$;
  while $t \leq T$ **do**
    Sample an action $a(t) \in \mathcal{A}$ using ϵ-greedy;
    Perform: IoTD cluster scheduling $I_{k_m}(t)$ and $(\Delta h_{UAV})(t)$;
    Compute ϕ from (6);
    Compute $\gamma_k(t)$ for each k from (4);
    Update: AoI for each IoTD;
    Compute the reward $r(t)$ using (9);
    Store Interaction: $s(t)$, $s(t+1)$, $a(t)$, and $r(t)$;
    if $t >$ *Min. Mem. Cap.* **then**
      Agent Learns;
      if $step \,\%\, T_f == 0$ **then**
        $\hat{\theta} \leftarrow \theta$;
      end
      Grab: Batch of interactions from memory of size B;
      Compute: Q_T and Loss;
      Update optimizer;
    end
    Transition to $s(t+1)$; $step = step + 1$;
  end
  Compute: $ASoA(K_{ep}) = -\frac{1}{TK}\mathbb{E}\left[\sum_{t=0}^{T-1} r(t)\right]$;
end

C. Proximal Policy Optimization (PPO)

PPO is a policy-based method that uses two deep NNs. The first network, the Actor, is what we train and it maintains the current policy $\pi_\theta(a|s)$. The Actor takes states in as input and

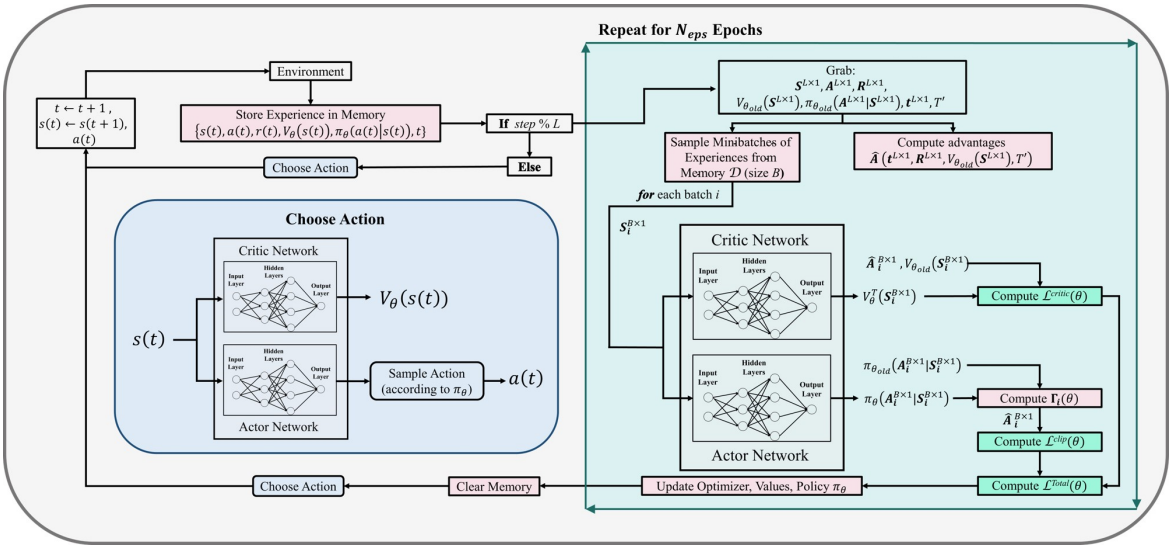

Fig. 4: PPO Architecture.

outputs a list of probabilities corresponding to each action in the action space. These probabilities form a distribution whereby an action can be sampled. The second network, the Critic, is used to "critique" the current policy of the Actor in terms of an estimated state-value, $V_\theta(s)$. After an action is performed, the Critic assesses whether the action taken has increased or reduced the expected feedback given the current state.

Unlike DQN, PPO only keeps a memory, denoted by $\mathcal{D}$, of experiences of size L. Once $\mathcal{D}$ is full, PPO iterates through a number of epochs over the collected data. In each epoch, the algorithm collects randomly sampled minibatches of the data in $\mathcal{D}$ where each batch holds B experiences, which are generated by the latest version of the policy, $\pi_{\theta_{old}}(a|s)$. Using the collected rewards and current state-values, $V_{\theta_{old}}$, PPO then uses the generalized advantage estimation (GAE) algorithm [41] to generate the advantage estimate, $\hat{A}$. The advantage estimate simply gives an evaluation on the existing policy. For each batch, we pass the collected states into the Actor and Critic networks to establish the new policy and the corresponding Critic target values, respectively.

The policy is updated using the stochastic gradient ascent (SGA) optimization method via the Adam optimizer in python. Once all the epochs have been executed, PPO clears its memory and proceeds interacting with the environment using the new policy $\pi_\theta(a|s)$ in the Actor. Fig. 4 shows the basic architecture of PPO. $\mathbf{S}^{L\times 1}$, $\mathbf{A}^{L\times 1}$, and $\mathbf{R}^{L\times 1}$ are the collected states, actions, and rewards in memory $\mathcal{D}$, respectively.

The pseudo-code of the proposed PPO method is summarized in Algorithm 2. The initialization process described in the DQN pesudo-code is the same for the PPO initialization, except for the hyperparameter settings which are shown in Table IIIb. At the start of each episode, K_{ep}, the UAV-RIS's initial height is reset and the AoI of the IoTDs is reset to zero. The algorithm then starts stepping through the environment where the state $s(t_0)$ holds the initial AoI, and SNR of the IoTDs and the UAV-RIS height location. The algorithm selects a valid action from the action space by observing the probabilities from the current policy, $\pi_\theta(a(t_0)|s(t_0))$ where the action $a(t_0)$ holds an indicator for switch/stay with the current IoTD cluster and hover/ascend/descend the UAV-RIS. Once the action is selected, we compute the current SNR of the IoTDs before the UAV-RIS adjusts its vertical location depending on the selected action. Once the SNRs are computed, the processes outlined in Section IV-A2 are followed for the fixed action space to perform the IoTD cluster scheduling. Once a cluster is selected for transmission, the UAV-RIS height is updated.

After executing the current action, the reward is then computed as the negative of the sum of AoI of all the IoTDs. The transition $\{s(t),\ a(t),\ r(t),\ V_\theta(s(t)),\ \pi_\theta(a(t)|s(t)),\ s(t+1),\ done(t)\}$ is then recorded in memory (where $done(t)$ indicates whether or not the episode duration has been reached). Once enough samples have been collected ($step > L$), then the algorithm enters the learning phase. We grab the states, acitons, advantage values, probabilities, rewards, and done vlaues under the old policy $\pi_{\theta_{old}}$. We first compute the advantage estimates according to [39] and sample a minibatch of experiences from memory. For each batch, we pass through the states in the minibatch to the actor and critic neural networks to generate the new policy π_θ and target advantage values, respectively, as described in Sections IV-C and IV-D. Using these values, the total loss can be computed according using Eqns. (14)-(16). Once the policy network is updated with the new NN parameters θ, the objective function is optimized via SGA using the Adam optimizer and then we repeat this process for a number of epochs, N_{eps}, to refine the weights. After completing this process, memory is cleared and we continue stepping through the environment. At the end of each episode, the average sum of AoI of the network is computed as $ASoA(K_{ep}) = -\frac{1}{TK}\mathbb{E}\left[\sum_{t=0}^{T-1} r(t)\right]$.

Algorithm 2: Proposed PPO Algorithm

Input: F, M_A, K, M, W_k activation patterns, Locations, Thresholds, Other Parameters.
Output: Avg. Sum of AoI ($ASoA$)
Initialization: PPO Hyper-parameters, Actor & Critic Networks, $MaxIter$, T.
for $K_{ep} = 1 : MaxIter$ **do**
  Set: UAV-RIS initial height $H[0]$, $AoI_k(0) = 1, \forall k$;
  while $t \leq T$ **do**
    Sample an action $a(t) \in \mathcal{A}$ using Actor;
    Perform: IoTD cluster scheduling $I_{k_m}(t)$ and $(\Delta h_{UAV})(t)$;
    Compute ϕ from (6);
    Compute $\gamma_k(t)$ for each k from (4);
    Update: AoI for each IoTD;
    Compute the reward $r(t)$ using (9);
    Store Interaction: $s(t)$, $s(t+1)$, $a(t)$, probabilities, values, and $r(t)$;
    if *t % L == 0* **then**
      Agent Learns using the PPO Objective Function;
      for $j = 1, .., N_{eps}$ **do**
        Grab: Past L interactions from memory;
        Compute the $\hat{A}$ estimates;
        Grab: Randomly sorted batches of past L interactions of size B;
        for `Each Batch` **do**
          Compute the new: log probabilities, probability ratio, and total loss w.r.t. θ;
          Pass through to NN optimizer;
          Map $\theta_{old} \leftarrow \theta_{new}$;
        end
        Clear memory;
      end
    end
    Transition to $s(t+1)$;
  end
  Compute: $ASoA(K_{ep}) = -\frac{1}{TK}\mathbb{E}\left[\sum_{t=0}^{T-1} r(t)\right]$;
end

D. NN Updates and the Importance of the Loss Functions

1) Loss Function Background: NNs are trained using optimization algorithms to find the optimal weights that map the inputs to the outputs. This is done by iteratively adjusting those weights along gradients that minimize a given loss function, $\mathcal{L}(\theta)$. Many types of loss functions exist including mean squared error (MSE) (L2-loss), mean absolute error (MAE) (L1-loss), log-cosh, cross-entropy, quantile, hinge, huber, KL divergence, marginal, binary, behavior cloning, etc. Depending on the type of problem, some loss functions may be more suitable to use than others as they affect the direction in which the model is reconditioned.

If $\mathcal{L}(\theta)$ is differentiable w.r.t. its parameters, gradient-descent based optimization is relatively efficient and often the most preferred in DRL since we aim to quickly find an optimal solution. The gradient descent procedure modifies the estimates opposite the direction of the gradient of $\mathcal{L}(\theta)$. That is, $\theta_i = \theta_{i-1} - \alpha \nabla_\theta \mathcal{L}(\theta; \theta_{i-1})$ where $\alpha > 0$ is the learning rate [42].

2) DQN Loss Function: DQN uses a function approximation method to approximate the original Bellman equation in Q-learning. The DQN algorithm was first presented in [43] where the Evaluation Q Network is updated according to the following MSE loss function:

$$\mathcal{L}^{Total}(\theta) = \mathbb{E}_{(s,a,r,s') \sim U(\mathcal{R})}[(Q_T - Q(s,a|\theta))^2] \quad (12)$$

where $U(\mathcal{R})$ denotes the uniformly sampled minibatch of experiences from the replay buffer. The gradient of the loss is backpropogated into the parameters, θ, of the Evaluation network. Q_T can be thought of as the temporal difference (TD) target and the value $Q_T - Q(s,a|\theta)$ is the TD error. TD, in this case, is used to predict a Q-value (Q_T) that depends on the Q-value of the future state. The MSE loss function will penalize the model when large errors are made by squaring them. In this way, the loss function minimizes the optimal Bellman residual. MSE loss is one of the most widely used loss functions in ML. However, MSE will usually suffer in robustness due to outliers and uncertainty.

3) PPO Loss Function: Unlike DQN, PPO is a policy-gradient method that involves optimizing the policy w.r.t. expected return (or reward) using gradient descent methods. The PPO loss function serves as the update guidance for the Actor and is defined in [41], and with the coefficient of the entropy function equal to zero, we get:

$$\mathcal{L}^{Total}(\theta) = \mathbb{E}\left[\mathcal{L}^{clip}(\theta) - c_1 \mathcal{L}^{critic}(\theta)\right] \quad (13)$$

where $\mathcal{L}^{clip}$ is the loss function of the Actor, $\mathcal{L}^{critic}$ is the loss function of the Critic, and c_1 is a loss coefficient. For each batch, $\mathcal{L}^{Total}(\theta)$ is computed and the NN parameters are updated. $\mathcal{L}^{clip}$ and $\mathcal{L}^{critic}$ are defined as follows.

$$\mathcal{L}^{clip}(\theta) = \mathbb{E}[\min(\boldsymbol{\Gamma_i}(\theta) \cdot \hat{\boldsymbol{A}}_{\boldsymbol{i}}^{B\times 1}, \text{clip}(\boldsymbol{\Gamma_i}(\theta), 1+\epsilon_{clip}, 1-\epsilon_{clip}) \cdot \hat{\boldsymbol{A}}_{\boldsymbol{i}}^{B\times 1})] \text{ and} \quad (14)$$

$$\mathcal{L}^{critic}(\theta) = \left[V_\theta\left(\mathbf{S}_i^{B\times 1}\right) - V_\theta^T(\mathbf{S}_i^{B\times 1})\right]^2, \quad (15)$$

where ϵ_{clip} is a parameter, $\mathbf{S}_i^{B\times 1}$ and $\mathbf{A}_i^{B\times 1}$ are the collection of states and actions in batch i of batch size B, respectively, $V_{\theta_{old}}(\mathbf{S}_i^{B\times 1})$ are the state-values corresponding to the states in batch i under the old policy, $V_\theta(\mathbf{S}_i^{B\times 1}) = \hat{\boldsymbol{A}}_{\boldsymbol{i}}^{B\times 1} + V_{\theta_{old}}(\mathbf{S}_i^{B\times 1})$ are the estimated state-values under the new policy, $V_\theta^T(\mathbf{S}_i^{B\times 1})$ are the target values from the Critic, $\boldsymbol{\Gamma_i}(\theta) = \frac{\pi_\theta(\mathbf{A}_i^{B\times 1}|\mathbf{S}_i^{B\times 1})}{\pi_{\theta_{old}}(\mathbf{A}_i^{B\times 1}|\mathbf{S}_i^{B\times 1})}$ holds the probability ratios of the new policy over the old policy, and $\hat{\boldsymbol{A}}_{\boldsymbol{i}}^{B\times 1}$ are the advantage estimates for the corresponding experience samples in batch i. The advantage estimates are given by [41]:

$$\hat{\boldsymbol{A}}(\mathbf{t}^{L\times 1}, \mathbf{R}^{L\times 1}, V_{\theta_{old}}(\mathbf{S}^{L\times 1})) = \delta(t) + (\gamma\lambda_p)\delta(t+1) + \cdots + (\gamma\lambda_p)^{T'-t-1}\delta(T'-1) \quad (16)$$

where T' is the end time of the current learning iteration, $\gamma \in [0,1]$ is the discount factor, λ_p accounts for variance in the model, and $\delta(t) = r(t) + \gamma V_{\theta_{old}}(s(t+1)) - V_{\theta_{old}}(s(t))$.

At each step of SGA, the objective function in (13) is maximized. To understand this, we discuss the objective function from the Trust Region Policy Optimization (TRPO): $\max_\theta \left(\mathbb{E}\left[\Gamma(\theta) \cdot \hat{A}\right]\right)$. This surrogate objective function is maximized at every learn iteration to ensure that the new policy defined by the expected discounted returns has a positive improvement from the old policy. In other words, the policy is learned by maximizing the expected returns. However, under this objective, each time we refine the old policy, the deviation between the new and old policy increases and thus, the variance in the estimation of the advantages increases. This lead the researchers in [41] to develop the modified objective in (14) with clipping the ratio term to be within $1 \pm \epsilon_{clip}$. This prevents the algorithm from making significant changes from the old policy to the new policy when learning updates occur, reducing the chances of instability in the model.

E. Complexity and Convergence Analysis

The complexity of DRL algorithms such as those implementing the Actor-Critic framework or DQN, is evaluated by the number of multiplications involved in each iteration. According to [44], [45], for DQN and PPO, the computational time complexity per time-step for a fully-connected deep NN is expressed by $\mathcal{O}\left(\sum_{j=0}^{J-1} n_j \cdot n_{j+1}\right)$, where n_j is the number of neurons in the j-th hidden layer and J is the number of layers.

Analytically, ensuring the convergence of any DRL algorithm is challenging since it can depend on the selection of the hyperparameters used. The number of neurons present in each hidden layer can affect the time execution per learning iteration and it may lead to under or overfitting the training data set [46].Thus, in this work, the convergence analysis will be evaluated via simulation studies (see Section V) under certain hyperparameter and environment settings.

V. Simulation and Performance Evaluation

In this section, we validate the performance of the developed algorithms via numerical simulations to gain insight into UAV-RIS aided wireless communications. We first evaluate the effectiveness of the fixed action space described in Section IV-A2 and the number of groupings of IoTDs. Second, we

TABLE III
SIMULATION AND PERFORMANCE EVALUATION PARAMETER SETTINGS

(a) ENVIRONMENT PARAMETERS

Notation	Value
(x, y)	$(-400, -100)$ m
$h_{UAV}(t_0)$	68 m
(x_m, y_m)	$(-3500, 300)$ m
h_{BS}	65 m
$[H_{U,min}, H_{U,max}]$	$[20, 120]$ m
β_0	-30 dB
P_T	30 dBm
$\sigma^2_{N_0}$	-138 dBm
α_B	2
f_c	24 GHz
η_{LOS}, η_{NLOS}	2.3, 34 dB
K_1, K_2	2, 10
F, M_A	64, 100
V_{Max}	1 m/s

(b) ALGORITHM HYPERPARAMETERS

Notation	DQN	PPO
$MaxIter$	5000	5000
α	0.001	0.01
γ	0.90	0.99
L	–	25
B	32	5
N_{eps}	–	4
T_f	100	–
ϵ_{init}	0.95	–
ϵ_{clip}	–	0.1
c_1	–	0.5
$Min.Mem.Cap$	2000	–
Activation Functions	ReLU	ReLU,Tanh
Num. of NN Hidden Layers	2	2
Number of Neurons	20, 20	20, 40

discuss the stability and convergence of the two algorithms. Third, we evaluate the adaptability of the proposed methods under different environmental conditions. The time-step size is set to $t_S = 1$ sec. and the episode duration is $T = 300$ sec. The environment parameters of the UAV-RIS aided wireless network are shown in Table IIIa. Unless otherwise specified, there are $K = 250$ IoTDs in the network and are assumed to be spatially spread according to the Uniform distribution within a grid of size 1×1 km^2 and are clustered based on Euclidean distances via the K-Means algorithm. There are also 130 buildings uniformly distributed between the BS to IoTDs, 10 of which are located within the IoTD grid. Hyperparameter values of PPO and DQN are listed in Table IIIb. We use feedforward NNs with Adam and Adagrad optimization architecture, which are extensions to the traditional gradient-descent methods using an adaptive learning rate to accelerate the optimization process.

A. Effect of Number of Clusters

Scalability is crucial to the design of wireless infrastructures to keep pace with the increasing numbers of IoTDs. The fixed action space is designed such that the size does not increase as the number of IoTDs increases. To demonstrate the performance of the developed DQN and PPO algorithms, we plot the ASoA vs. the number of episodes for different numbers of clusters of IoTDs shown in Figs. 5(a) and 5(b), respectively. As the number of clusters of IoTDs increases, the ASoA increases since less and less IoTDs will be grouped into a single cluster and the network starts appearing closer to a single-scheduled IoTD network. The convergence of DQN is not as stable as that of PPO, even after 5000 episodes. Both algorithms appear to approach similar ASoA values as the number of episodes increases. It is explicit that PPO achieves higher convergence rate as compared to DQN within the first 500 episodes.

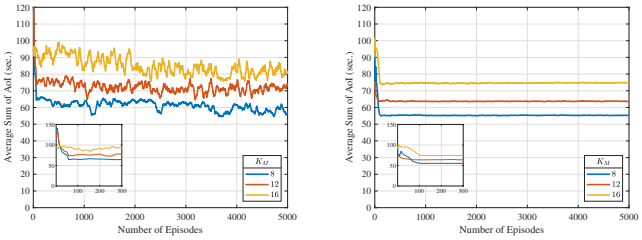

Fig. 5: (a) DQN and (b) PPO.

B. Stability and Convergence

To test the stability of the proposed methods, we ran the algorithms for 50 trials, where each trial spans 5000 episodes. Then, we have 50 samples for each episode where we compute the 95% confidence interval (CI). The results can be seen in Fig. 6 with $K_M = 15$ clusters. A random trial number is selected and the corresponding algorithm results are plotted using solid lines. The shaded regions show the respective 95% CI for each episode. The zoomed in windows provide further details on the performance of the algorithms. First, we note PPO's convergence is stable within the first few hundred episodes. Across all three windows, PPO maintains its stability close to within the confidence bands. DQN, on the other hand, takes longer to converge and struggles to stay within the confidence bands as the number of episodes increases.

A lower bound for ASoA is generated by relaxing the UAV velocity constraint. In each time slot, for all possible altitudes of UAV, the sum of AoI reduction is computed for every cluster of IoTDs. The maximum sum of AoI reduction is selected to compute the immediate reward in (9). According to the simulation results, the lower bound for ASoA is 66.45 and the converged ASoA (Fig. 6) is 79 and attains 18.88% above the lower bound.

C. Performance Under Diverse Environment Settings

1) Environment 1 – BS Location: In densely populated cities with tall office buildings or skyscrapers (such as New York City or Chicago), there can be many obstructions between the IoTDs and the nearest BS. Thus, in this section, we consider a large-scale scenario where the BS is located at different distances from the UAV-RIS. The results can be seen

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3246925

in Figs. 7 (a) and (e) for DQN and PPO, respectively. First, we note that the stability of DQN fluctuates similarly compared to the previous results in Fig. 5 (a). PPO on the other hand achieves about the same stability performance as in Fig. 5 (b) for all the BS location cases. Due to increased backhaul distances, slight changes in altitude adjustment only lead to very slight changes in the received SNR at the IoTDs for low altitudes. This means it becomes harder for DQN to learn and converge to an optimal policy since DQN cannot account for large changes in the policy. PPO's clipping function enables it to handle the slight variations in the environment since at every update, the algorithm cannot change the probability of any action (desired or undesired) by no more than ϵ_{clip}. In addition, at each update, the PPO objective function described in sections IV-C and IV-D3 aims to increase the probability of desired actions, thereby increasing the ratio $\Gamma(\theta)$, while the probability of bad actions decreases. Due to this feature, we see more stable results in the PPO performance. Location 3 will have the largest ASoA since this BS location is the furthest from the UAV-RIS. At location 2 is the closest to the UAV-RIS and we would expect it to have the lowest ASoA, however, this BS location is surrounded by buildings on all sides and thus the UAV-RIS works to find an optimal height such that the backhual link is not blocked off. Thus, location 2 achieves a similar ASoA to location 1.

2) Environment 2 – IoTD Distribution: In a realistic wireless network environment, IoTDs are not uniformly distributed within a given area. However, this feature is assumed in many prior works such as [13], [23], [33], [47]. Urban areas can have several wireless "hot spot" locations. In this section, we test the performance of the two algorithms when the IoTD locations are given by three different distributions: Uniform, Gaussian, and spatially inversed Gaussian, where the distributions are centered at the center of the grid (i.e., (500,500) m).

The results can be seen in Figs. 7 (b) and (f) for DQN and PPO, respectively. The Uniformly distributed IoTDs result in the lowest ASoA followed by spatially inverted Gaussian and Gaussian. The Gaussian and spatially inverted Gaussian distributions suffered a higher ASoA primarily due to the locations of the buildings within the IoTD grid. Buildings near the center of the grid penalized the Gaussian AoI while buildings near the edge of the grid penalized the spatially inverted Gaussian distribution results. Again we see that PPO rapidly learns and converges while DQN takes more time to train, but converges to similar values.

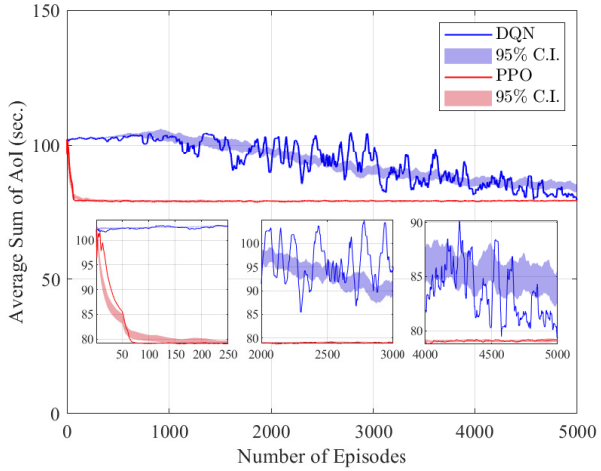

Fig. 6: Average Sum of AoI over T vs. number of episodes.

3) Environment 3 – IoTD Activation Pattern: In previous simulations, we assumed the IoTD activation pattern was uniformly distributed. In reality, the arrival times of packets are independent of each other. In discrete-time, the arrival times can be specified by the stochastic Bernoulli process, $\boldsymbol{P}$, such that the interarrival times, $Z_1, Z_2, \cdots$ (time between an arrival and the previous arrival of data packet information), are i.i.d. random variables. Let ν_i be the time of the ith arrival of $\boldsymbol{P}$ defined as $\nu_i = \sum_{j=1}^{i} Z_j$ where $Z_j \sim$ Geo(p) and p is the probability. This implies that $\boldsymbol{P}$ follows the memoryless property in that it restarts after each arrival time so that the next arrival time is independent of the past. Using the interarrival times, we can populate W_k (the activation vector) for each IoTD. By doing this, we can see the performance of the algorithms under more realistic IoTD activation patterns. Setting $p = \{0.4,\ 0.5,\ 0.8\}$, we expect the average number of activations of each IoTD to be around 40%, 50%, and 80%, respectively, of the time period T, i.e., around $\{120,\ 150,\ 240\}$ activations.

The results can be seen in Figs. 7 (c) and (g) for DQN and PPO, respectively. We first note that for both algorithms, the converged ASoA is higher for $p = \{0.4\}$ compared to results in the previous sections. This is expected since in the previous sections, due to the uniform distribution, on average, IoTDs were active around 50% of the time period (demonstrated by $p = 0.5$). For $p = \{0.8\}$, the ASoA is lower since the IoTDs are more active within the episode so they are more likely to have a successful transmission as long as their received SNR is above the threshold. Considering that the AoI is directly impacted by the IoTD activation, then the freshness of information is larger since status packet updates are not as frequent. PPO and DQN converge to similar values, but for most p cases, PPO converges within the first 500 episodes.

4) Environment 4 – UAV-RIS Location: In this environment setting, we test the performance of the algorithms when the horizontal location of the UAV-RIS is changed. The results can be seen in Figs. 7 (d) and (h) for DQN and PPO, respectively. The results of all four locations converge to nearly similar ASoA values. Compared to the performance of previous environments, we see that DQN and PPO achieve quick convergence with similar converged values. Practically, in terms of feasibility, this means that with just a few parameter adjustments, the UAV-RIS can offer significant aerial advantages in urban and rural regions where low-altitude applications are irrational due to obstacles. Even as the complexity of the environment increases with variable UAV-RIS location, PPO proves the capability of handling the complexity and finding an optimal policy for minimizing the ASoA.

VI. Conclusion

This work performs an Off-Policy vs. On-Policy comparison to better understand the design strategies of model-free DRL algorithms for UAV-RIS assisted large-scale IoT wireless networks. Using experimentally validated channel models, a generalized optimal RIS phase shift model is derived. DQN and PPO algorithms, as the Off-Policy and On-Policy candidates, respectively, are adopted to optimize the average sum of AoI of all IoTDs in the network. Through a comprehensive analysis of the two loss functions, PPO demonstrates to have finer control than DQN over the gradient ascent/descent procedures. In simulations, three metrics are used for evaluating the convergence performance of DQN and PPO: (1) impact of the size of the action space; (2) the stability and convergence of the algorithms; (3) and variance in performance under different environment settings. In each metric, PPO outperforms DQN for different environment settings: (1) PPO handles complicated action spaces more efficiently compared to DQN; (2) using CIs, as the number of episodes increased, PPO maintains relatively within a narrow confidence band since the first few hundred episodes, whereas DQN takes several thousand episodes to reach a narrow level of confidence; (3) over the different environment settings, PPO maintains stable performance for each case and has a faster convergence speed than DQN. To manifest the importance of 3D movement of the UAV-RIS, we consider the IoTD mobility in future work.

References

[1] J. Lin, W.-M. Chen, H. Cai, C. Gan, and S. Han, "MCUNetV2: Memory-Efficient Patch-based Inference for Tiny Deep Learning," *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[2] V. Berggren, R. Inam, L. Mokrushin, A. Hata, J. Jeong, S.-K. Mohalik, J. Forgeat, and S. Sorrentino, "Artificial Intelligence in Next-Generation Connected Systems," *Ericsson White Paper*, 2021, https://www.ericsson.com/en/reports-and-papers/white-papers/artificial-intelligence-in-next-generation-connected-systems.

[3] "Ericsson Mobility Report: Mobile Data Traffic Outlook," *Ericsson*, 2021, https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/mobile-traffic-forecast.

[4] D. Astely, P. Butovitsch, S. Faxer, and R. Larsson, "Ericsson Technology Review: The Role of Massive MIMO in 5G Networks - Meeting 5G Network Requirements with Massive MIMO," *Ericsson*, 2022, https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/using-massive-mimo-to-meet-5g-network-requirements.

[5] X. Yuan, Y.-J. A. Zhang, Y. Shi, W. Yan, and H. Liu, "Reconfigurable-Intelligent-Surface Empowered Wireless Communications: Challenges and Opportunities," *IEEE Wireless Communications*, vol. 28, no. 2, pp. 136–143, 2021, DOI: 10.1109/MWC.001.2000256.

[6] O. Özdogan, E. Björnson, and E. G. Larsson, "Using Intelligent Reflecting Surfaces for Rank Improvement in MIMO Communications," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 9160–9164, 2020, DOI: 10.1109/ICASSP40776.2020.9052904.

[7] X. Pei, H. Yin, L. Tan, L. Cao, Z. Li, K. Wang, K. Zhang, and E. Björnson, "RIS-Aided Wireless Communications: Prototyping, Adaptive Beamforming, and Indoor/Outdoor Field Trials," *IEEE Transactions on Communications*, vol. 69, no. 12, pp. 8627–8640, 2021, dOI: 10.1109/TCOMM.2021.3116151.

[8] A. C. Pogaku, D.-T. Do, B. M. Lee, and N. D. Nguyen, "UAV-Assisted RIS for Future Wireless Communications: A Survey on Optimization and Performance Analysis," *IEEE Access*, vol. 10, pp. 16 320–16 336, 2022, DOI: 10.1109/ACCESS.2022.3149054.

[9] Y. Chen, Y. Liu, M. Zeng, U. Saleem, Z. Lu, X. Wen, D. Jin, Z. Han, T. Jiang, and Y. Li, "Reinforcement Learning Meets Wireless Networks: A Layering Perspective," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 85–111, 2021, DOI: 10.1109/JIOT.2020.3025365.

[10] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019, DOI: 10.1109/COMST.2019.2916583.

[11] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of Information: An Introduction and Survey," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1183–1210, 2021, DOI: 10.1109/JSAC.2021.3065072.

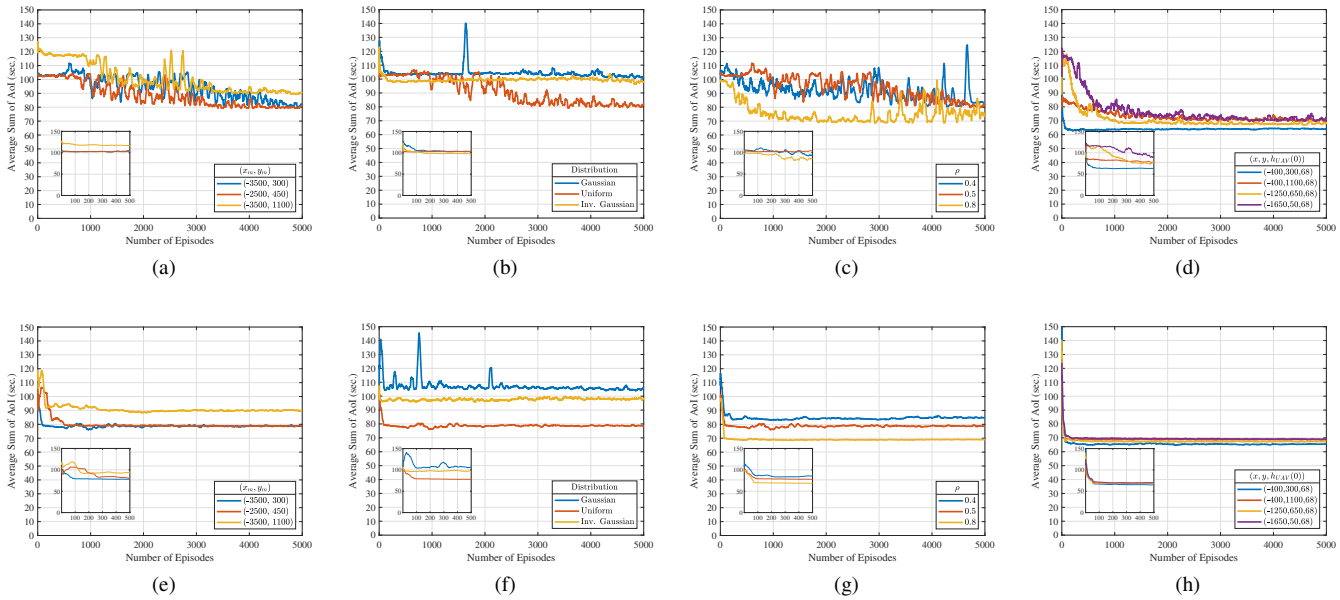

Fig. 7: DQN with (a) Environment 1, (b) Environment 2, (c) Environment 3, and (d) Environment 4 and PPO with (e) Environment 1, (f) Environment 2, (g) Environment 3, and (h) Environment 4.

[12] I. Kadota, A. Sinha, and E. Modiano, "Scheduling Algorithms for Optimizing Age of Information in Wireless Networks With Throughput Constraints," *IEEE/ACM Transactions on Networking*, vol. 27, no. 4, pp. 1359–1372, 2019, DOI: 10.1109/TNET.2019.2918736.

[13] Z. Liu, Z. Chen, L. Luo, M. Hua, W. Li, and B. Xia, "Age of Information-based Scheduling for Wireless Device-to-Device Communications using Deep Learning," *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2021, DOI: 10.1109/WCNC49053.2021.9417493.

[14] R. Talak, S. Karaman, and E. Modiano, "Improving Age of Information in Wireless Networks With Perfect Channel State Information," *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1765–1778, 2020, DOI: 10.1109/TNET.2020.2996237.

[15] Y. Inoue, H. Masuyama, T. Takine, and T. Tanaka, "A General Formula for the Stationary Distribution of the Age of Information and Its Application to Single-Server Queues," *IEEE Transactions on Information Theory*, vol. 65, no. 12, pp. 8305–8324, 2019, DOI: 10.1109/TIT.2019.2938171.

[16] A. Kosta, N. Pappas, and V. Angelakis, *Age of Information: A New Concept, Metric, and Tool*. NOW: Foundations and Trends in Networking, 2017, DOI: 10.1561/1300000060.

[17] Z. Fang, J. Wang, Y. Ren, Z. Han, H. V. Poor, and L. Hanzo, "Age of Information in Energy Harvesting Aided Massive Multiple Access Networks," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 5, pp. 1441–1456, 2022, DOI: 10.1109/JSAC.2022.3143252.

[18] M. A. Abd-Elmagid, A. Ferdowsi, H. S. Dhillon, and W. Saad, "Deep Reinforcement Learning for Minimizing Age-of-Information in UAV-Assisted Networks," *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2019, DOI: 10.1109/GLOBECOM38437.2019.9013924.

[19] M. A. Abd-Elmagid and H. S. Dhillon, "Average Peak Age-of-Information Minimization in UAV-Assisted IoT Networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 2003–2008, 2019, DOI: 10.1109/TVT.2018.2885871.

[20] J. Liu, X. Wang, B. Bai, and H. Dai, "Age-Optimal Trajectory Planning for UAV-Assisted Data Collection," *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 553–558, 2018, DOI: 10.1109/INFCOMW.2018.8406973.

[21] M. Chen, Y. Xiao, Q. Li, and K.-C. Chen, "Minimizing Age-of-Information for Fog Computing-Supported Vehicular Networks with Deep Q-learning," *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2020, DOI: 10.1109/ICC40277.2020.9149054.

[22] Z. Gong, Q. Cui, C. Chaccour, B. Zhou, M. Chen, and W. Saad, "Lifelong Learning for Minimizing Age of Information in Internet of Things Networks," *ICC 2021 - IEEE International Conference on Communications*, pp. 1–6, 2021, DOI: 10.1109/ICC42927.2021.9500646.

[23] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile Unmanned Aerial Vehicles (UAVs) for Energy-Efficient Internet of Things Communications," *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7574–7589, 2017, DOI: 10.1109/TWC.2017.2751045.

[24] M. H. Cheung, "Age of Information Aware UAV Network Selection," *2020 18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*, pp. 1–8, 2020.

[25] Q. Wu, S. Zhang, B. Zheng, C. You, and R. Zhang, "Intelligent Reflecting Surface-Aided Wireless Communications: A Tutorial," *IEEE Transactions on Communications*, vol. 69, no. 5, pp. 3313–3351, 2021, DOI: 10.1109/TCOMM.2021.3051897.

[26] B. Ning, Z. Chen, Z. Tian, X. Wang, C. Pan, J. Fang, and S. Li, "Joint Power Allocation and Passive Beamforming Design for IRS-Assisted Physical-Layer Service Integration," *IEEE Transactions on Wireless Communications*, vol. 20, no. 11, pp. 7286–7301, 2021, DOI: 10.1109/TWC.2021.3082530.

[27] H. Yang, Y. Zhao, Z. Xiong, J. Zhao, D. Niyato, K.-Y. Lam, and Q. Wu, "Deep Reinforcement Learning Based Intelligent Reflecting Surface for Secure Wireless Communications," *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1–6, 2020, DOI: 10.1109/GLOBECOM42002.2020.9322615.

[28] C. Huang, R. Mo, and C. Yuen, "Reconfigurable Intelligent Surface Assisted Multiuser MISO Systems Exploiting Deep Reinforcement Learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 8, pp. 1839–1850, 2020, DOI: 10.1109/JSAC.2020.3000835.

[29] J. Fang, Z. Yang, N. Anjum, Y. Hu, H. Asgari, and M. Shikh-Bahaei, "Secure Intelligent Reflecting Surface Assisted UAV Communication Networks," *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, 2021, DOI: 10.1109/ICCWorkshops50388.2021.9473641.

[30] S. Jiao, F. Fang, X. Zhou, and H. Zhang, "Joint Beamforming and Phase Shift Design in Downlink UAV Networks with IRS-Assisted NOMA," *Journal of Communications and Information Networks*, vol. 5, no. 2, pp. 138–149, 2020, DOI: 10.23919/JCIN.2020.9130430.

[31] Q. Zhang, W. Saad, and M. Bennis, "Reflections in the Sky: Millimeter Wave Communication with UAV-Carried Intelligent Reflectors," *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2019, DOI: 10.1109/GLOBECOM38437.2019.9013626.

[32] ——, "Distributional Reinforcement Learning for mmWave Communications with Intelligent Reflectors on a UAV," *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1–6, 2020, DOI: 10.1109/GLOBECOM42002.2020.9348040.

[33] M. Samir, M. Elhattab, C. Assi, S. Sharafeddine, and A. Ghrayeb, "Optimizing Age of Information Through Aerial Reconfigurable Intelligent Surfaces: A Deep Reinforcement Learning Approach," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 3978–3983, 2021, DOI: 10.1109/TVT.2021.3063953.

[34] I. Yildirim, A. Uyrus, and E. Basar, "Modeling and Analysis of Reconfigurable Intelligent Surfaces for Indoor and Outdoor Applications in Future Wireless Networks," *IEEE Transactions on Communications*, vol. 69, no. 2, pp. 1290–1301, 2021, dOI: 10.1109/TCOMM.2020.3035391.

[35] Y. Li, C. Yin, T. Do-Duy, A. Masaracchia, and T. Q. Duong, "Aerial Reconfigurable Intelligent Surface-Enabled URLLC UAV Systems," *IEEE Access*, vol. 9, pp. 140 248–140 257, 2021, DOI: 10.1109/ACCESS.2021.3119268.

[36] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP Altitude for Maximum Coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014, DOI: 10.1109/LWC.2014.2342736.

[37] Y. Han, W. Tang, S. Jin, C.-K. Wen, and X. Ma, "Large Intelligent Surface-Assisted Wireless Communication Exploiting Statistical CSI," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 8238–8242, 2019, DOI: 10.1109/TVT.2019.2923997.

[38] D.-W. Yue, H. H. Nguyen, and Y. Sun, "mmWave Doubly-Massive-MIMO Communications Enhanced With an Intelligent Reflecting Surface: Asymptotic Analysis," *IEEE Access*, vol. 8, pp. 183 774–183 786, 2020, DOI: 10.1109/ACCESS.2020.3029244.

[39] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency," *Foundations and Trends® in Signal Processing*, vol. 11, no. 3-4, pp. 154–655, 2017, DOI: 10.1561/2000000093.

[40] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. The MIT Press, 2018.

[41] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," 2017, DOI: 10.48550/ARXIV.1707.06347.

[42] C. Lei, *Deep Reinforcement Learning. In: Deep Learning and Practice with MindSpore. Cognitive Intelligence and Robotics.* Springer, Singapore, 2021, DOI: 10.1007/978-981-16-2233-5_10.

[43] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level Control Through Deep Reinforcement Learning," *Nature*, vol. 518, pp. 529–533, 2015, DOI: 10.1038/nature14236.

[44] J. Tan and W. Guan, "Resource allocation of fog radio access network based on deep reinforcement learning," *Engineering Reports*, vol. 4, no. 5, p. e12497, 2022, dOI: https://doi.org/10.1002/eng2.12497.

[45] M. Samir, C. Assi, S. Sharafeddine, and A. Ghrayeb, "Online Altitude Control and Scheduling Policy for Minimizing AoI in UAV-Assisted IoT Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 21, no. 7, pp. 2493–2505, 2022, dOI: 10.1109/TMC.2020.3042925.

[46] U. Challita, W. Saad, and C. Bettstetter, "Interference Management for Cellular-Connected UAVs: A Deep Reinforcement Learning Approach," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2125–2140, 2019, dOI: 10.1109/TWC.2019.2900035.

[47] A. Khalili, E. M. Monfared, S. Zargari, M. R. Javan, N. M. Yamchi, and E. A. Jorswieck, "Resource Management for Transmit Power Minimization in UAV-Assisted RIS HetNets Supported by Dual Connectivity," *IEEE Transactions on Wireless Communications*, vol. 21, no. 3, pp. 1806–1822, 2022, DOI: 10.1109/TWC.2021.3107306.