LSTM-characterized Deep Reinforcement Learning for Continuous Flight Control and Resource Allocation in UAV-assisted Sensor Network

Kai Li, *Senior Member, IEEE,* Wei Ni, *Senior Member, IEEE,* and Falko Dressler, *Fellow, IEEE*

***Abstract*—Unmanned aerial vehicles (UAVs) can be employed to collect sensory data in remote wireless sensor networks (WSN). Due to UAV's maneuvering, scheduling a sensor device to transmit data can overflow data buffers of the unscheduled ground devices. Moreover, lossy airborne channels can result in packet reception errors at the scheduled sensor. This paper proposes a new deep reinforcement learning based flight resource allocation framework (DeFRA) to minimize the overall data packet loss in a continuous action space. DeFRA is based on Deep Deterministic Policy Gradient (DDPG), optimally controls instantaneous headings and speeds of the UAV, and selects the ground device for data collection. Furthermore, a state characterization layer, leveraging long short-term memory (LSTM), is developed to predict network dynamics, resulting from time-varying airborne channels and energy arrivals at the ground devices. To validate the effectiveness of DeFRA, experimental data collected from a real-world UAV testbed and energy harvesting WSN are utilized to train the actions of the UAV. Numerical results demonstrate that the proposed DeFRA achieves a fast convergence while reducing the packet loss by over 15%, as compared to existing deep reinforcement learning solutions.**

***Index Terms*—Unmanned aerial vehicles, Flight trajectory, Resource allocation, Deep deterministic policy gradient, Long short-term memory, Experimental datasets**

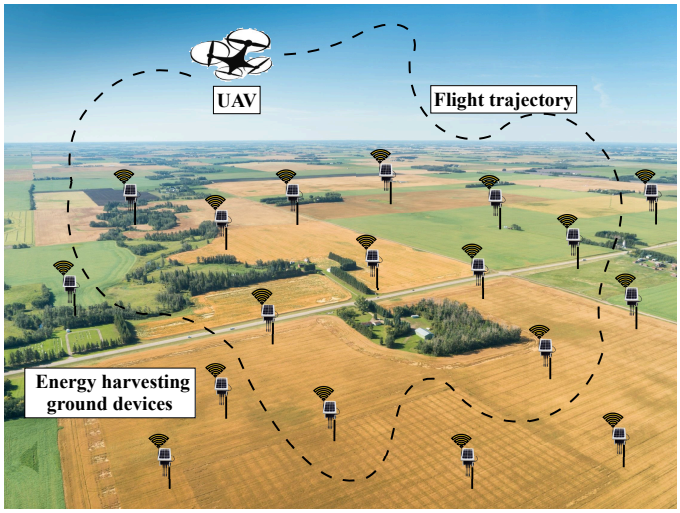


Fig. 1: A typical UAV-assisted WSN is deployed for monitoring crop growth conditions in smart farming. The UAV continually adjusts its heading and the cruising velocity along the trajectory, while selecting the ground devices to transmit data.

I. Introduction

WIRELESS sensor networks (WSN) have been widely studied for sustainable monitoring of remote, human-unfriendly environments, e.g., rural farmlands, forest, or disaster stricken areas [1]. In such harsh environments, terrestrial cellular infrastructures are unavailable or unreliable due to the lack of power supplies [2], [3]. Exploiting unmanned aerial vehicles (UAVs) as aerial data collectors for distributed ground devices can bring significant benefits in WSN [4]. A UAV can freely maneuver at high/low altitudes to achieve a line-of-sight (LoS) link with ground devices, thereby enabling a high data rate for the air-ground communications under all terrains [5].

Fig. 1 presents a typical UAV-assisted energy harvesting WSN in smart farming, where sensing devices monitor crop growth in a remote farmland, e.g., precipitation changes, soil moisture and acidity, and environment temperature [6]. The ground sensing device can be equipped with sun powered boards, or wind control generators, and renewable energy harvested from surroundings is used to energize its battery [7]. The harvested energy is stored to continue operation when the energy sources are not available [8]. Sensory data of the ground device is queued in its buffer, awaiting to be uploaded to the UAV. The UAV equipped with an inertial measurement unit (consisting of three-axis accelerometers, gyroscopes, and magnetometers) and a radio transceiver is employed to patrol over the target farmland. It can also be equipped with solar panels to charge its onboard lightweight rechargeable batteries [9]. Moreover, the UAV changes adaptively its heading and cruising velocity along the flight trajectory, while the ground devices are scheduled by the UAV to transmit data [10].

Due to time-varying data arrivals, the data queue lengths of the ground devices can be substantially different from each other. The ground device scheduled by the UAV for data collection has a short queue length, while other unscheduled ground devices can suffer from buffer overflows. The unscheduled ground devices may have to drop the newly arrived data if their buffers are already full. Moreover, scheduling a ground device, which undergoes a poor link quality, results in packet transmission errors.

K. Li is with Real-Time and Embedded Computing Systems Research Centre (CISTER), 4249–015 Porto, Portugal (E-mail: kai@isep.ipp.pt).

W. Ni is with the Digital Productivity and Services Flagship, Commonwealth Scientific and Industrial Research Organization (CSIRO), Sydney, Australia (E-mail: wei.ni@data61.csiro.au).

F. Dressler is with School of Electrical Engineering and Computer Science, TU Berlin, Berlin, Germany (E-mail: dressler@tkn.tu-berlin.de).

The battery levels and capacities of the ground devices and the UAV have a strong impact on the design of flight control. On the one hand, the flight of the UAV is powered by its battery, and so is the transmission of a ground device. On the other hand, the batteries of the UAV and ground devices are recharged by renewable sources, and the energy harvesting (i.e., recharging) processes are stochastic processes. In this sense, the flight trajectory of the UAV and its communication schedule with the ground devices heavily rely on the energy harvesting processes of the UAV and ground devices, while the energy harvesting processes are reflected by the changes of the battery levels at the UAV and ground devices. For this reason, the battery levels are important parameters for the optimization of the flight trajectory and communication schedule. The recharging stops when the battery levels reach the capacities (or in other words, the batteries are fully charged).

In practical scenarios, the instantaneous information of the data queue backlogs, battery levels, and link qualities between the UAV and the ground device is unlikely to be known. Therefore, it is critical to minimize data packet loss, which is resulted from data queue overflows and packet transmission errors, by jointly optimizing flight resource allocation (in terms of the trajectory planning of the UAV and the data collection scheduling).

In this paper, we investigate a deep reinforcement learning based flight resource allocation in a continuous action space, where network dynamics are predicted to train the actions of the UAV. The contributions of this paper are as follows:

1) A new onboard Deep Deterministic Policy Gradient (DDPG) based flight resource allocation framework (DeFRA) is proposed, where the network state consists of the battery levels and data queue backlogs of the ground devices, the timestamps of the UAV's visits to the devices, the battery levels of the UAV, and channel conditions between the ground devices and the UAV. DeFRA continuously learns online the actions of the UAV, i.e., its instantaneous heading, speed and selection of ground devices for data collection.

2) A new state characterization layer based on long short-term memory (LSTM) is developed in DeFRA to predict the time-varying airborne channels, energy and data arrivals at ground devices. The prediction is based on the reports of the ground devices when they are selected for data collection. The LSTM layer addresses the partial observability of the UAV on the states of the devices, approximating the obscure states of unselected devices at every instant for DDPG implementation. This is the first effort, to the best of our knowledge, to explore LSTM with DDPG to optimize the flight resource allocation to minimize data packet loss.

3) DeFRA is implemented in Google TensorFlow with Python 3.5. Experimental data of airborne channels and energy arrivals at the ground devices are collected from a real-world UAV testbed and energy harvesting-powered sensors. The state characterization layer enables the actions of the UAV to be trained in the presence of real-world network dynamics. The effectiveness of DeFRA is validated with the experimental data. Numerical results show that DeFRA achieves fast convergence while reducing the packet loss by over 14%, as compared to existing deep reinforcement learning solutions.

The rest of this paper is structured as follows. The related work on the UAV-assisted WSN and reinforcement learning based UAV networks is reviewed in Section II. Section III presents the system model. In Section IV, DeFRA is proposed to optimize the flight resource allocation. Section V demonstrates testbed setup, datasets collection, and numerical results. Section VI concludes the paper.

II. Related Work

This section reviews the related work on flight resource allocation in UAV-assisted WSNs.

A. UAV-assisted data collection

In [11], a nonorthogonal multiple access (NOMA)-based UAV-assisted data collection protocol is studied to improve the sum rate of multiple ground devices. The placement of the UAV is determined according to a channel hypergraph based sensor grouping and power control of NOMA. The UAV's flight trajectory, altitude, velocity, and data links with ground devices are designed in [12] to reduce the mission completion time of the UAV, where trajectory planning is modeled as a classic traveling salesman problem and the ground devices are divided into groups. A trajectory planning algorithm is developed to generate a visit order of the ground devices based on the groups. The authors of [13] aim to reduce the age of information in a UAV-assisted WSN, which consists of the data uploading time and the time elapsed since the UAV receives the data. A ground device association and trajectory planning strategy is developed to balance the uploading time and the UAV's cruising time. A trajectory planning strategy is studied to reduce the energy consumption of the UAV and/or ground devices, while accomplishing a data gathering tour [14]. The communication scheduling is formulated as a clustering problem, where the trajectory is planned by using a traveling salesman problem solution. In [15], probabilistic LoS channel models are used in the flight resource allocation to improve the average data collection rate. A hybrid offline-online method is studied to design the UAV's trajectory in an offline phase, while scheduling the transmission of the ground devices in an online phase.

The UAV is used as a flying data collector and wireless power source in UAV-assisted WSN, in [16]. The hovering waypoints and duration of the UAV are designed to extend the network lifetime under data collection and UAV energy consumption constraints. In [17], a TDMA-based scheduling model is developed to allow parallel transmissions of multiple wireless powered ground devices to improve the energy efficiency of the UAV. The scheduling model also allocates resources for clustering the ground devices, and determines the hovering time of the UAV and the wireless powering duration.

B. Reinforcement learning-based UAV networks

In [18], a UAV is employed as a jammer to help the legitimate UAV transmitter defend against ground eavesdroppers, where the legitimate UAV transmitter sends confidential information to the ground devices. The UAV jammer sends artificial

noise signals to the ground eavesdropper. Deep reinforcement learning is used to improve the secure capacity by learning the trajectory of the UAV, the transmit power and the jamming power. A reinforcement learning based training environment is developed for the altitude control of the UAV [19]. A learning architecture is presented, which utilizes a digital twin layer to reduce the effort required to implement the trained controllers.

Energy-efficient trajectory planning of a UAV is studied to provide fair communication coverage for the ground devices [20]. The trajectory planning is modeled by using mean field theory with a large state space. Since the trajectory planning of the UAV requires complex control strategies, deep reinforcement learning is applied to solve the problem for practical applications. Energy-efficient trajectory planning is also studied in edge computing networks to improve data freshness and accessibility to ground devices [21]. Deep reinforcement learning with experience replay is used to solve the energy-efficient UAV navigation problem under the constraints of the trajectory and age of information. Deep reinforcement learning is also used in [22] to improve the communication coverage, energy efficiency and connectivity of UAV networks.

In our earlier works [23], [24], scheduling strategies of a UAV are obtained by deep reinforcement learning to minimize the packet loss of a WSN, with consideration of battery levels and data queue lengths of ground devices. The training environment of deep reinforcement learning is generated by using a deep feed-forward neural network to approximate the Q-function for action inference. In [23], a DDPG based flight control scheme is developed in which the UAV carries out the trajectory planning actions in a continuous space. In [24], a deep Q-network (DQN) is used to determine the next waypoint of the UAV in a discrete action space, and the transmit powers of the ground devices.

In contrast, this paper focuses on a new deep reinforcement learning framework, where the state characterization layer is integrated with DDPG to learn online the actions of the UAV with real-world datasets of network dynamics. Particularly, the state characterization layer exploits a recurrent neural network (RNN), i.e., LSTM [25], to predict the time-varying channel conditions, data and energy arrivals at the ground devices for accurately training the DDPG. In addition, the solutions developed in [23] and [24] are simulated as benchmarks in this paper to assess the performance achievements of the proposed DeFRA, as will be shown in Section V.

III. System Model

In this section, we present the system model of the considered UAV-assisted WSN. Notations used in this paper are summarized in Table I.

N ground devices ($i \in [1, N]$) are deployed in a remote area of interest. Renewable energy, e.g., solar power, can be harvested to charge the battery of the ground device. Let $b_i(t) \leq E$ denote the battery level of device i at t, where E (in Joules) is the battery capacity of the ground device. Onboard sensors of the UAV can measure the battery level of the UAV, denoted by $b_{\mathrm{UAV}}(t)$.

The data queue length of the ground device is $\mathcal{Q}$ (in packets). The queue length at time t is $q_i(t) \in [1, \mathcal{Q}]$. The sensory data are randomly generated, thus, data arrivals at the ground devices are random with an unknown Poisson distribution. The data are queued in the buffer, and await to be collected by the UAV, following a first-in-first-out discipline. With the finite buffer size of the device, the newly arrived data packets have to be dropped if $q_i(t) = \mathcal{Q}$, i.e., the buffer overflows.

TABLE I: Notation and definition

Notations	Definitions
N	number of ground devices
$q_i(t)$	buffer length of i
$b_i(t)$	battery levels of i at time t
$\mathcal{Q}$	buffer size of the ground device
$\gamma_i(t)$	timespan of the ground device
$v(t)$	patrol velocity of the UAV
$b_{\mathrm{UAV}}(t)$	battery levels of the UAV
$g_i(t)$	link quality between the UAV and the ground device
δ	discount factor
a_α	actions of the UAV at state α
α, β	network states
M	number of episodes
ζ_{episode}	random process for action exploration
K	size of the minibacth in experience replay

The UAV maneuvers at a low altitude over the targeted field to collect the sensory data, where the LoS probability between the UAV and the ground devices can be known in [26]. Moreover, the UAV and the ground devices can apply channel reciprocity [27] to obtain the channel gain, once a ground device is scheduled to transmit. The transmit power of a ground device is a function of its transmit rate and its channel gain to the UAV [28], [29].

The coordinate of the UAV is $(x(t), y(t), z)$, and the UAV remains at the altitude of z meters [30]. With a safety consideration, the instantaneous speed of the UAV, denoted by $v(t)$, has to be between the minimum and the maximum speeds, i.e.,

$$V_{min} < |v(t)| \leq V_{max}. \tag{1}$$

Moreover, $\Delta v(t)$ and Δt are the acceleration of the UAV and the time for the UAV to fly from $(x(t), y(t), z)$ to $(x(t+1), y(t+1), z)$, respectively. The UAV is assumed not to move backward. The instantaneous speed and heading of the UAV are adjusted online according to the proposed DeFRA framework. The details are provided in the next section.

Furthermore, the propulsion energy consumption of the UAV can be obtained by [31]

$$\begin{aligned}\Delta E_{\mathrm{UAV}}(t) =& P_0\Big(1 + \frac{3v(t)^2}{\omega(t)^2}\Big) + P_0'\Big(\sqrt{1 + \frac{v(t)^4}{4v_0^4}} - \frac{v(t)^2}{2v_0^2}\Big)^{1/2} \\ &+ \frac{1}{2}\xi_{\mathrm{drag}}\rho_{\mathrm{air}}\xi_{\mathrm{rotor}}S_{\mathrm{rotor}}v(t)^3 \end{aligned} \tag{2}$$

where P_0 and P_0' are constants. $\omega(t)$ is the tip speed of the rotor blade. v_0 is the mean rotor induced velocity in hover. ξ_{drag} and ξ_{rotor} denote the fuselage drag ratio and rotor solidity, respectively. ρ_{air} and S_{rotor} denote the air density and rotor disc area, respectively.

IV. Deep Reinforcement Learning-Based Flight Control and Resource Allocation

In this section, we formulate the continuous online control problem of the UAV's flight and communication schedule. The proposed DeFRA employs onboard DDPG to minimize the overall data loss of the ground devices, where the instantaneous heading and speed of the UAV, and the selection of the ground devices are trained and optimized in a continuous action space. An LSTM-based state characterization layer is developed in DeFRA to help effectively predict the unobservable states of all ground devices, i.e., time-varying energy harvesting, data arrivals, and channel conditions, when the ground devices are beyond the coverage of the UAV.

A. State, Action, and Reward

Let $b_i(t)$ and $q_i(t)$ represent the battery level and the data buffer length of device i, respectively. $g_i(t)$ denotes the channel gain between the UAV and the ground device at time t. A device, e.g., the i-th ground device, is scheduled by the UAV to transmit data at time slot t. For estimating the energy and data arrivals at the unscheduled ground devices, a timespan parameter (denoted by $\gamma_i(t)$) is maintained at the UAV for the ground devices. $\gamma_i(t)$ increases by 1 if ground device i is not scheduled by the UAV; or $\gamma_i(t)$ returns to 0, otherwise.

The joint control of the UAV's maneuver and the communication schedule is a Markov decision process (MDP) in the presence of time-varying energy harvesting, packet arrival, and channel fading. The network state of the MDP consists of $b_i(t)$ and $q_i(t)$ ($i \in [1, N]$) of all ground devices, and $b_{\mathrm{UAV}}(t)$, $(x(t), y(t), z)$, $\gamma_i(t)$ and $g_i(t)$ of the UAV. The network state α can be given by

$$\alpha = \{b_{\mathrm{UAV}}(t), b_i(t), q_i(t), g_i(t), \gamma_i(t), (x(t), y(t), z); \forall i \in [1, N]\}. \quad (3)$$

Therefore, we have the battery level of the UAV at time t, which gives

$$b_{\mathrm{UAV}}(t) = b_{\mathrm{UAV}}(t-1) + \Delta b_{\mathrm{UAV}}(t) - \Delta E_{\mathrm{UAV}}(t), \quad (4)$$

where $\Delta b_{\mathrm{UAV}}(t)$ is the harvested solar power of the UAV at t. In particular, let B_{UAV} denote the battery level threshold for the UAV to return to the charging station. The UAV is required to hold the constraint $b_{\mathrm{UAV}}(t) \geq B_{\mathrm{UAV}}$. $\Delta E_{\mathrm{UAV}}(t)$ is the energy consumption of the UAV; see (2). Since $\Delta E_{\mathrm{UAV}}(t)$ depends on the speed $v(t)$ of the UAV, the battery level of the UAV $b_{\mathrm{UAV}}(t)$ in the network state depends on the cruise control of the UAV.

At state α, the action of the UAV, including the next location and speed of the UAV and the selected ground device for data collection, is written as

$$a_\alpha = ((x'(\alpha), y'(\alpha), z), (v_x(\alpha), v_y(\alpha)), i_\alpha), \quad (5)$$

where $(x'(\alpha), y'(\alpha), z)$ is the next location of the UAV. $(v_x(\alpha), v_y(\alpha))$ is projection of the speed of the UAV on x or y plane at state α. i_α indicates the selected ground device at state α. $a_\alpha \in \mathcal{A}$, and $\mathcal{A}$ collects all actions that the UAV can take to optimize the next location and speed of the UAV and the selected ground device for data collection.

The reward (or penalty) $L\{\beta|\alpha, a_\alpha\}$ measures the packet loss when the UAV carries out action a_α and the network state transits from α to β. In other words, $L\{\beta|\alpha, a_\alpha\}$ computes the number of dropped or lost packets during the state transition, resulting from both buffer overflows and channel fading.

B. Onboard DDPG

The proposed DeFRA is depicted in Fig. 2, which consists of the DDPG-based onboard deep reinforcement learning and the LSTM-based state characterization layer. DeFRA leverages the actor-critic neural network structure to develop the DDPG-based onboard deep reinforcement learning [32]. DeFRA trains the DDPG onboard at the UAV to optimize instantaneous heading and speed of the UAV, and the selection of the ground devices in a continuous action space, where the UAV has no a-priori information on the state transition probabilities, i.e., $\Pr\{\beta|\alpha\}$. The packet loss of all ground devices (i.e., network cost) is minimized over the large, continuous state and action spaces.

DDPG applies a policy gradient scheme that applies a stochastic behavior policy for exploration but estimates a deterministic target policy. The deterministic policy gradients of the DDPG enable to optimally update the current policy by deterministically mapping network states to a specific action of the UAV. Moreover, the replay memory of the UAV, denoted by Δ_{replay}, is used to store the experience tuple $\left(\alpha, \beta, a_\alpha, L\{\beta|\alpha, a_\alpha\}\right)$ at each training step. K minibatches of experience are randomly sampled from Δ_{replay} to train the DDPG onboard along with state α of the environment.

The UAV can only observe the states of itself and its scheduled ground device at any moment, including its data buffer lengths, battery levels, and channel gains. With ground device i_α selected at state α, the observed part of the network state at the UAV is $\{b_{\mathrm{UAV}}(\alpha), b_{i_\alpha}(\alpha), q_{i_\alpha}(\alpha), g_{i_\alpha}(\alpha), (x(\alpha), y(\alpha), z)\}$. The UAV evaluates the packet loss resulting from the device selection, based on the observed part of the network state. An experience replay from the replay memory in the DDPG complements the remaining part of the network state, i.e., the states of the unselected ground nodes at state α, which cannot be observed instantaneously at the UAV. The historical records in the experience replay memory (or predictions derived from a new LSTM-based characterization layer, as will be described in Section IV-C) have to be utilized for the packet loss evaluation. With the experience replay of the unscheduled ground devices (in addition to the observations of the selected ground devices), the UAV approximates the new states of the ground devices, evaluates lost data packets, and generate a piece of training experience.

With the continuous flight resource allocation, the action-value function $Q\{\alpha_t, a_\alpha\}$ is differentiable in respect of the actions of the UAV. This allows for the setup of a gradient-assisted training $\mu\{\alpha_t\}$ to optimize the action of the UAV. Instead of to minimize $Q\{\alpha_t, a_\alpha\}$, In particular, the proposed DeFRA approximates the optimal actions of the UAV for the flight resource allocation with $Q\{\alpha_t, \mu\{\alpha_t\}\}$, which refrains from exhaustively evaluating all the actions in DDPG.

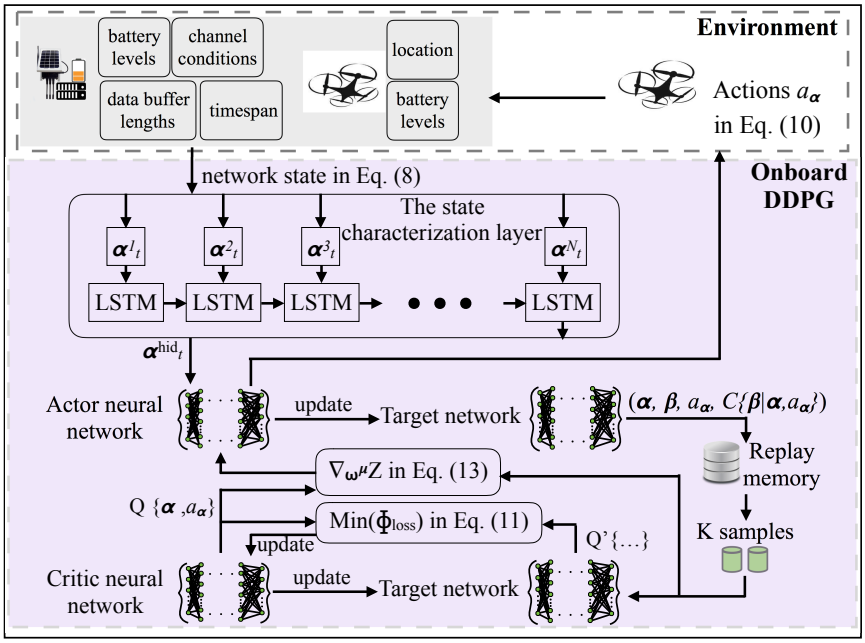


Fig. 2: The architecture of DeFRA, where the new state characterization layer predicts the time-varying network dynamics in the environment, while the onboard DDPG optimizes the continuous flight resource allocation.

To obtain $\mu\{\alpha_t\}$, the actor neural network takes the actions of setting $(x'(t), y'(t), z)$ and $(v_x(t), v_y(t))$, and the selection of device i_t $(1 \leq i_t \leq N)$. With the observed state α and action a_{α}, the optimal action-value function $Q\{\alpha_t, a_{\alpha}\}$ is approximated by the critic neural network, which obtains the expected overall data loss, i.e., the network cost. We denote $\mu\{\alpha_t|w^{\mu}\}$ and $\mu'\{\alpha_t|w^{\mu'}\}$ as the flight control and device selection policy of the actor neural network, and the target actor's policy, respectively. w^{μ} and $w^{\mu'}$ are their weights for the policy update. Fig. 2 depicts that the optimal $Q\{\alpha_k, a_{\alpha_k}\}$ is learned by the critic neural network, where K samples are taken from the experience memory. By adjusting the weight of the critic neural network w^{Q}, the critic neural network minimizes the approximation loss Φ_{loss} to minimize the following Bellman equation:

$$\Phi_{\text{loss}} = \frac{1}{K}\sum_{k}\Big(L\{\beta_k|\alpha_k, a_{\alpha_k}\} + \delta Q'\{\alpha_{k+1}, \mu'\{\alpha_{k+1}|w^{\mu'}\}| w^{Q'}\} - Q\{\alpha_k, a_{\alpha_k}|w^{Q}\}\Big)^2 \tag{6}$$

where δ is the discount factor. In the target critic neural network, $Q'\{\cdot\}$ gives the action-value function for evaluating the $\mu'\{\alpha_t|w^{\mu'}\}$.

The onboard DDPG conducted at the UAV aims to minimize the expected network cost, i.e., $\mathbb{E}[Q\{\alpha_t, a_{\alpha}\}]$. DeFRA updates the $\mu\{\alpha_t|w^{\mu}\}$ by applying the chain rule to obtain the expected data loss. Given the initialized distribution Z according to w^{μ}, we define the gradient of the DDPG policy as

$$\nabla_{w^{\mu}} Z \approx \mathbb{E}_{\alpha_t}\Big[\nabla_{w^{\mu}} Q\{\alpha_t, a_{\alpha}|w^{Q}\}|_{a_{\alpha}=\mu(\alpha_t|w^{\mu})}\Big] \tag{7}$$

Furthermore, Fig. 2 presents that K minibatches in the replay memory Δ_{replay} are used to train the actions of the UAV. Thus,

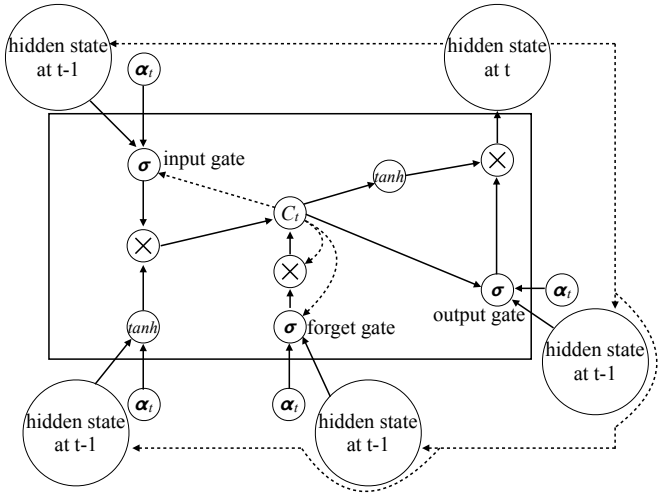


Fig. 3: The LSTM cell at the state characterization layer, where the historical information of α_t is taken as the input sequences. The hidden states α_t^{hid} are returned to make accurate prediction of the network dynamics.

by taking an average value of the sum of the gradients from the Δ_{replay}, we can obtain the $\nabla_{w^{\mu}} Z$ in (7), i.e.,

$$\nabla_{w^{\mu}} Z \approx \frac{1}{K}\sum_{k}\nabla_{a_{\alpha}} Q\{\alpha_k, a_{\alpha}|w^{Q}\}|_{a_{\alpha}=\mu(\alpha_k)} \times \nabla_{w^{\mu}}\mu\{\alpha_k|w^{\mu}\} \tag{8}$$

C. LSTM-based state characterization layer

The network dynamics resulting from time-varying data arrivals, energy harvesting, and channel fading, lead to unknown

network state transitions, increase learning uncertainties, and reduce learning accuracy. In particular, the UAV running the DeFRA onboard cannot observe the instantaneous, complete states of all the ground devices. It can only make the observation of a device, when the device is selected and transmits its state information to the UAV. The incomplete knowledge of the states of the devices can compromise the learning efficiency and accuracy of the DDPG-based DeFRA. For this reason, a state characterization layer is developed to predict the states of the devices which are not observable, and feed the predicted states into the DDPG-based decisions of flight resource allocation. The state characterization layer is based on LSTM.

LSTM is widely used in deep neural networks when the input data is time-varying, because of its ability to capture long-term (often unknown) dependencies of sequential data. LSTM consists of cell memory that stores the summary of the past input sequence, and the gating mechanism by which the information flow between the input, output, and cell memory is controlled. As shown in Fig. 2, the network states are fed into LSTM one by one (one at each step). The last hidden state α_t^{hid} is returned as the output of the state characterization layer.

Let o_t, C_t, f_t, and p_t denote the output gate, cell activation vectors, forget gate, and input gate of the LSTM layer at time t, respectively. According to the LSTM cell structure in Fig. 3, the LSTM processes the input sequence of α_t by adding new information into a memory, and using the gates that control the extent to which new information is memorized, old information is discarded, and current information is utilized. The hidden states α_t^{hid} is calculated by the following composite function.

$$\alpha_t^{\text{hid}} = o_t \tanh(C_t) \tag{9}$$

$$o_t = \sigma(W_o\alpha_t + W_o\alpha_{t-1}^{\text{hid}} + W_oC_t + e_o) \tag{10}$$

$$C_t = f_tC_{t-1} + p_t \tanh(W_c\alpha_t + W_c\alpha_{t-1}^{\text{hid}} + e_c) \tag{11}$$

$$f_t = \sigma(W_f\alpha_t + W_f\alpha_{t-1}^{\text{hid}} + W_fC_{t-1} + e_f) \tag{12}$$

$$p_t = \sigma(W_p\alpha_t + W_p\alpha_{t-1}^{\text{hid}} + W_pC_{t-1} + e_p) \tag{13}$$

where σ is the logistic sigmoid function, $\{W_o, W_c, W_f, W_p\} \in \mathbb{R}^{N\times 2N}$ is the weight matrix, and $\{e_o, e_c, e_f, e_p\} \in \mathbb{R}$ is the bias matrix.

The LSTM-based state characterization layer learns from past observations to adjust the weight and bias to predict future states of the devices (i.e., energy and data arrivals) and assist with the DDPG-based decisions. As illustrated in Fig. 2, we propose that for each ground device, an LSTM is maintained at the UAV. Whenever the UAV selects a device, the device reports its past and unreported states (associated with each of the time slots since the last report of the device). The reports are sequentially fed into to the LSTM as the input. By this means, the LSTM can obtain the complete (yet outdated) states of a device, based on which the future states of the device are predicted and exported to the DDPG.

Algorithm 1 summarizes the DeFRA with the LSTM-based characterization layer. The number of training episodes is M, where the length of each episode is t_{learning}. At every time step, the UAV takes an action a_α with a random process ζ_t for exploring the action space. Thus, we have

$$a_\alpha = \mu\{\alpha_t|w^\mu\} + \zeta_t. \tag{14}$$

Δ_{replay} is applied to store the experience of training flight control and the selection of device i, i.e., $(\alpha, \beta, a_\alpha, L\{\beta|\alpha, a_\alpha\})_t$. K minibatches are sampled in Δ_{replay} to minimize Φ_{loss}. Furthermore, DeFRA utilizes the sampled policy gradients in (8) to update the actor policy at the UAV. As the $\mu\{\alpha_t|w^\mu\}$ is optimized, the $Q'\{\cdot\}$ (in (6)) and $\mu'\{\alpha_t|w^{\mu'}\}$ onboard at the UAV are updated by

$$\begin{cases} w^{Q\prime} & \leftarrow \epsilon w^Q + (1-\epsilon)w^{Q\prime}; \\ w^{\mu\prime} & \leftarrow \epsilon w^\mu + (1-\epsilon)w^{\mu\prime}, \end{cases} \tag{15}$$

where the parameter ϵ is typically set to a small value such that the target networks are slightly updated. In our implementation, we set ϵ = 0.001.

DeFRA updates the Δ_{replay} based on the observation and evaluation of the actor and critic neural networks. Particularly,

Algorithm 1 Training of DeFRA

1. Initialize:
$\alpha, \beta \in \mathcal{S}$, $a_\alpha \in \mathcal{A}$, $b_{\text{UAV}}(t)$, t_{learning}, and Δ_{replay}.
The critic neural network $Q\{\alpha_t, a_\alpha|w^Q\}$ and the actor neural network $\mu\{\alpha_t|w^\mu\}$.
The target critic neural networks Q' with $w^{Q'} \leftarrow w^Q$. The target actor neural network μ' with $w^{\mu'} \leftarrow w^\mu$.
The state characterization layer with $\{W_o, W_c, W_f, W_p\}$ and $\{e_o, e_c, e_f, e_p\}$.
2. Learning:
for episode $1, \cdots, M$ **do**
  Train the state characterization layer based on the datasets $\rightarrow \alpha_t^{\text{hid}}$.
  State α is observed by the UAV.
  while $t \leq t_{\text{learning}}$ **do**
    Update $b_{\text{UAV}}(t)$ according to Eq. (4).
    if $b_{\text{UAV}}(t) >= B_{\text{UAV}}$ **then**
      The UAV carries out an action a_α to set $(x'(\alpha), y'(\alpha), z)$ and $(v_x(\alpha), v_y(\alpha))$, and select a ground device. $a_\alpha = \mu\{\alpha_t|w^\mu\} + \zeta_t$.
      The UAV calculates $L\{\beta_t|\alpha_t, a_\alpha\}$. A new state observation β is obtained.
      $(\alpha, \beta, a_\alpha, L\{\beta|\alpha, a_\alpha\})_t \rightarrow \Delta_{\text{replay}}$.
      K minibatches are randomly taken from the Δ_{replay} onboard at the UAV.
      $y_k = L\{\beta_k|\alpha_k, a_{\alpha_k}\} + \delta Q'\{\alpha_{k+1}, \mu'\{\alpha_{k+1}|w^{\mu'}\}|w^{Q'}\}$. Minimizing Φ_{loss} in (6).
      Based on (8), the actor policy is updated with the sampled policy gradients.
      With the optimized actor policy, $w^{Q'} \leftarrow \epsilon w^Q + (1-\epsilon)w^{Q'}$ and $w^{\mu'} \leftarrow \epsilon w^\mu + (1-\epsilon)w^{\mu'}$.
    else
      The UAV returns to the charging station.
    end if
  end while
end for

the training experience – in terms of the ground device selection, the data loss and the state information of all the unscheduled ground devices – is associated with the timespan in the network state, which is added to the Δ_{replay}. By performing the experience replay, DeFRA optimizes the actions of the UAV by learning online the latent energy and data arrival patterns, as well as the channel dynamics between the UAV and the ground device.

V. Implementation and Validation

In this section, we first present the implementation of DeFRA on Google TensorFlow, which is a symbolic math library based on dataflow and differentiable programming. Numerical results show the packet loss according to the training episodes and number of ground devices. The flight resource allocation achieved by DeFRA is also evaluated under different learning settings, and compared with existing deep reinforcement learning solutions.

A. Experimental datasets for training LSTM-based state characterization layer

A UAV-based communication testbed is built, as summarized in [33], where the UAV (as shown in Fig. 4(a)) patrols along a predetermined trajectory to relay sensory data of the ground devices. Outdoor experiments are conducted to measure the real-time channel gain between the UAV and the ground device. Fig. 4(b) plots 2500 data samples in the collected dataset, where the channel gains are dramatically effected by the movement of the UAV. The channel gain increases when the UAV gets closer to the ground device.

An energy harvesting-powered WSN [34] is deployed to monitor surrounding environmental information. As shown in Fig. 4(c), the sensor node is equipped with solar panels to charge its battery. Fig. 4(d) presents the voltage readings of the battery over 9 days. It can be observed that the battery is periodically charged with a high energy since the solar panel harvests energy during the day time.

The datasets of channel gains and solar charging voltages are used to train the state characterization layer of DeFRA. The datasets are firstly normalized in TensorFlow. Then, LSTM is implemented in Keras (the Python deep learning library [35]) to predict the future channel gain or solar charging energy. In addition, DDPG is configured in TensorFlow to minimize the training loss.

B. Implementation and Training of DeFRA

We implement the proposed DeFRA in Google TensorFlow with Python 3.5. TensorFlow is set up on a Linux workstation with 64-bit Ubuntu 18.04. DeFRA trains the flight resource allocation for 1000 episodes, during which a session is created in TensorFlow to enable DDPG with 2 hidden layers. The holder of the network state and the network cost is initialized to feed the knowledge of the current state and the next state to the tensors. The average loss values are computed across dimensions of the tensor, and the loss function Φ_{loss} is minimized. The experience replay memory with capacity of 10,000 training samples is created in DDPG, and stores the learning experiences, i.e., *(current state, next states, actions of the UAV, network cost)* at every step. Furthermore, the predicted channel gain and solar charging, which is trained by the state characterization layer, is memorized as hidden states, and used to update the next state in DDPG.

TABLE II: Simulation parameters

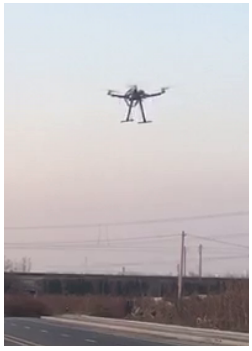
Parameters	Values
Battery capacity of the ground device (E)	800
Data buffer size ($\mathcal{Q}$)	100
Speed limit of the UAV (V_{max})	15
Air density in kg/m^3 (ρ_{air})	1.225
Rotor disc area in m^2 (S_{rotor})	0.79
Tip speed of the rotor blade ($\omega(t)$)	200
Fuselage drag ratio (ξ_{drag})	0.3
Rotor solidity (ξ_{rotor})	0.05
Mean rotor induced velocity in hover (v_0)	7.2
Battery level threshold of the UAV (B_{UAV})	100
Number of episodes (M)	1000
Discount factor (δ)	0.99

N ground devices (N is from 50 to 300) are uniformly distributed in the area of interest, which is a 1,000 m $\times$ 1,000 m square area. Each of the ground devices is equipped with a battery with capacity of 800 Joules, and the UAV is equipped with a battery with capacity of 250 Kilojoules. The speed limit of the UAV is 15 m/s. The number of epochs for training the state characterization layer is set to 10, 100, or 500. A training ratio is configured to control the amount of data in the datasets being utilized for the LSTM training in the state characterization layer. Moreover, the learning rate for the actor and critic in DDPG is 0.001, while the minibatch in Δ_{replay} has 100 samples. Table II specifies the configuration of simulation parameters.

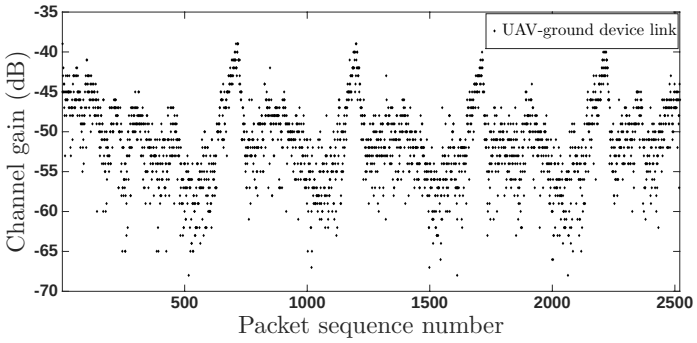
C. Performance of DeFRA

Fig. 5 plots the packet loss rate at each episode, given t_{learning} = 100, 400 and 800. The packet loss of DeFRA is high at the beginning of the learning process. With an increasing number of episodes, the acquired learning experience in the Δ_{replay} increases. The packet loss drops significantly in the first 400 episodes, and maintains a stable value afterward. The convergence of DeFRA is because network dynamics are predicted by the state characterization layer while the actions of the UAV are sufficiently trained by the actor and the critic neural networks in DDPG. Moreover, the packet loss rate of DeFRA (t_{learning} = 400 or 800) is slightly higher than the one with t_{learning} = 100. The reason is that a long t_{learning} extends the data generation of the ground devices, thus, more unscheduled ground devices suffer from buffer overflows.

Fig. 6 presents the prediction accuracy of the channel gain and energy harvesting, which is achieved by the state characterization layer in DeFRA. The difference value is calculated according to |the predicted value - the ground truth|, where the ground truth is the source data in the datasets. In Fig. 6(a), the state characterization layer with LSTM training epochs = 10 has the lowest prediction accuracy of the channel gain, while the one with 500 training epochs of the LSTM significantly reduces the difference value to 2 dB. This is also

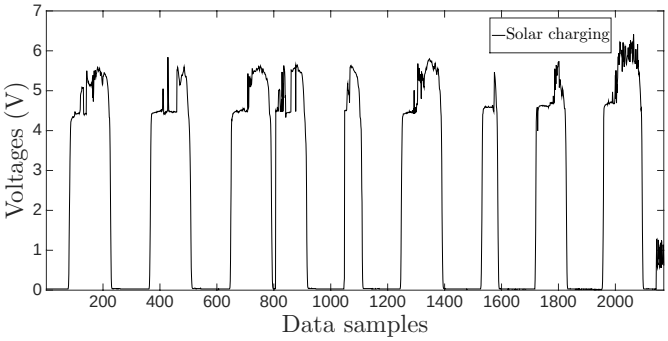


(a) The UAV patrols along the trajectory.



(b) The channel gain dataset.

(c) The energy harvesting ground device.



(d) Dataset on the solar charging voltages.

Fig. 4: Datasets of channel gains and solar charging voltages are collected from the real-world UAV (as shown in (a) [33]) and the ground sensing device (as shown in (c) [34]) to train the state characterization layer. 2500 data samples in the collected dataset are plotted in (b), and (d) presents the voltage readings of the battery over 9 days.

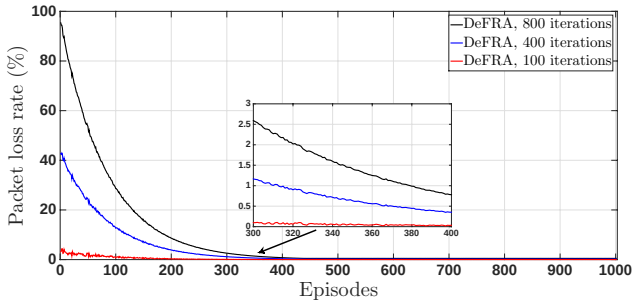


Fig. 5: Packet loss rate of DeFRA with regards to the training episodes.

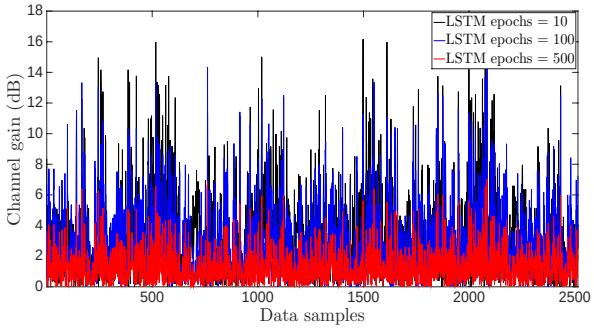
observed in Fig. 6(b), which shows the difference value of the solar charging voltage. The LSTM training epochs = 500 achieves the lowest difference between the prediction and the ground truth.

Fig. 7 plots the flight trajectories of the UAV with regards to different numbers of LSTM epochs and t_{learning} of DDPG. As observed, DeFRA persistently adjusts the trajectory of the UAV, where the actions of $(x'(\alpha), y'(\alpha), z)$ and $(v_x(\alpha), v_y(\alpha))$ are optimized in the continuous action space. In Fig. 7(a), the state characterization layer is unlikely to make accurate prediction of network dynamics due to a short LSTM training time. Thus, DeFRA hardly optimizes the flight resource allocation of the UAV. Moreover, a small number of t_{learning} in DeFRA result in insufficient experience in the replay memory, which gives rise to incomplete trajectory planning of the UAV. In Fig. 7(b), by extending the training of DeFRA, the state characterization layer and DDPG are adequately trained to minimize the approximation loss Φ_{loss}.

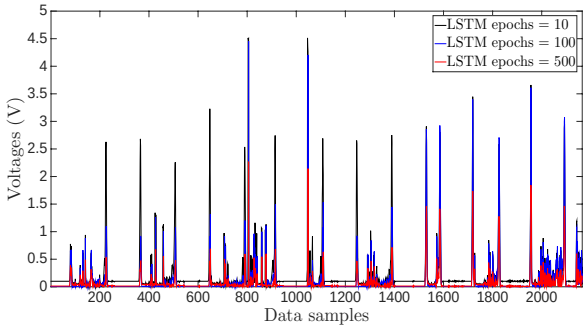
D. Performance comparison

For performance comparison, we compare the proposed DeFRA with two deep reinforcement learning based policies and two non-learning heuristics.

- DDPG based movement control (DDPG-MC) [23]. DDPG is carried out in the continuous action space for the trajectory planning of the UAV. Particularly, the network states in the training environment are randomly generated. In other words, DDPG-MC is trained with no predicted knowledge of network dynamics, which result from time-varying airborne channels and energy arrivals at the ground devices.
- Deep Q-Networks based flight resource allocation policy (DQN-FRA) [24], [36]. DQN-FRAS maintains two separate neural networks at the UAV, an evaluation DQN and a target DQN, which are alternatively updated to minimize the network cost. Since DQN is expected to the low dimensional discrete action space, the trajectory of the UAV is discretized as 50 waypoints in DQN-FRA.

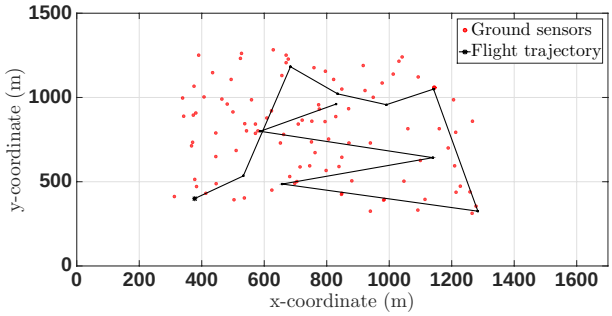


(a) Difference of the channel gain.

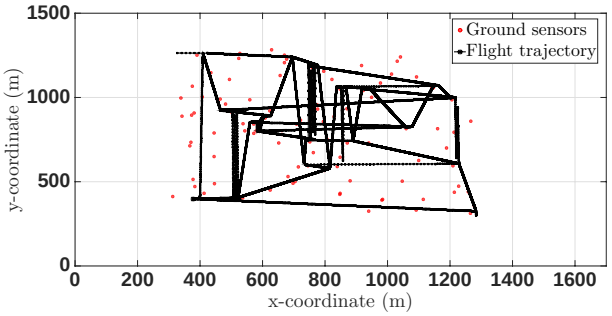


(b) Difference of the energy harvesting.

Fig. 6: The difference between the experimental ground truth and the predicted value which is achieved by DeFRA. The number of epochs of LSTM is 10, 100, or 500.



(a) LSTM epochs = 10, t_{learning} = 100.



(b) LSTM epochs = 500, t_{learning} = 800.

Fig. 7: The flight trajectories of the UAV with regards to different number of LSTM epochs and training iterations of DDPG.

- Channel aware waypoint selection (CAWS). This heuristic assumes that the UAV is aware of a-priori knowledge on the channel gains. The next waypoint of the UAV is designed to fly over and schedule the ground device which has the highest channel gain.
- Planned trajectory random scheduling (PTRS). 50 waypoints are predetermined to cover the targeted field. The UAV moves along the fixed trajectory, while one ground device is randomly scheduled to transmit. Namely, the trajectory planning and communication scheduling of PTRS are independent of the time-varying network states.

Fig. 8 plots the packet loss rate of DeFRA, DDPG-MC, and DQN-FRA, with the increase of training episodes. Without loss of generality, we take three representative configurations of the proposed DeFRA. We can see that in general, DeFRA and DDPG-MC achieve faster convergence than DQN-FRA. DeFRA achieves the smaller packet loss under the configuration of 100 LSTM epochs and the training ratio of 0.7 than it does under the other two considered configurations. The reason is that with more epochs and more training data, the state characterization layer of DeFRA can predict the time-varying channel and solar charging more effectively in the learning environment. Therefore, DDPG trains the actions of the UAV with the state information of all the ground devices to minimize the packet loss.

Fig. 9 depicts the packet loss rate of DeFRA, DDPG-MC,

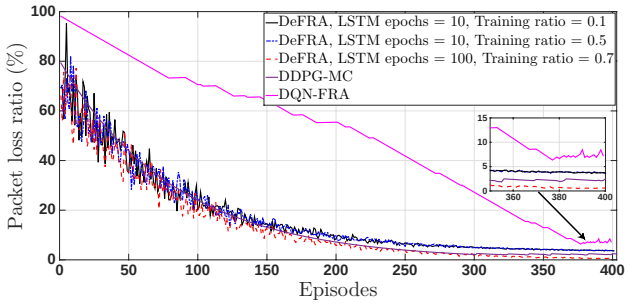


Fig. 8: Packet loss rate of DeFRA, DDPG-MC, and DQN-FRA, with regard to the training episodes.

DQN-FRA, CAWS, and PTRS, where $N \in [50, 300]$. In general, the packet loss rate grows with the network size since more ground devices have to buffer their data while one device is scheduled to transmit data. The deep reinforcement learning based policies, i.e., DeFRA, DDPG-MC, and DQN-FRA, outperform CAWS and PTRS since the deep neural networks explore every possible action of the UAV to minimize the packet loss. Particularly, the actor-critic based policies, i.e., DeFRA and DDPG-MC, achieve similar performance when N is smaller than 150 devices. When the number of ground devices is 300, the packet loss rate of the proposed DeFRA is

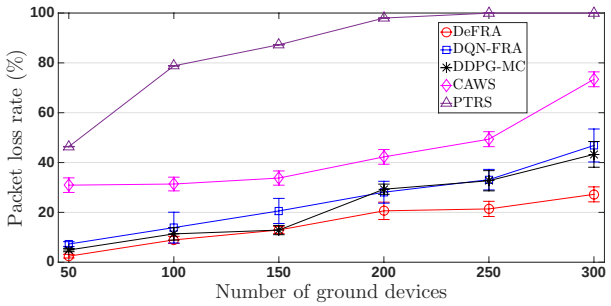


Fig. 9: Packet loss rate in regard to $N \in [50, 300]$. The error bar presents the standard deviation over 20 experiments.

about 15% and 19% lower than DDPG-MC and DQN-FRA. This is because DeFRA with the state characterization layer learns the network state dynamics of all the ground devices. By taking advantage of the precise prediction of LSTM, the hidden states stored in the experience replay memory are used to train the actions of the DDPG, which leads to the minimized approximation loss Φ_{loss}.

DDPG-MC and DQN optimize both UAV's trajectory and communication schedule with the ground devices in the current paper. In contrast, the DQN developed in [23] only optimized the communication schedule, where the trajectory of the UAV was given in prior.

E. Ablation study for the state characterization layer

The proposed DeFRA is compared with DDPG-MC in which the action of the UAV is trained without the LSTM-based state characterization layer. The other modules and configurations remain the same as described at the beginning of this section. Figs. 8 and 9 show that the proposed LSTM-based state characterization layer of DeFRA can effectively deal with the partial observability of the UAV on the states of the ground devices in the sense that it can help approximate the obscure states of unselected devices at every instant for the follow-on DDPG operation. Particularly, DeFRA with the LSTM-based state characterization layer achieves 15% lower packet loss rate than DDPG-MC, since historical information can be encoded in the hidden state of the LSTM cell to help make accurate prediction. DeFRA takes advantage of the prediction of the network states to train the actions of the UAV. Furthermore, the state characterization layer accelerates the convergence of DeFRA. This is due to the fact that the predicted network states enrich the training environment of DDPG, and the training time of the actor and the critic neural networks is shortened.

VI. Conclusions

This paper developed a new deep reinforcement learning based flight resource allocation framework, namely DeFRA, to minimize the overall data packet loss in a continuous action space. DeFRA based on DDPG jointly optimizes the instantaneous heading and cruising speed of the UAV, as well as the selection of ground devices for data collection. The new state characterization layer leverages LSTM to predict the time-varying airborne channels, and the data and energy arrivals at the ground devices. Experimental data was collected from the real-world UAV testbed and the energy harvesting-powered WSN, and utilized to train the actions of the UAV.

Acknowledgements

This work was partially supported by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology), within the CISTER Research Unit (UIDP/UIDB/04234/2020); also by national funds through the FCT, under CMU Portugal partnership, within project CMU/TIC/0022/2019 (CRUAV).

References

[1] D. K. Sah and T. Amgoth, "Renewable energy harvesting schemes in wireless sensor networks: A survey," *Information Fusion*, vol. 63, pp. 223–247, 2020.

[2] Z. Li, Y. Jiang, Y. Gao, L. Sang, and D. Yang, "On buffer-constrained throughput of a wireless-powered communication system," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 2, pp. 283–297, 2018.

[3] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 3, pp. 443–461, 2010.

[4] F. Qi, X. Zhu, G. Mang, M. Kadoch, and W. Li, "UAV network and IoT in the sky for future smart cities," *IEEE Network*, vol. 33, no. 2, pp. 96–101, 2019.

[5] D. Liu, Y. Xu, J. Wang, J. Chen, K. Yao, Q. Wu, and A. Anpalagan, "Opportunistic UAV utilization in wireless networks: Motivations, applications, and challenges," *IEEE Communications Magazine*, vol. 58, no. 5, pp. 62–68, 2020.

[6] P. Spachos and S. Gregori, "Integration of wireless sensor networks and smart UAVs for precision viticulture," *IEEE Internet Computing*, vol. 23, no. 3, pp. 8–16, 2019.

[7] H. Sharma, A. Haque, and Z. A. Jaffery, "Solar energy harvesting wireless sensor network nodes: A survey," *Journal of Renewable and Sustainable Energy*, vol. 10, no. 2, p. 023704, 2018.

[8] W.-K. Lee, M. J. Schubert, B.-Y. Ooi, and S. J.-Q. Ho, "Multi-source energy harvesting and storage for floating wireless sensor network nodes with long range communication capability," *IEEE Transactions on Industry Applications*, vol. 54, no. 3, pp. 2606–2615, 2018.

[9] B. Galkin, J. Kibilda, and L. A. DaSilva, "UAVs as mobile infrastructure: Addressing battery lifetime," *IEEE Communications Magazine*, vol. 57, no. 6, pp. 132–137, 2019.

[10] K. Li, W. Ni, E. Tovar, and A. Jamalipour, "Online velocity control and data capture of drones for the internet-of-things: An onboard deep reinforcement learning approach," *IEEE Vehicular Technology Magazine*, 2020.

[11] W. Chen, S. Zhao, R. Zhang, Y. Chen, and L. Yang, "UAV-assisted data collection with nonorthogonal multiple access," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 501–511, 2020.

[12] J. Li, H. Zhao, H. Wang, F. Gu, J. Wei, H. Yin, and B. Ren, "Joint optimization on trajectory, altitude, velocity, and link scheduling for minimum mission time in UAV-aided data collection," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1464–1475, 2019.

[13] P. Tong, J. Liu, X. Wang, B. Bai, and H. Dai, "UAV-enabled age-optimal data collection in wireless sensor networks," in *IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2019, pp. 1–6.

[14] M. B. Ghorbel, D. Rodríguez-Duarte, H. Ghazzai, M. J. Hossain, and H. Menouar, "Joint position and travel path optimization for energy efficient wireless data gathering using unmanned aerial vehicles," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2165–2175, 2019.

[15] C. You and R. Zhang, "Hybrid offline-online design for UAV-enabled data harvesting in probabilistic LoS channels," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 3753–3768, 2020.

[16] J. Baek, S. I. Han, and Y. Han, "Optimal UAV route in wireless charging sensor networks," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1327–1335, 2019.

[17] Y. Du, K. Yang, K. Wang, G. Zhang, Y. Zhao, and D. Chen, "Joint resources and workflow scheduling in UAV-enabled wirelessly-powered MEC for IoT systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 10 187–10 200, 2019.
[18] Y. Zhang, Z. Mou, F. Gao, J. Jiang, R. Ding, and Z. Han, "UAV-enabled secure communications by multi-agent deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 11 599–11 611, 2020.
[19] W. Koch, R. Mancuso, R. West, and A. Bestavros, "Reinforcement learning for UAV attitude control," *ACM Transactions on Cyber-Physical Systems*, vol. 3, no. 2, pp. 1–21, 2019.
[20] D. Chen, Q. Qi, Z. Zhuang, J. Wang, J. Liao, and Z. Han, "Mean field deep reinforcement learning for fair and efficient UAV control," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 813–828, 2020.
[21] S. F. Abedin, M. S. Munir, N. H. Tran, Z. Han, and C. S. Hong, "Data freshness and energy-efficient UAV navigation optimization: A deep reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, 2020.
[22] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 2059–2070, 2018.
[23] K. Li, W. Ni, and F. Dressler, "Continuous maneuver control and data capture scheduling of autonomous drone in wireless sensor networks," *IEEE Transactions on Mobile Computing*, 2021.
[24] K. Li, W. Ni, E. Tovar, and M. Guizani, "Joint flight cruise control and data collection in UAV-aided internet of things: An onboard deep reinforcement learning approach," *IEEE Internet of Things Journal*, 2020.
[25] A. Graves, "Long short-term memory," in *Supervised sequence labelling with recurrent neural networks*. Springer, 2012, pp. 37–45.
[26] 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.
[27] L. Xie, J. Xu, and Y. Zeng, "Common throughput maximization for UAV-enabled interference channel with wireless powered communications," *IEEE Transactions on Communications*, vol. 68, no. 5, pp. 3197–3212, 2020.
[28] Y. Emami, K. Li, and E. Tovar, "Buffer-aware scheduling for UAV relay networks with energy fairness," in *IEEE Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–5.
[29] K. Li, W. Ni, X. Wang, R. P. Liu, S. S. Kanhere, and S. Jha, "Energy-efficient cooperative relaying for unmanned aerial vehicles," *IEEE Transactions on Mobile Computing*, no. 6, pp. 1377–1386, 2016.
[30] C. R. Ashokkumar and G. W. York, "Observer based controllers for UAV maneuver options," in *AIAA guidance, navigation, and control conference*, 2016, p. 0643.
[31] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2329–2345, 2019.
[32] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.
[33] K. Li, N. Lu, P. Zhang, W. Ni, and E. Tovar, "Multi-drone assisted Internet of Things testbed based on bluetooth 5 communications," in *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2020, pp. 345–346.
[34] B. P. L. Lau, T. Chaturvedi, B. K. K. Ng, K. Li, M. S. Hasala, and C. Yuen, "Spatial and temporal analysis of urban space utilization with renewable wireless sensor network," in *IEEE/ACM International Conference on Big Data Computing Applications and Technologies (BDCAT)*. IEEE, 2016, pp. 133–142.
[35] N. Ketkar, "Introduction to keras," in *Deep learning with Python*. Springer, 2017, pp. 97–111.
[36] K. Li, W. Ni, B. Wei, and E. Tovar, "Onboard double Q-learning for airborne data capture in wireless powered IoT networks," *IEEE Networking Letters*, vol. 2, no. 2, pp. 71–75, 2020.

Kai Li (S'09–M'14–SM'20) received the B.E. degree from Shandong University, China, in 2009, the M.S. degree from The Hong Kong University of Science and Technology, Hong Kong, in 2010, and the Ph.D. degree in Computer Science from The University of New South Wales, Sydney, Australia, in 2014. Currently, he is a senior research scientist with Real-Time and Embedded Computing Systems Research Centre (CISTER), Portugal. He is also a CMU-Portugal research fellow, which is jointly supported by Carnegie Mellon University, U.S., and The Foundation for Science and Technology (FCT), Portugal. Prior to this, Dr. Li was a postdoctoral research fellow at The SUTD-MIT International Design Centre, The Singapore University of Technology and Design, Singapore (2014-2016). He was a visiting research assistant at ICT Centre, CSIRO, Australia (2012-2013). From 2010 to 2011, he was a research assistant at Mobile Technologies Centre with The Chinese University of Hong Kong. His research interests include machine learning, vehicular communications and security, resource allocation optimization, Cyber-Physical Systems, Internet of Things (IoT), and UAV networks.

Dr. Li has been serving as the Associate Editor for Elsevier Ad Hoc Networks Journal and IEEE Access Journal, and the Demo Co-chair for ACM/IEEE IPSN 2018.

Wei Ni (M'09-SM'15) received the B.E. and Ph.D. degrees in Electronic Engineering from Fudan University, Shanghai, China, in 2000 and 2005, respectively. Currently, he is a Group Leader and Principal Research Scientist at CSIRO, Sydney, Australia, and an adjunct professor at the University of Technology Sydney and an Honorary Professor at Macquarie University, Sydney. Prior to this, he was a Postdoctoral Research Fellow at Shanghai Jiaotong University from 2005-2008; Deputy Project Manager at the Bell Labs, Alcatel/Alcatel-Lucent from 2005-2008; and Senior Researcher at Devices R&D, Nokia from 2008-2009. His research interests include signal processing, stochastic optimization, as well as their applications to network efficiency and integrity.

Dr Ni is the Chair of IEEE Vehicular Technology Society (VTS) New South Wales (NSW) Chapter since 2020 and an Editor of IEEE Transactions on Wireless Communications since 2018. He served first the Secretary and then Vice-Chair of IEEE NSW VTS Chapter from 2015-2019, Track Chair for VTC-Spring 2017, Track Co-chair for IEEE VTCSpring 2016, Publication Chair for BodyNet 2015, and Student Travel Grant Chair for WPMC 2014.

Falko Dressler (F'17) received the MSc and PhD degrees from the Department of Computer Science, University of Erlangen, in 1998 and 2003, respectively. He is full professor and Chair for Data Communications and Networking at the School of Electrical Engineering and Computer Science, TU Berlin. Dr. Dressler has been associate editor-in-chief for IEEE Trans. on Mobile Computing and Elsevier Computer Communications as well as an editor for journals such as IEEE/ACM Trans. on Networking, IEEE Trans. on Network Science and Engineering, Elsevier Ad Hoc Networks, and Elsevier Nano Communication Networks. He has been chairing conferences such as IEEE INFOCOM, ACM MobiSys, ACM MobiHoc, IEEE VNC, IEEE GLOBECOM. He authored the textbooks Self-Organization in Sensor and Actor Networks published by Wiley & Sons and Vehicular Networking published by Cambridge University Press. He has been an IEEE Distinguished Lecturer as well as an ACM Distinguished Speaker. Dr. Dressler is an IEEE Fellow as well as an ACM Distinguished Member. He is a member of the German National Academy of Science and Engineering (acatech). He has been serving on the IEEE COMSOC Conference Council and the ACM SIGMOBILE Executive Committee. His research objectives include adaptive wireless networking (radio, visible light, molecular communications) and embedded system design (from microcontroller to Linux kernel) with applications in ad hoc and sensor networks, the Internet of Things, and cooperative autonomous driving systems.