A Fast UAV Trajectory Planning Framework in RIS-assisted Communication Systems with Accelerated Learning via Multithreading and Federating

Jun Huang, *Senior Member, IEEE,* Beining Wu, *Student Member, IEEE,* Qiang Duan, *Senior Member, IEEE,* Liang (Leon) Dong, *Senior Member, IEEE,* and Shui Yu, *Fellow, IEEE*

Abstract—Reconfigurable Intelligent Surface (RIS)-assisted Unmanned Aerial Vehicle (UAV) communications have been realized as essential to space-air-group system integration in the 6G technology landscape. Trajectory planning plays a crucial role in RIS-assisted UAV communications to face the challenges of UAV's limited power capacities and dynamic wireless channels. Existing solutions assume complete channel state information, focus on single-rotor UAVs, and rely heavily on time-consuming training processes for machine learning; thus, they lack applicability to deal with highly dynamic real-world scenarios. To fill these research gaps, we aim to characterize RIS-assisted UAV communications and design responsive and accurate UAV trajectory planning algorithms in this paper. We first develop a communication model with incomplete information and an energy consumption model for quadrotor UAVs. We then formulate UAV trajectory planning as an optimization problem to minimize UAV's energy consumption while maintaining communication throughput. To solve this problem, we design an acceleration framework, *FedX*, for reinforcement learning (RL) solvers and present two fast trajectory planning algorithms, FedSAC and FedPPO, as instantiations of the *FedX* framework. Our evaluation results indicate that the proposed framework is effective and efficient–more than 3 times faster with 5 agents and 7 times faster with 10 agents than standard RL algorithms, making it suitable for using RL solvers within wireless networks and mobile computing environments. We also discuss and identify the pros and cons of our proposed framework.

Index Terms—UAV, RIS, trajectory planning, reinforcement learning, training acceleration, federating.

1 Introduction

Unmanned aerial vehicles (UAV), also known as drones, are aircraft without a human pilot onboard that are either controlled remotely by an operator or programmed to fly autonomously [1]. The surge in drone popularity spans diverse sectors such as Infrastructure, Agriculture, Transport, Entertainment, Security, and Insurance [2]. Recent research shows that combining Reconfigurable Intelligent Surfaces (RIS) with UAV communications has become essential for connecting space and terrestrial networks. It involves relaying data between UAVs and ground terminals (GTs) using RIS. GTs, which are fixed or mobile stations on the ground, serve as communication points with UAVs. RIS are engineered surfaces that can manipulate electromagnetic waves by reflecting, absorbing, or focusing them in specific directions. When integrated with UAV communications, RIS can greatly enhance signal strength, coverage, and reliability by adjusting the propagation environment dynamically. In urban areas, RIS is particularly useful for overcoming obstacles, reducing signal interference, and extending the range of UAV communications. This new technology facilitates the implementation of space-air-ground integrated systems within the 6G technology landscape [3], [4].

While RIS-assisted UAV communications bring opportunities to new networking paradigms, they also impose challenges for the system design [5]. Although traditional designs for terrestrial systems could potentially be modified for UAV communications, the distinct nature of UAV systems requires a more customized approach. The limited power capacity of UAVs and the highly dynamic wireless channel states are two special features that make RIS-assisted UAV communications particularly challenging. UAV trajectory planning plays a crucial role in facing this challenge by optimally planning the UAV trajectory and scheduling the ground terminal connection in order to minimize the UAV energy consumption while maintaining the required data transmission rate. However, prior research on UAV trajectory planning has limitations in the following aspects.

First, most current works assumed that the channel state information is completely available [6]–[11]. Although there

This work is supported by the National Science Foundation under Grant Number 2348422.

Jun Huang and Beining Wu are with the Department of Electrical Engineering and Computer Science, South Dakota State University, Brookings, 57006 SD. Email: Jun.Huang@sdstate.edu, Wu.Beining@jacks.sdstate.edu.

Qiang Duan is with the Department of Information Sciences and Technology, The Pennsylvania State University, Abington, 19001 PA. Email: qxd2@psu.edu.

Liang (Leon) Dong is with the Department of Electrical and Computer Engineering, Baylor University, Waco, 76798 TX. Email: Liang_Dong@baylor.edu.

S. Yu is with the School of Computer Science, University of Technology Sydney, Ultimo, NSW 2007, Australia. E-mail: Shui.Yu@uts.edu.au.

Manuscript received XXX, 2024; revised XXX, 2024.

has been a significant amount of studies in channel measurements and modeling for UAV communications [12]–[14], the presumption of perfect channel information is too idealistic in real-world scenarios. Second, prior research in UAV communications primarily considered single-rotor UAVs, which leads to concerns about the applicability of the existing methods to the quadrotor UAV scenarios [15]–[20], which are becoming more common in various applications. Also, existing studies typically only consider the vertical ascension of UAVs, which does not reflect the settings of real-world operations. Third, although reinforcement learning (RL) techniques have been employed for UAV trajectory planning and yielded promising results [6]–[11], [15]–[18], [21]–[26], previous solutions lack sufficient consideration of the time performance thus leading to less responsive control of the UAV trajectory, which might not able to face the highly dynamic environment of UAV communications, as illustrated in our recent investigation [27].

In a nutshell, prior studies on this research topic present the following significant research gaps.

1) Current research assumes that the channel state information of the UAV-RIS-GT system is completely known by the UAV trajectory planner, which is too idealistic and impractical in real-world scenarios.
2) Existing trajectory planning algorithms are dedicated to the cases of single-rotor UAVs, leaving the quadrotor UAV scenarios grossly uninvestigated. In addition, they consider UAV's vertical ascent only, which does not reflect practical settings.
3) Previous RL-based solutions, have not sufficiently considered the time performance despite their potential for promising results. The current lack of a fast UAV trajectory planning framework is a significant issue that needs to be addressed to face the challenge of highly dynamic UAV communications.

To fill in the above research gaps, this work aims to characterize RIS-assisted UAV communications and design responsive and accurate UAV trajectory planning algorithms leveraging computing acceleration techniques for machine learning. Specifically, we make the following contributions.

- We develop a new channel model for UAV-RIS-GT communications in urban areas. To define how signals fade over a wireless link between two entities, the new model is established to characterize the wireless channel under the incomplete information assumption.
- We design a quadrotor UAV energy consumption model based on the single-rotor case to precisely describe the UAV's movement in any direction, and we formulate an optimization problem for UAV trajectory planning under multiple constraints.
- We devise a fast UAV trajectory framework by integrating multithreading and federated learning techniques for training acceleration. Based on this framework, we present two fast yet accurate RL algorithms. Evaluations are conducted to reveal the performance of the algorithms.

The remainder of this paper is organized as follows. Section 2 briefly summarizes the related studies. Section 3 describes the system model and formulates the problem. In Section 4, we present our proposed approach for the defined problem. We discuss the simulation results in Section 5 and draw the conclusion in Section 6.

2 Related Work

Recent studies have shown that UAV trajectory planning is formulated with other system parameters, such as rate/capacity [6], [8], [15], [22], [26], phase shift [10], [17], [23], energy [11], [24], [28], and beamforming [7], [9], [18], [25]. Together with the trajectory design, these system parameters lead to the defined optimization problems NP-hard.

Traditional optimization methods, such as convex [29] and multi-objective optimization [30], have been effective in solving NP-hard problems for UAV path planning while ensuring service quality but barely adapting to dynamic and complex environments due to high computational overhead. Heuristic algorithms like Ant Colony Optimization [31], A* Search [32], and Particle Swarm Optimization [33] offer efficiency but rely on predefined system models, limiting their adaptability to partial or localized information and often getting trapped in local optima. Also, machine learning approaches, such as hybrid neural networks [34], demonstrate high adaptability for UAV trajectory planning but demand significant computational resources and hyperparameter tuning, making them challenging to deploy in real-world scenarios.

To address these limitations, various RL algorithms have been proposed. Among these studies, DQN (Deep Q-Network) is considered the most straightforward and effective approach. In the paper [6], the authors proposed a DQN to solve the problem of maximizing broadcast secrecy rate in UAV-Empowered IRS (Intelligent Reflecting Surfaces) backscatter communications. Another work by Sun *et al.* [21] considered the age of information and designed a DQN for aerial IRS-assisted IoT networks. In a recent study [26], the authors designed a DQN to enhance the security performance of UAV-RIS reflection systems. Several other forms of DQN have also been proposed, such as centralized-declined DQN [24] and Decaying DQN [9].

Although DQN-based algorithms can produce acceptable results for UAV trajectory planning, the Double DQN (DDQN) and deep deterministic policy gradient (DDPG) algorithms can provide even more precise results. Mei *et al.* [17] proposed a DDQN and a DDPG algorithm to solve 3D trajectory and phase shift design for RIS-assisted UAV systems. In Zhang et al. [15], a DDQN-based approach was presented for the same purpose with capacity maximization. Truong *et al.* [10] recently used a DDPG algorithm for joint flying IRS trajectory and phase shift design. In [11], Nguyen *et al.* developed a DDPG algorithm for RIS-assisted UAV communications with wireless power transfer in IoT scenarios. To address the multi-objective optimization issue, a multi-objective DDPG solver was proposed in [7] for trajectory optimization and beamforming design.

As DDPG employs the actor-critic learning framework to improve the accuracy of the solution, contemporary studies tend to utilize this framework to design more effective RL

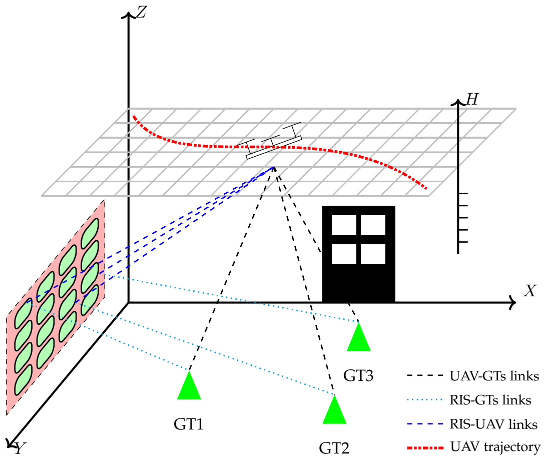


Fig. 1. System model of RIS-assisted UAV communications.

algorithms. In [8], a soft actor-critic reinforcement learning algorithm called DRRL (distributionally-robust RL) is developed. Qin *et al.* [35] proposed both centralized and decentralized SAC algorithms, effectively addressing the joint problem of UAV path planning and power allocation. Iacovelli *et al.* designed an actor-critic-inspired proximal policy optimization (PPO) for multi-UAV IRS-assisted communications [22]. In [11], a PPO was also developed. Dong *et al.* [36] optimized the UAV trajectory while considering channel state information based on the PPO algorithm. The authors of [23] presented a twin delayed deep deterministic policy gradient (TD3) algorithm for radio surveillance with a fixed-wing UAV to address the overestimate of Q-value by DDPG in the critic network. More recently, a multi-agent RL has been developed in [20] to optimize the energy consumption of the single-rotor UAV.

Our work differs from the studies mentioned above in three ways. First, we consider the communication model with the assumption of incomplete information, which is more general and applicable in real-world scenarios. Second, we develop a quadrotor UAV energy consumption model that is more commonly used and is expected to be more widespread in 6G communications. Third, we design a training acceleration framework for the RL solvers, which speeds up the training process, addressing a significant issue that current studies neglect.

3 System Model and Problem Formulation

3.1 Network Model

We consider a communication system that includes a UAV and a RIS connecting with K GTs (ground terminals) in an urban area, as shown in Fig. 1. The UAV acts as an aerial base station, while the RIS is positioned at the boundary of the service area to provide a line-of-sight connection to all GTs. Both the UAV and the ground user are equipped with a single antenna.

The RIS is made up of $M_c \times M_r$ passive reflection units (PRUs) arranged in a uniform planar array (UPA). The array consists of M_c PRUs spaced evenly at a distance of d_c meters, and M_r PRUs in each row also spaced evenly at a distance of d_r meters. To adjust the phase shift, each PRU applies an independent reflection coefficient that scatters the incoming signal with an amplitude $a \in [0,1]$ and a phase shift $\phi_{m_r,m_c} \in [-\pi,\pi)$. This means that the reflection coefficient $r_{m_r,m_c} = ae^{j\phi_{m_r,m_c}}$, where m_r belongs to the set of integers $1,2,\cdots,M_r$, and m_c belongs to the set of integers $1,2,\cdots,M_c$. The fixed reflection loss of the RIS is represented by a, while ϕ_{m_r,m_c} indicates the phase shift applied at the PRU (m_r,m_c).

Following the same convention of [28], we denote the length of a particular time slot as $\delta_t[n]$, and thus the overall flight time T is the sum of $\delta_t[n]$ for all n from 1 to N. The UAV's 3D path is represented by a sequence $\{\mathbf{q}[n] = [x[n], y[n], z[n]]^{\mathrm{T}}\}_{n=1}^{N}$, where $\mathbf{q}[n] = [x[n], y[n], z[n]]^{\mathrm{T}}$ denotes the 3D coordinates of the UAV at time slot n. The altitude that the UAV can fly at, denoted by H, must satisfy the safety regulations and is within the range $H_U^{\min} \leq z[n] \leq H_U^{\max}$. The locations of the ground terminals are fixed and denoted by $\mathbf{L}_k = [x_k, y_k, 0]^{\mathrm{T}}$, where $\mathbf{L}_k$ represents the coordinates of ground terminal k. The RIS is situated on a building wall at a certain altitude H_I, i.e., $\mathbf{L}_I = [x_I, 0, H_I]^{\mathrm{T}}$.

Let UG be the link between UAV and ground terminal k, UI be the link between UAV and RIS, IG be the link between RIS and ground terminal k, so we calculate the distance between the UAV and ground user k during time slot n as $d_k^{\mathrm{UG}}[n] = ||\mathbf{q}[n] - \mathbf{L}_k||$, the distance between the UAV and the RIS as $d^{\mathrm{UI}}[n] = ||\mathbf{q}[n] - \mathbf{L}_I||$ and the distance between the RIS and ground terminal k as $d_k^{\mathrm{IG}}[n] = ||\mathbf{L}_I - \mathbf{L}_k||$. The distances $d_k^{\mathrm{UG}}[n]$ and $d^{\mathrm{UI}}[n]$ remain constant within each time slot δ_t since the UAV's movement during δ_t is ignorable compared to these distances.

Due to the substantial path loss and reflection loss, we neglect the power of signals that undergo multiple reflections by the RIS [14].

3.2 Channel Model

We assume that the system utilizes orthogonal frequency division multiple access (OFDMA) and the total system bandwidth B is divided into N_f sub-carriers with sub-carriers spacing $\Delta f = \frac{B}{N_f}$. In time slot n, the channel vector between the UAV and the RIS on sub-carrier i can be given by [16]:

$$\mathbf{h}_i^{\mathrm{UR}}[n] = \sqrt{\frac{\beta_0}{(d^{\mathrm{UR}}[n])^2}}\, e^{-j2\pi i \Delta f \frac{d^{\mathrm{UR}}[n]}{c}} \mathbf{h}_{\mathrm{LoS}}^{\mathrm{UR}}[n], \tag{1}$$

where

$$\begin{aligned}\mathbf{h}_{\mathrm{LoS}}^{\mathrm{UR}}[n] = \Big[1, e^{-j2\pi f_c \frac{d_r \sin\theta^{\mathrm{UR}}[n] \sin\xi^{\mathrm{UR}}[n]}{c}}, \cdots,\\ e^{-j2\pi f_c (M_r-1)\frac{d_r \sin\theta^{\mathrm{UR}}[n]\sin\xi^{\mathrm{UR}}[n]}{c}}\Big]^{\mathrm{T}}\\ \otimes \Big[1, e^{-j2\pi f_c \frac{d_c \sin\theta^{\mathrm{UR}}[n]\sin\xi^{\mathrm{UR}}[n]}{c}}, \cdots,\\ e^{-j2\pi f_c (M_c-1)\frac{d_c \sin\theta^{\mathrm{UR}}[n]\sin\xi^{\mathrm{UR}}[n]}{c}}\Big]^{\mathrm{T}},\end{aligned} \tag{2}$$

β_0 is the channel power gain at the reference distance 1 m, c denotes the speed of light, and f_c is the carrier frequency. Variables $\theta^{\mathrm{UR}}[n]$ and $\xi^{\mathrm{UR}}[n]$ are the horizontal and vertical angles-of-arrival (AoAs) at the RIS with $\sin\theta^{\mathrm{UR}}[n] =$

$\frac{|z[n]-H_R|}{d^{\mathrm{UR}}[n]}$, $\sin\xi^{\mathrm{UR}}[n] = \frac{|x_R-x[n]|}{\sqrt{(x_R-x[n])^2+(y_R-y[n])^2}}$, and $\cos\xi^{\mathrm{UR}}[n] = \frac{|y_R-y[n]|}{\sqrt{(x_R-x[n])^2+(y_R-y[n])^2}}$.

We introduce the *Rician* fading model to characterize the links from the UAV to users and from the RIS to users. In time slot n, the channel vector between the RIS and user k on sub-carrier i can be written as

$$\begin{aligned}\mathbf{h}_{k,i}^{\mathrm{RG}}[n] = \sqrt{\frac{\beta_0}{(d_k^{\mathrm{RG}}[n])^{\alpha_k^{\mathrm{RG}}}}} \Bigg(& \sqrt{\frac{\kappa_k^{\mathrm{RG}}}{1+\kappa_k^{\mathrm{RG}}}} e^{-j2\pi i\Delta f \frac{d_k^{\mathrm{RG}}}{c}} \mathbf{h}_{k,\mathrm{LoS}}^{\mathrm{RG}} \\ & + \sqrt{\frac{1}{1+\kappa_k^{\mathrm{RG}}}} \tilde{\mathbf{h}}_{k,i}^{\mathrm{RG}}[n] \Bigg),\end{aligned} \tag{3}$$

where α_k^{RG} is the path loss exponent of the RIS-to-user link for user k, κ_k^{RG} is the Rician factor, $\tilde{\mathbf{h}}_{k,i}^{\mathrm{RG}}[n] \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{M_r M_c})$, $\mathbf{h}_{k,\mathrm{LoS}}^{\mathrm{RG}}$ is given by

$$\begin{aligned}\mathbf{h}_{k,\mathrm{LoS}}^{\mathrm{RG}} = & \Bigg[1, e^{-j2\pi f_c \frac{d_r \sin\theta_k^{\mathrm{RG}} \sin\xi_k^{\mathrm{RG}}}{c}}, \cdots, \\ & e^{-j2\pi f_c (M_r-1) \frac{d_r \sin\theta_k^{\mathrm{RG}} \sin\xi_k^{\mathrm{RG}}}{c}} \Bigg]^{\mathrm{T}} \\ & \otimes \Bigg[1, e^{-j2\pi f_c \frac{d_c \sin\theta_k^{\mathrm{RG}} \sin\xi_k^{\mathrm{RG}}}{c}}, \cdots, \\ & e^{-j2\pi f_c (M_c-1) \frac{d_c \sin\theta_k^{\mathrm{RG}} \sin\xi_k^{\mathrm{RG}}}{c}} \Bigg]^{\mathrm{T}},\end{aligned} \tag{4}$$

with θ_k^{RG} and ξ_k^{RG} are the horizontal and vertical angles-of-departure (AoDs) from the RIS to ground users. Note that we have $\sin\theta_k^{\mathrm{RG}} = \frac{H_R}{d_k^{\mathrm{RG}}}$, $\sin\xi_k^{\mathrm{RG}} = \frac{|x_k-x_R|}{\sqrt{(x_R-x_k)^2+(y_R-y_k)^2}}$, and $\cos\xi_k^{\mathrm{RG}} = \frac{|y_k-y_R|}{\sqrt{(x_R-x_k)^2+(y_R-y_k)^2}}$.

In time slot n, the channel between the UAV and the ground user k on sub-carrier i is:

$$\begin{aligned}h_{k,i}^{\mathrm{UG}}[n] = \sqrt{\frac{\beta_0}{(d_k^{\mathrm{UG}}[n])^{\alpha_k^{\mathrm{UG}}}}} \Bigg(& \sqrt{\frac{\kappa_k^{\mathrm{UG}}}{1+\kappa_k^{\mathrm{UG}}}} e^{-j2\pi i\Delta f \frac{d_k^{\mathrm{UG}}[n]}{c}} \\ & + \sqrt{\frac{1}{1+\kappa_k^{\mathrm{UG}}}} \tilde{h}_{k,i}^{\mathrm{UG}}[n] \Bigg),\end{aligned} \tag{5}$$

where α_k^{UG} denotes the path loss exponent of the UAV-to-user link for user k, κ_k^{UG} is the corresponding Rician factor, and $\tilde{h}_{k,i}^{\mathrm{UG}}[n] \sim \mathcal{CN}(0,1)$ is the scattering component of user k on sub-carrier i in time slot n.

The RIS reflection coefficient matrix in time slot n can be represented by

$$\mathbf{\Phi}[n] = \mathrm{diag}(\phi[n]) \in \mathbb{C}^{M_r M_c \times M_r M_c}, \tag{6}$$

where $\phi[n] = [e^{j,\phi_{1,1}[n]}, \cdots, e^{j,\phi_{m_r,m_c}[n]}, \cdots, e^{j,\phi_{M_r,M_c}[n]}] \in \mathbb{C}^{M_r M_c \times 1}$. Hence, the channel gain of link UAV-RIS-user k on sub-carrier i in time slot n can be given as

$$h_{k,i}^{\mathrm{URG}}[n] = a\left(\mathbf{h}_{k,i}^{\mathrm{RG}}\right)^{\mathrm{T}} \mathbf{\Phi}[n] \mathbf{h}_i^{\mathrm{UR}}[n]. \tag{7}$$

Note that measuring accurate channel state information (CSI) by each transceiver in practical settings is not trivial. We employ minimum mean square error (MMSE) estimation to address the imperfect CSI acquisition [37]. As such, the composite channel gain can be expressed as

$$g_{k,i}^{\mathrm{UG}} = \sqrt{1-\mathsf{P}}h' + \sqrt{\mathsf{P}}\tilde{h}, \tag{8}$$

where h' is the estimation of $h_{k,i}^{\mathrm{UG}}[n] + h_{k,i}^{\mathrm{URG}}[n]$, $\tilde{h}$ is the estimation error that is independent of h', and parameter P represents the estimation error variance, taking a constant value from 0 to 1.

The data rate of UAV is

$$R_{k,i}[n] = c_{k,i}[n] B \log_2\left(1 + \frac{p^{\mathrm{TX}} g_{k,i}^{\mathrm{UG}}}{\sigma^2}\right), \tag{9}$$

where p^{TX} is the fixed transmit power of the UAV, B is the bandwidth, σ is the noise variance, and $c_{k,i}[n] = \{0,1\}$ is used to indicate terminal k being served or not.

3.3 UAV Energy Consumption Model

While the UAV consumes energy for communications and task computations, the propulsion plays a dominant role in UAV energy consumption as a whole. To facilitate a tangible analysis, we assume that the estimation of energy consumption for communications and computations is constant, and we ignore the variation in energy consumption due to UAV acceleration/deceleration as long as the time slot for communications is short. Our model is primarily extended from [28], [38], and [39].

To extend the single-rotor UAV's power consumption model in [38] to a quadrotor UAV one, we make the following assumption [40]: 1) Every rotor is identical, and it is symmetrically distributed; 2) The weight assigned to each rotor is $\frac{W}{4}$; and 3) The thrust of each rotor is $\frac{T}{4}$ in hovering status. Thus, the total hovering power is

$$\begin{aligned}P_h^{\mathrm{quad}} &= 4\left(\frac{\delta_0}{8}\rho s_0 A_0 \Omega_0^3 R_0^3 + (1+k)\frac{(\frac{W}{4})^{3/2}}{\sqrt{2\rho A_0}}\right) \\ &= \underbrace{\frac{\delta_0}{2}\rho s_0 A_0 \Omega_0^3 R_0^3}_{\triangleq P_{B_0}} + \underbrace{(1+k)\frac{W^{3/2}}{2\sqrt{2\rho A_0}}}_{\triangleq P_{I_0}},\end{aligned} \tag{10}$$

where δ_0 denotes the profile drag coefficient, ρ accounts for air density (in $\mathrm{kg/m^3}$), s_0 represents rotor solidity, A_0 is the rotor disc area (in m^2), Ω_0 is the blade angular velocity (in radians/s), R_0 is the rotor radius (in m), k is the incremental correction factor to induced power, W is the UAV's total weight (in Newton).

According to the horizontal power expression for single-rotor UAV in [38] and Eq. (10), we have the total power consumption for quadrotor UAV in horizontal flight is

$$\begin{aligned}\bar{P}(\bar{V}) = & 4P_{B_0}\left(1 + \frac{3\bar{V}^2}{\Omega_0^2 R_0^2}\right) \\ & + 4P_{I_0}\left(\sqrt{1 + \frac{\bar{V}^4}{4v_0^4}} - \frac{\bar{V}^2}{2v_0^2}\right)^{\frac{1}{2}} + 2d_0 \rho s_0 A_0 \bar{V}^3.\end{aligned} \tag{11}$$

Fig. 2 displays our preliminary results using the same parameter configuration as [40] for this model. **An interesting fact is that as the UAV flies at a relatively low speed (less than 15 m/s), the total required power decreases with the increase in speed.**

Now, we consider the power consumption in vertical flight. We assume that $\hat{T}$ ($\check{T}$) and $\hat{D}$ ($\check{D}$) are the thrust and fuselage drag of the quadrotor UAV in vertical ascend

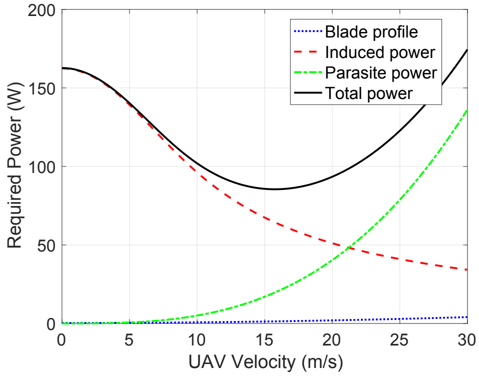


Fig. 2. Required power for quadrotor UAV in horizontal flight.

(descend). When the UAV ascends or descends at a constant speed, we have

$$\hat{T} - W = \hat{D} \tag{12}$$

in ascending and

$$W - \check{T} = \check{D} \tag{13}$$

in descending.

Let us look into the case of ascending first. According to the above force analysis, the following equation must be satisfied for each rotor on UAV

$$\hat{T}_0 = \frac{W}{4} + \frac{1}{2} S_{\mathrm{FP}\perp} \rho \hat{V}^2, \tag{14}$$

where $S_{\mathrm{FP}\perp}$ is the fuselage equivalent flat plate area in the vertical movement.

In line with [39], we have

$$\hat{P}_0(\hat{V}, \hat{T}_0) = \frac{P_h^{\mathrm{quad}}}{4} + \frac{1}{2}\hat{T}_0 \hat{V} + \frac{\hat{T}_0}{2}\sqrt{\hat{V}^2 + \frac{2\hat{T}_0}{\rho A_0}}. \tag{15}$$

So, the total power consumption of the quadrotor UAV is

$$\begin{aligned} \hat{P}(\hat{V}) &= 4\hat{P}_0(\hat{V}, \hat{T}_0) \\ &= P_h^{\mathrm{quad}} + \frac{1}{2} W \hat{V} + S_{\mathrm{FP}\perp} \rho \hat{V}^3 \\ &+ \left(\frac{W}{2} + S_{\mathrm{FP}\perp}\rho \hat{V}^2\right) \sqrt{\left(1 + \frac{S_{\mathrm{FP}\perp}}{A_0}\right)\hat{V}^2 + \frac{W}{2\rho A_0}}. \end{aligned} \tag{16}$$

Similarly, the total power consumption of quadrotor UAV in descending is

$$\begin{aligned} \check{P}(\check{V}) &= P_h^{\mathrm{quad}} + \frac{1}{2} W \check{V} - S_{\mathrm{FP}\perp} \rho \check{V}^3 \\ &+ \left(\frac{W}{2} - S_{\mathrm{FP}\perp}\rho \check{V}^2\right) \sqrt{\left(1 - \frac{S_{\mathrm{FP}\perp}}{A_0}\right)\check{V}^2 + \frac{W}{2\rho A_0}}. \end{aligned} \tag{17}$$

3.4 Problem Formulation

By Eqs (11), (16), and (17), we have the UAV's energy consumption model in time slot n as shown in

$$\begin{aligned} E[n] &= \delta_t[n] \Bigg(4P_{B_0}\left(1 + \frac{3\bar{V}^2}{\Omega_0^2 R_0^2}\right) \\ &+ 4P_{I_0}\left(\sqrt{1 + \frac{\bar{V}^4}{4v_0^4}} - \frac{\bar{V}^2}{2v_0^2}\right)^{1/2} + 2d_0\rho s_0 A_0 \bar{V}^3 \\ &+ P_h^{\mathrm{quad}} + \frac{1}{2} W \hat{V} + S_{\mathrm{FP}\perp}\rho \hat{V}^3 \\ &+ \left(\frac{W}{2} + S_{\mathrm{FP}\perp}\rho\hat{V}^2\right)\sqrt{\left(1 + \frac{S_{\mathrm{FP}\perp}}{A_0}\right)\hat{V}^2 + \frac{W}{2\rho A_0}} \Bigg), \end{aligned} \tag{18}$$

if UAV ascends in time slot n, and

$$\begin{aligned} E[n] &= \delta_t[n] \Bigg(4P_{B_0}\left(1 + \frac{3\bar{V}^2}{\Omega_0^2 R_0^2}\right) \\ &+ 4P_{I_0}\left(\sqrt{1 + \frac{\bar{V}^4}{4v_0^4}} - \frac{\bar{V}^2}{2v_0^2}\right)^{1/2} + 2d_0\rho s_0 A_0 \bar{V}^3 \\ &+ P_h^{\mathrm{quad}} + \frac{1}{2} W \check{V} - S_{\mathrm{FP}\perp}\rho \check{V}^3 \\ &+ \left(\frac{W}{2} - S_{\mathrm{FP}\perp}\rho\check{V}^2\right)\sqrt{\left(1 - \frac{S_{\mathrm{FP}\perp}}{A_0}\right)\check{V}^2 + \frac{W}{2\rho A_0}} \Bigg), \end{aligned} \tag{19}$$

if UAV descends in time time slot n. Here, $\bar{V} = \frac{\sqrt{(x[n+1]-x[n])^2 + (y[n+1]-y[n])^2}}{\delta_t[n]}$ and $\hat{V} = \check{V} = \frac{\sqrt{(z[n+1]-z[n])^2}}{\delta_t[n]}$.

Our goal is to minimize the energy consumption of the UAV in all time slots, which is formulated as

$$\begin{aligned} \min_{\mathbf{q}[n], c_{k,i}[n]} \quad & \sum_{n=1}^{N} E[n] \\ \text{s.t.} \quad & \sum_{k=1}^{K} c_{k,i}[n] \le 1 \\ & \sum_{n=1}^{N} \delta_t[n] R_{k,i}[n] \ge L_k \\ & \bar{V} \le \bar{V}_{\max} \\ & \hat{V}, \check{V} \le \tilde{V}_{\max} \\ & H_U^{\min} \le z[n] \le H_U^{\max} \end{aligned} \tag{20}$$

where the first constraint in (20) indicates that at most one terminal is served in each time slot, the second constraint ensures that the data transmission of each task with length L_k can be completed within the mission time of the UAV.

Note that the above optimization problem is non-convex and intractable due to the binary variable $c_{k,i}[n]$. This motivates us to seek RL techniques to solve it.

4 Proposed Approach

4.1 Markov Decision Process

We begin by modeling the UAV trajectory planning as an MDP.

4.1.1 State

In our implementation, the state space $s[n]$ includes not only the UAV position but also task-related information:

$$s[n] = [x[n], y[n], z[n], d_{\text{goal}}[n], R_{\text{remain}}[n]], \tag{21}$$

where:

- $[x[n], y[n], z[n]]$ represents the UAV's current position
- $d_{\text{goal}}[n]$ denotes the distance to the charging station
- $R_{\text{remain}}[n] = \sum_{k=1}^{K} D_k - \sum_{k=1}^{K} \sum_{n'=1}^{n} \delta_t R_{k,i}[n']$, represents the remaining data transmission tasks

4.1.2 Action

We define $\mathcal{A}$ as the action space of the RIS-assisted UAV system, which includes the horizontal and vertical movements of the UAV, the selection and scheduling of ground terminals (GTs), and the selection of time slot length. Specifically, it is defined as $a[n] = (l[n], h[n], c_{k,i}[n], \delta_t[n]) \in \mathcal{A} = \mathcal{L} \times \mathcal{H} \times \mathcal{C} \times \mathcal{T}$, where $l[n]$ and $h[n]$ being the UAV flying actions in horizontal and vertical dimensions in n-th time slot. $\mathcal{T} = [t_{\min} : 0.1 \text{ ms} : t_{\max}]$ is the space of the discrete flight times, from where $\delta_t[n]$ will be chosen as the discrete value between $t_{\min}$ and $t_{\min}$ with 0.1 ms as the step size. $\mathcal{C} = \{c_{k,i}[n], \forall k, i, n\}$ is the action space of GT scheduling.

Considering the flying actions of the UAV, assume that the UAV can only move to one of the adjacent cells from its current cell during a single time slot in the horizontal dimension or to an adjacent height level in the vertical dimension. Thus, the UAV's horizontal location $L[n+1] = [x[n+1], y[n+1]]^{\text{T}}$ in the next time slot is:

$$L[n+1] = L[n] + l[n], \tag{22}$$

where $l[n] \in \mathcal{L}$, and the horizontal action space $\mathcal{L}$ consists of 17 discrete choices: one option to remain stationary and 16 directions spaced evenly around a 360-degree circle, each separated by 15 degrees. This configuration allows the UAV to select from a full range of movement options in each time slot. Considering vertical flying, the UAV's vertical location H_{n+1} in next time slot can be defined as:

$$z[n+1] = z[n] + h[n], \tag{23}$$

where $h[n] \in \mathcal{H} \triangleq \{h_s, -h_s, 0\}$, with $\mathcal{H}$ being the vertical action space of the UAV including ascending, descending or remaining at its current height respectively.

4.1.3 Reward

In our model, the reward function $r(s[n], a[n])$ comprises two components, defined as follows:

$$r(s[n], a[n]) = \sum_{k=1}^{K} \sum_{n'=1}^{n+1} \frac{\omega \cdot \delta_t R_{k,i}[n']}{E[n']} - \mathrm{p}_0. \tag{24}$$

The first component: $\sum_{k=1}^{K} \sum_{n'=1}^{n+1} \frac{\omega \cdot \delta_t R_{k,i}[n']}{E[n']}$ represents the ratio of the cumulative data throughput from all ground terminals (GTs) up to time slot $n+1$ to the UAV's propulsion energy consumption. Here:

- $\delta_t R_{k,i}[n']$ denotes the amount of data transmitted by the k-th GT during time slot n',
- $E[n']$ represents the UAV's propulsion energy at time slot n', and
- ω is a weighting factor balancing throughput and energy consumption.

Algorithm 1 *FedX*

```
Initialize the number of agents $M$, the number of federated learning rounds $E$, the initial global parameters $w_G^0$, and the learning rate $\eta$;
Fork $M$ threads as $M$ agents for parallel training;
for $e \in \{1, 2, \cdots, E\}$ do
    for $m \in \{1, 2, \cdots, M\}$ do
        $w_m^e = w_G^e$;
    end for
    for $m \in \{1, 2, \cdots, M\}$ do
        Agent $m$ computes its local update by calling a RL algorithm X;
        Set $w_m^{e+1} = w_m^e - \eta \nabla L_m(w_m^e)$;
        Send $w_m^{e+1}$ to Aggregation process;
    end for
    Aggregation process receives $w_m^{e+1}$ from each agent $m$;
    Update global model using $w_G^{e+1} = \frac{\sum_{m=1}^{M} D_m w_m^{e+1}}{\mathcal{D}}$;
end for
```

This component encourages maximizing the cumulative data throughput relative to energy consumption, promoting resource-efficient and effective UAV trajectory planning.

The second component penalty term p_0 is designed as a function of the state rather than a constant:

$$p_0 = \lambda_A \cdot \psi_{\text{Data}}(R_{\text{remain}}[n]) + \lambda_B \cdot \psi_{\text{Bd}}(d_{\text{goal}}[n]), \tag{25}$$

where:

- $\psi_{\text{Data}}(R_{\text{remain}}[n]) = \frac{\sum_{k=1}^{K} D_k - \sum_{k=1}^{K} \sum_{n'=1}^{n} \delta_t R_{k,i}[n']}{\sum_{k=1}^{K} D_k}$, representing the normalized remaining data transmission ratio
- $\psi_{\text{Bd}}(d_{\text{goal}}[n]) = \frac{d_{\text{goal}}[n]}{d_{\max}}$, representing the normalized distance to the charging station
- λ_A and λ_B are scaling weights

This design ensures that both penalty components are normalized to $[0, 1]$ and connected to the state space. By using state-dependent penalties, the agent can optimize its trajectory by balancing remaining tasks and destination distance, providing smooth feedback to guide the learning process.

With the above MDP formulation, the design of RL solvers is straightforward. For example, a well-thought-out DDQN-based algorithm for UAV trajectory planning is given in [17]. However, its training process is extremely slow, although it may produce a satisfactory solution. Our prior investigation in [27] also highlights the challenge of slow training for deep reinforcement learning algorithms in wireless communication systems.

4.2 FedX: A Fast UAV Trajectory Planning Framework

To resolve this issue, we leverage the techniques of multi-threading and federated learning and propose a framework, called *FedX*, as shown in Algorithm 1. The idea underlying this algorithm is to fork multiple threads and treat each thread as an agent of federated learning to enable parallel training [41]. ***"X" in Algorithm 1 can be any RL solver***, including but not limited to DQN, DDQN, DDPG [42], TD3 [43], PPO [44], SAC [45] and other algorithms of the same kind.

Note that the proposed framework is different from conventional federated RL, where multiple agents independently interact with distinct parts of the environment. Instead, it features threads acting as agents interacting with the environment. *FedX* folks multiple threads and runs them in parallel for model training. These threads collaborate by aggregating their models, similar to the process in federated learning. *FedX* allows for centralized control and unified decision-making while benefiting from the parallelization and collaborative learning aspects of federated techniques. In addition, samples are distributed to each thread through an individual replay buffer, which stores experiences collected by the agent. Each thread independently accesses this replay buffer and extracts mini-batches of samples for training. This allows multiple threads to concurrently process different subsets of data. By operating on independent mini-batches, the threads can perform gradient updates in parallel, leveraging the diverse experiences stored in the replay buffer. The parallel processing not only speeds up the training process but also ensures that the model benefits from a wide variety of experiences. As a result, the proposed *FedX* can enhance the efficiency and effectiveness of model training and deployment.

4.3 FedSAC and FedPPO

To implement *FedX* as the optimizer for Problem (20), we instantiate *"X"* in Algorithm 1 as Soft Actor-Critic (SAC) and Proximal Policy Optimization (PPO).

SAC is an *off-policy* RL algorithm that offers significant advantages over other off-policy algorithms like TD3 and DDPG. It balances exploration and exploitation through entropy regularization [45]. In UAV trajectory planning, SAC ensures comprehensive exploration of different paths in complex environments, thus avoiding local optima. Furthermore, SAC boasts higher sample efficiency and a more stable training process, leading to faster convergence towards the optimal trajectory planning solution.

The framework of SAC is depicted in Algorithm 2. The complexity of the SAC algorithm is primarily determined by the update processes of the Q-network and the policy network. Suppose these networks have n layers, each with m neurons. The complexity of initialization (lines 1 to 5) is constant. The forward propagation for an action selection (lines 6 to 12) takes $O(n \cdot m^2)$ time. From line 13 to line 18, Sampling a mini-batch of transitions spends $O(B)$, and computing the target value y, which includes the forward propagation through two Q-networks and the policy network, is $O(B \cdot 3n \cdot m^2)$. In lines 19 and 20, the complexity of computing the losses L_{Q_1} and L_{Q_2} is $O(B \cdot m^2)$, and that of updating the parameters ϕ_1 and ϕ_2 via gradient descent is $O(2 \cdot n \cdot m^2)$. Computing the policy loss L_{π_θ} costs $O(B \cdot m^2)$ time, and updating the policy network parameters θ via gradient descent takes $O(n \cdot m^2)$ in lines 21 to 22. If the temperature parameter α is not fixed, the complexity of computing the temperature loss L_α and updating the parameter α is constant time (lines 23 and 26). In line 27, the time for the soft update of the target networks is $O(2 \cdot n \cdot m)$. Therefore, the overall complexity of the SAC algorithm for E episodes with N steps each is $O(E \cdot N \cdot n \cdot m^2)$, assuming the value of B is small.

Algorithm 2 SAC

```
 1: Initialize the replay memory $O$;
 2: Initialize actor network $\pi_\theta$ with parameters $\theta$;
 3: Initialize critic networks $Q_{\phi_1}$, $Q_{\phi_2}$ with parameters $\phi_1$, $\phi_2$;
 4: Initialize target networks $Q_{\phi_1'}$, $Q_{\phi_2'}$ with parameters $\phi_1'$, $\phi_2'$ (with
    $\phi_1' \leftarrow \phi_1$, $\phi_2' \leftarrow \phi_2$);
 5: Initialize temperature parameter $\alpha$;
 6: for episode $= 1, \dots, E$ do
 7:     Set $n = 1$, initialize the initial state $s(1)$;
 8:     while $n = 1, \dots, N$ and task $D_k$ is not finished do
 9:         Select action $a \sim \pi_\theta(a|s)$
10:         if (UAV out of desired region) and (UAV exceeding horizontal/vertical velocity) then
11:             Cancel the action and apply the penalty;
12:         end if
13:         Execute action $a$, observe reward $r$ and next state $s'$
14:         Store transition $(s, a, r, s')$ in replay buffer $O$
15:     end while
16:     Sample a random mini-batch of transitions $(s, a, r, s')$ from $O$
17:     Compute target value $y$:
            $y = r + \gamma \sum_{a'} \pi_\theta(a'|s') \Big[ \min\left(Q_{\phi_1'}(s', a'), Q_{\phi_2'}(s', a')\right) - \alpha \log \pi_\theta(a'|s') \Big];$
18:     where $a' \sim \pi_\theta(a'|s')$
19:     Update critic networks by minimizing the loss:
            $L_{Q_1} = \frac{1}{N} \sum \left(Q_{\phi_1}(s, a) - y\right)^2$
            $L_{Q_2} = \frac{1}{N} \sum \left(Q_{\phi_2}(s, a) - y\right)^2$
20:     Update parameters $\phi_1$ and $\phi_2$:
            $\phi_1 \leftarrow \phi_1 - \eta \nabla_{\phi_1} L_{Q_1}$
            $\phi_2 \leftarrow \phi_2 - \eta \nabla_{\phi_2} L_{Q_2}$
21:     Update actor network by minimizing the loss:
            $L_{\pi_\theta} = \frac{1}{N} \sum_{s,a} \pi_\theta(a|s) \left[\alpha \log \pi_\theta(a|s) - Q_{\phi_1}(s, a)\right]$
22:     Update parameters:
            $\theta \leftarrow \theta - \eta \nabla_\theta L_\pi$
23:     if temperature parameter $\alpha$ is not fixed then
24:         Update temperature parameter $\alpha$ by minimizing the loss:
                $L_\alpha = -\alpha \sum_a \pi_\theta(a|s) \left[\log \pi_\theta(a|s) + \mathcal{H}_{\text{target}}\right]$
25:         Update parameter:
                $\alpha \leftarrow \alpha - \eta \nabla_\alpha L_\alpha$
26:     end if
27:     Soft update target networks:
            $\phi_1' \leftarrow \tau \phi_1 + (1 - \tau) \phi_1'$
            $\phi_2' \leftarrow \tau \phi_2 + (1 - \tau) \phi_2'$
28:     $s \leftarrow s'$
29:     if done then
30:         break
31:     end if
32: end for
```

PPO is an *on-policy* RL algorithm with significant advantages over other on-policy algorithms, such as Trust Region Policy Optimization (TRPO) [46] and Advantage Actor-Critic (A2C) [47]. Using a clipping mechanism, PPO maintains stability and enhances performance during policy updates [44]. In UAV trajectory planning, PPO ensures that the UAV can efficiently learn optimal paths while reducing stability issues during training. The framework of PPO is illustrated in Algorithm 3.

The complexity of the PPO algorithm is dominated by

Algorithm 3 PPO

```
1:  Initialize actor network $\pi_\theta$ with parameters $\theta$;
2:  Initialize critic network $V_\phi$ with parameters $\phi$;
3:  Initialize replay buffer $\mathcal{D}$;
4:  Set learning rate $\eta$, clipping parameter $\epsilon$;
5:  for episode $= 1, \ldots, E$ do
6:      Initialize state $s_0$;
7:      while $n = 1, \ldots, N$ and task $D_k$ is not finished do
8:          Select action $a_t \sim \pi_\theta(a_t|s_t)$;
9:          if (UAV out of desired region) and (UAV exceeding horizontal/vertical velocity) then
10:             Cancel the action and apply the penalty;
11:         end if
12:         Execute action $a_t$, observe reward $r_t$ and next state $s_{t+1}$;
13:         Store transition $(s_t, a_t, r_t, s_{t+1})$ in replay buffer $\mathcal{D}$;
14:         $s_t \leftarrow s_{t+1}$;
15:         if $s_t$ is terminal then
16:             break;
17:         end if
18:     end while
19:     Compute advantage estimates $\hat{A}_t$;
20:     for $k = 1, \ldots, K$ do
21:         Sample a random mini-batch of transitions $(s_t, a_t, \hat{A}_t, \pi_\theta(a_t|s_t))$ from $\mathcal{D}$;
22:         Compute the ratio:
                $$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$$
23:         Compute the clipped objective:
                $$L^{\text{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right]$$
24:         Update actor network parameters:
                $$\theta \leftarrow \theta + \eta \nabla_\theta L^{\text{CLIP}}(\theta)$$
25:         Update critic network by minimizing the loss:
                $$L^V(\phi) = \frac{1}{N}\sum\left(V_\phi(s_t) - V_t^{\text{target}}\right)^2$$
26:         Update critic network parameters:
                $$\phi \leftarrow \phi - \eta \nabla_\phi L^V(\phi)$$
27:     end for
28: end for
```

the update processes of the actor and critic networks. Assuming these networks have n layers, each with m neurons, the complexity of initialization (lines 1 to 4) remains constant. The forward propagation for action selection (lines 5 to 11) takes $O(n \cdot m^2)$ time. Starting from line 19, computing the advantage estimates $\hat{A}_t$ using the critic network costs $O(N \cdot n \cdot m^2)$. In lines 12 to 22, sampling a mini-batch of size B takes $O(B)$ time, and computing the ratio $r_t(\theta)$, which involves forward propagation through the actor network, is $O(B \cdot n \cdot m^2)$. The complexity of computing the clipped objective $L^{\text{CLIP}}(\theta)$ (line 23) is $O(B)$. Updating the actor network parameters via gradient ascent takes $O(n \cdot m^2)$ (line 24). Computing the critic loss $L^V(\phi)$ is $O(B \cdot m^2)$, and updating the critic network parameters through gradient descent costs $O(n \cdot m^2)$ (lines 25 to 26). Therefore, the overall complexity of the PPO algorithm for E episodes with N steps each is $O(E \cdot (N + K \cdot B) \cdot n \cdot m^2)$, which simplifies to $O(E \cdot N \cdot n \cdot m^2)$ assuming B is small.

4.4 FedSAC vs. FedPPO

FedSAC excels in adaptability through its entropy maximization mechanism, which promotes broad exploration in high-dimensional spaces. However, its dependence on target networks and delayed updates can lead to parameter mismatches between local and global models in the asynchronous *FedX* setup, resulting in possibly lower update stability. In contrast, FedPPO ensures stability through its clipping mechanism and synchronization of actor and critic networks alongside their previous versions, avoiding the instability caused by delayed updates. While FedSAC offers superior exploration capabilities, FedPPO achieves more stable updates and faster convergence, making it reliable for tasks requiring scalability and consistent performance.

TABLE 1
Parameter settings for simulations.

Parameter	Value
Bandwidth B,	2MHz
GTs: K, Task: D_k	4, 1024 $\sim$ 2048Kb
$\bar{V}_{\max}, \tilde{V}_{\max}$	10m/s, 10m/s
t_{min}, t_{max}	1s, 2s
Flying height: h_{min}, h_{max}	60m, 200m
Time slots and episodes	1000, 60
Area size (width × depth × height)	500m × 500m × 300m
Channel power gain (β_0)	1
Speed of light (c)	3×10^8m/s
Carrier frequency (f_c)	2×10^9Hz (2GHz)
Noise power spectral density (N_0)	1×10^{-9}
Number of reflecting elements (M_r, M_c)	10, 10
Path loss exponent for RIS-to-user link (α_k^{RG})	2
Rician factor for RIS-to-user link (κ_k^{RG})	10
Path loss exponent for UAV-to-user link (α_k^{UG})	2
Rician factor for UAV-to-user link (κ_k^{UG})	10
Transmission power (p^{TX})	1W
Path loss factor (A, C)	1, 1
Noise power (σ)	$\sqrt{3.98 \times 10^{-12}}$
Number of sub-carriers (N_f)	64
Estimation error variance (P)	0.3
The positions of the GTs	$[100\text{m}, 100\text{m}]^{\text{T}}$, $[100\text{m}, 400\text{m}]^{\text{T}}$, $[400\text{m}, 100\text{m}]^{\text{T}}$, $[400\text{m}, 400\text{m}]^{\text{T}}$
The position of RIS	$w_R = [250\text{m}, 250\text{m}]^{\text{T}}$ with a height of 60m.

5 Performance Evaluation

In this section, we validate the effectiveness of FedSAC and FedPPO in an RIS-assisted UAV system through simulations. We compare the trajectory optimization of different algorithms and their acceleration performance. To ensure the fidelity of the results, we collect data by averaging the results from 100 simulations.

5.1 Parameter Settings

The simulation settings for the RIS-assisted UAV system are shown in Table 1.

Note that according to [48], the impact of the *Doppler* effect on the system can be safely ignored when the Doppler shift f_D is significantly smaller than the sub-carrier spacing Δf, as its influence becomes minimal and can reasonably be disregarded under these conditions.

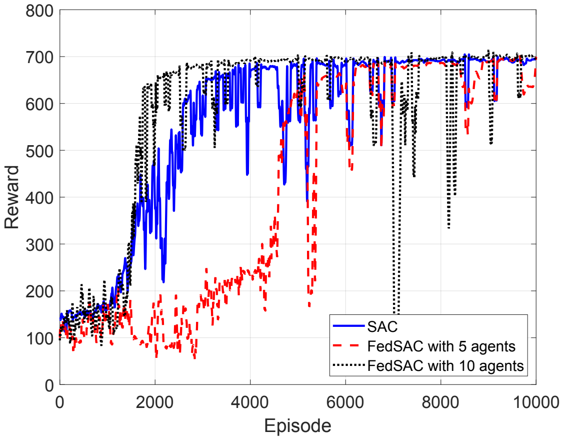


Fig. 3. Rewards of SAC and FedSAC.

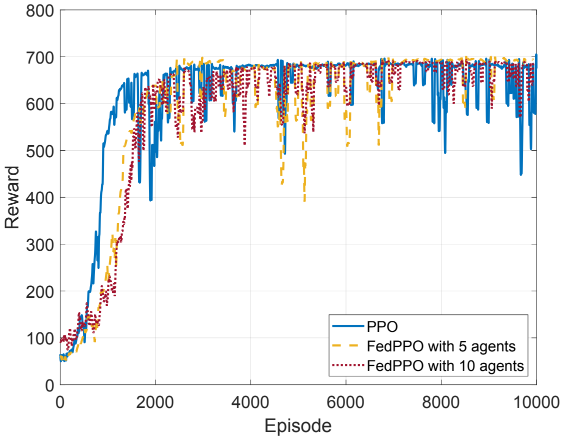


Fig. 4. Rewards of PPO and FedPPO.

With the above parameter settings, the sub-carrier spacing Δf and maximum Doppler shift $f_D^{\max}$ are calculated as:

$$\Delta f = \frac{B}{N_f} = \frac{2 \times 10^6\,\text{Hz}}{64} = 31.25\,\text{kHz}$$

and

$$f_D^{\max} = \frac{v}{c} f_c = \frac{10}{3 \times 10^8} \times 2 \times 10^9 = 66.67\,\text{Hz}.$$

Since $f_D^{\max} \ll \Delta f$, it is reasonable to assume that the Doppler effect has a negligible impact under these system parameters.

We set up the air-to-ground communication scenario based on the discussion in [16] and configure the propulsion model of the rotor UAV as described in [28], [38], and [39]. The initial position of the UAV is set to be $[0, 0, 200]$. To minimize energy consumption during exploration while encouraging the UAV to establish communication channels with ground terminals for data transmission, we introduce a scaling coefficient $\omega = 10$ in the reward function. The simulations were conducted in Python 3.10 to implement the Deep Neural Network (DNN) in the SAC and PPO algorithms.

In the SAC algorithm, the original network consists of a 3-layer structure with 64 neurons in each layer. The Rectified Linear Unit (ReLU) activation function is used in the hidden layers, and the Tangent Hyperbolic (Tanh) function is applied in the output layer. The Adam optimizer [49] is used to train the DNN, with its parameters randomly initialized following a zero-mean normal distribution.

For the PPO algorithm, both the original and target networks of the policy network consist of 2-layer DNNs. The first and second layers each have 128 neurons and utilize Tanh as the activation function, while the output layer uses Softmax. The Adam optimizer is applied to train the DNNs of the policy network.

5.2 Performance Metrics

5.2.1 Rewards

We examine the fluctuations of the reward functions throughout the training process. The reward function is critical to RL algorithms as it directly influences the agent's behavior and learning process. Observing the variations in the reward function offers insights into the convergence of the algorithms being evaluated.

5.2.2 Training Time

We evaluate the training times for FedSAC and FedPPO to explore their computational efficiency and practicality. This metric is crucial as accelerated training processes can significantly reduce time costs in practical scenarios. By comparing the time taken for training with and without the accelerated frameworks, we can highlight the advantages of our developed framework in terms of time performance.

5.2.3 UAV's Trajectory

We assess the effectiveness of the proposed algorithms through a qualitative analysis of UAV trajectories. The flight paths of the UAV in simulated scenarios visually demonstrate the performance of the algorithms in practical applications. This enables us to compare the variations in UAV trajectories planned by different algorithms and demonstrates the effectiveness of the *FedX* framework in UAV path planning.

5.2.4 System Performance

We also investigate the system performance in terms of throughput and energy consumption of the UAV. These two metrics reflect the quality of solutions obtained by our proposed algorithms, facilitating a direct comparison between the original algorithms (SAC and PPO) and their accelerated versions (FedSAC and FedPPO).

5.3 Results

Fig. 3 demonstrates the convergence of the SAC algorithm and its accelerated versions, FedSAC with 5 agents and FedSAC with 10 agents, across 10,000 training episodes. It is evident that all three configurations ultimately converge. Notably, the convergence rate of FedSAC with 5 agents is slower, gradually stabilizing around the 6000-th episode. This slower convergence is attributed to the use of asynchronous updates. With fewer threads initiated (such as FedSAC with 5 agents), each thread's weight update exerts a more significant impact on the global model yet occurs less frequently. This may result in infrequent updates to the

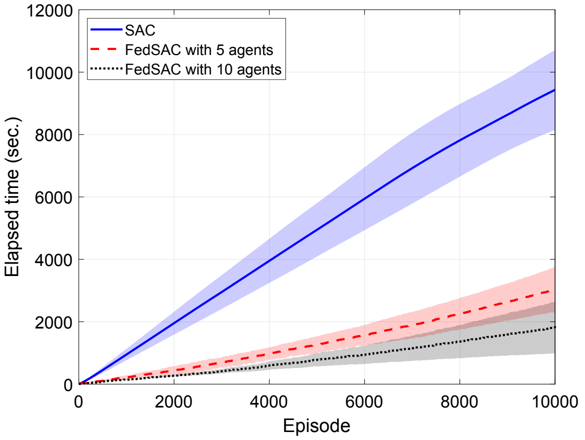


Fig. 5. Time comparison between SAC and FedSAC.

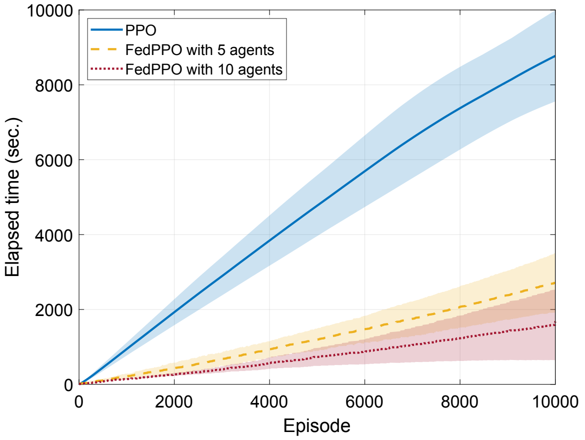


Fig. 6. Time comparison between PPO and FedPPO.

global model, thereby affecting the speed and efficiency of the learning process [1].

In contrast, increasing the number of threads to 10 (as in FedSAC with 10 agents) results in more frequent updates of the global model, even with asynchronous updates, which aids in faster convergence. However, in asynchronous updates, the completion times of local updates across different agents can vary significantly. As the number of agents increases, these temporal discrepancies become more pronounced, exacerbating parameter inconsistencies during global model aggregation. In addition, FedSAC aggregates four distinct networks: actor, critic, target actor, and target critic, causing instability, particularly under the asynchronous participation of a larger number of agents. Furthermore, SAC's entropy maximization mechanism, designed to enhance exploration, introduces greater update variability. This increased variability further impedes the convergence rate as the number of participating agents continues to grow.

Fig. 4 illustrates the convergence over 10,000 training episodes for the PPO algorithm and its accelerated versions, FedPPO with 5 agents and FedPPO with 10 agents. It is evident that all configurations converge quickly, with convergence occurring around 2000 episodes. Despite the varying numbers of threads initiated, the PPO algorithm maintains update stability by constraining the difference between new and old policies, thereby reducing the risk of significant performance degradation during updates. In comparison to the SAC algorithm and its associated accelerated algorithms, PPO can uphold update stability even in asynchronous environments [50], thus mitigating global model fluctuations caused by inconsistent learning processes among agents.

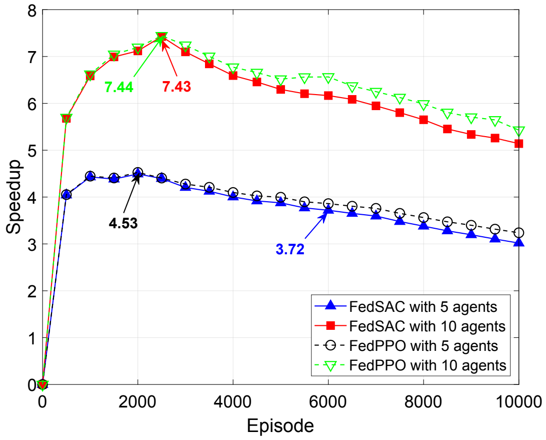


Fig. 7. Speedup performance.

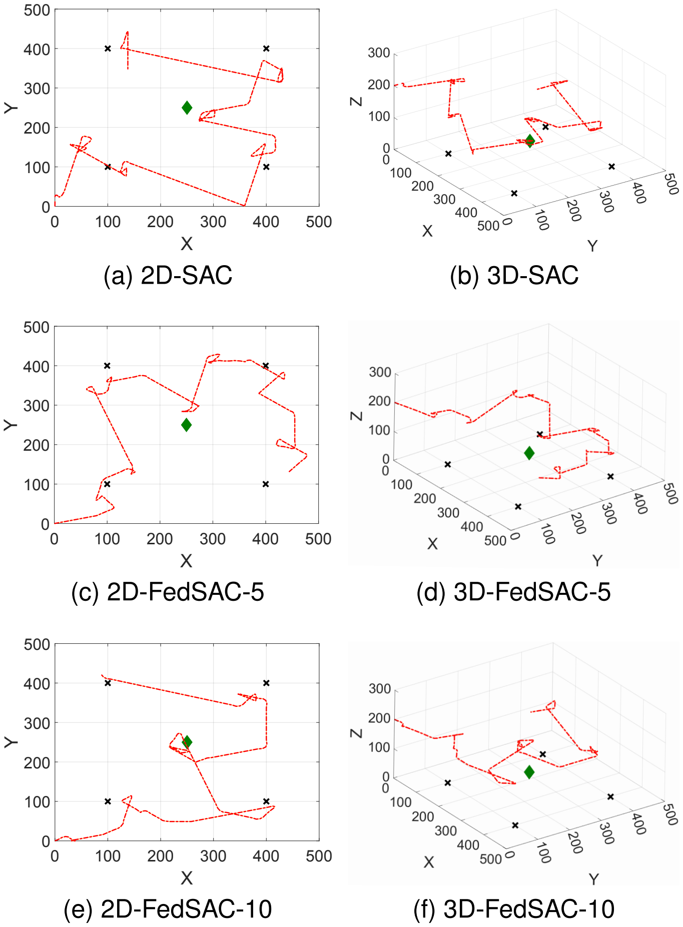
Fig. 5 and Fig. 6 show the average time required to complete 10,000 training episodes by SAC, FedSAC, PPO, and FedPPO algorithms. For each curve, the shaded areas represent the standard deviations. The results in these two figures faithfully demonstrate that the utilization of *FedX*, notably FedSAC, significantly reduced training durations. Such an enhancement highlights the efficiency of *FedX* in expediting the training process, making it a powerful framework for scenarios requiring rapid model updates.

Fig. 7 further confirms the effectiveness of *FedX* in terms of Speedup [2]. In the figure, the Speedup for FedSAC is approximately 3.72 and 7.43 for configurations with 5 and 10 agents, respectively, whereas for FedPPO, these ratios are about 4.53 and 7.44, respectively. In addition, the Speedup exhibits a trend of initially increasing and then decreasing as the number of training episodes grows. This can be attributed to the parallel operation of multiple agents in the initial training phases, especially when each agent starts with a relatively high initial communication or synchronization overhead. However, as training progresses into the middle and later stages, with the increase in data volume and the model nearing convergence, the update rate slows while overhead remains, resulting in a decline in Speedup [51].

Fig. 8 and Fig. 10 show a comparison of the 2D and 3D trajectories produced by the SAC and PPO algorithms along with their accelerated ones, FedSAC and FedPPO. It is clear to see that all the algorithms are capable of generating high-quality solutions. Specifically, we can observe that in order to establish stable communication links with GTs, UAVs

[1] Note that the time required to run the same episodes by different algorithms (SAC and FedSAC) is different. This can be observed in Fig. 5.

[2] Speedup, a.k.a. Amdahl's Law, refers to the performance improvement achieved by parallelizing a computational task. Speedup was given by Gene Amdahl, an alumnus of EECS@South Dakota State University.

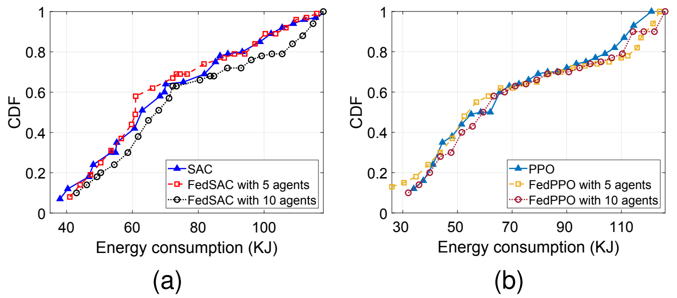


(a) 2D-SAC (b) 3D-SAC

(c) 2D-FedSAC-5 (d) 3D-FedSAC-5

(e) 2D-FedSAC-10 (f) 3D-FedSAC-10

Fig. 8. Trajectory comparison between SAC and FedSAC.

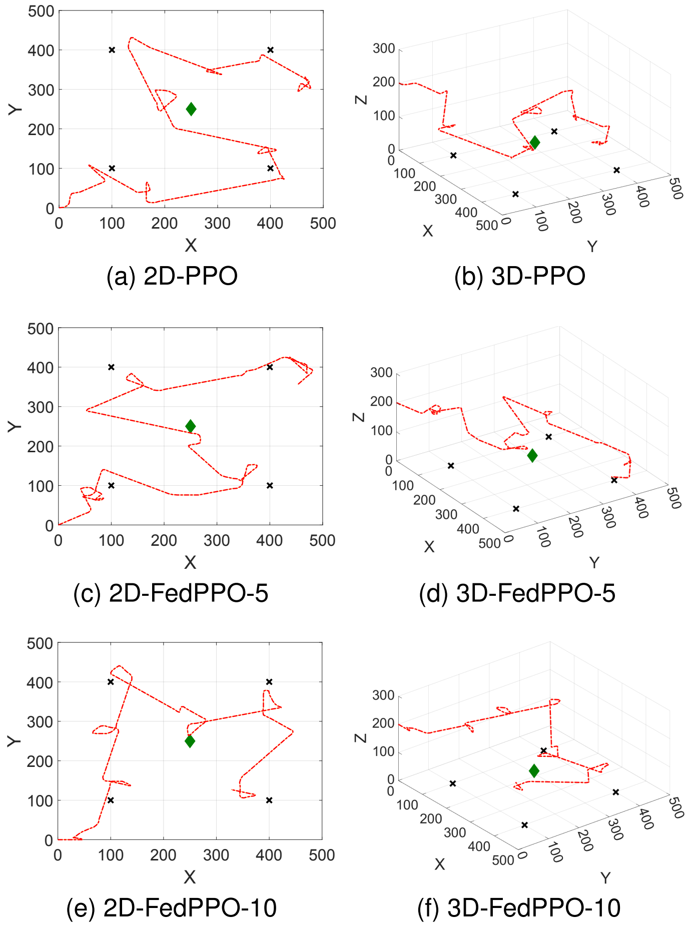


(a) (b)

Fig. 9. Comparison of energy consumption.

typically fly close to each GT and may descend to lower altitudes and hover as needed. This behavior primarily aims to optimize signal reception quality and enhance communication reliability, especially in complex practical settings.

Fig. 9 and 11 display the Cumulative Distribution Functions (CDF) of UAV energy consumption and throughput under different algorithms and their accelerated versions. The results demonstrate that the performance of both algorithms and their accelerated versions are almost the same. This observation further validates that our proposed framework significantly reduces training time while not compromising solution accuracy.

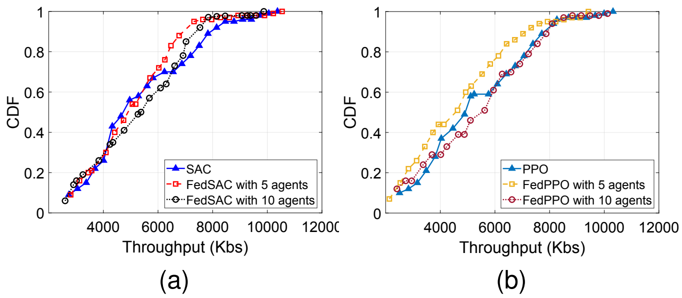


(a) 2D-PPO (b) 3D-PPO

(c) 2D-FedPPO-5 (d) 3D-FedPPO-5

(e) 2D-FedPPO-10 (f) 3D-FedPPO-10

Fig. 10. Trajectory comparison between PPO and FedPPO.



(a) (b)

Fig. 11. Comparison of throughput.

5.4 Discussion and Lesson Learned

5.4.1 Summary of Evaluation Results

We investigated the convergence properties, training duration, acceleration effects, trajectory quality, and comparative system performance of *FedX* and its implementations, FedSAC and FedPPO. The training results indicate that while all algorithms eventually converge, FedSAC and FedPPO exhibit significant advantages in accelerating the training process without compromising solution accuracy. Additionally, FedSAC and FedPPO demonstrate exceptional performance in planning UAV flight trajectories. The results show that these algorithms achieve similar system performance in terms of energy consumption and throughput, which indicates that UAVs can effectively approach each

Algorithm 4 *A* Algorithm for UAV Trajectory Planning*

```
Initialize environment parameters and UAV initial state (position, data served as zero, energy consumed as zero);
Initialize the open list with the start node and closed set as empty;
while $n < N$ do
    if open list is empty then return No feasible path found;
    end if
    Pop node with lowest $f_{\text{cost}}[n]$ from open list as current_node;
    if $\sum_{k=1}^{K} \sum_{n'=1}^{n+1} \delta_t R_{k,i}[n'] > \sum_{k=1}^{K} D_k$ then
        Retrieve and return path from the start node to current_node;
    end if
    for all possible actions in $\mathcal{A}'$ do
        Simulate the action to obtain the next state (position, data served, energy consumed);
        Calculate $g_{\text{cost}}[n]$ as cumulative cost from start to next state, including energy and penalties;
        Calculate $h_{\text{cost}}[n]$ as heuristic estimate to fulfill GT requirements from next state;
        Calculate $f_{\text{cost}}[n]$ as $g_{\text{cost}}[n] + h_{\text{cost}}[n]$;
        if next state is not in closed set then
            Create neighbor_node with next state, $f_{\text{cost}}[n]$, and action leading to it;
            Add neighbor_node to open list;
        end if
    end for
    Add current_node to closed set;
end whilereturn No feasible path found if max_iterations reached;
```

ground terminal and adjust their altitude to establish stable communication links, optimize signal reception quality, and enhance communication reliability.

Although the acceleration effects may lessen in the later stages of training, likely due to the slowdown in model update rates while the potential high communication and synchronization overheads remain, the accelerated training frameworks still effectively shorten the overall training time and significantly improve training efficiency.

5.4.2 Pros and Cons of *FedX*

The *FedX* framework proposed in this study has achieved remarkable results in UAV trajectory planning. The basic idea behind *FedX* is *trading-space-for-time*. It efficiently utilizes computational resources by forking multiple threads for parallel training and significantly accelerates the training process, enabling the model to converge in a shorter time. This is particularly suitable for wireless networks and mobile computing environments.

On the other hand, the assumption underlying *FedX* for training acceleration is that the dataset for training must be *homogeneous*. For those heterogeneous datasets, *FedX* needs to be redesigned, which is our future work. Moreover, in an asynchronous update environment, coordinating the updates from multiple agents becomes more complex, potentially making it challenging to ensure the stability of the global model.

5.4.3 Comparison with Non-RL Solution

We employed a model-based A* algorithm [32] for performance comparison, with results presented in Fig. 12 (including 2D and 3D trajectory plots) and Table 2, where we analyze UAV throughput and energy consumption. The pseudocode framework of the A* algorithm is shown in Algorithm 4.

The A* algorithm utilizes three key cost components to evaluate the efficiency of UAV trajectory planning:

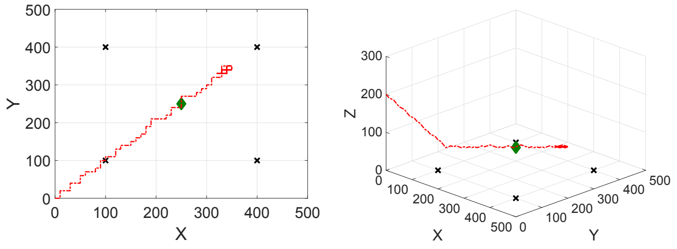
- f_{cost} (Total Cost): It is the sum of the actual cost incurred from the start node to the current node ($g_{\text{cost}}[n]$) and the estimated cost to reach the goal ($h_{\text{cost}}[n]$). The algorithm selects the node with the lowest $f_{\text{cost}}[n]$ at each step to explore:
$$f_{\text{cost}}[n] = g_{\text{cost}}[n] + h_{\text{cost}}[n].$$
- g_{cost} (Accumulated Cost): Representing the actual cumulative cost from the start node to the current node:
$$g_{\text{cost}}[n] = E[n] + p_{Bd},$$
where $E[n]$ denotes the energy consumption of the UAV in the current time slot, and p_{Bd} is the penalty for the UAV crossing the boundary. This setup allows $g_{\text{cost}}[n]$ to comprehensively reflect the actual operational cost of the UAV, taking into account both energy consumption and penalties for non-compliant flight behaviors, such as exceeding boundaries. It helps improve the efficiency and reliability of the path-planning process.
- h_{cost} (Heuristic Cost): The heuristic estimate of the remaining effort from the current node to the goal:
$$h_{\text{cost}}[n] = \sum_{k=1}^{K} D_k - \sum_{k=1}^{K} \sum_{n'=1}^{n+1} \delta_t R_{k,i}[n'],$$
where $\sum_{k=1}^{K} D_k$ represents the total data demand of all ground terminals, and $\sum_{k=1}^{K} \sum_{n'=1}^{n+1} \delta_t R_{k,i}[n']$ represents the total amount of data transmitted by the UAV to each ground terminal up to the current time slot. As shown, the heuristic function $h_{\text{cost}}[n]$ is expressed as the remaining data to be transmitted. This setup allows $h_{\text{cost}}[n]$ to reasonably estimate the remaining transmission tasks, helping the UAV plan its path more efficiently and prioritize routes that maximize data transmission.

The A* algorithm finds the optimal UAV trajectory through heuristic search. The idea is to select the node to expand at each step based on the cost function $f_{\text{cost}}[n]$, giving priority to expanding the node with the smallest cost. Each node state contains the current position of the UAV and the accumulated total energy consumption. At each step, the UAV's action space $\mathcal{A}'$ is traversed, and its definition is consistent with the action space in reinforcement learning, specifically:

$$a'(n) = (l_n, h_n, c_{k,n}, t_n^u) \in \mathcal{A} = \mathcal{L}_u \times \mathcal{H}_u \times \mathcal{C} \times \mathcal{T},$$

where l_n and h_n represent the UAV's movement directions in the horizontal and vertical dimensions, respectively. $\mathcal{T} = [t_{\min} : 0.5 \text{ ms} : t_{\max}]$ denotes the duration of the flight time slot, and t_n^u is a discrete value chosen between $t_{\min}$ and $t_{\max}$. $\mathcal{C} = \{c_{k,n}, \forall k, n\}$ represents the scheduling actions for the ground terminals.

We simplified the action space in the designed A* algorithm to improve search efficiency during heuristic search. In the horizontal dimension, the action space is restricted to five directions (forward, backward, left, right, and hover), and the time slot interval is adjusted to 0.5. The reduction in the search space guarantees that the A* algorithm can efficiently find optimal paths even in large-scale problems.



(a) 2D-A*.

(b) 3D-A*.

Fig. 12. Trajectory of A*.

TABLE 2
System performance comparison.

Algorithm	Energy (KJ)	Throughput (Kbs)	Energy efficiency (bits/J)
A*	53.38	3230.09	60.51
SAC	72.32	6580.25	90.98
SAC with 5 agents	76.54	6636.58	86.71
SAC with 10 agents	79.02	6592.88	83.02
PPO	76.12	6607.22	86.80
PPO with 5 agents	78.23	6742.74	86.19
PPO with 10 agents	79.53	6883.10	86.55

Fig. 12 shows the 2D and 3D trajectories of the UAV obtained by the A* algorithm. As shown in the figure, given the known system model, the A* algorithm selects the action with the lowest total cost (f_{cost}) at each step (e.g., hovering and circling at the lowest altitude) while ensuring the completion of data transmission tasks. However, the UAV demonstrates limited exploration, as it does not attempt to cover a wider spatial range. For example, the UAV does not choose to fly closer to the ground terminals to maximize data transmission rates; instead, it prioritizes energy saving. To some extent, such a behavior limits the overall optimization potential of its performance.

Table 2 compares the system performance between the A* algorithm and our proposed RL-based algorithms, including metrics such as energy consumption, throughput, and energy efficiency. It can be observed that the A* algorithm demonstrates significant optimization in terms of energy consumption, achieving lower energy usage compared to the RL methods. However, since the A* algorithm chooses the direction with the lowest cost function at each step, It does not sufficiently explore the global state, limiting action choices. This lack of exploration prevents the A* algorithm from fully utilizing the available space and resources to maximize data transmission rates, leading to lower overall energy efficiency compared to RL methods. While the A* algorithm excels in local optimization, it is less effective at finding globally optimal solutions in complex and dynamic environments compared to RL algorithms.

5.4.4 Impact of Flight Pattern

Fig. 13 illustrates a typical flight pattern comprising four phases: Ascend (from 0 to t_1), Communication (t_1 to t_2), Return (t_2 to t_3), and Descend (t_3 to t_4). Our work in this study only considers the flight and communication phases of the UAV trajectory (green area in Fig. 13) without accounting for the takeoff (Ascend), return ((t_2 to t_3)), and landing (Descend) phases. Recall that Fig. 2 provides a useful insight into the power consumption of UAVs under different speeds and conditions, offering critical data support for our subsequent research. In the next stage, we will incorporate the takeoff, landing, and return phases. By comprehensively analyzing the energy consumption and communication requirements of UAVs during these various flight phases, we aim to further optimize UAV path planning and energy management. This will contribute to the development of a more comprehensive and efficient UAV flight model.

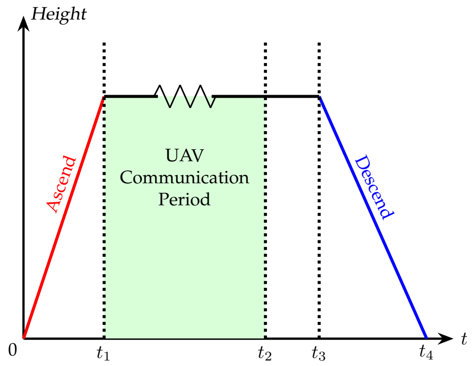


Fig. 13. A general UAV flight pattern.

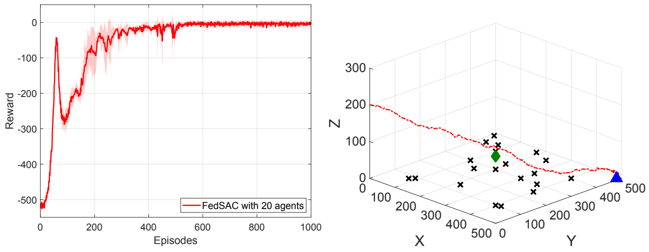
To validate the above idea, we deployed a charging station at coordinates (500, 500, 0) as a navigation target. The reinforcement learning framework was accordingly modified with:

- Reduced action space to five horizontal directions (stay, forward, backward, left, right) for improved training efficiency;
- Enhanced reward function incorporating target distance: $r(s[n], a[n])' = r(s[n], a[n]) + \Delta d$;
- Modified termination condition requiring the UAV to reach the charging station.

We conducted experiments with 20 agents and 20 ground terminals to test the scalability of our method. The results demonstrate stable convergence and robust performance in this larger-scale scenario, as shown in Fig. 14 and Fig. 15.

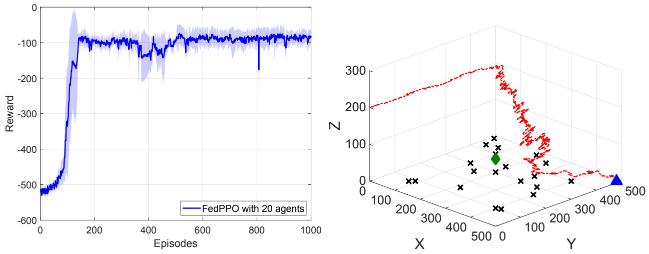
6 Conclusion

In this paper, we have investigated UAV trajectory planning in RIS-assisted UAV communications in urban areas. We have developed an incomplete information communication model and a quadrotor UAV energy consumption model and formulated the UAV's energy consumption problem toward an optimized UAV trajectory. To solve the problem, we have designed an acceleration framework, *FedX*, for reinforcement learning solvers. Two responsive and accurate trajectory planning algorithms, FedSAC and FedPPO, are developed. Our evaluation results show that the proposed framework is effective and efficient and, thus, is applicable to RL solvers in wireless network and mobile computing scenarios. We believe that our work stands out from previous studies by creating an impact on the field of UAV



(a) FedSAC rewards (b) FedSAC 3D trajectory

Fig. 14. FedSAC performance with 20 agents and 20 ground terminals.



(a) FedPPO rewards (b) FedPPO 3D trajectory

Fig. 15. FedPPO performance with 20 agents and 20 ground terminals.

communications. By developing novel models and new algorithms, researchers and practitioners can gain not only in-depth insight into the complex nature of UAV communications but also design fast machine learning algorithms by following our outcomes, paving the way for networking innovations in 6G.

References

[1] "Unmanned Aircraft Systems: Considerations for Law Enforcement Action," https://www.cisa.gov/topics/physical-security/unmanned-aircraft-systems/law-enforcement, accessed: 2024-06-25.

[2] "Science & Tech Spotlight: Drone Swarm Technologies," https://www.gao.gov/products/gao-23-106930, accessed: 2024-06-25.

[3] D. Zhou, M. Sheng, J. Li, and Z. Han, "Aerospace Integrated Networks Innovation for Empowering 6G: A Survey and Future Challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 975–1019, 2023.

[4] G. Geraci, A. Garcia-Rodriguez, M. M. Azari, A. Lozano, M. Mezzavilla, S. Chatzinotas, Y. Chen, S. Rangan, and M. D. Renzo, "What Will the Future of UAV Cellular Communications Be? A Flight From 5G to 6G," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 3, pp. 1304–1335, 2022.

[5] X. Cao, B. Yang, C. Huang, C. Yuen, M. D. Renzo, D. Niyato, and Z. Han, "Reconfigurable intelligent surface-assisted aerial-terrestrial communications via multi-task learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 10, pp. 3035–3050, 2021.

[6] S. Han, J. Wang, L. Xiao, and C. Li, "Broadcast Secrecy Rate Maximization in UAV-Empowered IRS Backscatter Communications," *IEEE Transactions on Wireless Communications*, vol. 22, no. 10, pp. 6445–6458, 2023.

[7] K. Guo, M. Wu, X. Li, H. Song, and N. Kumar, "Deep Reinforcement Learning and NOMA-Based Multi-Objective RIS-Assisted IS-UAV-TNs: Trajectory Optimization and Beamforming Design," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 9, pp. 10 197–10 210, 2023.

[8] J. Zhao, Y. Zhu, X. Mu, K. Cai, Y. Liu, and L. Hanzo, "Simultaneously Transmitting and Reflecting Reconfigurable Intelligent Surface (STAR-RIS) Assisted UAV Communications," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 10, pp. 3041–3056, 2022.

[9] X. Liu, Y. Liu, and Y. Chen, "Machine Learning Empowered Trajectory and Passive Beamforming Design in UAV-RIS Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2042–2055, 2021.

[10] T. P. Truong, V. D. Tuong, N.-N. Dao, and S. Cho, "FlyReflect: Joint Flying IRS Trajectory and Phase Shift Design Using Deep Reinforcement Learning," *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 4605–4620, 2023.

[11] K. K. Nguyen, A. Masaracchia, V. Sharma, H. V. Poor, and T. Q. Duong, "RIS-Assisted UAV Communications for IoT With Wireless Power Transfer Using Deep Reinforcement Learning," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 5, pp. 1086–1096, 2022.

[12] W. Khawaja, I. Guvenc, D. W. Matolak, U.-C. Fiebig, and N. Schneckenburger, "A Survey of Air-to-Ground Propagation Channel Modeling for Unmanned Aerial Vehicles," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2361–2391, 2019.

[13] X. Cheng, Z. Huang, and L. Bai, "Channel Nonstationarity and Consistency for Beyond 5G and 6G: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 3, pp. 1634–1669, 2022.

[14] Y. Zeng, Q. Wu, and R. Zhang, "Accessing From the Sky: A Tutorial on UAV Communications for 5G and Beyond," *Proceedings of the IEEE*, vol. 107, no. 12, pp. 2327–2375, 2019.

[15] H. Zhang, M. Huang, H. Zhou, X. Wang, N. Wang, and K. Long, "Capacity Maximization in RIS-UAV Networks: A DDQN-Based Trajectory and Phase Shift Optimization Approach," *IEEE Transactions on Wireless Communications*, vol. 22, no. 4, pp. 2583–2591, 2023.

[16] Z. Wei, Y. Cai, Z. Sun, D. W. K. Ng, J. Yuan, M. Zhou, and L. Sun, "Sum-Rate Maximization for IRS-Assisted UAV OFDMA Communication Systems," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2530–2550, 2021.

[17] H. Mei, K. Yang, Q. Liu, and K. Wang, "3D-Trajectory and Phase-Shift Design for RIS-Assisted UAV Systems Using Deep Reinforcement Learning," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 3, pp. 3020–3029, 2022.

[18] L. Wang, K. Wang, C. Pan, and N. Aslam, "Joint Trajectory and Passive Beamforming Design for Intelligent Reflecting Surface-Aided UAV Communications: A Deep Reinforcement Learning Approach," *IEEE Transactions on Mobile Computing*, vol. 22, no. 11, pp. 6543–6553, 2023.

[19] M.-L. Tham, Y. J. Wong, A. Iqbal, N. B. Ramli, Y. Zhu, and T. Dagiuklas, "Deep Reinforcement Learning for Secrecy Energy-Efficient UAV Communication with Reconfigurable Intelligent Surface," in *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, 2023, pp. 1–6.

[20] S. Liu, G. Sun, J. Li, S. Liang, Q. Wu, P. Wang, and D. Niyato, "UAV-enabled Collaborative Beamforming via Multi-Agent Deep Reinforcement Learning," *IEEE Transactions on Mobile Computing*, pp. 1–18, 2024.

[21] Q. Sun, J. Niu, X. Zhou, T. Jin, and Y. Li, "AoI and Data Rate Optimization in Aerial IRS-Assisted IoT Networks," *IEEE Internet of Things Journal*, pp. 1–1, 2023.

[22] G. Iacovelli, A. Coluccia, and L. A. Grieco, "Multi-UAV IRS-Assisted Communications: Multi-Node Channel Modeling and Fair Sum-Rate Optimization via Deep Reinforcement Learning," *IEEE Internet of Things Journal*, pp. 1–1, 2023.

[23] X. Yuan, S. Hu, W. Ni, X. Wang, and A. Jamalipour, "Deep Reinforcement Learning-Driven Reconfigurable Intelligent Surface-Assisted Radio Surveillance With a Fixed-Wing UAV," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 4546–4560, 2023.

[24] V. Vishnoi, P. Consul, I. Budhiraja, S. Gupta, and N. Kumar, "Deep Reinforcement Learning Based Energy Consumption Minimization for Intelligent Reflecting Surfaces Assisted D2D Users Underlaying UAV Network," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2023, pp. 1–6.

[25] Y. Qi, Z. Su, Q. Xu, and D. Fang, "Joint Beamforming and Trajectory Optimization for UAV-Assisted Double IRS Secure Transmission System: A Deep Reinforcement Learning Approach," in *2023 IEEE International Conference on Metaverse Computing, Networking and Applications (MetaCom)*, 2023, pp. 504–509.

[26] J. Sun, H. Zhang, X. Wang, M. Yang, J. Zhang, H. Li, and C. Gong, "Leveraging UAV-RIS Reflects to Improve the Security Performance of Wireless Network Systems," *IEEE Networking Letters*, vol. 5, no. 2, pp. 81–85, 2023.

[27] J. Huang, C.-C. Xing, S. Gu, and E. Baker, "Drop Maslow's Hammer or Not: Machine Learning for Resource Management in D2D Communications," *ACM SIGAPP Applied Computing Review*, vol. 22, no. 1, p. 5–14, April 2022.

[28] 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.

[29] M. Eskandari and A. Savkin, "Trajectory Planning for UAVs Equipped With RISs to Provide Aerial LoS Service for Mobile Nodes in 5G/Optical Wireless Communication Networks," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 6, pp. 8216–8221, 2023.

[30] M. Abdel-Basset, R. Mohamed, K. M. Sallam, I. M. Hezam, K. Munasinghe, and A. Jamalipour, "A Multiobjective Optimization Algorithm for Safety and Optimality of 3-D Route Planning in UAV," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 60, no. 3, pp. 3067–3080, 2024.

[31] X. Yu, W.-N. Chen, T. Gu, H. Yuan, H. Zhang, and J. Zhang, "ACO-A*: Ant Colony Optimization Plus A* for 3-D Traveling in Environments With Dense Obstacles," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 617–631, 2019.

[32] V. Roberge, M. Tarbouchi, and G. Labonté, "Fast Genetic Algorithm Path Planner for Fixed-Wing Military UAV Using GPU," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 5, pp. 2105–2117, 2018.

[33] Z. Yu, Z. Si, X. Li, D. Wang, and H. Song, "A Novel Hybrid Particle Swarm Optimization Algorithm for Path Planning of UAVs," *IEEE Internet of Things Journal*, vol. 9, no. 22, pp. 22 547–22 558, 2022.

[34] A. B. M. Adam, X. Wan, M. A. M. Elhassan, M. S. A. Muthanna, A. Muthanna, N. Kumar, and M. Guizani, "Intelligent and Robust UAV-Aided Multiuser RIS Communication Technique With Jittering UAV and Imperfect Hardware Constraints," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 8, pp. 10 737–10 753, 2023.

[35] Y. Qin, Z. Zhang, X. Li, W. Huangfu, and H. Zhang, "Deep Reinforcement Learning Based Resource Allocation and Trajectory Planning in Integrated Sensing and Communications UAV Network," *IEEE Transactions on Wireless Communications*, vol. 22, no. 11, pp. 8158–8169, 2023.

[36] R. Dong, B. Wang, K. Cao, J. Tian, and T. Cheng, "Secure Transmission Design of RIS Enabled UAV Communication Networks Exploiting Deep Reinforcement Learning," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 6, pp. 8404–8419, 2024.

[37] M. Tuchler, A. Singer, and R. Koetter, "Minimum Mean Squared Error Equalization Using A Priori Information," *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 673–683, 2002.

[38] A. R. S. Bramwell, G. Done, and D. Balmford, *Bramwell's Helicopter Dynamics*, 2nd ed. Washington, DC, USA: Butterworth-Heinemann, 2001.

[39] A. Filippone, *Flight Performance of Fixed and Rotary Wing Aircraft*. Washington, DC, USA: Butterworth-Heinemann, 2006.

[40] H. Gong, B. Huang, B. Jia, and H. Dai, "Modeling Power Consumptions for Multirotor UAVs," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 6, pp. 7409–7422, 2023.

[41] Q. Duan, J. Huang, S. Hu, R. Deng, Z. Lu, and S. Yu, "Combining Federated Learning and Edge Computing Toward Ubiquitous Intelligence in 6G Network: Challenges, Recent Advances, and Future Directions," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2892–2950, 2023.

[42] 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," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: http://arxiv.org/abs/1509.02971

[43] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," *CoRR*, vol. abs/1802.09477, 2018. [Online]. Available: http://arxiv.org/abs/1802.09477

[44] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *CoRR*, vol. abs/1707.06347, 2017. [Online]. Available: http://arxiv.org/abs/1707.06347

[45] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, Stockholm, SWEDEN, Jul. 2018, pp. 1861–1870.

[46] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust Region Policy Optimization," in *32nd International Conference on Machine Learning. (ICML)*, Lille, FRANCE, Jul. 2015, pp. 1889–1897.

[47] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous Methods for Deep Reinforcement Learning," in *33rd International Conference on Machine Learning. (ICML)*, New York, NY, Jun. 2016, pp. 1928–1937.

[48] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

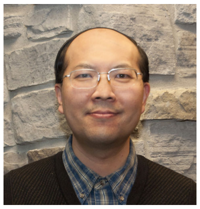
[49] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[50] R. Grande, T. Walsh, and J. How, "Sample Efficient Reinforcement Learning with Gaussian Processes," in *International Conference on Machine Learning (ICML)*, Bejing, PEOPLES R CHINA, Jun. 2014, pp. 1332–1340.

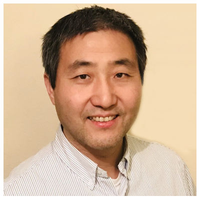
[51] W. Liu, L. Chen, Y. Chen, and W. Zhang, "Accelerating Federated Learning via Momentum Gradient Descent," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 8, pp. 1754–1766, 2020.

Jun Huang (M'12-SM'16) received a Ph.D. degree (with honors) from the Institute of Network Technology, Beijing University of Posts and Telecommunications, China, in 2012. He is now an Assistant Professor in the Department of Electrical Engineering and Computer Science (EECS) at South Dakota State University. Before that, he was a non-tenure track faculty member at Baylor University. He held a full professor appointment at Northwestern Polytechnical University and Chongqing University of Posts and Telecommunications in China from 2015 to 2021. Dr. Huang was a Visiting Scholar at the University of British Columbia, a Research Fellow at the South Dakota School of Mines & Technology and the University of Texas at Dallas, and a Guest Professor at the National Institute of Standards and Technology. He was the recipient of the Outstanding Research Award (Tier I) from CQUPT in 2019, the Best Paper Award from EAI Mobimedia in 2019, Outstanding Service Award from ACM RACS in 2017, 2018, and 2019, Best Paper Nomination from ACM SAC in 2014, and Best Paper Award from AsiaFI 2011. He is the Technical Editor of ACM SIGAPP Applied Computing Review and an Associate Editor of Elsevier Digital Communications and Networks and ICT Express. He guest-edited several special issues in IEEE/ACM journals. He also chaired and co-chaired multiple conferences in the communications and networking areas and organized numerous workshops at major IEEE and ACM events. He is a Senior Member of the IEEE.

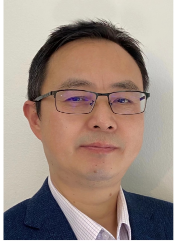
Beining Wu (Student Member, IEEE) received his BS degree in mathematics and applied mathematics from Anhui Normal University, Wuhu, China in 2024. He is currently pursuing a Ph.D. degree in Computer Science at South Dakota State University (SDSU), Brookings, United States. His research interests include wireless communications, UAV networks, and reinforcement learning.



Qiang Duan (Senior Member, IEEE) is a Professor of Information Sciences and Technology at Pennsylvania State University Abington College. His general research interests include computer networking, distributed systems, and artificial intelligence, with recent research focusing on network virtualization and softwarization, network-edge-cloud convergence, federated and split learning, and ubiquitous intelligence in future Internet. Prof. Duan has published four monographs, six book chapters, and 120+ refereed journal articles and conference papers. He has served on the editorial boards as an editor/associate editor for multiple research journals and has been involved in organizing numerous international conferences as a TPC member and track/session chair. Prof. Duan received his Ph.D. in Electrical Engineering from the University of Mississippi in 2003. He is a Senior Member of the IEEE.



Liang (Leon) Dong (Senior Member, IEEE) received his B.S. degree in applied physics with a minor in computer engineering from Shanghai Jiao Tong University, China, in 1996, and his M.S. and Ph.D. degrees in electrical and computer engineering from the University of Texas at Austin in 1998 and 2002, respectively. Since 2011, he has been with Baylor University, where he is currently an Associate Professor of Electrical and Computer Engineering. His research interests include digital signal processing, wireless communications and networking, cyber-physical systems and security, and AI/ML applications in signal processing and communications. Dr. Dong's work has been supported by NSF, DoD, NASA, and industry partners such as L3Harris, Intel, and ExxonMobil. He has extensive industry experience in smart antenna communications systems and wireless networking technologies. Previously, he held academic positions at Western Michigan University and was a Visiting Researcher at Stanford University. Dr. Dong is a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE) and a Member of the American Physical Society (APS).



Shui Yu (IEEE F'23) obtained his PhD from Deakin University, Australia, in 2004. He is a Professor of School of Computer Science, Deputy Chair of University Research Committee, University of Technology Sydney, Australia. His research interest includes Cybersecurity, Network Science, Big Data, and Mathematical Modelling. He has published five monographs and edited two books, more than 500 technical papers at different venues, such as IEEE TDSC, TPDS, TC, TIFS, TMC, TKDE, TETC, ToN, and INFOCOM. His current h-index is 76. Professor Yu promoted the research field of networking for big data since 2013, and his research outputs have been widely adopted by industrial systems, such as Amazon cloud security. He is currently serving the editorial boards of IEEE Communications Surveys and Tutorials (Area Editor) and IEEE Internet of Things Journal (Editor). He served as a Distinguished Lecturer of IEEE Communications Society (2018-2021). He is a Distinguished Visitor of IEEE Computer Society, and an elected member of Board of Governors of IEEE VTS and ComSoc, respectively. He is a member of ACM and AAAS, and a Fellow of IEEE.