FedTD3: An Accelerated Learning Approach for UAV Trajectory Planning

Beining Wu[1], Jun Huang[1(✉)], and Qiang Duan[2]

[1] EECS Department, South Dakota State University, Brookings, SD 57006, USA
Wu.Beining@jacks.sdstate.edu, Jun.Huang@sdstate.edu
[2] IST Department, The Pennsylvania State University, Abington, PA 19001, USA
qxd2@psu.edu

Abstract. Uncrewed Aerial Vehicle (UAV) trajectory planning has been realized to have a significant impact on precision agriculture in enabling more efficient monitoring of crops and soil through optimized flight paths and data collection. While current learning-based algorithms may yield promising results, their training process and the system's highly dynamic channel cause these algorithms to be extremely slow. To address this issue, we design a learning acceleration framework with an efficient algorithm, FedTD3. Our main contributions include a channel model that characterizes UAV-GS (Ground Sensors) communications features in rural areas, a quadrotor UAV energy consumption model for its movement in any direction, and FedTD3–an accelerated RL solver to deal with the problem of time efficiency and sustainability. We perform thorough evaluations to validate the algorithm's performance. The results show that our design can achieve a significant speedup.

Keywords: UAV · Trajectory planning · Reinforcement learning · TD3 · Federated learning

1 Introduction

Uncrewed aerial vehicles (UAV), also known as drones, are aircraft without a human pilot onboard that are controlled remotely by an operator or programmed to fly autonomously [2]. Recent studies have indicated that UAV communications present tremendous benefits in precision agriculture, especially in the rural region of the Midwest [1]. By communicating with ground sensors, UAVs collect data such as crop health, growth patterns, and environmental conditions to provide a more complete and accurate map of the agricultural environment. Moreover, an optimized UAV flying path can help farmers make informed decisions in real time, leading to the optimal application of water, fertilizers, and pesticides, reducing waste and environmental impact. Also, with the automation of communication between UAVs and ground sensors, continuous monitoring and response, streamlining operations, and reduced manual interventions can be realized to achieve more sustainable and productive agricultural practices.

Z. Cai et al. (Eds.): WASA 2025, LNCS 15686, pp. 13–24, 2025.
https://doi.org/10.1007/978-981-96-8725-1_2

Therefore, investigating UAV communications and its trajectory planning is imperative, not only because UAV communications have become an indispensable component for interconnecting space and terrestrial networks to implement the *space-air-ground* integrated systems in 6G [15], but also because they have a significant impact on reshaping precision agriculture, enabling more efficient crop/soil monitoring through optimized flight paths and data collection.

While traditional models used for terrestrial systems could theoretically be adapted for UAV communications, the unique nature of UAV systems demands a more individualized approach. Unlike conventional ground-based communication systems, UAVs operate at relatively higher altitudes, which calls for a customized model to accurately characterize their signal propagation features. Such an adaption is crucial in understanding communication dynamics at various altitudes. At present, there has been a significant amount of studies in channel measurements and modeling specifically for UAV communications [11]. However, prior research in UAV communications was primarily designed for single-rotor UAVs, leading to a significant gap in research for quadrotor UAV scenarios. This gap can be even more obvious with the increasing use of quadrotor UAVs across various sectors, where existing works fail to meet their requirements [8]. In addition, existing studies typically only take into account the vertical ascension of UAVs, which does not reflect the settings of real-world operations.

For UAV trajectory planning, current works formulate the problem as a co-design with system parameters, such as rate/capacity [3], phase shift [13], energy [12], and beamforming [7]. The integration of trajectory design with various system parameters gives rise to the defined optimization problems NP-hard. Consequently, various Reinforcement Learning (RL) algorithms have been proposed to tackle these optimization problems. Typical RL algorithms include Deep Deterministic Policy Gradient (DDPG) [7], Soft Actor-Critic (SAC) [3], Proximal Policy Optimization (PPO) [10], and Twin-delayed DDPG (TD3) [13]. Although these algorithms may yield promising results, *we argue that they deliberately neglect the time performance of their solutions.* This can be ascribed to our recent investigation in [9] that the RL-based algorithm's training process and the system's highly dynamic channel can result in the algorithm being extremely slow, even in a simple toy scenario.

The goal of this paper is to model the UAV-GS (Ground Sensors) communications precisely, design efficient trajectory planning algorithms, and conduct system evaluations to bridge the above research gaps. Our contributions of this paper are summarized as follows.

- We develop a system model that includes: 1) a channel model that characterizes UAV-GS communications features in rural areas and 2) a quadrotor UAV energy consumption model for its movement in any direction. Based on the system model, we formulate the optimization problem for UAV trajectory planning.
- We present a learning acceleration framework, under which an efficient and lightweight RL algorithm for UAV trajectory planning to address the challenges of time efficiency and sustainability is designed. Our main endeavor

is to guarantee the designed algorithm can be responsive and accurate for complex UAV operations.
– We perform evaluations of the UAV trajectory planning algorithm to validate its performance. We believe that the research efforts made in this work will provide empirical evidence and feedback to refine theoretical designs, ensuring that these algorithms are not only theoretically sound but also practically viable and ready for industry adoption.

The remainder of this paper is organized as follows. Section 2 presents the system model and formulates the problem. In Sect. 3, the proposed approach for the problem is given. The simulation results are provided in Sect. 4. We conclude the paper in Sect. 5.

2 System Model

We consider a communication system involving a UAV connecting with K Ground Sensors (GS) distributed across a farmland. The UAV functions as an aerial base station. Both the UAV and GTs are equipped with a single antenna.

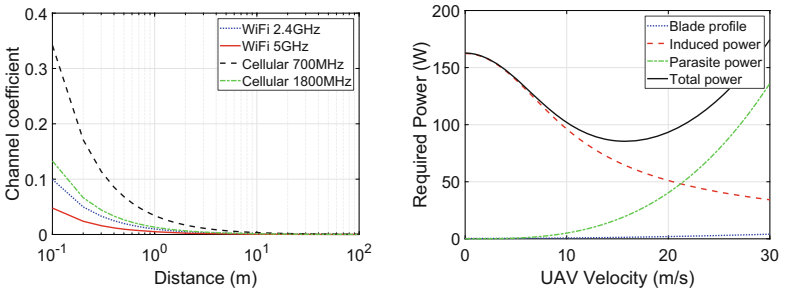
2.1 UAV-Ground-Sensors Communication Model

The general wireless channel model can be written as $h = \sqrt{\alpha(d)}\tilde{h}$. Here, $\alpha(d)$ represents the large-scale channel attenuation, which includes distance-dependent path loss and shadowing. The variable d signifies the distance between the UAV and GS. The complex random variable $\tilde{h}$ accounts for small-scale fading caused by multipath propagation, with $E\left[|\tilde{h}|^2\right] = 1$. In rural areas, the effects of shadowing and small-scale fading disappear due to free space transmission. As such, we have $|\tilde{h}| = 1$, and the channel power is simplified as $\alpha(d) = \left(\frac{\lambda}{4\pi d}\right)^2$, where λ is the carrier wavelength and $\tilde{\alpha}_0 \triangleq (\lambda/4\pi)^2$ is the channel power at the reference distance of 1 m. With this model, the channel power is fully determined by the distance between the UAV and GS, which can be accurately predicted given known locations. Figure 1a illustrates our initial findings utilizing this model, showcasing various potential transmission technologies and wireless bands.

The data rate of UAV is

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

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



(a) Channel model (b) Required power

Fig. 1. Preliminary results on channel and power characteristics for UAV communications and flight.

2.2 Energy Consumption Model for UAV

We focus on the UAV's energy consumption model, emphasizing propulsion, which is the dominant factor. We assume that the energy required for communication and computation tasks remains constant. Additionally, we neglect the minor fluctuations in energy consumption due to UAV acceleration or deceleration as long as the communication time slot is relatively short. The model is partially adapted from earlier studies [6].

For a quadrotor UAV, the energy consumption during hovering can be divided into two components:

1) The power consumed due to the aerodynamic drag of the UAV's rotors:

$$P_{\mathrm{B}} = \frac{\delta_0}{2}\rho s_0 A_0 \Omega_0^3 R_0^3, \tag{2}$$

2) The power required to balance the UAV's weight and maintain lift:

$$P_{\mathrm{I}} = (1+k)\frac{W^{3/2}}{2\sqrt{2\rho A_0}}. \tag{3}$$

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

According to Eqs. (2) and (3), due to the varying effects of horizontal flight on the different power components, including increased aerodynamic drag on the rotors, the non-linear variation of induced power with speed, and the growing parasite power caused by air resistance, we will now present the corrected power for the aerodynamic drag of the rotor $P_{\mathrm{B_h}} = 4P_{B_0}\left(1 + \frac{3\bar{V}^2}{\Omega_0^2 R_0^2}\right)$, the induced power correction $P_{\mathrm{I_h}} = 4P_{I_0}\left(\sqrt{1 + \frac{\bar{V}^4}{4v_0^4}} - \frac{\bar{V}^2}{2v_0^2}\right)^{1/2}$, and the parasite drag power

$P_{\mathrm{D_h}} = 2d_0\rho s_0 A_0 \bar{V}^3$, respectively. Based on this, we can derive the total power consumption for the UAV in horizontal flight:

$$\bar{P}(\bar{V}) = P_{\mathrm{B_h}} + P_{\mathrm{I_h}} + P_{\mathrm{D_h}}. \tag{4}$$

Figure 1b displays our preliminary results using the same parameter configuration as [6] for this model.

Next, we analyze the power consumption during vertical flight. Let $\hat{F}$ and $\check{F}$ represent the upward and downward forces, and $\hat{R}$ and $\check{R}$ denote the drag during ascent and descent, respectively. Next, taking the ascent process as an example, the thrust of the UAV during ascent satisfies:

$$\hat{F}_0 = \frac{W}{4} + \frac{1}{2} S_{\mathrm{EQ}\perp} \rho \hat{V}^2, \tag{5}$$

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

In line with [5], we have

$$\hat{P}_0(\hat{V}, \hat{F}_0) = \frac{P_h^{\mathrm{quad}}}{4} + \frac{1}{2}\hat{F}_0\hat{V} + \frac{\hat{F}_0}{2}\sqrt{\hat{V}^2 + \frac{2\hat{F}_0}{\rho A_0}}. \tag{6}$$

Since the UAV is a quadrotor, we can easily derive the total power consumption during ascent $\hat{P}(\hat{V}) = 4\hat{P}_0\left(\hat{V}, \hat{F}_0\right)$ and descent $\check{P}(\check{V}) = 4\hat{P}_0\left(\hat{V}, \hat{F}_0\right) - 2S_{\mathrm{EQ}\perp}\rho\check{V}^3$.

2.3 Problem Formulation

Following the same convention of [14], we make the assumption that the duration of flight for a particular time slot, denoted by $\delta_t[n]$, as well as the overall flight time T, which is the sum of $\delta_t[n]$ for all n from 1 to N. The UAV's 3D path is represented by a sequence $\left\{\mathbf{q}[n] = [x[n], y[n], z[n]]^{\mathrm{T}}\right\}_{n=1}^{N}$, 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 $H_U^{\min} \le z[n] \le H_U^{\max}$. The locations of the ground sensors are fixed and denoted by $\mathbf{L}_k = [x_k, y_k, 0]^{\mathrm{T}}$, where $\mathbf{L}_k$ represents the coordinates of GS k.

Based on the aforementioned settings for the flight time slots and positions, we can separately derive the UAV's speed during horizontal flight $\bar{V} = \frac{\sqrt{(x[n+1]-x[n])^2+(y[n+1]-y[n])^2}}{\delta_t[n]}$, as well as its speed during ascent and descent $\hat{V} = \check{V} = \frac{\sqrt{(z[n+1]-z[n])^2}}{\delta_t[n]}$.

Based on the UAV's flight speed, flight time slots, and total power consumption, we can derive the UAV's energy consumption model:

$$E[n] = \delta_t[n]\left(\bar{P}(\bar{V}) + \hat{P}_0(\hat{V})\right), \tag{7}$$

if the UAV descends during the n-th time slot, we have:

$$E[n] = \delta_t[n]\left(\bar{P}(\bar{V}) + \check{P}(\check{V})\right). \tag{8}$$

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

$$\begin{aligned} \min_{\mathbf{q}[n], c_{k,i}[n]} \quad & \sum_{n=1}^{N} E[n] \\ \text{s.t.} \quad & \sum_{k=1}^{K} c_{k,i}[n] \leq 1 \\ & \sum_{n=1}^{N} \delta_t[n] R_{k,i}[n] \geq L_k \\ & \bar{V} \leq \bar{V}_{\max} \\ & \hat{V}, \check{V} \leq \tilde{V}_{\max} \\ & H_U^{\min} \leq z[n] \leq H_U^{\max} \end{aligned} \tag{9}$$

where the first constraint in (9) indicates that at most one GS 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 our problem takes GS locations $\mathbf{L}_k$ and tasks L_k as inputs, and outputs the optimized UAV trajectory. The 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.

3 Proposed Approach

We start with modeling the UAV trajectory planning as an MDP:

1. *State*: the current position of UAV in time slot n, $s[n]$;
2. *Action*: the UAV trajectory, flight time, and ground terminals scheduling choices that have been taken in time slot n, $a[n]$;
3. *Reward*: the reward of state-action pair $(s[n], a[n])$ is defined as $r(s[n], a[n]) = \sum_{k=1}^{K} \sum_{n'=1}^{n+1} \frac{\delta_t R_{k,i}[n]}{E[n]} - \lambda d' - \mathrm{p}_0$, and the Q-value of each action is given as $Q(s[n], a[n]) = \mathbb{E}\left[\sum_{n'=n}^{N} \gamma r(s[n'], a[n']) | s(n), a(n)\right]$, p_0 is a penalty employed for the second constraint in (9). $\gamma \in (0, 1]$ is the discount factor.

3.1 FedX: A Flower-Based Accelerated Learning Framework

To resolve this issue, we leverage the federated learning technique provided by the Flower and propose an accelerated framework called *FedX*, as shown in Algorithm 1. The key idea behind this algorithm is to utilize the Flower

Algorithm 1: FedX with Flower

Data: Number of clients M, federated learning rounds E, initial global parameters w_G^0, learning rate η

Initialize Flower server and M clients
for $e \leftarrow 1$ *to* E **do**
  for *each client* $m \in \{1, \ldots, M\}$ *in parallel* **do**
    Server sends global model w_G^e to client m
    Client m initializes local model $w_m^e = w_G^e$
    Client m computes local update using RL algorithm X
    Client m updates model and sends w_m^{e+1} to server
  end
  Server receives updates w_m^{e+1} from each client m
  Server aggregates: $w_G^{e+1} = \frac{\sum_{m=1}^{M} D_m w_m^{e+1}}{\mathcal{D}}$
  Server sends updated global model w_G^{e+1} to all clients
end

platform to manage multiple clients (they can be processes, threads within a process, hosts, or mobile terminals), where each client acts as an agent in the federated learning process to enable parallel training [4]. *"X" in Algorithm 1 can be any RL solver, including but not limited to DQN, DDQN, DDPG, TD3, and other algorithms of the same kind.*

Note that the UAV's action space combines flight parameters and scheduling decisions. Discrete RL methods (DQN/DDQN) would sacrifice precision, while DDPG suffers from Q-value overestimation for continuous actions. We, therefore, employ TD3 to optimally handle this complex action space.

3.2 FedTD3

TD3 is an off-policy DRL technique using dual critics to mitigate DDPG's overestimation bias. Algorithm 2 shows our system framework. TD3 uses actor networks $\pi(s \mid \vartheta_\pi)$, critic networks $Q(s, a \mid \vartheta_{q_i})$, and their target versions. It employs delayed policy updates and target policy smoothing with clipped noise $\zeta \sim \text{clip}\left(\mathcal{N}(0, \sigma_\zeta^2), -c, c\right)$ to enhance stability. Transitions are stored in a replay buffer and randomly sampled to break sequential correlations during training.

For each mini-batch, the target value $\mathcal{Y}$ is computed using the reward r received from the environment and the bootstrapped estimate of the future value of the state-action pair, discounted by the factor γ. The target value is calculated as follows:

$$\mathcal{Y} = r + \gamma \min_{i=1,2} \left\{ Q_{\vartheta_i'}(s_{t+1}, \bar{a}) \right\}, \tag{10}$$

where $\min_{i=1,2} \left\{ Q_{\vartheta_i'} \right\}$ refers to the minimum of the two target critic networks, and $\bar{a}$ represents the action chosen at the next time step s_{t+1}, which has been smoothed by adding random noise ζ to prevent the deterministic policy from overfitting to specific actions.

Algorithm 2: TD3 Algorithm

Data: Episodes E, steps per episode N, buffer size $N_{\mathcal{M}}$, batch size $N_{\mathcal{D}}$, discount γ, learning rate η, noise ζ
Result: Optimal action $a(n) = (x_n^h, x_n^{dh}, x_n^{dv}, c_{k,n}, \delta_t[n])$

Initialize actor network π_{ϑ_π}, target actor $\pi_{\vartheta'_\pi}$
Initialize critic networks $Q_{\vartheta_{q_1}}$, $Q_{\vartheta_{q_2}}$ and targets $Q_{\vartheta'_{q_1}}$, $Q_{\vartheta'_{q_2}}$
Initialize replay buffer $\mathcal{M}$ with size $N_{\mathcal{M}}$
for *episode* $\leftarrow 1$ *to* E **do**
    Set $n = 1$, initialize state $s(1)$
    while $n \leq N$ *and task* L_k *not finished* **do**
        Select action $a \sim \pi_{\vartheta_\pi}(a|s)$
        if *UAV out of region or exceeding velocity* **then**
            Cancel action and apply penalty
        end
        Execute action a, observe reward r and next state s'
        Store transition (s, a, r, s') in buffer $\mathcal{M}$
        Sample mini-batch from $\mathcal{M}$
        Compute target value $\mathcal{Y}$ using (10)
        Update critics by minimizing (11)
        if *every two critic updates* **then**
            Update actor using (12)
        end
        Update target networks using (13)
        $n \leftarrow n + 1$
    end
end

Once the target value $\mathcal{Y}$ is computed, the two critic networks $Q(s, a \mid \vartheta_{q_1})$ and $Q(s, a \mid \vartheta_{q_2})$ are updated by minimizing the Bellman loss. The loss function for each critic is given by:

$$L(\vartheta_{q_i}) = \text{argmin}_{\vartheta_{q_i}} \frac{1}{N_{\mathcal{D}}} \sum_{j=1}^{N_{\mathcal{D}}} \left(\mathcal{Y} - Q_{\vartheta_{q_i}}(s_t, a_t)\right)^2. \tag{11}$$

The critics are trained by minimizing the difference between the predicted Q-value and the target value $\mathcal{Y}$, ensuring accurate value estimation.

After every two critic updates, the actor network $\pi(s \mid \vartheta_\pi)$ is updated using the deterministic policy gradient algorithm. The actor aims to maximize the expected return by adjusting its policy according to the first critic network Q_{q_1}. The policy gradient can be expressed as:

$$\nabla_{\vartheta_\pi} J(\vartheta_\pi) = \frac{1}{N_{\mathcal{D}}} \sum_{j=1}^{N_{\mathcal{D}}} \nabla_a Q_{q_1}(s_t, a_t) \nabla_{\vartheta_\pi} \pi(s_t \mid \vartheta_\pi). \tag{12}$$

In this process, the gradient of the critic with respect to the action is used to update the actor's parameters ϑ_π. By following the gradient of the critic, the actor is guided toward selecting actions that maximize the expected return.

To maintain stability during training, TD3 applies delayed updates to the target networks. The parameters of the target actor and target critics are updated with soft updates, controlled by a Polyak averaging coefficient τ, ensuring that the target networks evolve more smoothly:

$$\begin{aligned} \vartheta_{\pi'} &\leftarrow \tau\vartheta_\pi + (1-\tau)\vartheta_{\pi'} \\ \vartheta_{q_i'} &\leftarrow \tau\vartheta_{q_i} + (1-\tau)\vartheta_{q_i'}. \end{aligned} \tag{13}$$

The complexity of TD3 is $O(n \cdot m^2)$ per network forward pass, where n is the layer count and m is the neuron count. With mini-batch operations and network updates, the overall complexity for E episodes with N steps becomes $O(E \cdot N \cdot n \cdot m^2)$.

4 Performance Evaluation

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

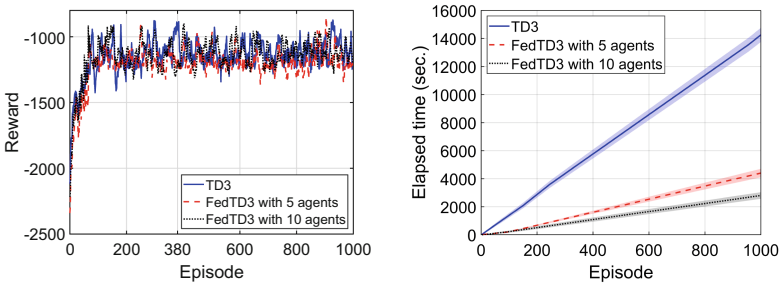
We set up an air-to-ground communication scenario where the UAV's initial position is set at $[0, 0, 0]$. At each stage, the UAV operates to achieve an optimal balance between communication quality and energy efficiency. The simulation was conducted in Python 3.10 to implement the deep neural networks (DNNs) used in the TD3 algorithm.

Table 1. Parameter settings for simulations.

Parameter	Value
Bandwidth B	2 MHz
GTs: K, Task: L_k	4, $1024 \sim 2048$ Kb
$\bar{V}_{\max}, \tilde{V}_{\max}$	10 m/s, 10 m/s
t_{min}, t_{max}	1 s, 3 s
Flying height: h_{min}, h_{max}	0, 200 m
Time slots and episodes	600, 1000
Area size (width × depth × height)	500 m × 500 m × 200 m
Transmission power (p^{TX})	1 W
Noise power (σ)	$\sqrt{3.98 \times 10^{-12}}$
The positions of the GTs	$[100\,\mathrm{m}, 100\,\mathrm{m}]^T$, $[100\,\mathrm{m}, 400\,\mathrm{m}]^T$, $[400\,\mathrm{m}, 100\,\mathrm{m}]^T$, $[400\,\mathrm{m}, 400\,\mathrm{m}]^T$

In the TD3 algorithm, the Actor network consists of a four-layer structure. The first layer contains 128 neurons, followed by two residual blocks, each consisting of two fully connected layers and ReLU activation functions. The output

layer uses a Tangent Hyperbolic (Tanh) function to ensure the actions are within the required range. The Critic network also has a four-layer structure, starting with 128 neurons, followed by two residual blocks, and a final fully connected layer to output the estimated Q-value. The Adam optimizer is used to train the neural networks, with weights initialized using Xavier uniform initialization.



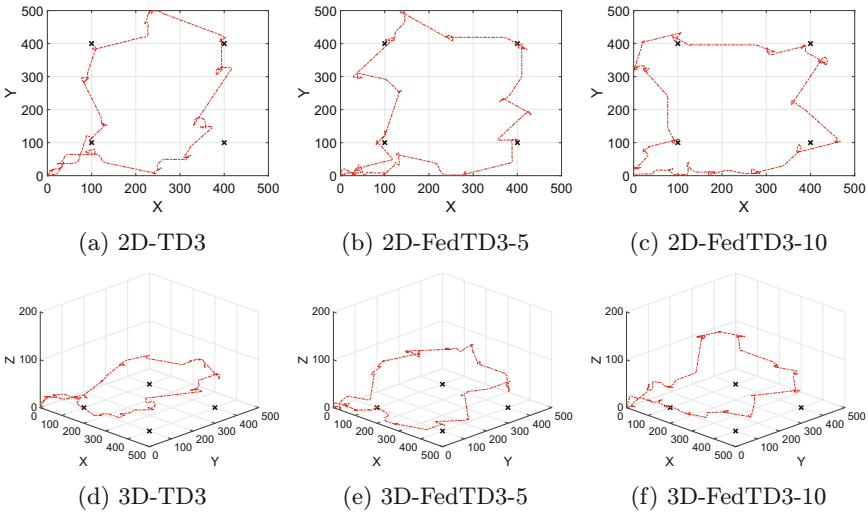
(a) Reward (b) Training time

Fig. 2. Rewards and time comparisons of TD3 and FedTD3.

Figure 2 shows the rewards and training time of the TD3, FedTD3 with 5 agents, and FedTD3 with 10 agents over 1,000 episodes. Figure 2a shows that the algorithms converge around the 380th episode. Notably, the convergence of the TD3 and FedTD3 are very similar. This indicates that the strong performance of FedTD3 can be attributed to the synchronous updates used by the Flower-based federating implementation, which ensures stability in model updates during the learning process. Figure 2b illustrates the average time required by the TD3 and FedTD3 to complete 1,000 episodes of training. For each curve, the shaded area represents the standard deviation. The results in this figure faithfully demonstrate that the adoption of *FedX* significantly reduced training durations.

Figure 3 illustrates a comparison of 2D and 3D trajectories generated by the TD3 and FedTD3. It can be observed that, as the UAV takes off from the origin, it must reach the minimum required altitude to ensure safety and avoid low-altitude flight. After reaching the minimum flight altitude, the UAV typically flies close to each GT to establish a stable communication link, potentially descending to a lower altitude or hovering as needed. The underlying reason for this behavior is to optimize signal reception quality while minimizing energy consumption.

Figure 4 shows the cumulative distribution function (CDF) of UAV energy consumption and throughput. The results indicate that the performance of the TD3 and FedTD3 is essentially the same. This demonstrates that our proposed framework can significantly reduce training time without compromising solution accuracy.



(a) 2D-TD3 (b) 2D-FedTD3-5 (c) 2D-FedTD3-10

(d) 3D-TD3 (e) 3D-FedTD3-5 (f) 3D-FedTD3-10

Fig. 3. Trajectory comparison between TD3 and FedTD3.

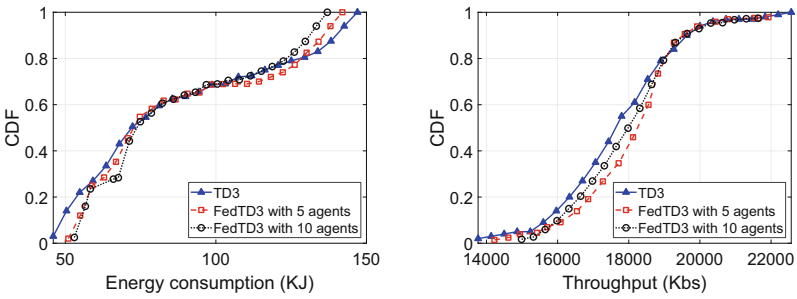


Fig. 4. System performance comparison.

5 Conclusion

In this work, we proposed FedTD3, a novel algorithm for UAV trajectory planning in precision agriculture, addressing slow training and dynamic communication challenges. Our model improves time efficiency and sustainability by incorporating realistic UAV-GS communication and energy consumption models. The distributed nature of our federated approach enables seamless scalability to larger networks with increasing numbers of agents and diverse environmental conditions. Evaluations showed significant speedup over conventional methods, which makes our solution highly effective for rural environments. Future work can extend this approach to more complex and multi-UAV scenarios, in order to offer a scalable solution for UAV-based agricultural practices.

Acknowledgments. This work was supported by the National Science Foundation under CNS-2348422.

References

1. Drones For Farmers Makes Midwest Startup A Flying Success. https://www.forbes.com/sites/indiana/2017/01/30/drones-for-farmers-makes-midwest-startup-a-flying-success/?sh=7aed4a5f37dc. Accessed 30 Feb 2025
2. Unmanned Aircraft Systems: Considerations for Law Enforcement Action. https://www.cisa.gov/topics/physical-security/unmanned-aircraft-systems/law-enforcement. Accessed 06 Mar 2025
3. Cao, Y., Luo, Y., Yang, H., Luo, C.: UAV-based emergency communications: an iterative two-stage multiagent soft actor-critic approach for optimal association and dynamic deployment. IEEE Internet Things J. **11**(16), 26610–26622 (2024)
4. Duan, Q., Huang, J., Hu, S., Deng, R., Lu, Z., Yu, S.: Combining federated learning and edge computing toward ubiquitous intelligence in 6g network: challenges, recent advances, and future directions. IEEE Commun. Surv. Tutor. **25**(4), 2892–2950 (2023)
5. Filippone, A.: Flight Performance of Fixed and Rotary Wing Aircraft. Butterworth-Heinemann, Washington, DC (2006)
6. Gong, H., Huang, B., Jia, B., Dai, H.: Modeling power consumptions for multirotor UAVs. IEEE Trans. Aerosp. Electron. Syst. **59**(6), 7409–7422 (2023)
7. Guo, K., Wu, M., Li, X., Song, H., Kumar, N.: Deep reinforcement learning and NOMA-based multi-objective RIS-assisted IS-UAV-TNs: trajectory optimization and beamforming design. IEEE Trans. Intell. Transp. Syst. **24**(9), 10197–10210 (2023)
8. Huang, J., Wu, B., Duan, Q., Dong, L., Yu, S.: A fast UAV trajectory planning framework in ris-assisted communication systems with accelerated learning via multithreading and federating. IEEE Trans. Mob. Comput., 1–16 (2025)
9. Huang, J., Xing, C.C., Gu, S., Baker, E.: Drop Maslow's Hammer or not: machine learning for resource management in D2D communications. ACM SIGAPP Appl. Comput. Rev. **22**(1), 5–14 (2022)
10. Iacovelli, G., Coluccia, A., Grieco, L.A.: Multi-uav irs-assisted communications: multinode channel modeling and fair sum-rate optimization via deep reinforcement learning. IEEE Internet Things J. **11**(3), 4470–4482 (2024)
11. Sun, M., Bai, L., Huang, Z., Cheng, X.: Multi-modal sensing data-based real-time path loss prediction for 6g uav-to-ground communications. IEEE Wirel. Commun. Lett. **13**(9), 2462–2466 (2024)
12. Vishnoi, V., Consul, P., Budhiraja, I., Gupta, S., Kumar, N.: Deep reinforcement learning based energy consumption minimization for intelligent reflecting surfaces assisted D2D users underlaying UAV network. In: IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 1–6 (2023)
13. Yuan, X., Hu, S., Ni, W., Wang, X., Jamalipour, A.: Deep reinforcement learning-driven reconfigurable intelligent surface-assisted radio surveillance with a fixed-wing UAV. IEEE Trans. Inf. Forensics Secur. **18**, 4546–4560 (2023)
14. Zeng, Y., Xu, J., Zhang, R.: Energy minimization for wireless communication with rotary-wing UAV. IEEE Trans. Wirel. Commun. **18**(4), 2329–2345 (2019)
15. Zhou, D., Sheng, M., Li, J., Han, Z.: Aerospace integrated networks innovation for empowering 6G: a survey and future challenges. IEEE Commun. Surv. Tutor. **25**(2), 975–1019 (2023)