Reinforcement Learning-Based Decode-and-Forward UAV Relay Trajectory Optimization

Aniket Subbanwar*, Ojas Joshi*, and Amit Agarwal
Department of Electronics and Communication Engineering
Visvesvaraya National Institute of Technology (VNIT), Nagpur, India
Email: aniketsubbanwar05@gmail.com, ojasdjoshi23@gmail.com, amit.nitk07@gmail.com

***Abstract*—Unmanned Aerial Vehicles (UAVs) are promising relay platforms due to their flexible deployment and high probability of line-of-sight (LoS) connectivity. This paper compares three deep reinforcement learning (DRL) algorithms—Proximal Policy Optimization (PPO), Soft Actor-Critic (SAC), and Recurrent PPO with LSTM memory—for joint UAV trajectory and energy optimization in UAV based relay systems. The problem formulated is a non-convex optimization problem that minimizes UAV propulsion energy while satisfying Quality of Service (QoS) and mobility constraints under realistic 3GPP channel conditions. Simulation results show that all methods achieve over 99% QoS satisfaction. SAC exhibits the fastest convergence, whereas the proposed Recurrent PPO achieves the lowest energy consumption (44.72 kJ), reducing energy usage by 5.1% compared with PPO. These results highlight the trade-off between convergence speed and energy efficiency in DRL-based UAV relay optimization.**

***Index Terms*—UAV relay, trajectory optimization, deep reinforcement learning, PPO, SAC, Recurrent PPO, LSTM, energy efficiency, 3GPP channel model, QoS.**

I. Introduction

The exponential growth of mobile data traffic in next-generation wireless networks has intensified the demand for higher data rates and improved quality of service (QoS) [1]. When ground base stations (GBSs) become congested, deploying unmanned aerial vehicles (UAVs) as aerial relays or base stations offers a flexible and rapidly deployable solution [2]. Owing to their adaptive positioning, low deployment latency, and high probability of line-of-sight (LoS) connectivity, UAVs are particularly attractive for emergency communications, temporary coverage, and rural connectivity [3].

Despite these advantages, UAV relay systems are constrained by limited onboard battery capacity and restricted ground user (GU) transmission energy. Joint optimization of UAV energy consumption and communication performance under QoS requirements leads to a challenging mixed-integer nonlinear programming (MINLP) problem due to nonlinear channel behavior, propulsion power characteristics, dynamic obstacles, and the coupling between trajectory and communication variables [4].

Early research focused on UAV placement and energy-efficient operation. Lyu *et al.* [3] investigated optimal UAV deployment, while Zeng *et al.* [4] developed a practical propulsion power model that revealed the inefficiency of both hovering and high-speed flight. To accurately characterize wireless links, recent studies have adopted realistic channel models incorporating probabilistic LoS/NLoS propagation [5], Rician and Rayleigh fading [6], and the measurement-based 3GPP TR 38.901 path-loss framework [7].

Deep reinforcement learning (DRL) has recently emerged as an effective tool for UAV trajectory optimization. Representative works include multi-UAV path planning under interference constraints [8], energy-efficient trajectory design using actor–critic methods [9], cooperative UAV relay networks [10], and attention-based DRL for joint trajectory optimization in 5G/6G systems [11]. Unlike conventional optimization techniques, DRL can learn efficient policies directly through interaction with the environment, making it well suited for non-convex and partially observable UAV communication scenarios.

In this work, we formulate a UAV trajectory optimization problem aimed at minimizing the total energy consumption while ensuring that the minimum required data rate is achieved. In addition, we impose constraints on the mobility of the UAV, allowing it to move freely in a three-dimensional space. Our work compares multiple RL-based models. We also propose a recurrent proximal policy optimization (Recurrent PPO) method which we observe minimizes the energy consumed by the UAV. Specifically, this paper addresses trajectory optimization by implementing three methods: PPO, soft-actor critic (SAC), and Recurrent PPO in a 3D simulation environment with shadowing and Rician/Rayleigh fading. This paper makes the following contributions:

1) We formulate a 3GPP-compliant UAV trajectory optmization problem focusing on energy minimization incorporating spatially correlated shadowing and Rayleigh/Rician fading. Further, rate constraint and UAV movement constraint are imposed.
2) We solve the proposed problem using three DRL based methods and compare these three DRL algorithms: PPO (continuous action space), SAC (continuous, off-policy), and Recurrent PPO (continuous, LSTM memory).
3) We provide a thorough comparative analysis of training convergence, data rate, energy consumption, energy efficiency, and QoS satisfaction along with their trajectory visualizations.

II. System Model and Problem Formulation

We consider a relay system comprising a mobile ground user (GU), a UAV relay, and a fixed base station (BS). The

* Aniket Subbanwar and Ojas Joshi contributed equally to this work.

UAV operates within a 500×500 m^2 area at altitudes between 25 m and 125 m over 200 time slots and employs decode-and-forward relaying. The end-to-end rate is determined by the bottleneck link, i.e., the minimum rate of the UAV–GU and UAV–BS links. Additionally, the UAV must avoid nine cylindrical obstacles throughout the mission.

A. 3GPP-PathLoss Model

1) Path Loss: For urban LoS links (3GPP UMa TR 36.777):

$$L_{\text{UMa}} = 28 + 22\log_{10}(d_{3\text{D}}) + 20\log_{10}\left(\tfrac{f_c}{1000}\right) + \text{CF}, \quad (1)$$

$$\text{CF} = 1.0005\times10^{-4} h_{\text{UAV}}^2 - 0.0286 h_{\text{UAV}} + 10.5169. \quad (2)$$

[7]. For NLoS urban links: L_{NLoS}=$32.4+20\log_{10}(f_c/1000)+30\log_{10}(d_{3\text{D}})$ [12].

2) Shadowing: Log-normal shadowing[1] $X_\sigma \sim \mathcal{N}(0,\sigma^2)$, evolves via a spatially correlated autoregressive model updated every 5 steps:

$$X_\sigma(t+1) = \rho(\Delta d)\cdot X_\sigma(t) + \sqrt{1-\rho^2(\Delta d)}\cdot Z, \quad (3)$$

where $\rho(\Delta d)=\exp(-\Delta d/d_{\text{corr}})$, d_{corr}=20 m, $Z\sim\mathcal{N}(0,\sigma^2)$.

3) Small-Scale Fading: For LoS links, Rician fading with K-factor K_{dB}=10 dB:

$$h = \sqrt{\tfrac{K}{K+1}} + \sqrt{\tfrac{1}{K+1}}\,(h_r + j h_i), \quad h_r, h_i \sim \mathcal{N}(0, \tfrac{1}{\sqrt{2}}), \quad (4)$$

with power gain $|h|^2$. For NLoS links the fading switches to Rayleigh fading (K=0). Total path loss combines all effects:

$$PL_{\text{total}}\,[\text{dB}] = PL_{\text{path}} + X_\sigma - 10\log_{10}(|h|^2). \quad (5)$$

B. SNR and Data Rate

The received SNR for transmit power P and noise power N_0: $\text{SNR} = \frac{P}{PL_{\text{total}}\cdot N_0}$. The achievable spectral efficiency follows Shannon's formula:

$$C = \log_2(1+\text{SNR}), \quad R = C\cdot BW/1000 \text{ kbps}, \quad (6)$$

with BW=20 MHz. The bottleneck rate is:

$$R_{e2e}(t) = \min\{R_{GU\to UAV}(t),\ R_{UAV\to BS}(t)\} \quad (7)$$

C. UAV Energy Model

Adopting the rotary-wing model from [4], propulsion power at velocity v:

$$P_{\text{UAV}}(v) = P_0\left(1+\tfrac{3v^2}{v_{\text{tip}}^2}\right) + P_i\sqrt{\sqrt{1+\tfrac{v^4}{4v_0^4}} - \tfrac{v^2}{2v_0^2}} + \tfrac{1}{2}d_0\rho s A\, v^3, \quad (8)$$

where the three terms model, blade-profile drag, induced power, and parasitic power, respectively.P_0 and P_i are constants of blade profile power and induced power in hovering Per-slot energy: $E_{\text{UAV}}(t)$=$P_{\text{UAV}}(v(t))\cdot\Delta t$. GU transmits at fixed $P_{\text{GU}}^{\max}$ mW, so $E_{\text{GU}}(t)$=$P_{\text{GU}}^{\max}\cdot\Delta t$ [13].

[1]The log normal shadowing is applied to capture residual signal variations due to environmental clutter such as vehicles, vegitation, foliage, and irregular surface reflections.

D. Problem Formulation

$$P1: \min_{\mathbf{q}(t)} \quad E_{\text{total}} = \sum_t E_{\text{UAV}}(t) \quad (9)$$

$$\text{s.t.} \quad R_{\text{e2e}}(t) \ge R_{\text{req}}\ \forall t \quad (\text{C1})$$

$$0 \le x(t), y(t) \le L\ \forall t \quad (\text{C2})$$

$$H_{\min} \le z(t) \le H_{\max}\ \forall t \quad (\text{C3})$$

$$\|\mathbf{q}(t) - \mathbf{q}_k^{\text{obs}}\|_{xy} \ge r_{\text{obs}}\ \forall k,t \quad (\text{C4})$$

where $q_k^{\text{obs}} = (x_k, y_k, 0)$ is the horizontal center of the k–th obstacle, r_{obs} obstacle radius and R_{req} [kbps] is minimum required required rate. Problem (P1) is a non-convex optimization problem that minimizes the total UAV propulsion energy over N time slots. Constraint (C1) ensures a minimum end-to-end rate of R_{req} kbps, while (C2) and (C3) bound the UAV velocity and altitude within feasible limits. Constraint (C4) guarantees obstacle avoidance by maintaining a safe distance from all obstacles. The integer component arises from the binary LoS/NLoS indicator in the path-loss model. Due to the non-convexity and the coupling between communication and trajectory variables, a DRL-based approach is adopted to solve (P1).

III. DRL Framework: Shared MDP Structure

All three algorithms solve the same Markov Decision Process $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$.

A. Reward Function

The reward at step t is:

$$r(t) = r_{\text{QoS}}(t) + r_{\text{energy}}(t) + r_{\text{rate}}(t) + r_{\text{survival}} - p_{\text{constraint}}(t), \quad (10)$$

$$r_{\text{QoS}} = \begin{cases} +10 & R_{\text{e2e}} \ge R_{\text{req}} \\ -20\frac{R_{\text{req}} - R_{\text{e2e}}}{R_{\text{req}}} & \text{otherwise,} \end{cases} \quad (11)$$

$$r_{\text{energy}} = -10\left(\tfrac{E_{\text{UAV}}}{10^3}\right), \quad (12)$$

$$r_{\text{rate}} = 2\,(C(t) - 2), \quad (13)$$

$$p_{\text{constraint}} = 50\cdot\mathbf{I}_{\text{altitude}} + 200\cdot\mathbf{I}_{\text{collision}} \quad (14)$$

where I_{altitude} and $I_{\text{collision}}$ are indicator functions for constraint violations. The survival bonus $r_{\text{survival}} = 0.1$ encourages longer trajectories. The term r_{rate} is included to provide continuous feedback on improvements in the end-to-end rate. If only the QoS reward is used, the agent receives the same penalty whenever $R_{\text{e2e}} < R_{\text{req}}$, regardless of whether its actions improve the link quality. As a result, the learning signal becomes sparse and less informative. The r_{rate} term addresses this issue by rewarding incremental rate improvements, guiding the UAV toward satisfying the QoS constraint more efficiently.

B. Learning Algorithms

Method 1: Proximal Policy Optimization (PPO)

1) Algorithm: PPO is an on-policy actor-critic algorithm [14] that collects rollouts with the current policy, performs n_{epoch} gradient updates, then discards the data. Its clipped surrogate objective is:

$$\mathcal{L}^{\text{CLIP}}(\theta) = \hat{\mathbb{E}}_t\Big[\min\Big(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1{-}\epsilon, 1{+}\epsilon)\hat{A}_t\Big)\Big], \tag{15}$$

where $r_t(\theta){=}\pi_\theta(a_t|s_t)/\pi_{\theta_{\text{old}}}(a_t|s_t)$ and $\epsilon{=}0.2$.

The full objective combines clipped policy loss, value loss, and entropy bonus:

$$\mathcal{L}(\theta, \phi) = \mathcal{L}^{\text{CLIP}} - c_v\mathcal{L}^{\text{VF}} + c_e\mathcal{L}^{\text{ENT}}. \tag{16}$$

2) Action Space and Architecture: PPO uses a continuous action space. At each time step the agent outputs a 3d real valued vector denoting the position of UAV at that moment. The actor outputs a categorical distribution via softmax; the critic produces a scalar. State dimension: 16. Both networks: Input(13)→FC(256)→ReLU→FC(256)→ReLU→Head. Rollout: 2048 steps, 10 epochs per rollout, batch size 64, $\alpha{=}3{\times}10^{-4}$.

Method 2: Soft Actor-Critic (SAC)

3) Algorithm: SAC is an off-policy maximum-entropy actor-critic algorithm [15]. It augments the return with an entropy term:

$$J(\pi) = \sum_t \mathbb{E}_{(s_t,a_t)\sim\pi}[r(s_t, a_t) + \alpha\,\mathcal{H}(\pi(\cdot|s_t))]\,. \tag{17}$$

Two Q-networks and target networks mitigate overestimation via the clipped double-Q trick. Critic loss (for $i \in \{1, 2\}$):

$$\mathcal{L}_Q(\phi_i) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\big[(Q_{\phi_i}(s, a) - y)^2\big]\,, \tag{18}$$

$$y = r + \gamma\Big(\min_i Q_{\phi'_i}(s', \tilde{a}') - \alpha\log\pi_\theta(\tilde{a}'|s')\Big)\,. \tag{19}$$

Target networks are updated via Polyak averaging: $\phi'_i \leftarrow \tau\phi_i + (1{-}\tau)\phi'_i$, $\tau{=}0.005$. The policy is trained to maximise entropy-regularized Q-values:

$$\mathcal{L}_\pi(\theta) = \mathbb{E}_{s\sim\mathcal{D},\tilde{a}\sim\pi_\theta}\Big[\alpha\log\pi_\theta(\tilde{a}|s) - \min_i Q_{\phi_i}(s, \tilde{a})\Big]\,. \tag{20}$$

The temperature α is tuned automatically via:

$$\mathcal{L}(\alpha) = \mathbb{E}_{\tilde{a}\sim\pi_\theta}\big[-\alpha(\log\pi_\theta(\tilde{a}|s) + \bar{\mathcal{H}})\big]\,, \tag{21}$$

with target entropy $\bar{\mathcal{H}}{=}-\dim(\mathcal{A}){=}-3$ [15].

4) Action Space and Architecture: SAC uses a *continuous* action space $\mathbf{a}{=}[\delta x, \delta y, \delta z]{\in}[-1, 1]^3$, scaled to $\pm 10\,\text{m}$ per axis. The policy outputs parameters $(\boldsymbol{\mu}, \log\boldsymbol{\sigma})$ of a diagonal Gaussian; actions are sampled via the reparameterization trick. Asymmetric architecture [400, 300] (state dim 16): Actor: Input(13)→FC(400)→FC(300)→(3×2 outputs). Critic: Input(16)→FC(400)→FC(300)→scalar.

Method 3: Recurrent PPO with LSTM

TABLE I
SYSTEM PARAMETERS

Parameter	Value
f_c, B	2400 MHz, 20 MHz
P, N_0	35 dBm (3.16 W), -94 dBm
R_{req}, σ	2000 kbps, 8 dB
K-factor, $P_{\text{GU}}^{\max}$	10 dB, 200 mW
(ω_0, ω_1), Δt	(0.3, 0.7), 1 s
T, Operational area	200 slots, 500×500 m^2
Altitude range, r_{obs}	25–125 m, 15 m
P_0, P_i	79.86 W, 88.63 W
v_{tip}, v_0	7.2 m/s, 4.03 m/s
d_0, ρ	0.6, 1.225 kg/m^3
s, A	0.05, 0.503 m^2

5) Motivation: The standard Markovian assumption is violated in our setting: instantaneous fading realizations, future GU movement, and current shadowing values are unobservable from a single state snapshot. Recurrent PPO (sb3-contrib) replaces the feedforward backbone with an LSTM [16] that maintains a hidden state $\mathbf{h}_t$ across time steps:

$$\mathbf{h}_t, \mathbf{c}_t = \text{LSTM}(s_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}), \tag{22}$$

$$\pi_\theta(a_t|s_{1:t}) \approx \pi_\theta(a_t|\mathbf{h}_t), \quad V_\phi(s_{1:t}) \approx V_\phi(\mathbf{h}_t). \tag{23}$$

6) LSTM-Augmented PPO: Training proceeds on sequences of length $n_{\text{steps}}{=}512$ using BPTT. The per-sequence objective is:

$$\mathcal{L}^{\text{seq}}_{\text{CLIP}}(\theta) = \frac{1}{T_{\text{seq}}}\sum_{t=1}^{T_{\text{seq}}}\min\Big(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1{-}\epsilon, 1{+}\epsilon)\hat{A}_t\Big)\,.$$

Both actor and critic carry independent LSTM modules (`enable_critic_lstm=True`), allowing the value function to condition on history for more accurate credit assignment.

7) Enhanced State Space: Recurrent PPO extends the base 13-feature state to 16 dimensions with three history-exploiting features:

$$\hat{R}(t) = \tanh(R_{\text{e2e}}(t)/10), \tag{24}$$

$$u_x(t) = (x_{\text{user}}(t) - x_{\text{user}}(t{-}1))/10, \tag{25}$$

$$u_y(t) = (y_{\text{user}}(t) - y_{\text{user}}(t{-}1))/10. \tag{26}$$

$\hat{R}(t)$ enables the LSTM to detect gradual channel deterioration (shadow zones); u_x, u_y allow the LSTM to learn predictive user-following behaviour. Here $x_{\text{user}}(t)$ and $y_{\text{user}}(t)$ denote the x- and y-coordinates of the user at time slot t, respectively.

IV. SIMULATION RESULTS AND DISCUSSIONS

All environments are custom OpenAI Gymnasium implementations that use the same physical models. The GU starts out at a random point on the ground plane and moves 10 m in a random cardinal direction every 20 steps. Every time an episode resets, shadowing channels are forced to start over. All experiments use Stable-Baselines3 (PPO, SAC) and sb3-contrib (Recurrent PPO) [17]. We set total timesteps

TABLE II
PERFORMANCE COMPARISON OF DRL ALGORITHMS (2.25M STEPS)

Metric	PPO	RecPPO	SAC
Ep. Reward (mean)	3453.4	3366.6	**3469.7**
Ep. Length (steps)	196.72	**197.64**	196.51
Avg. Data Rate ($\times 10^3$ kbps)	**132.7**	127.44	130.52
Avg. UAV Energy/step (J)	255.4	**227.7**	245.4
Total UAV Energy (kJ)	47.09	**44.72**	45.86
QoS Satisfaction (%)	**99.75**	99.69	99.08

(2.25M), approximate number of episodes (11,250 at 200 steps/episode), hardware (standard CPU training without GPU acceleration), approximate wall-clock time per algorithm, random seed used (seed=42 for all experiments), and the Stable-Baselines3 version (1.8.0). A custom `MetricsCallback` logs per-episode various metrics- average data rate, average UAV energy, QoS satisfaction and energy efficiency. Other than these, training metrics- average Reward per episode and average episode length are also logged using TensorBoard. Key system parameters are listed in Table I.

Fig. 1 show that all three algorithms—PPO, SAC, and Recurrent PPO—learn the task well and over time reach high reward values. SAC, on the other hand, converges the fastest. It quickly raises its reward during the first training phase and then levels off around a high value. PPO also works well, but it takes a little longer to converge than SAC. In the end, though, it gets a similar final reward. Recurrent PPO, on the other hand, learns much more slowly, with rewards that slowly increase and convergence that takes longer.

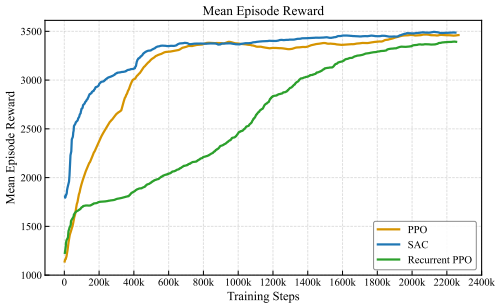

Fig. 1. Training convergence (episodic reward vs. timesteps) for PPO, SAC, and Recurrent PPO. Training convergence: episodic reward (smoothed with window=50) vs. training timesteps for PPO (blue), SAC (orange), and Recurrent PPO (green) over 2.25M steps. SAC achieves the fastest convergence, while Recurrent PPO shows gradual but sustained improvement.

Fig. 2 shows that the average data rate curves for all three algorithms—PPO, SAC, and Recurrent PPO—can get better over time. PPO gets the highest final data rate because it improves quickly in the beginning of training and then stays strong. SAC is very close to PPO, with slightly lower but stable data rate values and smooth convergence behavior. Recurrent PPO, on the other hand, starts off poorly, with a noticeable drop in data rate during the first few training sessions, and its improvement trend is much slower.

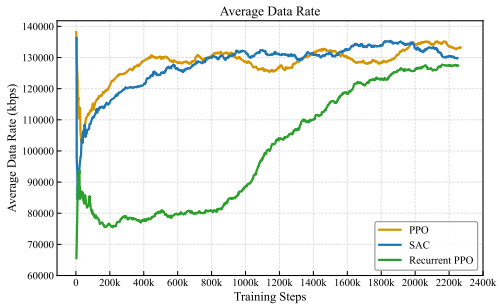

Fig. 2. Average data rate (kbps) vs. timesteps.

Fig. 3 shows the average amount of energy used by the UAV per episode. The decreasing nature of curves shows that the UAV successfully learns to optimize the energy and achieve a stable position quickly during training. All three algorithms learning to stabilize its position with minimum energy consumption but at different rates with SAC showing the fastest reduction and Recurrent PPO the slowest, but all three converging to the same constant energy.

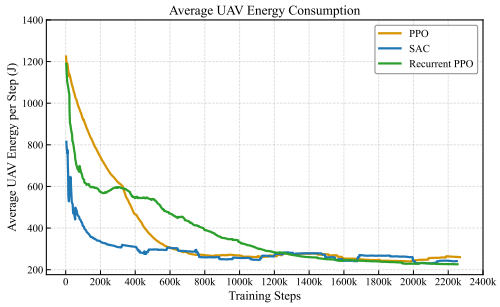

Fig. 3. Average UAV energy per time slot (J) vs. timesteps.

Energy efficiency is defined as the ratio of the average achievable data rate to the average weighted energy consumption of the UAV. Fig. 4 shows how energy-efficient the three proposed algorithms are over the entire training period. SAC converges the fastest of all the algorithms. In the early stages of training, it quickly reaches a high efficiency plateau because its off-policy experience replay mechanism lets it reuse past transitions efficiently. PPO shows steady and consistent progress throughout the entire training period. The way it moves suggests that the algorithm hasn't fully converged by the end of training and could benefit from more training. Recurrent PPO shows the slowest initial learning because LSTM architecture adds extra computational work. The energy efficiency remains low due to the considerable propulsion power during hovering and flight. Also the bottleneck constraint limits the achievable data rate.

Fig. 5 shows how the trained PPO agent moved during a typical evaluation episode. The top view (X-Y plane) shows that the agent successfully moves from its starting point to the moving ground user while avoiding all five cylindrical obstacles. This shows that it can plan a path that doesn't

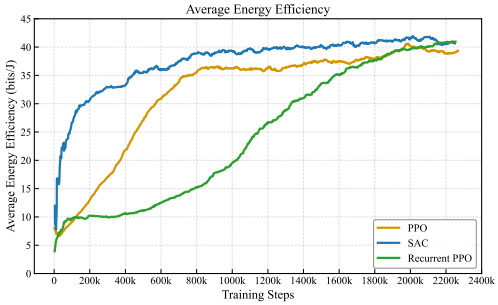

Fig. 4. Energy efficiency (kbps/J) vs. training timesteps.

hit anything in the 500 m × 500 m operational arena. The 3D trajectory plot shows that the UAV quickly rises to a height close to the maximum allowed height of 125 m (average height: 123.4 m) and stays at this height for the rest of the mission.

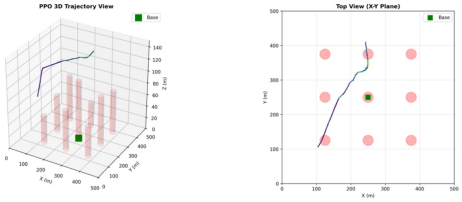

Fig. 5. Trajectory for PPO during a representative evaluation episode. (Left) Top-down (X-Y) view showing the path while avoiding all cylindrical obstacles (circles). (Right) 3D trajectory showing altitude profile; the UAV ascends to 125 m and maintains high altitude for LoS connectivity.

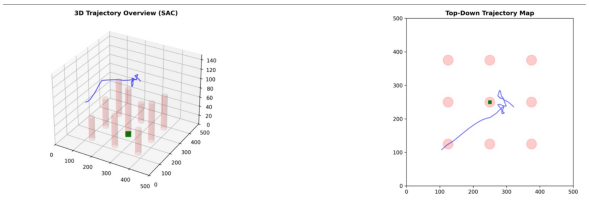

Fig. 6. UAV trajectory for SAC during a representative evaluation episode. SAC produces smoother, more direct paths consistent with its faster policy convergence and off-policy training

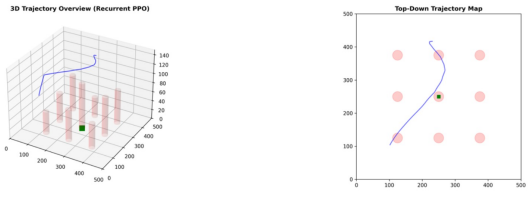

Fig. 7. UAV trajectory for Recurrent PPO during a representative evaluation episode. The LSTM memory enables anticipatory path adjustments; the trajectory shows tighter tracking of the moving GU compared to PPO and SAC, contributing to lower overall energy.

V. Conclusion

This paper proposes a deep reinforcement learning framework for energy-efficient UAV trajectory optimization under realistic channel conditions. The task is formulated as a continuous control problem to maintain reliable communication while minimizing movement. PPO, SAC, and Recurrent PPO are evaluated using reward, data rate, and energy metrics. All methods learn effective policies, but SAC achieves the fastest and most stable convergence. PPO offers competitive performance with moderate convergence, while Recurrent PPO converges slower due to its higher complexity but reaches similar final performance. However, provide minimum UAV energy consumption. This study is limited to simulation-based evaluation under 3GPP-compliant channel models and assumes perfect UAV state information without control latency. Future work will focus on hardware validation, transfer learning for faster adaptation, and multi-agent UAV relay optimization.

References

[1] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, pp. 134–142, 2020.

[2] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, 2016.

[3] J. Lyu, Y. Zeng, R. Zhang, and T. J. Lim, "Placement optimization of UAV-mounted mobile base stations," *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 604–607, Mar. 2017.

[4] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, Apr. 2019.

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

[6] A. A. Khuwaja, Y. Chen, N. Zhao, M. S. Alouini, and P. Dobbins, "A survey of channel modeling for UAV communications," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2804–2821, 2018.

[7] S. J. Seah et al., "Empirical channel models for UAV communication: A comparative study," *IEEE Access*, vol. 12, pp. 96740–96756, 2024.

[8] U. Challita, W. Saad, and C. Bettstetter, "Interference management for cellular-connected UAVs: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2125–2140, 2019.

[9] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, 2018.

[10] J. Hu et al., "Cooperative Internet of UAVs: Distributed trajectory design by multi-agent deep reinforcement learning," *IEEE Trans. Commun.*, vol. 68, no. 11, pp. 6807–6821, 2020.

[11] S. Iranmanesh et al., "An energy-efficient deep reinforcement learning framework for joint UAV trajectory optimization in logistics-assisted 5G/6G cellular networks," *IEEE Trans. Veh. Technol.*, 2026.

[12] ETSI, "Study on channel model for frequencies from 0.5 to 100 GHz (3GPP TR 38.901 version 15.0.0)," *ETSI TR 138 901 V15.0.0*, Jul. 2018.

[13] J. Zhang et al., "Energy consumption minimization for UAV-assisted network in hotspot area," *Drones*, vol. 9, no. 3, 2025.

[14] J. Schulman et al., "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.

[15] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. ICML*, 2018.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-Baselines3: Reliable reinforcement learning implementations," *J. Mach. Learn. Res.*, vol. 22, no. 268, pp. 1–8, 2021.