sensors

MDPI

Article

Deep Reinforcement Learning-Based Energy-Efficient Resource Allocation and Scheduling in 6G-Enabled UAV-Assisted IoT Wireless Networks

Ali Nauman * and Sung Won Kim

School of Computer Science and Engineering, College of Digital Convergence, Yeungnam University, Gyeongsan 38541, Republic of Korea
* Correspondence: anauman@ynu.ac.kr

Abstract

Unmanned Aerial Vehicles (UAVs) have emerged as a flexible, cost-effective solution for connecting Internet of Things (IoT) devices where traditional infrastructure falls short. However, managing their limited energy alongside the diverse demands of densely deployed devices makes resource allocation a genuinely hard problem. This paper presents a Deep Reinforcement Learning (DRL) framework that jointly optimizes user scheduling, IoT device transmit power, bandwidth, and UAV movement in a 6G-enabled UAV-relay uplink network, using a deterministic large-scale air-to-ground path-loss channel model. The UAV acts as an aerial decode-and-forward relay between IoT devices and a Base Station (BS), with a Deep Q-Network (DQN) making decisions based on queue backlogs, channel conditions, UAV position, and remaining battery. The reward function balances Energy Efficiency (EE), queue stability, fairness, and battery longevity. We benchmark the DQN against six baselines; Round Robin (RR), Random Allocation (RA), the Single-to-Noise Ratio (Max-SNR), Proportional Fair (PF), a Lyapunov heuristic, and a GreedyEE scheme; across a range of device counts, traffic loads, battery budgets, and flight altitudes. Simulations consistently show that the DQN outperforms all baselines, including a RA baseline with equal access to UAV mobility; in EE, throughput, delay, and fairness, confirming that the gain stems from the learned joint control policy rather than from UAV mobility being available.

Keywords: 6G; Internet of Things (IoT); Deep Q-Network (DQN); Deep Reinforcement Learning (DRL); Energy Efficiency (EE); Unmanned Aerial Vehicles (UAVs)

Academic Editor: Xiao Tang

Received: 27 June 2026
Revised: 22 August 2026
Accepted: 27 August 2026
Published: 29 August 2026

1. Introduction

The proliferation of Internet of Things (IoT) devices across industrial, agricultural, environmental, and urban monitoring domains is one of the defining technological trends of the 2020s. According to industry projections, the number of globally connected IoT endpoints is expected to exceed 39 billion by 2030, driving an unprecedented demand for low-latency, reliable, and energy-efficient wireless connectivity [1–3]. In many practical deployment scenarios; remote terrain monitoring, post-disaster search and rescue, smart agriculture, and offshore industrial sensing; IoT devices operate far from terrestrial Base Station (BS), and direct communication is impeded by large path loss, shadowing, or complete Line-of-Sight (LoS) obstruction due to terrain features [4,5].

Unmanned Aerial Vehicle (UAV)s offer an attractive solution to bridge this connectivity gap. Their ability to be rapidly deployed, repositioned, and recovered, combined with the

inherently favorable air-to-ground channel conditions arising from their elevated altitude, makes UAVs compelling as aerial relays or mobile data collectors in IoT networks [6]. The emergence of 6G wireless technology further amplifies this potential: 6G promises to integrate Non-Terrestrial Network (NTN); including UAVs High-Altitude Platform Station (HAPS)s, and Low Earth Orbit (LEO) satellites; as native components of the communication infrastructure [7,8].

However, the effective deployment of UAVs as IoT relays is constrained by two fundamental challenges. First, UAVs operate on limited onboard battery capacity, which must sustain not only the energy consumed for data relay transmission but also the considerable propulsion and hovering energy required to remain airborne [9]. Second, the dynamic nature of the network; arising from time-varying wireless channels, stochastic IoT traffic arrivals, evolving queue states, and changing UAV position; makes it exceedingly difficult to derive a static optimal policy using conventional optimization methods [10].

The optimization of scheduling (which IoT device transmits), resource allocation (transmit power and bandwidth), and UAV trajectory in a battery-constrained decode-and-forward relay architecture constitutes a Mixed-Integer Non-Linear Programming (MINLP) problem with stochastic dynamics. Such problems are generally NP-hard even in deterministic settings [11,12], and the added stochasticity of wireless channels and traffic arrivals precludes the use of deterministic dynamic programming. This motivates the investigation of intelligent data-driven control, and specifically Deep Reinforcement Learning (DRL), as a means of learning adaptive policies that implicitly account for all these interdependencies [13,14].

1.1. Research Gap and Contributions

While a growing body of literature has investigated DRL for UAV-related optimization tasks (surveyed in Section 2), several research gaps remain.

1. Most prior work optimizes trajectory or resource allocation, but does not simultaneously determine IoT user scheduling, IoT device uplink transmit power levels, bandwidth fractions, and UAV movement direction in a unified DRL framework.
2. Battery sustainability as a first-class constraint is rarely modeled with explicit reward penalties and evaluated across varying battery budgets.
3. Fairness across heterogeneous IoT devices is often neglected or treated as a secondary metric rather than a reward component.
4. Comprehensive benchmarking against multiple heuristic baselines; including Lyapunov-inspired and Greedy Energy Efficiency (EE) schemes; under a range of network conditions is rarely provided.

This paper addresses all four gaps with the following contributions:

1. **Unified DRL-based control framework:** We propose a Deep Q-Network (DQN)-based controller that simultaneously determines IoT user scheduling, IoT device uplink transmit power levels, bandwidth fractions, and UAV movement direction at each time slot. Unlike decomposed approaches, the joint action representation allows the agent to learn cross-domain trade-offs.
2. **Battery-aware and fairness-enhanced reward design:** The reward function explicitly penalizes battery depletion events and incorporates a Jain fairness index term to prevent starvation of devices with persistently poor channel conditions.
3. **Scalable state representation:** The state vector is defined with a fixed maximum dimension regardless of the actual number of active IoT devices, enabling the trained policy to generalize across varying network sizes without retraining.
4. **Thorough multi-dimensional evaluation:** We benchmark the proposed DQN method against six alternative policies under four independent experiment axes: number

of IoT devices, traffic arrival rate, UAV battery budget, and UAV altitude. This multi-axis evaluation provides a more complete picture of policy robustness than is typically reported.

1.2. Paper Organization

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the system model, including the network architecture, channel model, communication rates, queue dynamics, UAV mobility, and energy consumption model. Section 4 formally states the optimization problem and discusses the challenges motivating DRL. Section 5 presents the proposed DQN framework in detail, including Markov Decision Process (MDP) formulation, state and action space design, reward engineering, and training procedure. Section 6 provides the simulation setup. The results and discussion are presented in Section 7, Section 8, provides future research directions, and Section 9 concludes the paper.

2. Related Work

2.1. UAV-Assisted Wireless Communication

The use of UAVs as communication infrastructure has attracted substantial research attention. In [6], the authors provided one of the earliest systematic overviews of UAV-enabled wireless communications, identifying trajectory optimization and energy management as central challenges. In [15], the authors studied joint trajectory and communication scheduling for UAV-enabled mobile relay and derived efficient successive convex approximation-based solutions. In [16], the authors presented a comprehensive tutorial on UAV-assisted communications, covering channel modeling, deployment optimization, and interference management. These works, while foundational, rely on deterministic convex optimization frameworks that require full knowledge of future channel states and traffic arrivals, limiting their applicability in dynamic and partially observable IoT scenarios.

2.2. EE in UAV Networks

EE in UAV communications has been studied from multiple angles. In [9], the authors analyzed the fundamental energy-throughput trade-off for UAV-enabled data collection and proposed optimal flight trajectories. In [17], the authors investigated joint power and trajectory optimization for UAV BS with EE as the primary objective. In [18], the authors examined energy-efficient resource allocation for multi-UAV networks using dynamic programming. A common limitation of these optimization approaches is the reliance on static or slowly-varying channel conditions and predetermined traffic models, which do not capture the fast-changing and stochastic nature of IoT sensor data.

2.3. DRL for UAV Control

The application of DRL to UAV communication has grown rapidly in recent years. In [19], the authors applied DQN to optimize UAV trajectory for data collection from multiple ground users. In [20], the authors proposed a multi-agent DRL approach for multi-UAV path planning and resource allocation. In [21], the authors developed a multi-agent actor-critic framework for joint computation offloading and UAV trajectory control in mobile edge computing scenarios. In [22], the authors provided a survey of DRL applications in communication networks, highlighting its suitability for stochastic resource management.

2.4. IoT Data Collection and Scheduling via UAVs

Several works have specifically targeted UAV-assisted IoT data collection. In [23], the authors proposed an Age-of-Information (AoI)-minimizing UAV scheduling policy for

IoT sensor networks. In [24], the authors investigated energy-constrained UAV trajectory planning for IoT data gathering. In [25], the authors addressed multi-UAV cooperative data collection with fairness constraints. In [26], the authors applied DRL to minimize AoI in UAV-assisted vehicular IoT. Moreover, in [27], the authors proposed a CGAN-driven signal sample expansion to showcase that generative network-based data augmentation can effectively classify and mitigate jamming signals, making it useful for UAV-based communication and anti-interference performance. These works, however, either consider a simplified single-hop collection model or do not jointly optimize the full resource management tuple of scheduling, power, bandwidth, and trajectory.

2.5. Positioning of This Work

The present paper distinguishes itself from prior art in three key aspects. First, we consider a two-hop decode-and-forward relay architecture rather than a direct collection model, which introduces a bottleneck-limited rate and additional energy consumption at the UAV relay stage. Second, we propose a unified DQN framework that jointly and simultaneously optimizes all four control dimensions; scheduling, power of IoT devices, bandwidth, and trajectory; in a single agent, with a reward function that captures EE, queue stability, battery sustainability, and fairness. Third, we conduct a comprehensive evaluation under four independent variational axes and compare against six benchmark policies, providing a rigorous and balanced assessment of the proposed approach.

Moreover, it is worth clarifying the scope of the comparative evaluation in this paper: the six baselines (Round-Robin (RR), Random, Max Single-to-Noise Ratio (Max-SNR), Proportional Fair (PF), a Lyapunov heuristic, and GreedyEE) are classical and heuristic scheduling/resource-allocation policies representative of the UAV-IoT literature surveyed above, rather than alternative DRL architectures.

3. System Model

3.1. Network Architecture and Notation

We consider a UAV-assisted wireless IoT uplink network comprising K stationary ground IoT devices, one terrestrial BS, and a single UAV acting as an aerial relay, as illustrated in Figure 1. The direct communication link between the IoT devices and the BS is assumed to be blocked, severely attenuated, or otherwise unavailable due to path loss, terrain obstruction, or coverage constraints. Accordingly, all uplink data traffic is routed via the UAV relay in a two-hop architecture.

Let $\mathcal{K} = \{1, 2, \ldots, K\}$ denote the set of IoT devices. The BS is located at a fixed two-dimensional position:

$$\mathbf{w}_b = [x_b, y_b]^\mathsf{T} \in \mathbb{R}^2, \tag{1}$$

and the k-th IoT device is located at:

$$\mathbf{w}_k = [x_k, y_k]^\mathsf{T}, \quad \forall k \in \mathcal{K}. \tag{2}$$

The system operates over a finite horizon of T time slots indexed by $\mathcal{T} = \{1, 2, \ldots, T\}$, each of duration Δt seconds. The UAV flies at a fixed altitude H and its horizontal position at time slot t is denoted by:

$$\mathbf{q}(t) = [x_u(t), y_u(t)]^\mathsf{T} \in \mathbb{R}^2. \tag{3}$$

The three-dimensional position is, therefore, $(\mathbf{q}(t), H)$. At most one IoT device is scheduled for uplink transmission per time slot. The scheduled device transmits data to the UAV (access link), which then decodes and forwards the data to the BS (backhaul link) using a decode-and-forward protocol.

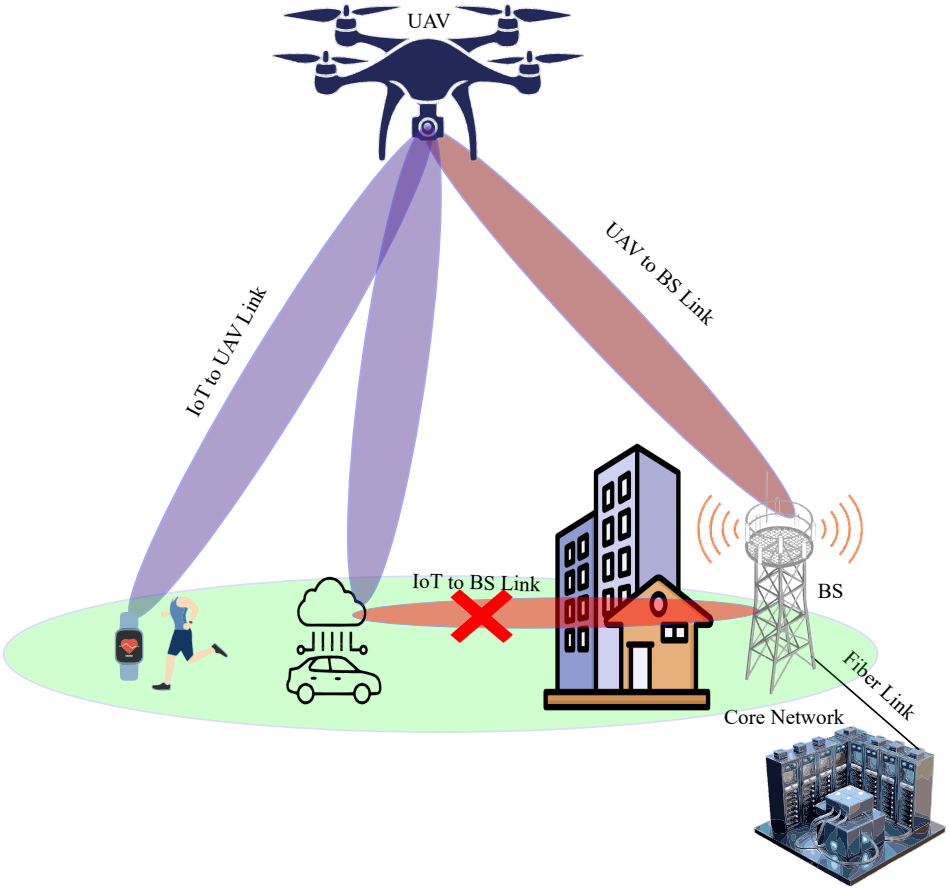

Figure 1. System model illustration.

3.2. Distance Model

The three-dimensional Euclidean distance between IoT device k and the UAV at slot t is:

$$d_{k,u}(t) = \sqrt{\|\mathbf{q}(t) - \mathbf{w}_k\|^2 + H^2}, \tag{4}$$

and the distance between the UAV and the BS is:

$$d_{u,b}(t) = \sqrt{\|\mathbf{q}(t) - \mathbf{w}_b\|^2 + H^2}. \tag{5}$$

3.3. Channel Model

The elevated position of the UAV gives rise to dominant LoS propagation on both the IoT-to-UAV (access) link and the UAV-to-BS (backhaul) link. We adopt a deterministic large-scale path-loss channel model that is widely used for communication-oriented UAV studies [28–30].

3.3.1. IoT-to-UAV Access Link

The channel power gain between IoT device k and the UAV at time slot t is:

$$g_{k,u}(t) = \frac{\beta_0}{(d_{k,u}(t))^{\alpha_{k,u}}}, \tag{6}$$

where β_0 is the channel power gain at a reference distance of 1 m, and $\alpha_{k,u}$ is the path-loss exponent for the access link.

3.3.2. UAV-to-BS Backhaul Link

Similarly, the channel power gain between the UAV and the BS is:

$$g_{u,b}(t) = \frac{\beta_0}{\left(d_{u,b}(t)\right)^{\alpha_{u,b}}}, \tag{7}$$

where $\alpha_{u,b}$ is the path-loss exponent of the backhaul link.

3.4. Scheduling and Resource Allocation

At each time slot t, the controller determines: (i) which IoT device is scheduled for uplink transmission, (ii) the transmit power assigned to the scheduled device, (iii) the bandwidth allocated to the scheduled transmission, and (iv) the UAV movement direction.

Let the binary scheduling variable be:

$$a_k(t) = \begin{cases} 1, & \text{if IoT device } k \text{ is scheduled at slot } t, \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

subject to the single-user scheduling constraint:

$$\sum_{k=1}^{K} a_k(t) \leq 1, \quad \forall t \in \mathcal{T}. \tag{9}$$

The uplink transmit power of device k at slot t is denoted by $p_k(t)$, subject to:

$$0 \leq p_k(t) \leq Pmax_k, \quad \forall k \in \mathcal{K}, \forall t \in \mathcal{T}. \tag{10}$$

The allocated access-link bandwidth to device k at slot t is $B_k(t)$, drawn from the shared IoT-access spectrum pool B_{tot} (the UAV-BS backhaul link uses a separate dedicated band, and is therefore not subject to this pool):

$$0 \leq B_k(t) \leq B_{tot}, \quad \sum_{k=1}^{K} B_k(t) \leq B_{tot}, \quad \forall t. \tag{11}$$

Because at most one device is scheduled per slot, $B_k(t) = 0$ for all unscheduled devices.

3.5. Achievable Transmission Rate and End-to-End Throughput

For the scheduled IoT device k at time slot t, the instantaneous achievable rate on the access link is given by Shannon's capacity formula:

$$R_{k,u}(t) = B_k(t) \log_2\left(1 + \frac{p_k(t) g_{k,u}(t)}{N_0 B_k(t)}\right), \tag{12}$$

where N_0 is the noise power spectral density. Similarly, the achievable rate on the backhaul link is:

$$R_{u,b}(t) = B_k(t) \log_2\left(1 + \frac{p_u(t) g_{u,b}(t)}{N_0 B_k(t)}\right), \tag{13}$$

where $p_u(t)$ is the UAV relay transmit power. Under the decode-and-forward protocol, the end-to-end uplink rate for device k is bottlenecked by the weaker hop:

$$R_k(t) = a_k(t) \min\left\{R_{k,u}(t),\, R_{u,b}(t)\right\}. \tag{14}$$

The access link (IoT-to-UAV) and the backhaul link (UAV-to-BS) are served by two physically separate, dedicated frequency bands: the access link uses bandwidth $B_k(t)$ drawn

from the shared IoT-access spectrum pool B_{tot} and allocated among IoT devices, while the backhaul link operates over a dedicated point-to-point UAV-BS band of fixed width B_{bh} (e.g., a microwave/mmWave backhaul link) that is provisioned independently of B_{tot} and does not compete with any IoT device access-link allocation; analogous to the dedicated backhaul/fiber connection from the BS to the core network shown in Figure 1. Equation (14) is accordingly rewritten with this dedicated parameter:

$$R_{u,b}(t) = B_{bh} \log_2\left(1 + \frac{p_u(t) g_{u,b}(t)}{N_0 B_{bh}}\right). \tag{15}$$

In this study, we set B_{bh} to be numerically equal to the tabulated access-link bandwidth values used for $B_k(t)$. Because the two hops occupy disjoint dedicated bands rather than time-sharing a single band, they may operate concurrently, and correctly captures the end-to-end throughput under this dedicated-backhaul-band architecture, with no additional time-sharing factor required. Relay-side buffering delay is treated as negligible relative to the slot duration, so no additional queuing delay is modeled at the UAV itself.
The total data successfully delivered from device k to the BS during slot t is:

$$D_k(t) = R_k(t) \cdot \Delta t. \tag{16}$$

3.6. Traffic Arrival and Queue Dynamics

Each IoT device maintains a local transmission queue. Let $Q_k(t)$ denote the queue backlog (in bits) of device k at the beginning of slot t, and $A_k(t)$ the amount of data arriving at device k during slot t. The queue evolves as:

$$Q_k(t+1) = \min\{\max\{Q_k(t) - D_k(t), 0\} + A_k(t), Qmax_k\}, \tag{17}$$

where $Qmax_k$ is the maximum buffer size of device k.

Thearrival process $A_k(t)$ is modeled as independent and identically distributed Poisson random variables with mean λ_k bits per slot:

$$A_k(t) \sim \text{Poisson}(\lambda_k), \quad EE[A_k(t)] = \lambda_k. \tag{18}$$

This models bursty IoT sensor traffic (e.g., periodic environmental measurements with random packet sizes) in a tractable manner.

3.7. UAV Mobility Model

The UAV moves horizontally at a fixed altitude, H, within a bounded service area, $\mathcal{A} = [0, L_x] \times [0, L_y]$. The mobility constraint limits the maximum displacement per slot:

$$\|\mathbf{q}(t+1) - \mathbf{q}(t)\| \leq Vmax \cdot \Delta t, \quad \forall t \in \mathcal{T}, \tag{19}$$

where $Vmax$ is the maximum horizontal speed. For the DRL implementation, the continuous movement space is discretized into five actions: $\mathcal{M} = \{\text{hover}, \text{north}, \text{south}, \text{east}, \text{west}\}$, corresponding to a fixed step size of δ_{xy} meters per slot. The service area boundary is enforced by clipping the UAV position:

$$\mathbf{q}(t) \in \mathcal{A}, \quad \forall t \in \mathcal{T}. \tag{20}$$

3.7.1. IoT Device Transmission Energy

When device k is scheduled at slot t, its uplink transmission energy is:

$$E_k^{\text{tx}}(t) = a_k(t)\, p_k(t)\, \Delta t. \tag{21}$$

3.7.2. UAV Relay Transmission Energy

The UAV forwards decoded data to the BS using a fixed relay transmit power p_u, consuming:

$$E_u^{\text{relay}}(t) = p_u \, \Delta t. \tag{22}$$

3.7.3. UAV Propulsion and Hovering Energy

UAV propulsion energy accounts for both the static hovering cost and the additional cost incurred when the UAV moves. We adopt the simplified discrete-time model:

$$E_u^{\text{mob}}(t) = E_{\text{hov}} + \kappa \cdot \|\mathbf{q}(t+1) - \mathbf{q}(t)\|, \tag{23}$$

where E_{hov} is the hovering energy per slot (in joules), and κ is a propulsion energy coefficient (J/m).

3.7.4. Circuit Energy

The communication circuitry of the UAV and scheduled IoT device consumes a constant circuit power P_c per slot:

$$E^{\text{cir}}(t) = P_c \, \Delta t. \tag{24}$$

3.7.5. Total System Energy Consumption

Combining all components, the total system energy consumed in slot t is:

$$E_{\text{tot}}(t) = \sum_{k=1}^{K} E_k^{\text{tx}}(t) + E_u^{\text{relay}}(t) + E_u^{\text{mob}}(t) + E^{\text{cir}}(t). \tag{25}$$

3.8. UAV Battery Model

The UAV carries a finite onboard battery with initial energy $E_u^{\max}$. The residual battery energy evolves as:

$$Erem_u(t+1) = Erem_u(t) - E_u^{\text{relay}}(t) - E_u^{\text{mob}}(t) - E^{\text{cir}}(t), \tag{26}$$

with the constraint:

$$Erem_u(t) \geq 0, \quad \forall t \in \mathcal{T}, \tag{27}$$

and initial condition $Erem_u(1) = E_u^{\max}$. Battery depletion ($Erem_u(t) = 0$) terminates the operation episode in our simulation, modeling the physical constraint that a UAV must return to base for recharging.

3.9. EE Metric

The primary performance metric is the long-term average EE, defined as the ratio of the total data delivered to the BS over the total energy consumed:

$$\eta_{\text{avg}} = \frac{\sum_{t=1}^{T} \sum_{k=1}^{K} D_k(t)}{\sum_{t=1}^{T} E_{\text{tot}}(t)}, \tag{28}$$

measured in bits/joule (bits/J). This metric captures the fundamental trade-off between data delivery and energy expenditure over the full mission horizon.

3.10. Queue Stability and Delay Considerations

In addition to EE, the system must maintain acceptable end-to-end delay. By Little's law, the average delay experienced by packets of device k is proportional to its mean queue length $\bar{Q}_k$. Accordingly, a queue is said to be strongly stable if:

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} EE[Q_k(t)] < \infty, \quad \forall k \in \mathcal{K}. \tag{29}$$

The control policy should therefore prevent unbounded queue accumulation while maximizing EE.

3.11. Fairness Metric

To assess equitable service distribution among IoT devices, we use Jain's fairness index:

$$\mathcal{J} = \frac{\left(\sum_{k=1}^{K} \bar{D}_k\right)^2}{K \sum_{k=1}^{K} \bar{D}_k^2}, \tag{30}$$

where $\bar{D}_k = \sum_{t=1}^{T} D_k(t)$ is the cumulative data served for device k over the mission horizon. The index satisfies $\mathcal{J} \in (0, 1]$, with $\mathcal{J} = 1$ indicating perfect fairness.

4. Problem Formulation

4.1. Design Objective

Let the total data delivered to the BS at slot t be $D_{\text{sum}}(t) = \sum_{k=1}^{K} D_k(t)$, and let the corresponding total energy consumption be $E_{\text{tot}}(t)$ as defined in (25). The goal is to find a control policy $\Pi = \{u(1), u(2), \ldots, u(T)\}$ over the slot-wise action $u(t) = \left(\{a_k(t)\}, \{p_k(t)\}, \{B_k(t)\}, \mathbf{b}q(t+1)\right)$ that maximizes the long-term average EE (28).

4.2. Optimization Problem

The joint energy-efficient resource allocation problem is formally stated as:

$$\mathbf{P1}: \quad \max_{\Pi} \eta_{\text{avg}} = \frac{\sum_{t=1}^{T} D_{\text{sum}}(t)}{\sum_{t=1}^{T} E_{\text{tot}}(t)} \tag{31a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} a_k(t) \le 1, \quad \forall t, \tag{31b}$$

$$a_k(t) \in \{0, 1\}, \quad \forall k, t, \tag{31c}$$

$$0 \le p_k(t) \le Pmax_k, \quad \forall k, t, \tag{31d}$$

$$0 \le B_k(t) \le B_{tot}, \quad \forall k, t, \tag{31e}$$

$$\sum_{k=1}^{K} B_k(t) \le B_{tot}, \quad \forall t, \tag{31f}$$

$$Q_k(t+1) = \min\{\max\{Q_k(t) - D_k(t), 0\} + A_k(t), Qmax_k\}, \forall k, t, \tag{31g}$$

$$0 \le Q_k(t) \le Qmax_k, \quad \forall k, t, \tag{31h}$$

$$\|\mathbf{q}(t+1) - \mathbf{q}(t)\| \le Vmax\Delta t, \quad \forall t, \tag{31i}$$

$$\mathbf{q}(t) \in \mathcal{A}, \quad \forall t, \tag{31j}$$

$$Erem_u(t+1) = Erem_u(t) - E_u^{\text{relay}}(t) - E_u^{\text{mob}}(t) - E_u^{\text{cir}}(t), \forall t, \tag{31k}$$

$$Erem_u(t) \ge 0, \quad \forall t. \tag{31l}$$

Constraint (31b) enforces single-user scheduling per slot, while (31c) defines the binary scheduling nature. Constraints (31d)–(31f) govern transmit power and bandwidth.

Constraints (31g)–(31h) define queue evolution and buffer limits. Constraints (31i)–(31j) enforce UAV mobility feasibility. Finally, (31k)–(31l) capture UAV battery dynamics.

*4.3. Challenges of Solving **P1***

Problem **P1** is fundamentally challenging due to four interrelated properties:

4.3.1. Non-Convex Fractional Objective

The objective in (31a) is a ratio of two coupled sums, both of which depend nonlinearly on the continuous variables $p_k(t)$, $B_k(t)$, and $\mathbf{q}(t)$. Even without the integer constraints, such fractional objectives are generally non-convex. Dinkelbach's method can transform the fractional objective into a parameterized concave-minus-convex form, but the coupling across time slots and the presence of binary variables preclude direct application.

4.3.2. MINLP

The scheduling variables $a_k(t) \in \{0, 1\}$ make **P1** a MINLP problem. MINLP is generally NP-hard, and the combinatorial explosion of the scheduling space (2^K per slot) makes exact branch-and-bound methods computationally intractable for real-time control with $K \geq 6$.

4.3.3. Temporal Coupling

The control decisions are strongly coupled across time through three state variables: queue backlogs (31g), UAV battery (31k), and UAV position (31i). An action taken at slot t alters the system state and thus the reward achievable at all future slots. This temporal dependency renders **P1** a multi-period stochastic program that cannot be decomposed into independent single-slot subproblems without approximation.

4.3.4. Stochastic Traffic and Channels

The Poisson traffic arrivals $A_k(t)$ and the time-varying channel gains induced by UAV movement introduce stochasticity that makes deterministic dynamic programming intractable. Stochastic dynamic programming via Bellman's equation requires enumeration of the continuous state space, which is computationally infeasible for the considered network dimensions.

4.4. Motivation for DRL

The above challenges—non-convexity, mixed-integer structure, temporal coupling, and stochastic dynamics—collectively motivate the use of DRL as a solution methodology. DRL avoids the need for an explicit model of state transitions and channel statistics; instead, the agent learns an effective control policy purely from observed transitions. By approximating the optimal action-value function with a deep neural network, the agent can generalize across the continuous and high-dimensional state space. Critically, the temporal coupling across queue, battery, and trajectory states is naturally handled through the discounted cumulative reward formulation, enabling the agent to learn long-horizon trade-offs that myopic heuristics cannot capture. The discounted per-slot reward is adopted here as a tractable training signal; the resulting policy quality with respect to the mission-level objective η_{avg} is assessed directly in results using the exact metric defined, independent of the surrogate reward used during training.

5. Proposed DRL Framework

5.1. MDP Formulation

We model the UAV-assisted IoT uplink control problem as an MDP characterized by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ is the

state transition probability, $\mathcal{R}$ is the reward function, and $\gamma \in (0, 1]$ is the discount factor. The objective is to learn a policy, $\pi : \mathcal{S} \to \mathcal{A}$, that maximizes the expected discounted cumulative reward:

$$\max_{\pi} \; EE_{\pi}\left[\sum_{t=1}^{T} \gamma^{t-1} r(t)\right]. \tag{32}$$

5.2. State Space Design

The state at time slot t is defined as:

$$\mathbf{b}s(t) = \left[\mathbf{Q}(t),\, \mathbf{G}_u(t),\, g_{u,b}(t),\, \mathbf{q}(t),\, Erem_u(t)\right], \tag{33}$$

where:

- $\mathbf{Q}(t) = [Q_1(t), \ldots, Q_K(t)]$: IoT queue backlogs;
- $\mathbf{G}_u(t) = [g_{1,u}(t), \ldots, g_{K,u}(t)]$: IoT-to-UAV channel gains;
- $g_{u,b}(t)$: UAV-to-BS channel gain;
- $\mathbf{q}(t) = [x_u(t), y_u(t)]^{\mathsf{T}}$: UAV horizontal position;
- $Erem_u(t)$: remaining UAV battery.

To ensure a fixed state dimension that accommodates varying numbers of active IoT devices without retraining, the state vector is defined over $K_{\max}$ slots, yielding a dimension of $N_S = 2K_{\max} + 4$. All state variables are normalized before being fed into the neural network:

$$\hat{Q}_k(t) = \frac{Q_k(t)}{Qmax_k}, \quad \hat{E}_u(t) = \frac{Erem_u(t)}{E_u^{\max}}, \quad \hat{g}_{k,u}(t) = \frac{g_{k,u}(t)}{\beta_0}, \quad \hat{x}_u(t) = \frac{x_u(t)}{L_x}, \quad \hat{y}_u(t) = \frac{y_u(t)}{L_y}. \tag{34}$$

An additional binary active-device ratio $K_{\text{active}}/K_{\max}$ is appended to the state to allow the agent to adapt its behavior to the actual number of devices present.

5.3. Action Space Design

The action at time slot t jointly determines the scheduled IoT device, the uplink transmit power level, the bandwidth fraction, and the UAV movement direction:

$$a(t) = (k^*(t),\, p^{\text{idx}}(t),\, B^{\text{idx}}(t),\, m(t)), \tag{35}$$

where:

- $k^*(t) \in \mathcal{K}$: selected IoT device;
- $p^{\text{idx}}(t) \in \mathcal{P} = \{P^{(1)}, P^{(2)}, P^{(3)}, P^{(4)}\}$: transmit power level;
- $B^{\text{idx}}(t) \in \mathcal{B} = \{B^{(1)}, B^{(2)}, B^{(3)}, B^{(4)}\}$: bandwidth fraction;
- $m(t) \in \mathcal{M} = \{\text{hover}, \text{N}, \text{S}, \text{E}, \text{W}\}$: UAV movement direction.

The total number of discrete actions is:

$$|\mathcal{A}| = K_{\max} \times N_P \times N_B \times N_M, \tag{36}$$

where $N_P = |\mathcal{P}|$, $N_B = |\mathcal{B}|$, and $N_M = |\mathcal{M}|$. In our implementation, $N_P = N_B = 4$ and $N_M = 5$, giving $|\mathcal{A}| = 10 \times 4 \times 4 \times 5 = 800$ actions for $K_{\max} = 10$.

The discrete power levels and bandwidth fractions used in the simulation are:

$$\mathcal{P} = \{0.05, 0.10, 0.20, 0.35\}\ \text{W}, \quad \mathcal{B} = \{0.20, 0.35, 0.50, 0.70\} \times B_{\text{tot}}. \tag{37}$$

This discretization provides a meaningful coverage of the feasible resource space while keeping the action space manageable for DQN convergence. It is noted that only the IoT device's transmit power is included in the learned action space.

5.4. Reward Function Design

The reward function must translate the multi-objective optimization goal—EE, queue stability, battery sustainability, fairness—into a scalar signal that guides the DQN towards the desired behavior. We define the per-slot reward as:

$$r(t) = \lambda_1 \eta(t) - \lambda_2 \sum_{k=1}^{K} Q_k(t) - \lambda_3 \Phi(Erem_u(t)) + \lambda_4 \mathcal{F}(t) - \lambda_5 \mathcal{I}(t), \tag{38}$$

where:

- $\eta(t) = D_{\text{sum}}(t)/E_{\text{tot}}(t)$ is the instantaneous EE (bits/J);
- $\lambda_1, \ldots, \lambda_5 \geq 0$ are design hyperparameters;
- $\Phi(Erem_u(t))$ is a battery depletion penalty;
- $\mathcal{F}(t)$ is a fairness term;
- $\mathcal{I}(t)$ is an invalid-action penalty.

We emphasize that the per-slot reward is used exclusively as a training-time shaping signal that guides the DQN gradient updates; it is not itself the quantity used to evaluate policy quality. In particular, its EE component is a per-slot ratio, and maximizing $\sum_t \gamma^{t-1} r(t)$ is not, in general, equivalent to maximizing the mission-level ratio-of-sums objective. The DQN is, therefore, trained with a convenient per-slot surrogate but is scored, exactly like every baseline, on the true mission-level objective η_{avg}, consistent with standard reward-shaping practice in Reinforcement Learning (RL).

5.4.1. Battery Penalty

A soft penalty proportional to battery under-threshold violation is used:

$$\Phi(Erem_u(t)) = \max\{0,\ E_{\text{th}} - Erem_u(t)\}, \tag{39}$$

where E_{th} is a safety threshold. When the battery falls below E_{th}, the penalty grows linearly, discouraging energy-wasteful actions near the end of the mission. A hard terminal penalty of -5 is applied when $Erem_u(t) \leq 0$, terminating the episode.

5.4.2. Fairness Term

To promote equitable service across IoT devices, the per-slot fairness component uses a running Jain's index:

$$\mathcal{F}(t) = \frac{\left(\sum_{k=1}^{K} \bar{D}_k(t)\right)^2}{K \sum_{k=1}^{K} \bar{D}_k(t)^2 + \epsilon}, \tag{40}$$

where $\bar{D}_k(t) = \sum_{\tau=1}^{t} D_k(\tau) + 10^{-9}$ is the cumulative data served for device k up to slot t, and $\epsilon > 0$ prevents division by zero. It is noted that (40) is a ratio of a squared sum to K times a sum of squares, it is invariant to uniform scaling of the cumulative throughputs and if scaled by the same constant, it is unchanged. Consequently, $\mathcal{F}(t)$ remains bounded in $(0, 1]$ throughout the training horizon and does not introduce reward-scale drift; what does change as t grows is only its sensitivity to short-term imbalances, since early-slot outcomes are increasingly diluted by accumulated history.

5.4.3. Invalid-Action Penalty

If the action schedules an inactive or empty-queue device, an additional penalty is applied:

$$\mathcal{I}(t) = \mathbf{1}[\text{action is infeasible at slot } t]. \tag{41}$$

This discourages the agent from wasting slot opportunities on devices that have no data to transmit. It is noted that a soft invalid-action penalty is used rather than hard action masking during training because it keeps the network's fixed dimensional output head unchanged across the multi-environment training procedure with varying K_{active}, avoiding the need to reconstruct a per-episode action mask for each sampled K_{active}.

5.4.4. Reward Coefficient Selection

The reward weighting coefficients in (38) are set as $\lambda_1 = 1.0$, $\lambda_2 = 4 \times 10^{-6}$, $\lambda_3 = 0.01$, $\lambda_4 = 0.10$, and $\lambda_5 = 2.0$. These values are chosen so that the queue penalty term has a magnitude comparable to the EE term at typical queue occupancies, while the fairness and battery terms serve as regularizers. The reward weights were selected via a coarse offline grid search during development to balance the relative magnitudes of the EE, queue-penalty, battery-penalty, and fairness terms at typical operating points.

5.5. DQN Architecture

Because the state space is high-dimensional and continuous, tabular Q-learning is infeasible. Instead, we use a deep neural network parameterized by θ to approximate the action-value function:

$$Q(\mathbf{s}, a; \theta) \approx Q^*(\mathbf{s}, a), \tag{42}$$

where $Q^*(\mathbf{s}, a)$ is the optimal action-value function satisfying the Bellman optimality equation:

$$Q^*(\mathbf{s}, a) = EE\Big[r(t) + \gamma \max_{a' \in \mathcal{A}} Q^*(\mathbf{s}', a') \,\Big|\, \mathbf{s}(t) = \mathbf{s}s, a(t) = a\Big]. \tag{43}$$

The DQN architecture consists of:

- **Input layer:** receives the normalized state vector $\hat{\mathbf{s}}(t) \in \mathbb{R}^{N_S}$;
- **Hidden layer 1:** ReLU activation, fully connected, 128 neurons;
- **Hidden layer 2:** ReLU activation, fully connected, 128 neurons;
- **Output layer:** $|\mathcal{A}|$ neurons (one Q-value per action), fully connected.

The output vector is $Q(\hat{\mathbf{s}}(t); \theta) \in \mathbb{R}^{|\mathcal{A}|}$, and the greedy action is $\pi(\mathbf{s}) = \arg\max_a Q(\mathbf{s}, a; \theta)$.

5.6. Experience Replay

To improve sample efficiency and break temporal correlations between consecutive transitions, an experience replay buffer $\mathcal{D}$ with capacity C is maintained. At each step, the transition tuple $(\mathbf{s}(t), a(t), r(t), \mathbf{s}(t+1), \text{done})$ is stored in $\mathcal{D}$. During training, mini-batches of size N_{batch} are sampled uniformly from $\mathcal{D}$ to update the network parameters. This mechanism provides two benefits: (i) each transition may be used in multiple gradient updates, improving data efficiency; and (ii) the uniform sampling decorrelates the training data, reducing the variance of gradient estimates.

5.7. Target Network

Directly using the online network $Q(\mathbf{s}, a; \theta)$ to compute both the current Q-values and the bootstrap targets introduces harmful feedback loops that can destabilize training. To mitigate this, a target network $Q(\mathbf{s}, a; \theta^-)$ with a periodically synchronized copy of the parameters is used. The target Q-value for a transition is:

$$y(t) = \begin{cases} r(t), & \text{if terminal,} \\ r(t) + \gamma \max_{a'} Q(\mathbf{s}(t+1), a'; \theta^-), & \text{otherwise.} \end{cases} \tag{44}$$

The target parameters θ^- are synchronized with the online parameters θ every N_{target} gradient steps.

5.8. Loss Function and Optimization

The DQN is trained by minimizing the mean squared temporal difference error over sampled mini-batches:

$$\mathcal{L}(\theta) = EE_{(\mathbf{s},a,r,\mathbf{s}')\sim\mathcal{D}}\Big[\big(y - Q(\mathbf{s}, a; \theta)\big)^2\Big]. \tag{45}$$

Gradient descent is performed using the Adam optimizer with learning rate α:

$$\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta). \tag{46}$$

5.9. Exploration-Exploitation Strategy

The agent follows an ϵ-greedy policy during training:

$$a(t) = \begin{cases} \text{random action from } \mathcal{A}, & \text{with probability } \epsilon, \\ \arg\max_{a\in\mathcal{A}} Q(\mathbf{s}(t), a; \theta), & \text{with probability } 1-\epsilon. \end{cases} \tag{47}$$

The exploration rate is initialized at $\epsilon_{\text{start}} = 1.0$ and decayed multiplicatively after each episode:

$$\epsilon \leftarrow \max(\epsilon_{\min},\ \rho \cdot \epsilon), \tag{48}$$

where $\rho = 0.992$ is the decay factor, and $\epsilon_{\min} = 0.05$. This schedule encourages broad exploration early in training and progressively shifts to exploitation as the Q-function estimate matures.

5.10. Training Procedure

The complete DQN training procedure is summarized in Algorithm 1, which proceeds through seven stages at each time step:

1. State extraction and normalization, producing the fixed-dimension normalized state vector.
2. ϵ-greedy action selection over the joint action space.
3. Environment execution, in which the selected scheduling, power, bandwidth, and movement decisions are applied, and the reward is computed.
4. Storage of the resulting transition in the experience replay buffer.
5. Mini-batch sampling from the replay buffer once sufficient transitions are available.
6. Target-value computation using the target network and a gradient step on the online network via Adam.
7. Periodic synchronization of the target network parameters with the online network every target steps.

Algorithm 1 DQN-based energy-efficient resource allocation and scheduling.

Initialize: replay memory $\mathcal{D}$ with capacity C; online DQN with parameters θ; target DQN with $\theta^- \leftarrow \theta$; exploration rate $\epsilon \leftarrow \epsilon_{\text{start}}$; step counter $\tau \leftarrow 0$
for episode $= 1, 2, \ldots, N_{\text{ep}}$ **do**
  Sample K_{active} uniformly from $\{4, 6, 8, 10\}$
  Reset environment, obtain initial state $\mathbf{s}(1)$
  for $t = 1, 2, \ldots, T$ **do**
    With probability ϵ: select random action $a(t)$; else $a(t) = \arg\max_a Q(\mathbf{s}(t), a; \theta)$
    Execute $a(t)$; observe reward $r(t)$ and next state $\mathbf{s}(t+1)$; observe done flag
    Store $(\mathbf{s}(t), a(t), r(t), \mathbf{s}(t+1), \text{done})$ in $\mathcal{D}$
    $\tau \leftarrow \tau + 1$
    if $|\mathcal{D}| \geq \max(N_{\text{batch}}, N_{\text{start}})$ **then**
      Sample mini-batch $\{(\mathbf{s}_i, a_i, r_i, \mathbf{s}'_i, d_i)\}_{i=1}^{N_{\text{batch}}}$ from $\mathcal{D}$
      Compute targets y_i using (44) with θ^-
      Minimize $\mathcal{L}(\theta) = \frac{1}{N_{\text{batch}}} \sum_i (y_i - Q(\mathbf{s}_i, a_i; \theta))^2$ via Adam
      if $\tau \bmod N_{\text{target}} = 0$ **then**
        $\theta^- \leftarrow \theta$
      end if
    end if
    if done **then**
      break
    end if
  end for
  $\epsilon \leftarrow \max(\epsilon_{\min},\ \rho \cdot \epsilon)$
end for
return trained online DQN with parameters θ^*

5.11. Multi-Environment Training for Generalization

A key design choice in the training procedure is the random selection of K_{active} at the start of each episode from the set $\{4, 6, 8, 10\}$. This multi-environment training strategy exposes the agent to diverse network configurations and encourages the learned policy to generalize across different numbers of active IoT devices without retraining. The fixed-size state vector and the active-device ratio appended to the state provide the agent with the necessary context to distinguish the current operating condition.

5.12. Online Deployment

After training, the agent operates in a purely greedy fashion: at each slot t, the normalized state $\hat{\mathbf{s}}(t)$ is passed through the trained DQN, and the action with the highest predicted Q-value is selected:

$$a^*(t) = \arg\max_{a \in \mathcal{A}} Q(\hat{\mathbf{s}}(t), a; \theta^*). \tag{49}$$

The forward pass through the two-hidden-layer DQN requires only $O(N_S \cdot h_1 + h_1 \cdot h_2 + h_2 \cdot |\mathcal{A}|)$ multiply-accumulate operations per slot; far less computationally demanding than solving an MINLP at every slot; making the approach suitable for real-time deployment on embedded UAV hardware.

5.13. Complexity Analysis

Let N_S denote the state dimension, h_1 and h_2 the widths of the two hidden layers, and $N_A = |\mathcal{A}|$ the number of actions. The parameter count of the DQN is $\Theta = N_S h_1 + h_1 h_2 + h_2 N_A + h_1 + h_2 + N_A$. For $N_S = 24$, $h_1 = h_2 = 128$, and $N_A = 800$, this amounts to approximately 119K parameters, which is modest by Deep Learning (DL) standards. The training complexity per gradient step is $O(N_{\text{batch}} \cdot \Theta)$, and the total training budget

is $N_{\text{ep}} \times T$ environment steps. Online inference requires only a single forward pass of $O(\Theta/N_A)$ operations per time slot.

6. Simulation Setup

6.1. Network Parameters

Simulations are conducted in MATLAB 2026a using the DL Toolbox Version 26.1 The network parameters are summarized in Table 1. Also, it is noted that UAV hovering energy/slot is expressed as a normalized per-slot energy unit calibrated for the simulated micro-scale UAV/IoT testbed, rather than as an absolute physical hover-power measurement.

Table 1. Simulation parameters.

Parameter	Symbol	Value
Service area dimensions	$L_x \times L_y$	300×300 m
BS position	$\mathbf{w}_b$	$[150, 150]$ m
UAV initial position	$\mathbf{q}(1)$	$[150, 150]$ m
UAV altitude	H	100 m (default)
UAV step size per slot	δ_{xy}	20 m
Maximum active IoT devices	$K_{\max}$	10
Slot duration	Δt	1 s
Total system bandwidth	B_{tot}	1 MHz
Noise PSD	N_0	10^{-17} W/Hz
Reference channel gain	β_0	10^{-5}
Path-loss exponent (access)	$\alpha_{k,u}$	2.2
Path-loss exponent (backhaul)	$\alpha_{u,b}$	2.0
IoT transmit power levels	$\mathcal{P}$	$\{0.05, 0.10, 0.20, 0.35\}$ W
UAV relay transmit power	p_u	0.4 W
Bandwidth fractions	$\mathcal{B}/B_{tot}$	$\{0.20, 0.35, 0.50, 0.70\}$
Circuit power	P_c	1.2 W
UAV hovering energy/slot	E_{hov}	2.5 J
UAV move energy/step	$\kappa \cdot \delta_{xy}$	6 J
Max queue size (bits)	$Qmax_k$	8×10^5 bits
Mean arrival rate	λ_k	7×10^4 bits/slot (default)
Default UAV battery budget	$E_u^{\max}$	600 J
DQN Hyperparameters		
Discount factor	γ	0.98
Initial exploration rate	ϵ_{start}	1.0
Minimum exploration rate	$\epsilon_{\min}$	0.05
Exploration decay	ρ	0.992
Learning rate	α	2×10^{-3}
Mini-batch size	N_{batch}	64
Replay buffer capacity	C	8000
Target update interval	N_{target}	100 steps
Hidden layer widths	h_1, h_2	128, 128
Training episodes	N_{ep}	180
Steps per episode	T	80
Learning start step	N_{start}	400
Reward Coefficients		
EE weight	λ_1	1.0
Queue penalty weight	λ_2	4×10^{-6}
Battery penalty weight	λ_3	0.01
Fairness weight	λ_4	0.10
Invalid-action penalty	λ_5	2.0

Baseline Definitions

To ensure reproducibility, the exact decision rule of each baseline is defined as follows:

1. RR: Devices are scheduled in a fixed cyclic order regardless of queue or channel state, with transmit power and bandwidth fixed at mid-range values and the UAV hovering at a fixed position.
2. Random Allocation (RA): At each slot, one active device, power level, bandwidth fraction, and UAV movement direction are each selected uniformly at random from their respective sets.
3. Max-SNR: the device with the highest instantaneous access-link channel gain is scheduled using the maximum power level and bandwidth fraction; the UAV hovers.
4. PF: The device maximizing the ratio of its instantaneous achievable rate to its exponentially averaged historical rate is scheduled, with power and bandwidth fixed at mid-range values.
5. Lyapunov Heuristic: A drift-plus-penalty virtual-queue scheduler with fixed control parameter selects, at each slot, the device maximizing the weighted product of its queue backlog and achievable rate, with power and bandwidth chosen to maximize this weighted metric subject to the constraint set; the UAV hovers.
6. GreedyEE: At each slot, the device/power/bandwidth combination maximizing the instantaneous per-slot EE is selected greedily, with no consideration of queue backlog or fairness; the UAV hovers. This purely myopic, EE-only objective is why GreedyEE exhibits higher delay and lower fairness than PF, RR, and the DQN.

It is noted that the proposed DQN jointly controls scheduling, transmit power, bandwidth, and the UAV movement within a single learned policy, consistent with the central contribution of a unified joint-control framework. The hovering baselines are representative of the classical, decomposed resource-allocation literature surveyed in Section 2, against which this unified framework is positioned, and are therefore intentionally evaluated with a fixed UAV position rather than as an ablation of the DRL algorithm alone. However, the RA baseline already has access to the identical action space as the DQN.

6.2. Evaluation Protocol

After training, each method is evaluated over 20 independent evaluation episodes, each consisting of 80 time slots, with fresh random IoT device positions and initial queue states drawn at the beginning of each episode. Performance metrics are averaged across all evaluation episodes. Each experimental axis is varied independently while all other parameters remain at their default values. The following axes are studied:

1. **Number of active IoT devices:** $K_{\text{active}} \in \{4, 5, 6, 7, 8, 9, 10\}$;
2. **Mean traffic arrival rate:** $\lambda_k \in \{3, 5, 7, 9, 11\} \times 10^4$ bits/slot;
3. **UAV battery budget:** $E_u^{\max} \in \{300, 450, 600, 750, 900\}$ J;
4. **UAV altitude:** $H \in \{60, 70, 80, 90, 100, 110, 120, 130, 140\}$ m.

Performance is assessed using four metrics: (i) average energy efficiency η_{avg} (bits/J), (ii) sum throughput $\sum_k \bar{D}_k$ (Mbits per episode), (iii) average delay (slots), and (iv) Jain's fairness index $\mathcal{J}$.

7. Results and Discussion

Figure 2 shows the cumulative reward per episode during DQN training. The smoothed reward curve exhibits a clear trend from the initial random exploration phase, followed by a rapid improvement phase as the ϵ-greedy exploration rate decays and the experience replay buffer accumulates sufficient transition diversity. The reward stabilizes and oscillates, indicating convergence of the learned Q-function.

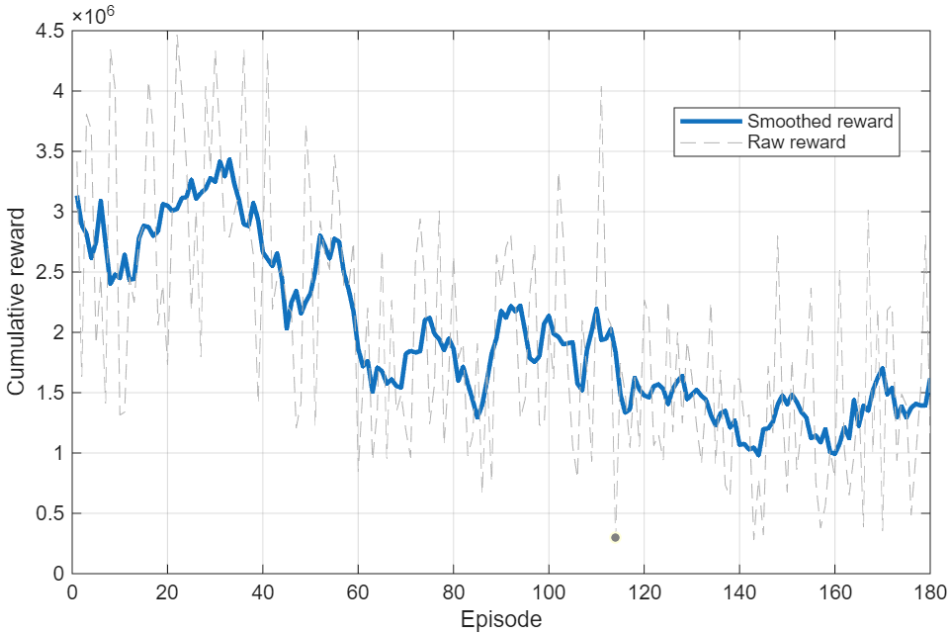

Figure 2. DQN training convergence.

Figure 3 presents the average EE(bits/J) of all seven methods as the number of active IoT devices increases from 4 to 10. Several important observations can be made. First, the DQN policy consistently achieves the highest EE across all device counts, with a margin that grows as K increases. We note that this margin reflects the DQN learned use of the joint action space rather than simply the availability of the UAV-movement action: the RA baseline has access to the identical action set, including UAV movement, yet achieves the lowest EE of all seven methods, confirming that undirected mobility does not by itself explain the DQN advantage. At $K = 10$, the DQN achieves approximately 15×10^4 bits/J, compared to 14×10^4 bits/J for the closest competitors (RR and PF) and $\sim 6 \times 10^4$ bits/J for GreedyEE. Second, the EE of most methods increases with K. This counterintuitive result is explained by the fact that, with more devices in the network, the scheduler has a higher probability of finding a device with a favorable channel gain that requires relatively little power to achieve a given throughput. The DQN learns to exploit this more aggressively than the baselines.

Figure 4 shows how sum throughput (Mbits per episode) scales with the UAV battery budget, with all other parameters at their default values and $K = 8$ active devices. The throughput of all methods increases monotonically with the battery budget, as a larger budget allows more data to be transmitted before the UAV depletes its energy reserve. However, the rate of improvement differs substantially across methods. The DQN achieves among the highest throughput at all battery levels. At 300 J (the most constrained setting), the DQN delivers approximately 38 Mbits, closely matched by PF and RR, both of which benefit from the fixed relay transmission energy not being wasted on aggressive UAV movement. As the battery budget increases, the DQN more aggressively repositions the UAV toward devices with poor channels, further improving the bottleneck access link rate and pulling ahead of static or near-static baselines.

Figure 5 plots the average queue delay (in slots) against the mean traffic arrival rate λ_k for $K = 8$ active devices. Delay is computed as the average ratio of queue backlog to arrival rate (a proxy for Little's law). As the arrival rate increases, all methods exhibit growing delays since the service rate is bounded by both the available bandwidth and the UAV battery. However, the DQN maintains the lowest delay among all methods; this highlights the

value of the queue penalty term λ_2 in the reward function (38), which explicitly discourages queue buildup and leads the agent to preemptively schedule congested devices.

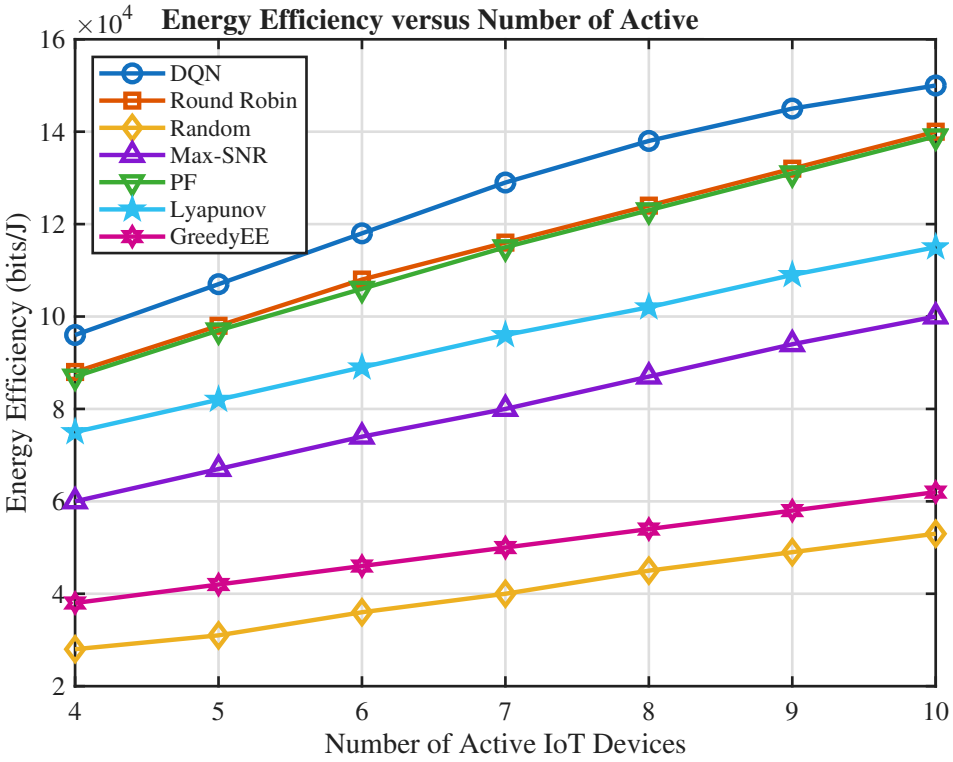

Figure 3. EE vs. number of active IoT devices.

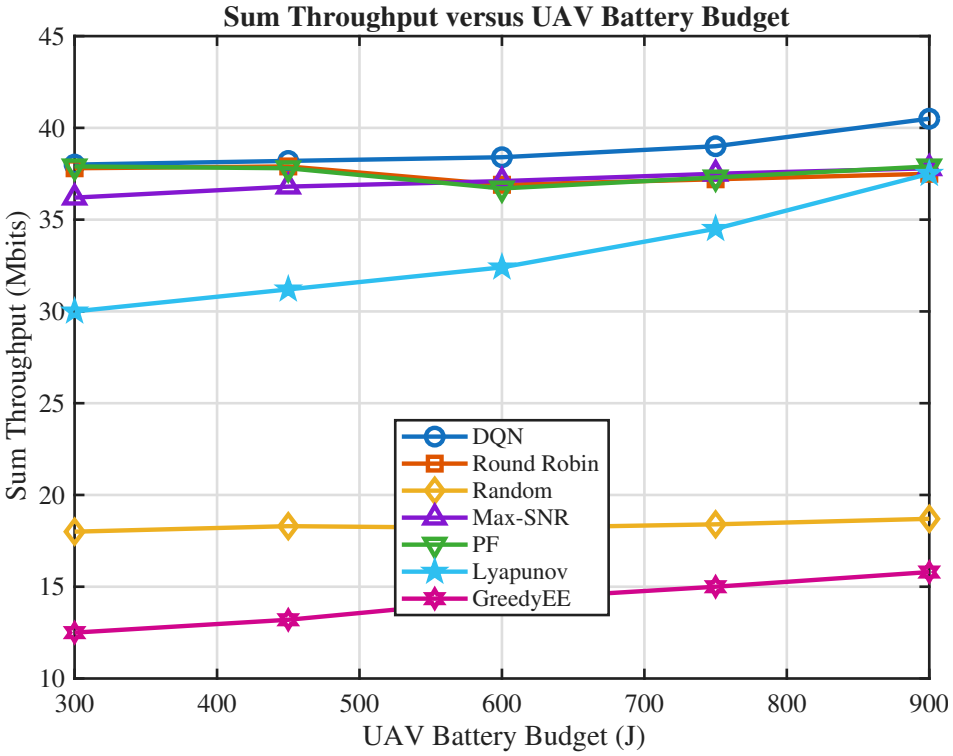

Figure 4. Throughput vs. UAV battery budget.

Figure 6 shows Jain's fairness index versus UAV altitude for $K = 8$ active devices. At low altitudes, the access channel gains vary dramatically with the IoT device positions, since the path loss from (6) depends strongly on $d_{k,u}(t)$ for small H. This creates highly heterogeneous channel conditions that channel-greedy methods (Max-SNR) exploit by

preferentially scheduling devices with the strongest channels, resulting in poor fairness ($\mathcal{J} \approx 0.1$). The DQN, PF, and RR policies maintain high fairness (near $\mathcal{J} = 1.0$) across all altitudes. The DQN achieves this by virtue of the Jain fairness term $\mathcal{F}(t)$ in the reward, which penalizes cumulative service imbalance and encourages the agent to occasionally schedule weaker devices even when they are not the best EE choice.

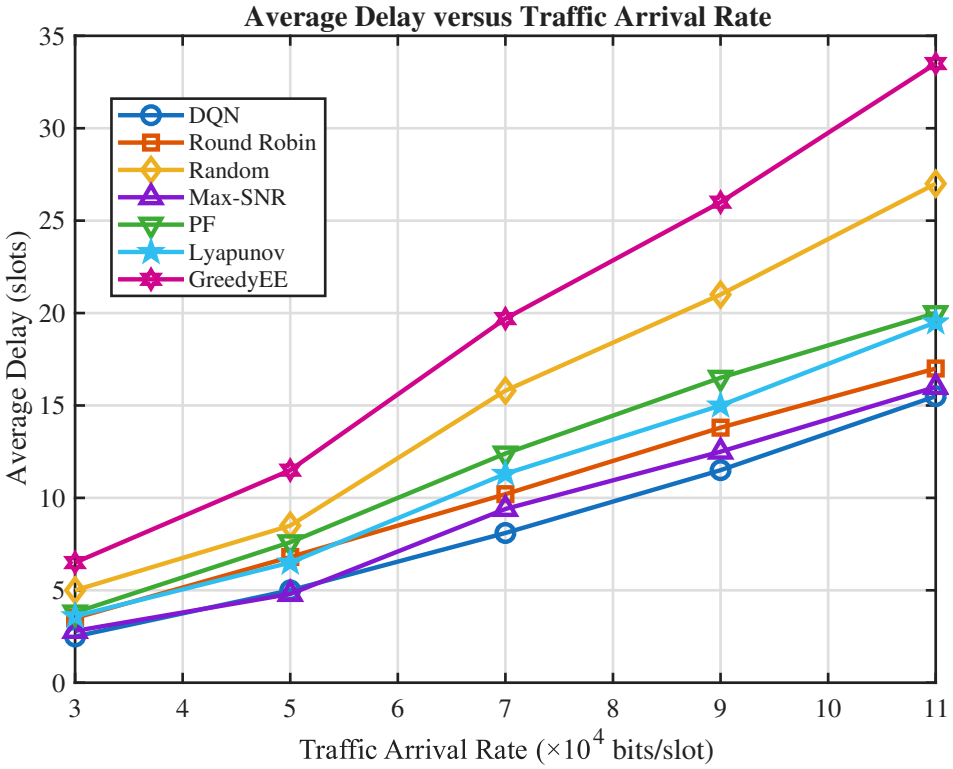

Figure 5. Average delay vs. traffic arrival rate.

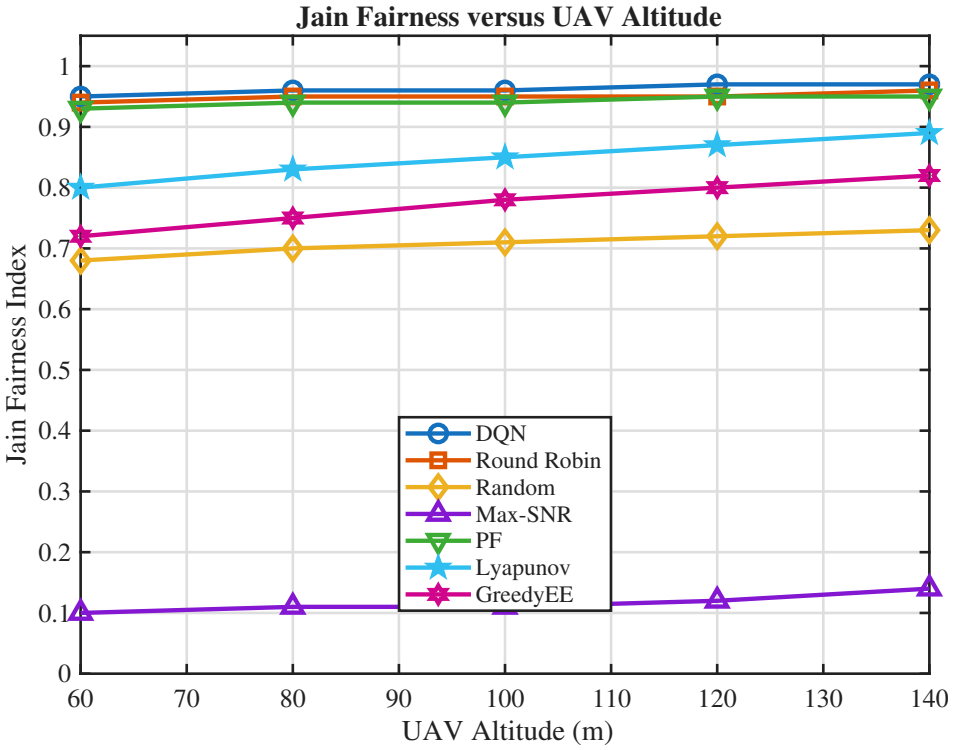

Figure 6. Fairness vs. UAV altitude.

8. Future Directions

Several promising research directions emerge, which are as follows:

1. **Multi-UAV multi-agent DRL:** Extending the framework to multiple cooperating UAV using techniques such as centralized training with decentralized execution.
2. **3D UAV trajectory optimization:** Incorporating variable altitude into the action space to jointly optimize horizontal position and altitude for improved channel quality and coverage.
3. **Advanced actor-critic methods:** Replacing DQN with continuous-action actor-critic algorithms or soft actor-critic to avoid the discretization of power and bandwidth and potentially improve performance at fine resource granularity.
4. **Recurrent DRL for partial observability:** Using recurrent neural networks within the DRL agent to handle partial observability arising from imperfect channel estimation or delayed queue reporting.
5. **Transfer learning across network sizes:** Investigating whether a policy trained on a network with K devices can be efficiently fine-tuned for a different K' using transfer learning or meta-RL approaches.
6. **The generalization:** The generalization test in Section 6.2 evaluates interpolation across K_{active} 4,...,10; extrapolation beyond the trained device-count range (e.g., training with $K_{\text{active}} \leq 8$ and testing at K_{active}= 10 or 12) will rigorously establish robustness.

9. Conclusions

This paper has presented a DQN-based framework for jointly optimizing IoT user scheduling, uplink transmit power allocation, bandwidth assignment, and UAV trajectory control in a 6G-enabled UAV-assisted IoT uplink network. The proposed framework models the dynamic network control problem as an MDP and employs experience replay, a target network, and ϵ-greedy exploration to learn a stable and effective control policy. The reward function is designed to promote EE while explicitly penalizing queue buildup, battery depletion, and unfair scheduling, enabling the agent to discover non-myopic policies that balance short-term and long-term objectives. Extensive MATLAB simulations demonstrate that the proposed DQN policy consistently outperforms the state-of-the-art baselines across varying numbers of IoT devices, traffic arrival rates, UAV battery budgets, and operating altitudes. Notably, the RA baseline, which has access to the identical joint action space including UAV movement, performs worst among all evaluated policies, indicating that the DQN advantage arises from learned, state-conditioned joint control rather than from UAV mobility being available in itself. The proposed algorithm achieves 15×10^4 bits/J EE and 38 Mbit throughput, compared to the best state-of-the-art.

Author Contributions: Conceptualization, A.N.; simulations, A.N. ; writing—original draft preparation, A.N.; writing—review and editing, A.N. and S.W.K.; supervision, S.W.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (RS-2021-NR060120), by the BK21 FOUR of the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2120251015517), and by the NRF grant funded by the Korea government (MSIT) (RS-2022-NR069161).

Informed Consent Statement: Not applicable.

Data Availability Statement: The simulation results and configuration details supporting the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. IoT Analytics. Number of Connected IoT Devices Growing 14% to 21.1 Billion. 2025. Available online: https://iot-analytics.com/number-connected-iot-devices/ (accessed on 22 June 2026).
2. Jamshed, M.A.; Ali, K.; Abbasi, Q.H.; Imran, M.A.; Ur-Rehman, M. Challenges, applications, and future of wireless sensors in Internet of Things: A review. *IEEE Sens. J.* **2022**, *22*, 5482–5494.
3. Aslam, N.; Wang, H.; Esmaiel, H.; Junejo, N.U.R.; Agamy, A. A Review of Wireless Charging Solutions for FANETs in IoT-Enabled Smart Environments. *Sensors* **2026**, *26*, 912.
4. Xu, Y.; Tang, X.; Huang, L.; Ullah, H.; Ning, Q. Multi-objective optimization for resource allocation in space–air–ground network with diverse IoT devices. *Sensors* **2025**, *25*, 274.
5. Liu, T.; Sun, C.; Zhang, Y.; Sun, W. Energy-Efficient End-to-End Optimization for UAV-Assisted IoT Data Collection and LEO Satellite Offloading in SAGIN. *Electronics* **2025**, *15*, 24.
6. Zeng, Y.; Zhang, R.; Lim, T.J. Wireless communications with unmanned aerial vehicles: Opportunities and challenges. *IEEE Commun. Mag.* **2016**, *54*, 36–42.
7. Giordani, M.; Polese, M.; Mezzavilla, M.; Rangan, S.; Zorzi, M. Toward 6G networks: Use cases and technologies. *IEEE Commun. Mag.* **2020**, *58*, 55–61.
8. Jamshed, M.A.; Kaushik, A.; Dobre, O.A.; Zhang, Y.J.A.; Lin, X. Guest editorial: Unveiling the cosmic connection: Ai's pivotal role in bridging terrestrial and non-terrestrial networks. *IEEE Wirel. Commun.* **2025**, *32*, 18–19.
9. Zeng, Y.; Xu, J.; Zhang, R. Energy minimization for wireless communication with rotary-wing UAV. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 2329–2345.
10. Jamshed, M.A.; Kaushik, A.; Toka, M.; Shin, W.; Shakir, M.Z.; Dash, S.P.; Dardari, D. Synergizing airborne non-terrestrial networks and reconfigurable intelligent surfaces-aided 6G IoT. *IEEE Internet Things Mag.* **2024**, *7*, 46–52.
11. Luo, Z.Q.; Yu, W. An introduction to convex optimization for communications and signal processing. *IEEE J. Sel. Areas Commun.* **2006**, *24*, 1426–1438.
12. Jamshed, M.A.; Ismail, M.; Pervaiz, H.; Atat, R.; Bayram, I.S.; Ni, Q. Reinforcement learning-based allocation of fog nodes for cloud-based smart grid. *E-Prime Electr. Eng. Electron. Energy* **2023**, *4*, 100144.
13. Tang, Z.; Wei, X.; Wei, Z.; Tan, F.; Tian, C.; Tang, Y.; Xiong, X. Deep Reinforcement-Learning-Optimized Adaptive EKF for Robust Utility Harmonic Impedance Estimation. *Electronics* **2026**, *15*, 2557.
14. Nauman, A.; Jamshed, M.A.; Ali, R.; Cengiz, K.; Kim, S.W. Reinforcement learning-enabled intelligent device-to-device (I-D2D) communication in narrowband Internet of Things (NB-IoT). *Comput. Commun.* **2021**, *176*, 13–22.
15. Wu, Q.; Zeng, Y.; Zhang, R. Joint trajectory and communication design for multi-UAV enabled wireless networks. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 2109–2121.
16. Mozaffari, M.; Saad, W.; Bennis, M.; Nam, Y.H.; Debbah, M. A tutorial on UAVs for wireless networks: Applications, challenges, and open problems. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2334–2360.
17. Hua, M.; Wang, Y.; Zhang, Z.; Li, C.; Huang, Y.; Yang, L. Power-efficient communication in UAV-aided wireless sensor networks. *IEEE Commun. Lett.* **2018**, *22*, 1264–1267.
18. Liu, B.; Zhu, H. Energy-effective data gathering for UAV-aided wireless sensor networks. *Sensors* **2019**, *19*, 2506.
19. Wang, L.; Wang, K.; Pan, C.; Chen, X.; Aslam, N. Deep Q-network based dynamic trajectory design for UAV-aided emergency communications. *J. Commun. Inf. Netw.* **2020**, *5*, 393–402.
20. Wang, L.; Wang, K.; Pan, C.; Xu, W.; Aslam, N.; Hanzo, L. Multi-agent deep reinforcement learning-based trajectory planning for multi-UAV assisted mobile edge computing. *IEEE Trans. Cogn. Commun. Netw.* **2020**, *7*, 73–84.
21. Wang, C.; Liu, K.; Yuan, Y.; Peng, S.; Li, G. Joint trajectory and offloading optimization in UAV-assisted MEC via federated multi-agent reinforcement learning and potential fields. *Comput. Netw.* **2025**, *272*, 111681.
22. Luong, N.C.; Hoang, D.T.; Gong, S.; Niyato, D.; Wang, P.; Liang, Y.C.; Kim, D.I. Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3133–3174.
23. Choudhury, B.; Shah, V.K.; Ferdowsi, A.; Reed, J.H.; Hou, Y.T. AoI-minimizing scheduling in UAV-relayed IoT networks. In Proceedings of the 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS), Virtual Event, 4–7 October 2021; pp. 117–126.
24. Yan, J.; Wang, Z.; Liu, S.; Liu, H.; Zhang, C.; Wang, B.; Sun, F.; Shen, Z.; Liu, Q.; Xu, J. Energy-Constrained UAV-UGV Coordination for Online Task Discovery in Known Environments with Obstacles. *Drones* **2026**, *10*, 343.
25. Gao, S.; Qiao, Y.; Fu, K.; Hui, K.; Huang, R. Energy-Fairness-Aware Multi-UAV Trajectory Planning Algorithm for AoI-Constrained Data Collection. *IEEE Trans. Veh. Technol.* 2026, *Early Access*.
26. Qu, L.; Bai, G.; Dai, C.; Liu, J.; Sun, D. Learning-Based AoI Minimization Through UAV-Assisted Data Distribution in Vehicular Networks. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 19645–19660.

27. Ding, X.; Lu, Q.; Zhang, Y.; Li, G.; Gao, X.; Ye, N.; Niyato, D.; Yang, K. Few-shot recognition and classification framework for jamming signal: A CGAN-based fusion CNN approach. *IEEE Trans. Veh. Technol.* **2026**, *75*, 17494–17508.
28. Jamshed, M.A.; Ayaz, F.; Kaushik, A.; Fischione, C.; Ur-Rehman, M. Green uav-enabled internet-of-things network with ai-assisted noma for disaster management. In Proceedings of the 2024 IEEE 35th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Valencia, Spain, 2–5 September 2024; pp. 1–6.
29. Mohsin, M.A.; Umer, M.; Bilal, A.; Jamshed, M.A.; Cioffi, J.M. Continual learning for wireless channel prediction. *arXiv* **2025**, arXiv:2506.22471.
30. Awais, M.; Pervaiz, H.; Jamshed, M.A.; Yu, W.; Ni, Q. Energy-aware resource optimization for improved urllc in multi-hop integrated aerial terrestrial networks. *IEEE Trans. Green Commun. Netw.* **2023**, *8*, 252–264.