Energy-Efficient Data Collection and Task Offloading Optimization in Heterogeneous Multi-Tier UAV Systems via Deep Reinforcement Learning

Jiayi Tang, Xuting Duan, *Member, IEEE,* Jianshan Zhou, Kaige Qu, *Member, IEEE,* Ivan Wang-Hei Ho, *Senior Member, IEEE,* Daxin Tian, *Fellow, IEEE*

***Abstract*—Unmanned aerial vehicles (UAVs) have gained considerable attention in data collection due to their mobility and flexibility. These capabilities are crucial in time-sensitive missions (e.g., disaster response, military reconnaissance). In such cases, tasks are often subject to tight deadlines and require timely access to information. To address these challenges, this paper investigates collaborative data collection and task offloading in multi-UAV systems, aiming to maximize mission area coverage while minimizing total energy consumption. To overcome the limited computing power of data collection UAVs, we propose a heterogeneous multi-tier UAV system. In this design, an assisted UAV with strong computing capabilities is introduced to handle data offloading and processing. This enhances energy efficiency and enables timely task execution. Consequently, we develop an integrated optimization model to jointly design trajectory planning and task offloading under communication, energy, and deadline constraints. We propose a deep reinforcement learning algorithm called data collection optimized proximal policy optimization (DCOPPO). This approach optimizes both UAV trajectories and offloading decisions. Simulation results demonstrate that DCOPPO significantly outperforms baseline DRL approaches in terms of energy efficiency and task completion performance.**

***Index Terms*—UAV, data collection, deep reinforcement learning, task offloading, path planning, energy efficient.**

I. Introduction

UNMANNED aerial vehicles (UAVs) have been increasingly employed in data collection tasks due to their mobility, flexibility, and ease of deployment. They have shown great potential in a wide range of applications, such as environmental monitoring [1], precision agriculture [2], disaster assessment [3], and military reconnaissance [4]. In these missions, UAVs must gather raw data and perform computationally intensive tasks, such as image processing, object detection, and trajectory adjustment. However, these operations require significant processing power and energy, which are often limited by the hardware constraints of onboard systems. To address this limitation, offloading certain tasks to more capable devices is necessary to reduce latency, save energy, and ensure timely task execution. Therefore, this motivates the need for efficient task offloading and coordination strategies in UAV networks.

In data collection missions, path planning is critical to completing tasks efficiently. Common approaches include rule-based strategies and optimization-based methods [5]–[7]. These methods can generate feasible trajectories when the environment is known and the task is stable. However, they often lack the ability to respond quickly to unexpected events or dynamic conditions. Even with optimized trajectories, UAVs are constrained by limited battery capacity and onboard computing resources. They cannot perform computationally intensive tasks such as image recognition or object detection for long periods or at high frequency.

Task offloading helps relieve the computation bottleneck of UAVs in computationally intensive tasks. It transfers part of the workload to more powerful nodes, reducing local energy use and delay. Common modes include ground-based and airborne edge computing. Both modes can be jointly optimized with trajectory planning to improve task efficiency under limited resources [8]. However, task offloading and trajectory optimization are interdependent. Traditional methods often face high computational complexity in joint decision-making [9]. They also lack the ability to adapt quickly to changes in task requirements and environmental conditions. As a result, finding a globally optimal strategy in dynamic scenarios remains challenging.

In recent years, deep reinforcement learning (DRL) has been widely applied to UAV path planning and task coordination [10]. It offers online learning, dynamic policy updates, and the ability to search for near-optimal solutions in high-dimensional state spaces. DRL can adjust flight paths and task allocation strategies as the environment changes. This improves task completion efficiency and system adaptability. However, most existing studies assume homogeneous UAV systems, where all UAVs have the same communication and computation capabilities [11], [12]. In heterogeneous multi-tier UAV systems, different UAV types vary in computing power,

This work was supported by the National Natural Science Foundation of China under Grant 62173012, Grant 62432002, Grant U2433202 and Grant U22A2046, the Fundamental Research Funds for the Central Universities (Beihang Ganwei Action Plan Key Program) under Grant JK2024-19. (*Corresponding author: Xuting Duan.*)

Jiayi Tang, Xuting Duan, Jianshan Zhou, Kaige Qu, Daxin Tian are with the State Key Lab of Intelligent Transportation System, Beijing Key Laboratory for Cooperative Vehicle Infrastructure Systems and Safety Control, School of Transportation Science and Engineering. Beihang University, Beijing 100191, China (e-mail: duanxuting@buaa.edu.cn).

Xuting Duan and Daxin Tian are also with the Zhongguancun Laboratory. Beijing 100081, China.

Ivan Wang-Hei Ho is with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China.

energy capacity, and task roles. Ignoring these differences can limit the effectiveness of joint optimization strategies.

To improve the coordination efficiency of multi-UAV systems in complex missions, this paper proposes a heterogeneous multi-tier system. In this design, an assisted UAV with stronger computing power supports computationally heavy tasks. Moreover, a joint optimization framework is developed to design both trajectories and offloading strategies. To solve the problem under dynamic and constrained conditions, we propose a DRL-based solution. The contributions of this article are summarized as follows.

1) We design a multi-tier UAV system consisting of multiple data collection UAVs and one assisted UAV with enhanced computing capabilities. Data collection UAVs are responsible for collecting data within the target area, while the assisted UAV helps process computationally intensive tasks offloaded by the data collection UAVs. This collaborative system reduces the burden on onboard resources and effectively addresses the computation bottlenecks in data collection missions.
2) We formulate a joint optimization model for trajectory planning and task offloading. The objective is to maximize task completion and minimize overall energy consumption. The model considers deadline constraints, energy limits, and obstacle avoidance. To solve it, we develop a DRL algorithm called data collection proximal policy optimized (DCOPPO). It learns both optimal flight paths and offloading strategies through policy updates, improving task efficiency and resource utilization.
3) We evaluate the proposed method through extensive simulations. In a typical time-sensitive area data collection task, DCOPPO outperforms baseline algorithms across multiple metrics. Specifically, it achieves 100% final coverage while DQN and A2C reach only 97.96%, and reduces total energy consumption by 19.07% compared to DQN. These results confirm the method's strong performance in task execution, path planning, and resource utilization, making it suitable for multi-tier UAV system coordination in dynamic environments.

The structure of this paper is organized as follows: Section II reviews the related work in the field. Section III provides a detailed description of the system model, including the mobility model, communication model, and computation model of UAVs, and formulates a joint optimization problem. Section IV presents the optimization algorithm framework based on PPO. Section V evaluates the performance of the proposed method through simulation experiments and provides a comparative analysis with existing approaches. Finally, Section VI concludes the paper and discusses potential directions for future research.

II. Related Work

In early studies, UAVs commonly performed data collection tasks by following predefined paths. Jiao et al. [5] used exact cellular decomposition and fixed-direction sweeps to guide UAVs for image collection. Similarly, another study [6] applied a line-sweep path to complete 3D terrain reconstruction. To enhance mission efficiency, Cabreira et al. [7] proposed an energy-aware spiral path planning method. However, these approaches mainly relied on rule-based trajectories, which often resulted in low collection efficiency. To overcome this limitation, recent studies have introduced optimization algorithms to design more efficient data collection strategies [13]–[20]. Wang et al. modeled the data collection task as a traveling salesman problem and optimized the visiting order to reduce overall mission time [14]. Hu et al. [15] and Xu et al. [16] used iterative optimization to generate UAV trajectories with higher coverage efficiency and shorter flight distances. Liu et al. [17] jointly optimized UAV deployment, flight paths, and device access sequences using a heuristic algorithm. However, most of these methods rely on static or pre-known task information, such as device locations [18] and channel conditions [20]. These works lack the ability to adjust flight paths in real time when facing dynamic environmental changes. This often leads to reduced task efficiency and higher energy consumption in unpredictable scenarios. Moreover, in time-sensitive data collection scenarios, efficient task execution also relies on timely data processing. These studies primarily focus on flight path design without jointly considering computation offloading during the mission. In contrast, we formulate a joint trajectory planning and computation offloading optimization framework, enabling UAVs to adapt dynamically while balancing task efficiency and processing latency.

In UAV networks, computation offloading is introduced to enhance task execution by supporting data processing beyond onboard capabilities. Recent studies have jointly considered trajectory planning and task offloading to improve overall mission performance. Currently, task offloading and path planning in UAV networks are commonly implemented under two architectures. The first involves cooperation with ground infrastructure. Guo et al. [21] offloaded tasks from UAVs to ground-based mobile edge computing servers and jointly optimized UAV trajectories to improve overall responsiveness and task efficiency. Suganya et al. [22] proposed a dynamic offloading strategy based on edge resource awareness. UAVs choose edge platforms based on link quality and server workload, and optimize their trajectories accordingly. These approaches help reduce the computational burden on UAVs, but their effectiveness is limited by the availability of fixed infrastructure. Ground stations often fail to cover remote or disaster-affected areas, which restricts the flexibility of UAV deployment. Another architecture involves the use of airborne relays or assistive UAVs [23]–[26]. Qi et al. [23] offloaded tasks to computing-capable UAVs and adopted a connected dominating set to enhance task assignment and path planning. Luo et al. [24] developed a multi-UAV scheduling framework based on a heuristic algorithm. They assigned tasks and planned routes based on real-time computing availability and service load. Ma et al. [25] proposed a 3D swarm deployment method where assistive UAVs act as both communication relays and edge computing nodes. Airborne offloading architectures provide greater mobility. However, they involve complex decision-making as multiple factors must be balanced. The joint optimization of trajectory planning and task offloading in such systems is often computationally intensive.

Traditional methods struggle to adapt efficiently to rapidly changing environments, highlighting the need for an online approach capable of learning near-optimal policies in real time.

In recent years, deep reinforcement learning (DRL) has gained increasing attention in UAV applications. DRL enables online learning, dynamic policy adaptation, and efficient processing of high-dimensional states. These capabilities make it well suited for trajectory planning and task coordination in complex, changing environments. For example, Liu et al. [27] proposed a path planning method for a single UAV with edge computing. The UAV uses Deep Q-Network to generate flight paths for serving multiple ground devices. Building on this, Zhao et al. [28] introduced multiple UAVs into the offloading process. A multi-agent DRL framework enables cooperative offloading and resource allocation, improving overall task performance. Furthermore, Li et al. [29] jointly optimized task assignment and trajectory planning. This coordination enhanced scheduling efficiency in complex environments. In [30], safety constraints were added to path planning. The method combines graph search and DRL to support both obstacle avoidance and target coverage in dynamic scenarios. However, most existing studies still assume that all UAVs have the same computing and communication abilities [31]. This setting simplifies system modeling and computation but overlooks the differences in UAV capabilities. In multi-UAV missions, it is difficult to achieve high system efficiency without division of roles and cooperation. To overcome this limitation, this paper introduces a heterogeneous multi-tier UAV system, where high-computation assisted UAVs collaborate with resource-constrained data collection UAVs. This design leverages role differentiation to improve coordination efficiency and overall system performance in dynamic environments.

III. System Model

In this paper, the goal is to complete the data collection task by achieving full area coverage. To support time-sensitive tasks, the system deploys two types of UAVs: data collection UAVs and an assisted UAV. Data collection UAVs focus on image capture and sensing, but they lack the computing power and battery capacity required for extensive processing. To address this issue, the assisted UAV with expanded computing capability handles offloaded activities and does real-time data processing. The assisted UAV can carry edge computing modules, such as NVIDIA Jetson or embedded GPU platforms. It has moderate endurance and stable communication ability. In the absence of ground base stations, it can act as a temporary airborne MEC node. This multi-tier UAV system improves task responsiveness while increasing overall system efficiency. It is particularly well suited for tasks such as disaster recovery, search and rescue, and emergency monitoring.

To simplify planning and analysis, we represent the mission area as a rectangular zone, as shown in Fig. 1. Without loss of generality, the region is partitioned into a grid map made of square cells of side length l, resulting in an $L \times W$ grid. Each grid cell represents a subregion that a UAV can cover from a fixed altitude. The task period is divided into T time slots ($\mathcal{T} = \{1, 2, 3, \ldots, T\}$). In each slot t ($t \in \mathcal{T}$),

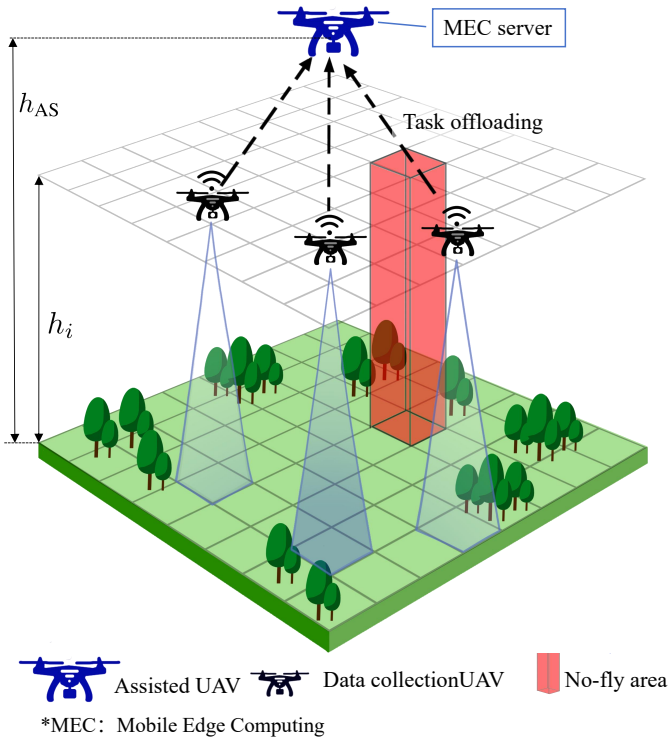


Fig. 1: System model of proposed heterogeneous multi-tier UAV system.

UAV's position and task state is updated. The system consists of I data collection UAVs ($\mathcal{I} = \{1, 2, 3, \ldots, I\}$), denoted as U_i ($i \in \mathcal{I}$), and one assisted UAV U_{AS} with enhanced computing capabilities. In a three-dimensional Cartesian coordinate system, the position of UAV U_i at time t is given by $\mathbf{q}_i[t] = (x_i[t], y_i[t], h_i)$ and the position of the assisted UAV U_{AS} is $\mathbf{q}_{\mathrm{AS}}[t] = (x_{\mathrm{AS}}[t], y_{\mathrm{AS}}[t], h_{\mathrm{AS}})$. For simplicity, we assume that all UAVs operate at fixed altitudes during the mission: h_i for each U_i and h_{AS} for U_{AS}. This reduces the complexity of the model and makes learning more efficient. Similar settings can be found in prior work [32], [33]. Our framework can be extended to variable altitudes by adding altitude levels to the action space.

All UAVs take off from fixed launch points at the beginning of the mission. The data collection UAVs start from a corner of the area, and the assisted UAV starts from the center. This avoids overlap at launch and makes coordination easier. The mission environment may include inaccessible or restricted zones, such as towering buildings or no-fly zones. Fig. 1 shows these regions as obstacles (red blocks in the grid). We denote the set of such inaccessible or restricted zones as $\mathcal{F}$. The key symbols and their respective definitions are summarized in Table I.

During the flight mission, UAV positions evolve over time and are updated using the formulas as follows

$$\mathbf{q}_i[t+1] = \begin{cases} \mathbf{q}_i[t] + \mathbf{a}_{s,i}[t], & \psi_i[t] = 1, \\ \mathbf{q}_i[t], & \psi_i[t] = 0, \end{cases} \tag{1}$$

where $\mathbf{q}_i[t]$ denotes the position of U_i at time t, and $\mathbf{a}_{s,i}[t]$ is its corresponding motion vector. $\psi_i[t] = 1$ indicates that U_i

TABLE I: SYMBOLS AND DEFINITIONS

Symbol	Definition
$q_i[t]$	Position vector of UAV i at time t
$q_{AS}[t]$	Position of the assisted UAV at time t
$p_f[t]$	Takeoff power consumption at time t
$p_l[t]$	Landing power consumption at time t
p_h	Hovering power consumption
$p_r[t]$	Horizontal flight power at time t
$V[t]$	Vertical speed of UAV at time t
$v[t]$	Horizontal speed of UAV at time t
$E_{F,i}[t]$	Flight energy consumption of UAV i
$a_i[t]$	Task offloading decision of UAV i
b_z	Data workload of cell z
c_i	Computation load per bit for UAV i
f_i	CPU frequency of UAV i
$t_{loc,i}$	Local computing time of UAV i
$E_{loc,i}[t]$	Local computing energy of UAV i
k_i	Capacitance coefficient of UAV i
$h_{i,AS}[t]$	Channel gain between UAV i and assisted UAV
$d_{i,AS}[t]$	Distance between UAV i and assisted UAV
$R_{i,AS}^{\max}[t]$	Maximum offloading data rate
$P_{i,AS}$	Transmission power of UAV i
$E_{off,i}[t]$	Offloading energy of UAV i
$E_i[t]$	Total energy of UAV i
$E_{AS}[t]$	Total energy of assisted UAV

is in flight, otherwise it has completed the task and landed.

The motion state transition of UAVs can be described as

$$\psi_i[t+1] = \begin{cases} 0, & \mathbf{a}_{s,i}[t] = [0,0,-h]^T \ \vee \ \psi_i[t] = 0, \\ 1, & \text{otherwise}, \end{cases} \tag{2}$$

where h denotes the flight altitude of the UAV, $\psi_i[t+1]$ is the motion state of the U_i at time $t+1$.

A. Motion Model

The energy consumption of UAVs typically consists of four stages: takeoff, landing, hovering and horizontal flight. The motion characteristics of UAVs during these four stages are represented as follows.

1) Takeoff Stage: The power consumption during the takeoff stage is given by [34]

$$p_{\mathrm{f}}[t] = \frac{W}{2V[t]} + \frac{W}{2}\sqrt{V[t]^2 + \frac{2W}{\rho\pi R^2}}, \tag{3}$$

where W is the total weight of the UAV, $V[t]$ is the vertical speed at time t, ρ is the air density, and R is the rotor radius.

2) Landing Stage: The power consumption during the landing stage is given by

$$p_{\mathrm{l}}[t] = \frac{W}{2V[t]} - \frac{W}{2}\sqrt{V[t]^2 - \frac{2W}{\rho\pi R^2}}. \tag{4}$$

3) Hovering Stage: The power consumption during the hovering stage is given by [27]

$$p_{\mathrm{h}} = \frac{\sqrt{W^3}}{\sqrt{2\rho\pi R^2}}. \tag{5}$$

4) Horizontal Flight Stage: The power consumption during horizontal flight is expressed as [35]

$$\begin{aligned} p_{\mathrm{r}}[t] = & p_0\left(1 + \frac{3v[t]^2}{U_{\mathrm{tip}}^2}\right) + p_i\left(\sqrt{1 + \frac{v[t]^4}{4v_0^4}}\right), \\ & - \frac{v[t]^2}{2v_0^2} + \frac{1}{2}d_0\rho s A v[t]^3 \end{aligned} \tag{6}$$

where $v[t]$ is the flight speed at time t , p_0 is the basic power consumption, U_{tip} is the rotor tip speed, p_i is the induced power, v_0 is the induced velocity, d_0 is the parasite drag coefficient, s is the rotor solidity and A is the reference area of the UAV.

Thus, the total flight energy consumption of U_i at time t can be expressed as

$$E_{F,i}[t] = \left(p_{\mathrm{f}}[t] + p_{\mathrm{l}}[t] + p_{\mathrm{h}}[t] + p_{\mathrm{r}}[t]\right) t_s, \tag{7}$$

where t_s represents the duration of each stage.

B. Local Computation Model

Some computing tasks can be partitioned locally and then partially processed on remote servers. Offloadable tasks, such as image recognition and video stream analysis, have been extensively studied in edge computing settings [36]–[38]. This study focuses on time-sensitive data collection missions. For example, in disaster response and emergency monitoring, UAVs must collect and interpret picture data in a timely manner and return results as soon as possible.

To enable efficient processing, the assisted UAV is modeled as an aerial edge node. We assume that each grid cell $z \in \mathcal{Z} = \{1, 2, \ldots, Z\}$ generates a data workload of size b_z. The data collection UAV offloads part of the task to the assisted UAV, with the offloaded portion denoted as $a_i b_z$, while the remaining part $(1 - a_i)b_z$ is processed locally. The variable $a_i \in [0, 1]$ denotes the offloading decision of data collection UAV U_i.

Considering that the z-th grid is covered by UAV U_i, the delay of local computation is determined by the proportion of the tasks computed locally, task data size, computational requirements of the tasks, and computational capacity, which can be expressed as

$$t_{\mathrm{loc},i} = \frac{(1 - a_i)b_z c_i}{f_i}, \tag{8}$$

where c_i is the number of CPU cycles required per bit of task data, and f_i is the local computational capacity (in cycles per second) of U_i.

The energy consumption for local computation is based on the dynamic voltage and frequency scaling (DVFS) model, where the energy consumption is proportional to the computational capacity and the computation time. The formula is given as [39]

$$E_{\mathrm{loc},i}[t] = k_i f_i^2 t_{\mathrm{loc},i}. \tag{9}$$

Substituting the delay formula $t_{\mathrm{loc},i}$ into the energy consumption formula, it can be expanded as

$$E_{\mathrm{loc},i}[t] = k_i f_i^2 \frac{(1 - a_i)b_z c_i}{f_i} = k_i f_i (1 - a_i) b_z c_i, \tag{10}$$

where $E_{\text{loc},i}$ represents the energy consumption of U_i for local computation (in joules), k_i is the effective capacitance coefficient of the processor, reflecting the energy efficiency of the chip.

C. Channel Model

We use a channel model that considers both line-of-sight (LoS) and non-line-of-sight (NLoS) situations. It accounts for signal blockage caused by obstacles. As a result, it better reflects random variations in real-world communication. The model also captures link uncertainty but stays simple to use. The channel gain $h_{i,\text{AS}}[t]$ between U_i and U_{AS} is divided into LoS and NLoS states, and can be calculated as [40]

$$h_{i,\text{AS}}[t] = \begin{cases} \frac{\beta_0}{d_{i,\text{AS}}[t]^{\alpha_{\text{LoS}}}}, & \text{LoS} \\ \frac{\beta_0}{d_{i,\text{AS}}[t]^{\alpha_{\text{NLoS}}}} 10^{-\eta_e/10}, & \text{NLoS} \end{cases} \tag{11}$$

where β_0 denotes the channel power gain measured at the reference distance, $d_{i,\text{AS}}[t]$ is the distance between U_i and U_{AS} at time slot t, α_{LoS} and α_{NLoS} are the path loss exponents for LoS and NLoS states, respectively, and η_e represents shadowing effects, following a normal distribution $\mathcal{N}(0, \sigma_e^2)$.

Based on the channel gain, the SNR can be expressed as

$$\text{SNR}_{i,\text{AS}}[t] = \frac{P_{i,\text{AS}} h_{i,\text{AS}}[t]}{\sigma^2}, \tag{12}$$

where $P_{i,\text{AS}}$ is the transmission power of UAV, and σ^2 is the noise power. Thus, the data rate is given by

$$R_{i,\text{AS}}^{\max}[t] = B \log_2 \left(1 + \text{SNR}_{i,\text{AS}}[t]\right), \tag{13}$$

where B denotes the channel bandwidth allocated for communication between data collection UAV U_i and the assisted UAV U_{AS}.

In the communication scenario, the probabilities of LoS and NLoS states have a significant impact on channel modeling. The probability of the LoS state is determined by the geometric characteristics between the UAV and ground equipment and can be expressed as [41]

$$\begin{cases} P_{\text{LoS}} = \dfrac{1}{1 + \kappa \exp\left(-\mu(\theta - \kappa)\right)}, \\ P_{\text{NLoS}} = 1 - P_{\text{LoS}}, \end{cases} \tag{14}$$

where θ is the elevation angle between U_i and U_{AS}, and κ, μ are environment-related constants.

Combining the probabilities of LoS and NLoS states, the channel gain at time t is expressed as

$$\bar{h}_{i,\text{AS}}[t] = P_{\text{LoS}} \frac{\beta_0}{d_{i,\text{AS}}[t]^{\alpha_{\text{LoS}}}} + P_{\text{NLoS}} \frac{\beta_0}{d_{i,\text{AS}}[t]^{\alpha_{\text{NLoS}}}} 10^{-\eta_e/10}. \tag{15}$$

Based on the above formulas, the energy consumption during data transmission is calculated as

$$E_{\text{off},i}[t] = P_{i,\text{AS}} \frac{a_i b_z}{B \log_2 \left(1 + \frac{P_{i,\text{AS}} h_{i,\text{AS}}[t]}{\sigma^2}\right)}. \tag{16}$$

D. Overall Energy Consumption Model

In this study, we model the energy consumption of UAVs from three aspects: flight, local computation, and task offloading. The data collection UAV U_i has an overall energy consumption at time step t expressed as

$$E_i[t] = E_{F,i}[t] + E_{\text{loc},i}[t] + E_{\text{off},i}[t]. \tag{17}$$

Similarly, the energy consumption of the assisted UAV U_{AS} is modeled as

$$E_{\text{AS}}[t] = E_{F,\text{AS}}[t] + E_{\text{loc,AS}}[t], \tag{18}$$

where $E_{F,\text{AS}}[t]$ denotes the flight energy of the assisted UAV, and $E_{\text{loc,AS}}[t]$ represents the energy consumed for processing the offloaded tasks. These components follow the same formulation as those for data collection UAVs, but with parameters tailored to the assisted UAV's enhanced capabilities.

E. Problem Formulation

In this paper, we present a joint optimization strategy for coordinating the flight paths and task offloading decisions of a proposed multi-tier UAV system. The goal of this study is to increase collaboration efficiency in complex mission situations. The optimization focuses on two important goals: (1) ensuring timely task completion by maximizing effective area coverage during the mission duration, and (2) reducing total system energy consumption to improve resource usage.

To achieve these goals, we formulate the problem as a joint optimization of trajectory planning, task assignment, and offloading strategies. The task completion reward and coverage reward are used to quantify performance. These two reward components will be formally defined later in the *markov decision process* (MDP) formulation, specifically in (27) and (28). The energy term accounts for all energy consumed by data collection UAVs and the assisted UAV, including flight, computation, and communication. Based on this formulation, the optimization problem is formulated as follows

$$\begin{aligned} \max_{a_i[t], q_i[t], q_{\text{AS}}[t]} \sum_{i=1}^{I} \sum_{t=1}^{T} & \left(R_{\text{a}}(i,t) + \gamma R_{\text{c}}(i,t)\right) \\ & - \lambda \left(E_i[t] + E_{\text{AS}}[t]\right) \end{aligned} \tag{19}$$

$$\text{s.t.} \quad 0 \le a_i[t] \le 1, \quad \forall i \in \mathcal{I},\, t \in \mathcal{T} \tag{19a}$$

$$\mathbf{q}_i[t] \ne \mathbf{q}_j[t], \quad \forall i, j \in \mathcal{I},\, i \ne j,\, t \in \mathcal{T} \tag{19b}$$

$$\mathbf{q}_i[t] \notin \mathcal{F}, \quad \forall i \in \mathcal{I},\, t \in \mathcal{T} \tag{19c}$$

$$0 \le x_i[t] \le L,\ 0 \le y_i[t] \le W, \quad \forall i \in \mathcal{I},\, t \in \mathcal{T} \tag{19d}$$

$$h_{\min} \le h_i[t] \le h_{\max}, \quad \forall i \in \mathcal{I},\, t \in \mathcal{T} \tag{19e}$$

$$v_i[t] \le v_{\max}, \quad \forall i \in \mathcal{I},\, t \in \mathcal{T} \tag{19f}$$

$$V_i[t] \le V_{\max}, \quad \forall i \in \mathcal{I},\, t \in \mathcal{T} \tag{19g}$$

$$E_{r,i}[t] \ge E_{\min}, \quad \forall i \in \mathcal{I},\, t \in \mathcal{T} \tag{19h}$$

$$t_c \le t_{\max}, \tag{19i}$$

where the coefficient γ adjusts the weight of the coverage reward. The coefficient λ is the penalty for energy consumption.

Constraint (19a) ensures that the task offloading ratio for each UAV remains within the valid range. Constraint (19b) ensures that no two UAVs occupy the same position at the same time, thus avoiding collisions. Constraint (19c) prevents UAVs from entering the no-fly zones $\mathcal{F}$, ensuring safe operations. Constraint (19d) keeps all UAVs within the defined rectangular area during flight. Constraint (19e) restricts UAV altitudes between $h_{\min}$ and $h_{\max}$ for safety and task accuracy. Constraint (19f) limits UAV speeds to a maximum value $v_{\max}$ to prevent unstable motion. Constraint (19g) restricts the vertical takeoff and landing speed of UAVs to ensure safe altitude transitions. Constraint (19h) restricts the UAV's energy level to stay above the safe threshold $E_{\min}$, ensuring continuous operation. Constraint (19i) ensures the mission is completed within the allowed time budget $t_{\max}$.

It is important to note that the formulated problem is a mixed-integer nonlinear program. The decision variables are the continuous offloading ratio $a_i[t]$, and the discrete UAV positions $q_i[t]$ and $q_{\text{AS}}[t]$. The objective function includes nonlinear components from energy models as well as communication terms based on UAV trajectory. These characteristics make the problem non-convex and challenging to solve using typical optimization approaches. To solve this challenge, we use a DRL-based approach that learns effective decision-making mechanisms by interacting with the environment.

IV. PPO-BASED ALGORITHM

In this section, the DCOPPO algorithm is proposed to solve the joint trajectory planning and task offloading problem in multi-tier UAV system under energy, communication, and time restrictions.

A. MDP Formulation

In this system, multiple UAVs jointly optimize their trajectories and task offloading strategies. The problem involves multiple UAVs and dynamic mission objectives. These characteristics make it suitable to model the system as a *markov decision process* (MDP). An MDP typically consists of three key components: the state space $\mathcal{S}$, the action space $\mathcal{A}$, and the reward function $\mathcal{R}$. The state space describes the environment status and task progress at each time step. The action space defines the available decisions for each UAV, such as movement directions and offloading choices. The reward function quantifies the immediate feedback based on task completion and resource consumption.

At each time slot, the system first observes the current state. Then, each UAV selects an action based on its current policy, such as moving, offloading a task, or executing local computation. After action execution, the system updates the state and assigns a reward to each UAV. This process repeats over time, allowing each UAV to improve its policy through interaction with the environment and gradually approach optimal performance. Based on this framework, each UAV can learn effective strategies by interacting with the environment and updating its policy over time. The state space, action space, and reward function of the system are defined as follows.

1) State Space $\mathcal{S}$: The state space captures key system information to support strategy learning and optimization. To this end, we define separate state representations for the data collection UAVs and the assisted UAV. These include position, remaining energy, task processing status, and motion state. The state space of U_i is represented as s_t, which includes the following sets

$$s_t = \{\{\mathbf{q}_i[t]\}, \{E_{r,i}[t]\}, \{T_i[t]\}, \{\psi_i[t]\} \mid i = 1, 2, ..., I\}, \tag{20}$$

where $\{\mathbf{q}_i[t]\}$ is the set of positions of U_i, $\{E_{r,i}[t]\}$ is the set of remaining energy of U_i, $\{T_i[t]\}$ is the set of task amounts executed by U_i at time t, $\{\psi_i[t]\}$ is the set of motion states of U_i.

The state space of the assisted UAV is represented as $s_{AS,t}$, which includes the following sets

$$s_{AS,t} = \{\{\mathbf{q}_{AS}[t]\}, \{E_{r,AS}[t]\}, \{T_{AS}[t]\}, \{\psi_{AS}[t]\}\}, \tag{21}$$

where $\{\mathbf{q}_{AS}[t]\}$ is the set of positions of the U_{AS}, $\{E_{r,AS}[t]\}$ is the set of remaining energy of the U_{AS}, $\{T_{AS}[t]\}$ is the set of task amounts handled by the U_{AS} at time t, $\{\psi_{AS}[t]\}$ is the set of motion states of the U_{AS}.

2) Action Space $\mathcal{A}$: The modeling of the action space should reflect not only the UAVs' movement decisions in space, but also their strategies for processing computation tasks. Therefore, the action space of a data collection UAV is defined as follows

$$\mathcal{A} = \left\{ \underbrace{\begin{bmatrix}0\\0\\0\end{bmatrix}}_{\text{hover}}, \underbrace{\begin{bmatrix}l\\0\\0\end{bmatrix}}_{\text{east}}, \underbrace{\begin{bmatrix}0\\l\\0\end{bmatrix}}_{\text{north}}, \underbrace{\begin{bmatrix}-l\\0\\0\end{bmatrix}}_{\text{west}}, \underbrace{\begin{bmatrix}0\\-l\\0\end{bmatrix}}_{\text{south}}, \underbrace{\begin{bmatrix}0\\0\\-h_i\end{bmatrix}}_{\text{land}}, \underbrace{[\alpha]}_{\text{offload ratio}} \right\}, \tag{22}$$

where l denotes the side length of each grid cell, which determines the UAV's movement step size in the horizontal plane.

And then, the action space of the assisted UAV is defined as follows

$$\mathcal{A}_{AS} = \left\{ \underbrace{\begin{bmatrix}0\\0\\0\end{bmatrix}}_{\text{hover}}, \underbrace{\begin{bmatrix}l\\0\\0\end{bmatrix}}_{\text{east}}, \underbrace{\begin{bmatrix}0\\l\\0\end{bmatrix}}_{\text{north}}, \underbrace{\begin{bmatrix}-l\\0\\0\end{bmatrix}}_{\text{west}}, \underbrace{\begin{bmatrix}0\\-l\\0\end{bmatrix}}_{\text{south}}, \underbrace{\begin{bmatrix}0\\0\\-h_{AS}\end{bmatrix}}_{\text{land}} \right\}. \tag{23}$$

3) Safety Controller: In data collection missions, UAV collisions with obstacles or other UAVs during flight can lead to mission failure, equipment damage, and resource waste. Therefore, it is essential to introduce a safety control mechanism. It helps filter out high-risk actions and ensures system stability and reliability. Compared to existing rule-based methods [42], the safety controller proposed in this work is more intelligent and flexible. It evaluates the safety of candidate actions by considering multiple risk probabilities, including collision risk, energy depletion, and environmental hazards. In addition, its risk weights and safety thresholds are adjustable, making it adaptable to diverse mission requirements and operational

strategies. Therefore, the adjusted UAV action $a_{s,i}[t]$ is defined as follows

$$\mathbf{a}_{s,i}[t]\begin{cases}[0,0,0]^T, & \Gamma_i[t] > \Gamma_{\text{th}} \\ [0,0,0]^T, & \mathbf{q}_i[t] + \mathbf{a}_i[t] \in \mathcal{F} \\ [0,0,0]^T, & \mathbf{q}_i[t] + \mathbf{a}_i[t] = \mathbf{q}_j[t],\ \forall j \neq i \\ [0,0,-h]^T, & E_{r,i}[t+1] \leq E_l \\ \mathbf{a}_i[t], & \text{otherwise}\end{cases} \tag{24}$$

where Γ_{th} is the risk threshold used to filter unsafe actions, E_l is the minimum energy level required to ensure safe return or emergency landing. A dynamic risk score $\Gamma_i[t]$ is introduced to perform real-time risk evaluation

$$\Gamma_i[t] = \omega_{\text{c}} P_{\text{c}}[t] + \omega_{\text{e}} P_{\text{e}}[t] + \omega_{\text{en}} P_{\text{en}}[t], \tag{25}$$

where $P_{\text{c}}[t]$ is the probability of U_i colliding with other UAVs, $P_{\text{e}}[t]$ is the probability of U_i running out of energy, $P_{\text{en}}[t]$ is the probability of UAV entering a hazardous area, $\omega_{\text{c}}, \omega_{\text{e}}, \omega_{\text{en}}$represent adjustable weights for risk factors.

4) Reward Function $\mathcal{R}$: In this system, UAVs complete area coverage and task offloading within limited time and energy budgets. Therefore, the reward function should reflect a balance between task performance and resource consumption. In addition, to adapt to complex and dynamic environments, the reward design must ensure both real-time responsiveness and scalability. Based on these considerations, we design the following reward function to quantify the utility of UAV system's behavior at every time step. The reward function consists of three components: task completion reward R_{a}, grid coverage reward R_{c}, and energy consumption penalty R_{e}. The overall expression is given by

$$R = R_{\text{a}} + R_{\text{c}} + R_{\text{e}}. \tag{26}$$

The task completion reward R_{a} encourages UAVs to complete all tasks in mission area as quickly as possible

$$R_{\text{a}} = \begin{cases} R_{\text{p}}, & \text{if all tasks in the grid are completed,} \\ 0, & \text{otherwise.}\end{cases} \tag{27}$$

where R_{p} represents the fixed reward granted upon completing all tasks.

The grid coverage reward R_{c} encourages data collection UAVs to maximize the coverage of grid cells throughout the mission duration. This design ensures reward uniqueness. A grid cell is rewarded only once when it is visited for the first time and data collection is successful. The grid coverage reward R_{c} can be expressed as

$$R_{\text{c}} = \gamma \sum_{t=1}^{T} \sum_{i=1}^{I} \mathbf{1}_{\text{g}}(i,t), \tag{28}$$

where γ is the reward coefficient for each newly covered grid cell. The indicator function $\mathbf{1}_{\text{g}}(i,t)$ returns 1 only if the cell is uncovered before and is now successfully covered by U_i at time t; otherwise, it returns 0.

Energy consumption is an important factor affecting mission efficiency. An energy consumption penalty is included to dynamically account for energy usage

$$R_{\text{e}} = -\left(\sum_{t=1}^{T} \sum_{i=1}^{I} E_i[t] + E_{\text{AS}}[t] \right). \tag{29}$$

B. PPO-Based Algorithm

To efficiently solve the joint optimization problem of trajectory planning and task offloading in multi-UAV systems, we propose a DRL algorithm named DCOPPO. In this framework, we adopt proximal policy optimization (PPO) as the policy update method. Compared to traditional policy gradient algorithms, PPO achieves faster convergence and better generalization, making it well suited for dynamic and constrained UAV collaboration environments. By combining the actor-critic architecture with PPO's surrogate objective, DCOPPO enables effective and stable policy improvement. The overall framework of the DCOPPO algorithm is depicted in Fig. 2.

The actor and critic networks are the key components of this algorithm. The actor network generates the current policy π_{θ_k}, which outputs an action based on the observed state s_t. The critic network estimates the state value $V_\theta(s_t)$, which serves as a baseline to evaluate the expected long-term reward. This value helps compute the advantage function used for policy updates. The critic's feedback guides the actor in adjusting π_θ toward better-performing actions.

Based on the interaction between the actor and critic network, the PPO algorithm updates the policy by optimizing a surrogate objective. Its goal is to improve task rewards while keeping policy changes stable and controlled. The surrogate objective function is defined as [43]

$$L_{\text{PPO}}(\theta) = \hat{E}_t \left[\min \left(r_t(\theta)\hat{A}_t^\lambda, \text{clip}(r_t(\theta), 1-\delta, 1+\delta)\hat{A}_t^\lambda \right) \right], \tag{30}$$

where $r_t(\theta)$ is the ratio between the current policy and the old policy, $\hat{A}_t^\lambda$ is the generalized advantage estimate at time t, and δ is the clipping threshold that limits the size of each policy update. This clipping mechanism helps PPO avoid overly large updates and improves training stability. The clip function is defined as

$$\text{clip}(r, 1-\delta, 1+\delta) = \begin{cases} 1-\delta, & \text{if } r < 1-\delta, \\ r, & \text{if } 1-\delta \leq r \leq 1+\delta, \\ 1+\delta, & \text{if } r > 1+\delta, \end{cases} \tag{31}$$

which constrains the policy ratio $r_t(\theta)$ within a trust region. It prevents the policy from changing too aggressively in a single update step.

The advantage function is used to evaluate the superiority of an action in a given state. PPO uses generalized advantage estimation (GAE) to reduce variance and improve learning efficiency. GAE calculates the advantage function by accumulating weighted temporal difference (TD) errors over multiple time steps, thus reducing variance while maintaining low bias. The advantage function is defined as

$$\hat{A}_t^\lambda = \sum_{l=0}^{N-1} (\gamma\lambda)^l \delta_{t+l}, \tag{32}$$

where N is the number of time steps in a minibatch, $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$ is the TD error, γ is the discount factor, and λ is the smoothing parameter of GAE. By using GAE, PPO effectively reduces variance in the advantage estimate, thus improving the learning efficiency of the policy.

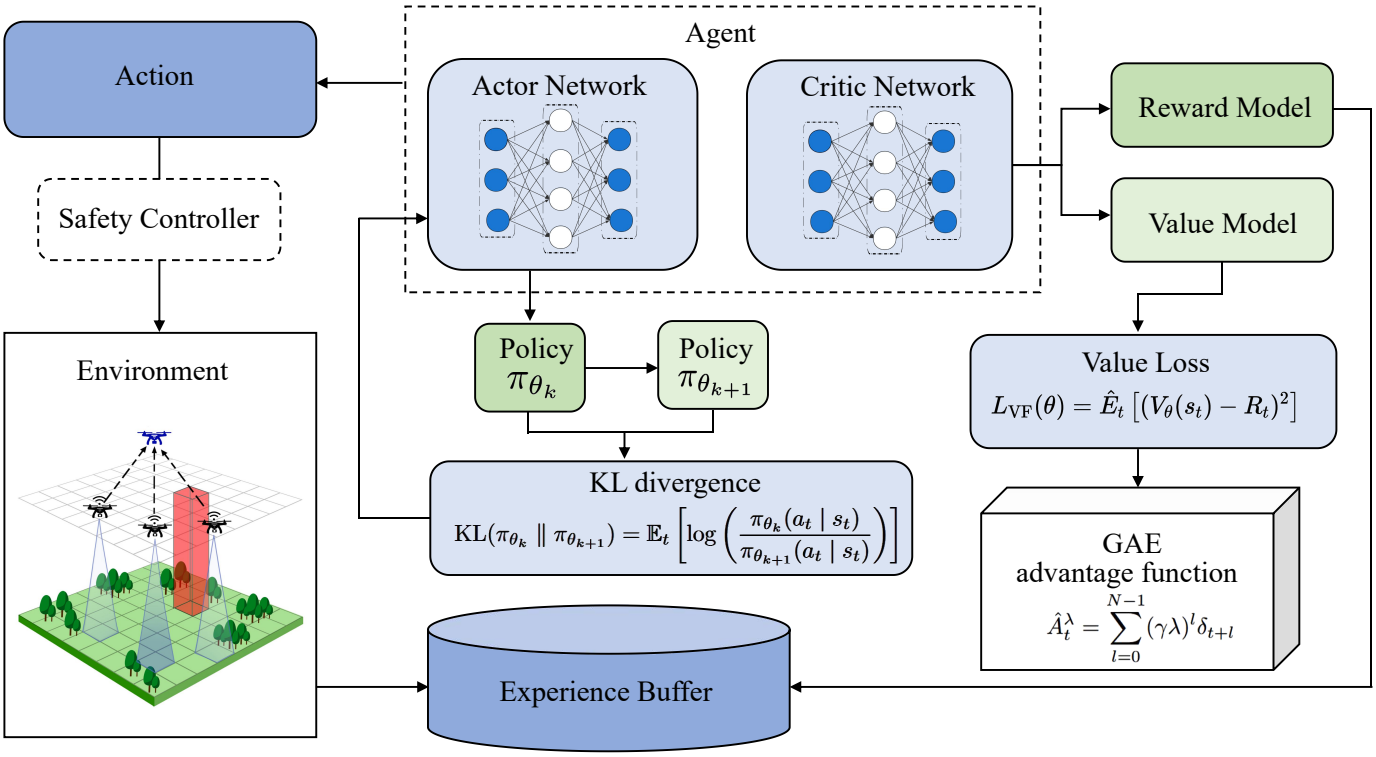


Fig. 2: Framework of the DCOPPO algorithm.

In our framework, policy updates are central to improving UAV coordination. By optimizing the surrogate objective function L_{PPO}, the policy adapts to better handle trajectory planning and task offloading decisions. The policy update is performed by the following formula [44]

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta L_{\mathrm{PPO}}(\theta_k), \tag{33}$$

where θ_k represents the current policy parameter, α is the learning rate that controls the step size for each update. By performing gradient ascent on the surrogate objective function, PPO gradually improves the policy, allowing the multi-UAV system to perform more efficiently and stably in complex task environments.

The value function loss L_{VF} is used to optimize the state value function, further improving the quality of the policy. This loss function quantifies the mismatch between the estimated return $V_\theta(s_t)$ of a UAV system state s_t and the actual accumulated mission reward. The value function loss is defined as

$$L_{\mathrm{VF}}(\theta) = \hat{E}_t\left[(V_\theta(s_t) - R_t)^2\right], \tag{34}$$

where $V_\theta(s_t)$ is the state value function under the current policy, and R_t is the return starting from time step t, typically the accumulated reward. By optimizing the value function loss, PPO improves the fitting of the state value function, which further enhances the policy's performance.

To ensure the stability of the training process, PPO introduces kullback–leibler (KL) divergence to limit the magnitude of each policy update. KL divergence measures the difference between the current and old policies, helping to control the scope of policy updates. The formula for KL divergence is

$$\mathrm{KL}(\pi_{\theta_k} \parallel \pi_{\theta_{k+1}}) = \mathbb{E}_t\left[\log\left(\frac{\pi_{\theta_k}(a_t \mid s_t)}{\pi_{\theta_{k+1}}(a_t \mid s_t)}\right)\right]. \tag{35}$$

By introducing KL divergence, PPO ensures that the policy change is not too large at each update, thereby enhancing the stability of the training process. Alg. 1 summarizes the training process of the PPO algorithm.

V. Numerical Results

A. Simulation Parameter Settings

This section describes the simulation setup and results adopted to evaluate the proposed DCOPPO algorithm. A UAV flight simulation environment is built on Python and employs a DRL framework. A three-layer fully connected neural network is built, with nodes in each layer [256, 256, 256]. The policy network and value network have identical architectures, and the ReLU function is utilized for activation. The discount factor is set to $\gamma_u = 0.98$, the learning rate is set to $lr = 0.001$, and the PPO clipping parameter is set to 0.2.

A $140\,\mathrm{m} \times 140\,\mathrm{m}$ rectangular target area is simulated and discretized into uniform square grid cells, each with a side length of $20\,\mathrm{m}$. The safety controller monitors each UAV's behavior to ensure that they are flying normally in the region. If an action does not match safety requirements (for example, putting the UAV into a no-fly zone or colliding with another UAV), it is denied and a hover action is used to ensure safety. The data collection UAVs take off from a fixed point at (0, 0). The assisted UAV starts from the center of the map at

Algorithm 1 PPO-Based DCOPPO Algorithm

Require: Initialize actor network θ and critic network ω;
Initialize: Construct the policy network and value network.

For each episode episode $= 1$ to N_{episodes}:
  Reset task matrix and UAV positions.
for each data collection UAV = 1 to I **do**
  Observe state $s_t = [q_i, E_{r,i}, T_i, \psi_i]$.
  Generate action $\mathcal{A}_t$.
end for
for each assisted UAV **do**
  Observe state $s_{AS,t} = [q_{\text{AS}}, E_{\text{r,AS}}, T_{\text{AS}}, \psi_{\text{AS}}]$.
  Generate action $\mathcal{A}_{AS}$.
end for
Compute composite reward R_t.
Optimization step:
Optimize $L_{\text{PPO}}(\theta)$ for actor network using R_t and $\hat{A}_t^{\lambda}$.
Compute critic loss $L_{\text{VF}}(\theta)$.
Update critic network by minimizing $L_{\text{VF}}(\theta)$.
if all tasks are completed **then**
  Exit current episode.
end if
if reward converges **then**
  Terminate training early.
end if

TABLE II: LIST OF SIMULATION PARAMETERS

Variable	Description	Value
I	Number of U_i	3
I_{AS}	Number of U_{AS}	1
h_i	Flight height of U_i	60 meters
h_{AS}	Flight height of U_{AS}	100 meters
l	Length of each grid cell in the map	20 meters
δ	Time slot duration	2 seconds
c	CPU cycles required to process data per bit	10^3 cycles/bit
f_i	CPU frequency of data collection UAVs	3×10^9 cycles/s
k_i	Chip effective capacitance factor	10^{-27}
B	Communication bandwidth of UAVs	10^7 Hz
U_{tip}	Tip speed of UAV rotor	60 m/s
d_0	Aircraft body drag ratio	0.5017
s	Solidity of rotor	0.0832
r	Rotor radius of UAV	0.3 meters

point (70,70). The experimental setting contains a no-fly area cube placed below 100m height, with its center at coordinates (90, 90). Data collection UAVs adhere to safety controller principles to avoid entering this area. The DCOPPO method divides each simulation period into time slots of predetermined lengths. Other parameters in the simulation are summarized in Table II, according to previous works [45]–[47].

To verify the effectiveness of the proposed method, we compare DCOPPO with three baseline algorithms:

1) **Advantage Actor-Critic (A2C)**: This approach is a reinforcement learning algorithm that uses the actor-critic architecture. A2C computes updates directly using the policy gradient and value function, reducing variance by calculating the advantage function [48], [49]. A2C is a stable policy gradient approach capable of directly optimizing policies and effectively handling both discrete and continuous action spaces.

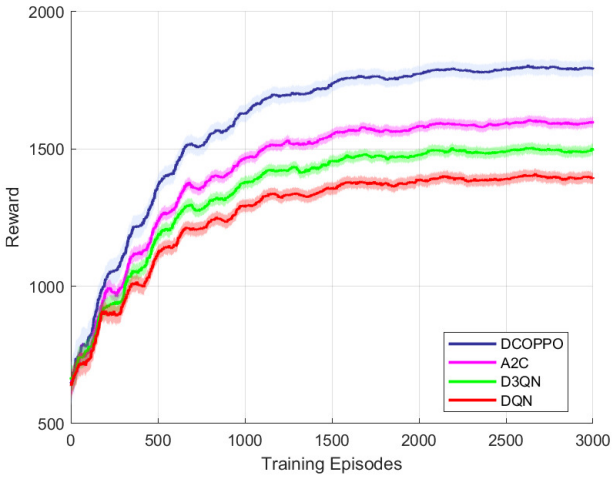


Fig. 3: Convergence performances.

2) **Deep Q-Network (DQN)**: A traditional DRL approach that blends Q-learning and deep neural networks. DQN uses deep neural networks to approximate the Q-value function, making it useful for challenges with discrete action spaces [50].
3) **Double Deep Q-Network (D3QN)**: This approach improves on the classic DQN. D3QN uses a second Q-network to calculate target Q-values, leading to more accurate estimations and improved learning stability and convergence [51].

The task objective is to cover the target area while minimizing energy consumption, under the condition of satisfying constraints. The goal is to achieve cooperative operation of UAVs in dynamic environments while maintaining maximum task efficiency. Performance is measured using parameters such convergence speed, area coverage, revisit frequency, and energy efficiency.

B. Convergence Performance

Fig. 3 shows the average reward values over multiple periods for the DCOPPO algorithm and the baseline methods. We compare the convergence speeds of four algorithms. In the training of 3000 episodes, significant differences in convergence speed and final reward values were observed. DCOPPO and A2C demonstrated faster convergence during training. In contrast, the reward values of D3QN and DQN exhibited significant fluctuations, especially in the early stages of training. In terms of final rewards, DCOPPO achieved the highest reward value at the end of training. These results indicate that DCOPPO and A2C provide superior task performance under extended training conditions through faster convergence and higher final rewards.

C. Performance Evaluation

Fig. 4 shows the flight trajectories of data collection UAVs under different algorithms. The trajectory based on DCOPPO demonstrates a more optimized path, enabling the UAV to

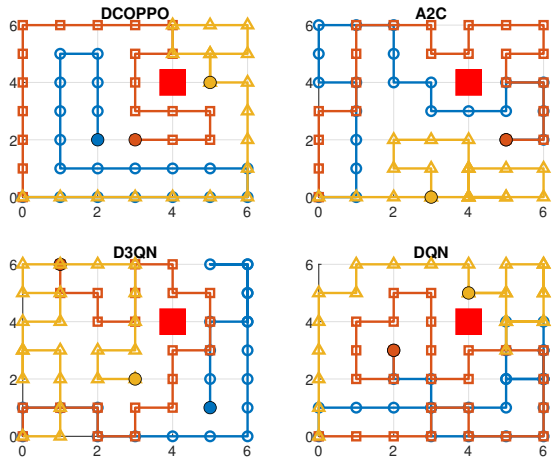


Fig. 4: Trajectories of data collection UAVs under different algorithms.

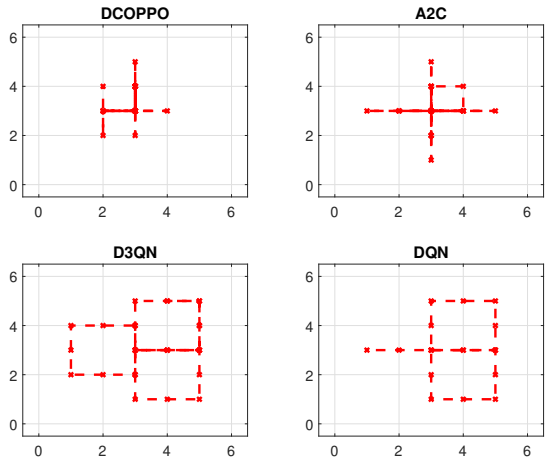


Fig. 5: Trajectories of assisted UAV under different algorithms.

cover the task area more effectively while avoiding unnecessary detours. Although D3QN completes the coverage task, its trajectory shows greater irregularity and redundant movements. Similarly, A2C and DQN also exhibit less efficient path planning.

Fig. 5 presents the trajectory of the assisted UAV. Clearly, the path of the assisted UAV is more flexible, able to adjust its position based on real-time task changes. More importantly, the trajectory of the assisted UAV highlights its crucial role in a multi-UAV system: effectively assisting with data processing and task coordination, and adjusting its position according to task objectives. This design makes the assisted UAV more flexible and adaptable. It plays a key role in improving overall task efficiency in complex environments.

Fig. 6 demonstrates the coverage performance in the simulation area under different algorithms. DCOPPO achieves high coverage with fewer redundant passes. This is made possible by precise path optimization, which improves the overall efficiency of UAV resource usage in complex environments.

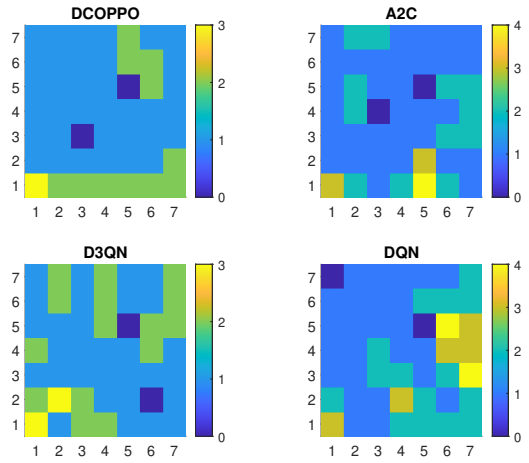


Fig. 6: Heatmaps of UAV coverage performance under different algorithms.

D3QN also achieves full coverage and performs well in certain subregions, but its overall efficiency lags behind DCOPPO. The non-optimality of its path planning leads to wasted energy and time, especially in areas at the edges of the region or areas with excessive coverage. In contrast, A2C and DQN exhibit noticeably lower coverage rates and frequent redundant coverage behaviors. Over extended task durations, redundant movements directly affect the efficient utilization of resources.

Fig. 7 plots the change in coverage rate over time for different algorithms. In the early stages of training, DQN and A2C show faster initial coverage than DCOPPO and D3QN. This may be due to their stronger exploratory behavior in the early episodes, allowing them to cover certain areas more quickly. However, excessive exploration may cause instability in path planning during later stages. This often prevents the policy from converging and completing tasks effectively under complex conditions. These algorithms fail to reach the same global coverage level as DCOPPO in the later stages. As a result, DQN and A2C reach a final coverage rate of 97.96%, while DCOPPO and D3QN achieve full coverage at 100%.

Fig. 8 shows the average offloading ratio of data collection UAVs over time. DCOPPO adjusts the offloading behavior dynamically throughout the mission. In the early phase, some data collection UAVs carry heavier workloads and tend to offload more. This helps improve response time and reduce local computation burden. In the later stage, the ratio drops to save energy for the assisted UAV. In contrast, DQN shows unstable patterns with large fluctuations. A2C keeps the ratio low throughout, which may increase the local load. These results demonstrate that DCOPPO achieves more adaptive and intelligent offloading control over time. It also leads to better energy balance and task responsiveness.

Fig. 9 shows the cumulative energy consumption of the data collection UAVs over time under the four algorithms. DCOPPO demonstrates superior energy control, with significantly lower consumption than the baselines. By the end of the mission, DCOPPO consumes approximately 382 J, while

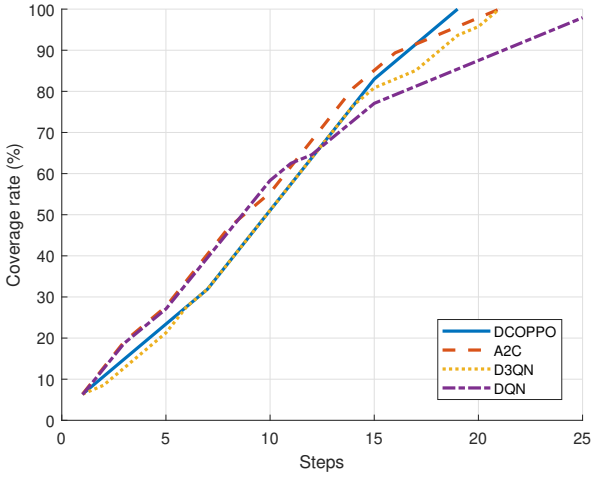


Fig. 7: Coverage rate progression of UAVs under different algorithms.

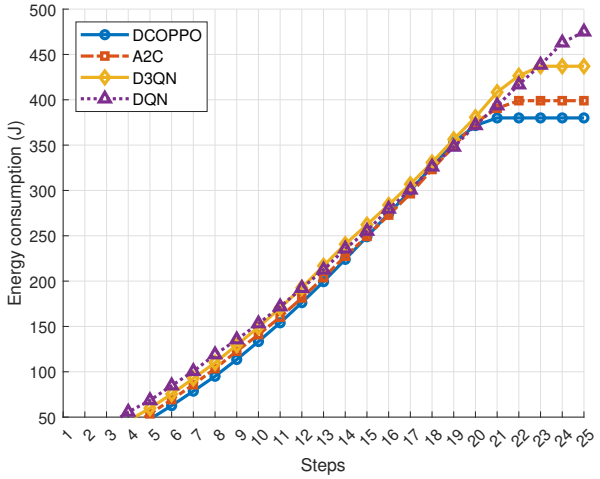


Fig. 9: Cumulative energy consumption of data collection UAVs over time.

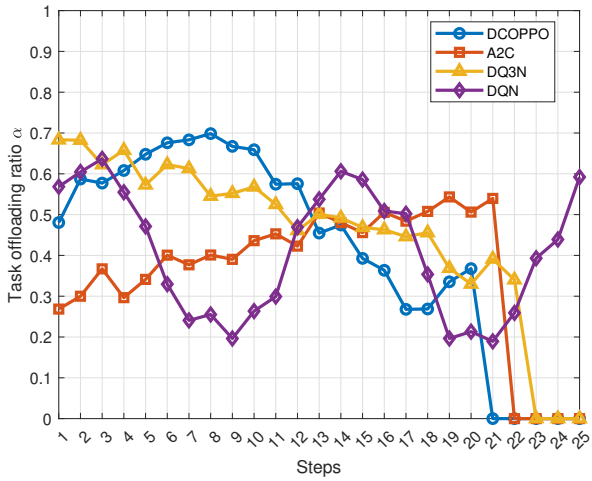


Fig. 8: Offloading ratio of data collection UAVs over time.

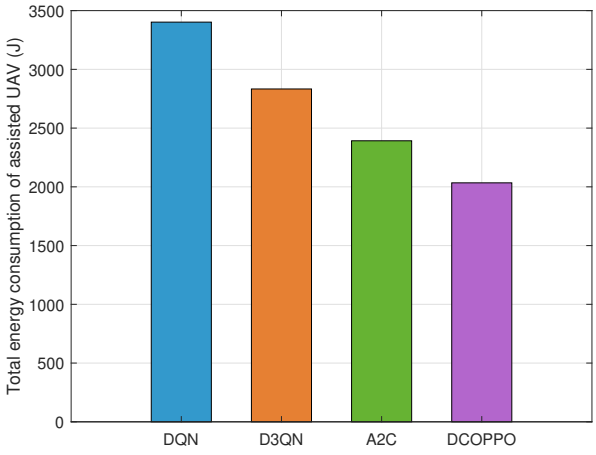


Fig. 10: Total energy consumption of the assisted UAV under different algorithms.

A2C and D3QN consume about 398 J and 443 J, respectively. DQN records the highest usage at 472 J. Compared to DQN, DCOPPO reduces energy consumption by around 19.07%, and achieves savings of 4.02% and 13.77% compared to A2C and D3QN, respectively. A2C performs moderately, lying between DCOPPO and D3QN in terms of energy efficiency but shows less consistent performance. These results indicate that DCOPPO helps data collection UAVs avoid redundant movements and improves their operational efficiency. It balances task execution and energy usage more effectively, which is critical for long-duration and resource-constrained UAV missions.

Fig. 10 shows the total energy consumption of the assisted UAV under four different algorithms. DCOPPO achieves the lowest energy usage, around 2040 J. In contrast, DQN leads to the highest consumption, exceeding 3400 J. D3QN and A2C fall in between, with approximately 2830 J and 2390 J, respectively. Compared to DQN, DCOPPO reduces the assisted UAV's energy consumption by about 40%. This result highlights that DCOPPO manages task offloading efficiently and avoids unstable delegation to the assisted UAV. As a result, it reduces the UAV's energy burden and improves energy balance across the system, enabling longer operation under limited resources.

Fig. 11 shows how the average energy consumption changes with different numbers of data collection UAVs. From 2 to 5 UAVs, all algorithms show a downward trend in energy usage. This suggests that more UAVs help share the workload and reduce individual consumption. However, when the number increases to 6 or 7, the energy cost rises again for most methods. This may result from higher coordination complexity or overlapping trajectories. Among all methods, DCOPPO maintains the lowest energy consumption across all UAV counts. It reaches its minimum at 5 UAVs, around 340 J, clearly outperforming A2C, D3QN, and DQN. In contrast, DQN shows the highest energy use, especially at 7 UAVs. These results confirm that DCOPPO adapts better to larger-

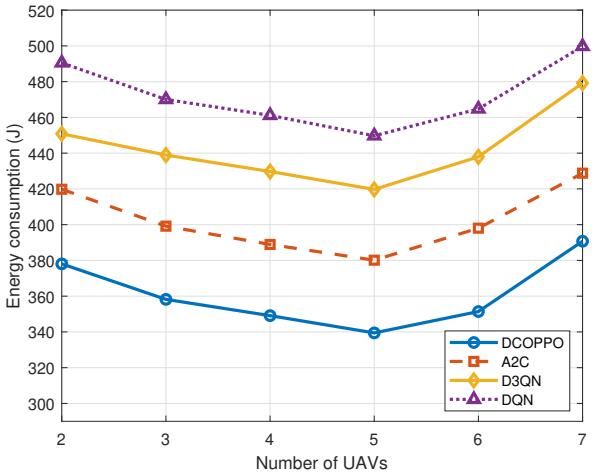


Fig. 11: Comparison of energy consumption with different numbers of data collection UAVs.

scale coordination. It balances resources more effectively and remains energy-efficient as task size grows.

VI. Conclusion

This paper presented a heterogeneous multi-tier UAV system to enhance data collection and task offloading in time-sensitive missions. The proposed system improved coordination efficiency by using an assisted UAV with higher computing power, which reduced onboard load and supported real-time processing. We formulated a joint optimization problem to maximize task performance while minimizing energy consumption. The model considered UAV trajectories, task offloading ratios, energy limits, communication constraints, and mission deadlines. To address this problem in dynamic environments, we proposed a DRL-based algorithm called DCOPPO. Simulation results showed that our method achieved full coverage and consumed less energy than baseline approaches.

While the current framework focuses on area coverage and data collection, it is not limited to these tasks. By redesigning the reward function, the method can be extended to other scenarios such as search and rescue, target tracking, or information relay. Energy efficiency may vary under different task types, especially those involving prolonged hovering or frequent data transmission. In addition, altitude control may also influence system performance in some cases. Exploring these extensions is a valuable direction for future work.

References

[1] C. Graham, I. O'Connor, L. Broderick, M. Broderick, O. Jensen, and H. Lally, "Drones can reliably, accurately and with high levels of precision, collect large volume water samples and physio-chemical data from lakes," *Sci. Total Environ.*, vol. 824, p. 153875, 2022.

[2] P. Velusamy, S. Rajendran, R. K. Mahendran, S. Naseer, M. Shafiq, and J. Choi, "Unmanned aerial vehicles (UAV) in precision agriculture: Applications and challenges," *Energies*, vol. 15, no. 1, pp. 217–235, 2022.

[3] B. Ramesh, R. Callender, B. F. Zaitchik, M. Jagger, S. Swarup, and J. M. Gohlke, "Adverse health outcomes following hurricane harvey: A comparison of remotely-sensed and self-reported flood exposure estimates," *GeoHealth*, vol. 7, no. 4, p. Art. no. e2022GH000710, 2023.

[4] D. C. Schedl, I. Kurmi, and O. Bimber, "An autonomous drone for search and rescue in forests using airborne optical sectioning," *Sci. Robot.*, vol. 6, no. 55, p. eabg1188, 2021.

[5] Y.-S. Jiao, X.-M. Wang, H. Chen, and Y. Li, "Research on the coverage path planning of UAVs for polygon areas," in *Proc. 5th IEEE Conf. Ind. Electron. Appl. (ICIEA)*. IEEE, 2010, pp. 1467–1472.

[6] M. Torres, D. A. Pelta, J. L. Verdegay, and J. C. Torres, "Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction," *Expert Syst. Appl.*, vol. 55, pp. 441–451, 2016.

[7] T. M. Cabreira, C. Di Franco, P. R. Ferreira, and G. C. Buttazzo, "Energy-aware spiral coverage path planning for UAV photogrammetric applications," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3662–3668, 2018.

[8] Y. Chen, K. Li, Y. Wu, J. Huang, and L. Zhao, "Energy efficient task offloading and resource allocation in air-ground integrated MEC systems: A distributed online approach," *IEEE Trans. Mob. Comput.*, vol. 23, no. 8, pp. 8129–8142, 2023.

[9] L. He, G. Sun, Z. Sun, P. Wang, J. Li, S. Liang, and D. Niyato, "An online joint optimization approach for QoE maximization in UAV-enabled mobile edge computing," in *Proc. IEEE INFOCOM 2024*. IEEE, 2024, pp. 101–110.

[10] H. Yu, S. Leng, and F. Wu, "Joint cooperative computation offloading and trajectory optimization in heterogeneous UAV-swarm-enabled aerial edge computing networks," *IEEE Internet Things J.*, vol. 11, no. 10, pp. 17 700–17 711, 2024.

[11] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam, and L. Hanzo, "Multi-agent deep reinforcement learning-based trajectory planning for multi-UAV assisted mobile edge computing," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 1, pp. 73–84, 2020.

[12] H. Hao, C. Xu, W. Zhang, S. Yang, and G.-M. Muntean, "Joint task offloading, resource allocation, and trajectory design for multi-UAV cooperative edge computing with task priority," *IEEE Trans. Mob. Comput.*, vol. 23, no. 9, pp. 8649–8663, 2024.

[13] Z. Yang, S. Bi, and Y.-J. A. Zhang, "Dynamic offloading and trajectory control for UAV-enabled mobile edge computing system with energy harvesting devices," *IEEE Trans. Wireless Commun.*, vol. 21, no. 12, pp. 10 515–10 528, 2022.

[14] D. Wang, J. Tian, H. Zhang, and D. Wu, "Task offloading and trajectory scheduling for UAV -enabled MEC networks: An optimal transport theory perspective," *IEEE Wireless Commun. Lett.*, vol. 11, no. 1, pp. 150–154, 2021.

[15] Z. Hu, F. Zeng, Z. Xiao, B. Fu, H. Jiang, and H. Chen, "Computation efficiency maximization and QoE-provisioning in UAV-enabled MEC communication systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1630–1645, 2021.

[16] B. Xu, Z. Kuang, J. Gao, L. Zhao, and C. Wu, "Joint offloading decision and trajectory design for UAV -enabled edge computing with task dependency," *IEEE Trans. Wireless Commun.*, vol. 22, no. 8, pp. 5043–5055, 2022.

[17] L. Liu, A. Wang, G. Sun, and J. Li, "Maximizing data gathering and energy efficiency in UAV-assisted iot: A multi-objective optimization approach," *Comput. Netw.*, vol. 235, p. 109986, 2023.

[18] B. Liu, Y. Wan, F. Zhou, Q. Wu, and R. Q. Hu, "Resource allocation and trajectory design for miso UAV-assisted MEC networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4933–4948, 2022.

[19] X. Tang, H. Zhang, R. Zhang, D. Zhou, Y. Zhang, and Z. Han, "Robust trajectory and offloading for energy-efficient UAV edge computing in industrial internet of things," *IEEE Trans. Ind. Informat.*, vol. 20, no. 1, pp. 38–49, 2023.

[20] M. Samir, S. Sharafeddine, C. M. Assi, T. M. Nguyen, and A. Ghrayeb, "UAV trajectory planning for data collection from time-constrained iot devices," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 34–46, 2019.

[21] F. Guo, H. Zhang, H. Ji, X. Li, and V. C. Leung, "Joint trajectory and computation offloading optimization for UAV-assisted MEC with NOMA," in *Proc. IEEE INFOCOM Workshops*. IEEE, 2019, pp. 1–6.

[22] B. Suganya, R. Gopi, A. R. Kumar, and G. Singh, "Dynamic task offloading edge-aware optimization framework for enhanced UAV operations on edge computing platform," *Sci. Rep.*, vol. 14, no. 1, p. 16383, 2024.

[23] X. Qi, J. Chong, Q. Zhang, and Z. Yang, "Collaborative computation offloading in the multi-UAV fleeted mobile edge computing network via

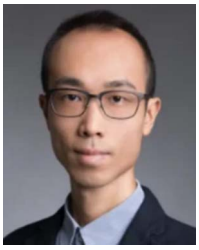
connected dominating set," *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 10 832–10 848, 2022.
[24] Y. Luo, W. Ding, and B. Zhang, "Optimization of task scheduling and dynamic service strategy for multi-UAV-enabled mobile-edge computing system," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 3, pp. 970–984, 2021.
[25] B. Ma, H. Kuang, S. Liu, and C. Li, "UAV assisted cellular network traffic offloading: Joint swarm, 3D deployment, and user allocation optimization based on a data-aware method," *Comput. Netw.*, vol. 231, p. 109812, 2023.
[26] J. Xiong, J. Li, J. Li, S. Kang, C. Liu, and C. Yang, "Probability-tuned market-based allocations for UAV swarms under unreliable observations," *IEEE Trans. Cybern.*, vol. 53, no. 11, pp. 6803–6814, 2022.
[27] Q. Liu, L. Shi, L. Sun, J. Li, M. Ding, and F. Shu, "Path planning for UAV-mounted mobile edge computing with deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5723–5728, 2020.
[28] N. Zhao, Z. Ye, Y. Pei, Y.-C. Liang, and D. Niyato, "Multi-agent deep reinforcement learning for task offloading in UAV-assisted mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 6949–6960, 2022.
[29] B. Li, R. Yang, L. Liu, J. Wang, N. Zhang, and M. Dong, "Robust computation offloading and trajectory optimization for multi-UAV-assisted MEC: A multiagent drl approach," *IEEE Internet Things J.*, vol. 11, no. 3, pp. 4775–4786, 2023.
[30] B. Khamidehi and E. S. Sousa, "Reinforcement-learning-aided safe planning for aerial robots to collect data in dynamic environments," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 13 901–13 912, 2022.
[31] F. Song, H. Xing, X. Wang, S. Luo, P. Dai, Z. Xiao, and B. Zhao, "Evolutionary multi-objective reinforcement learning based trajectory control and task offloading in UAV-assisted mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 22, no. 12, pp. 7387–7405, 2022.
[32] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3747–3760, 2017.
[33] M. M. U. Chowdhury, S. J. Maeng, E. Bulut, and I. Güvenç, "3-d trajectory optimization in UAV-assisted cellular networks considering antenna radiation pattern and backhaul constraint," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 5, pp. 3735–3750, 2020.
[34] H. Gong, B. Huang, B. Jia, and H. Dai, "Modeling power consumptions for multirotor UAVs," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 6, pp. 7409–7422, 2023.
[35] F. S. Abkenar, P. Ramezani, S. Iranmanesh, S. Murali, D. Chulerttiyawong, X. Wan, A. Jamalipour, and R. Raad, "A survey on mobility of edge computing networks in iot: State-of-the-art, architectures, and challenges," *IEEE Commun. Surv. Tutor.*, vol. 24, no. 4, pp. 2329–2365, 2022.
[36] S. Zhang, N. Chen, Z. Qian, J. Wu, and S. Lu, "Real-time proportional computation offloading via deep reinforcement learning," in *Proc. 25th IEEE Int. Conf. Parallel Distrib. Syst. (ICPADS)*. IEEE, 2019, pp. 414–421.
[37] Y. Matsubara and M. Levorato, "Neural compression and filtering for edge-assisted real-time object detection in challenged networks," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*. IEEE, 2021, pp. 2272–2279.
[38] S. Wan, S. Ding, C. Chen, J. Yang, Q. Liu, and Y. Mao, "Edge computing enabled video segmentation for real-time traffic monitoring in internet of vehicles," *Pattern Recognit.*, vol. 121, p. 108146, 2021.
[39] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, 2018.
[40] S. Bi and Y. J. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4177–4190, 2018.
[41] Y. Guan, S. Zou, H. Peng, W. Ni, Y. Sun, and H. Gao, "Cooperative UAV trajectory design for disaster area emergency communications: A multiagent ppo method," *IEEE Internet Things J.*, vol. 11, no. 5, pp. 8848–8859, 2023.
[42] H. Bayerlein, M. Theile, M. Caccamo, and D. Gesbert, "Multi-UAV path planning for wireless data harvesting with deep reinforcement learning," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 1171–1187, 2021.
[43] U. Saha, A. Jawad, S. Shahria, and A. H.-U. Rashid, "Proximal policy optimization-based reinforcement learning approach for dc-dc boost converter control: A comparative evaluation against traditional control techniques," *Heliyon*, vol. 10, no. 18, 2024.
[44] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
[45] M. Li, N. Cheng, J. Gao, Y. Wang, L. Zhao, and X. Shen, "Energy-efficient UAV-assisted mobile edge computing: Resource allocation and trajectory optimization," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3424–3438, 2020.
[46] K. Ravichandran, S. Ananthan, I. Chopra, and B. Hein, "Active rotor controls for vibration reduction and performance enhancement," in *Proc. AHS Aeromechanics Spec. Conf.*, San Francisco, CA, USA, 2010.
[47] Z. Yu, Y. Gong, S. Gong, and Y. Guo, "Joint task offloading and resource allocation in UAV-enabled mobile edge computing," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3147–3159, 2020.
[48] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*. PMLR, 2016, pp. 1928–1937.
[49] N. Zhang, J. Wang, and M. Xiao, "Deep reinforcement learning trading strategy based on lstm-a2c model," in *Proc. Int. Conf. Adv. Algorithms Neural Netw. (AANN)*, vol. 12285. SPIE, 2022, pp. 281–287.
[50] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[51] Z. Zhu, C. Hu, C. Zhu, Y. Zhu, and Y. Sheng, "An improved dueling deep double-q network based on prioritized experience replay for path planning of unmanned surface vehicles," *J. Mar. Sci. Eng.*, vol. 9, no. 11, p. 1267, 2021.

Jiayi Tang received the B.Eng. degree in transportation engineering from Beijing University of Technology, Beijing, China, in 2023. She is currently pursuing the M.Eng. degree in traffic information engineering and control at Beihang University, Beijing, China. Her research interests include unmanned systems and intelligent transportation systems.

Xuting Duan (Member, IEEE) received the Ph.D. degree in traffic information engineering and control from Beihang University, Beijing, China. He is currently a Professor with the School of Transportation Science and Engineering, Beihang University. His research interests are focused on the theories of V2X communications and collaborative intelligence and their systematic engineering applications in three-dimensional intelligent transportation systems, connected and autonomous vehicles.

Jianshan Zhou received the Ph.D. degree in traffic information engineering and control from Beihang University, Beijing, China. He is currently an Assistant Professor with the School of Transportation Science and Engineering, Beihang University. His current research interests include wireless communication, artificial intelligence systems, and intelligent transportation systems.

Kaige Qu (Member, IEEE) received the BS degree in communication engineering from Shandong University, Jinan, China, in 2013, theMS degree in integrated circuits engineering and electrical engineering from Tsinghua University, Beijing, China, and KU Leuven, Leuven, Belgium, in 2016, and the PhD degree in electrical and computer engineering from the University of Waterloo, Waterloo, Canada, in 2021. Since February 2021, she has been a postdoctoral fellow with the Department of Electrical and Computer Engineering, University of Waterloo. She is currently an associate professor with the school of transportation science and engineering with Beihang University. Her research interests include network slicing, edge intelligence, machine learning for wireless networks, connected autonomous vehicles, and digital twin assisted network automation.



Ivan Wang-Hei Ho (Senior Member, IEEE) received the BEng and MPhil degrees in information engineering from The Chinese University of Hong Kong, Hong Kong, in 2004 and 2006, respectively, and the PhD degree in electrical and electronic engineering from the Imperial College London, London, U.K., in 2010. He was a research intern with the IBM Thomas J. Watson Research Center, Hawthorne, NY, USA, and a postdoctoral research associate with the System Engineering Initiative, Imperial College London. In 2010, he co-founded P2 Mobile Technologies Ltd., where he was the chief research and development engineer. He is currently an associate professor with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong. His research interests include wireless communications and networking, specifically in vehicular networks, intelligent transportation systems, and Internet of Things (IoT). Dr. Ho primarily invented the MeshRanger series wireless mesh embedded system, which received the Silver Award in Best Ubiquitous Networking at the Hong Kong ICT Awards 2012. His work on indoor positioning and IoT also received the Gold Medal at the International Trade Fair Ideas and Inventions New Products (iENA) in Germany, in 2019, and the Gold Medal with the Organizer's Choice Award in the International Invention Innovation Competition in Canada (iCAN) in 2020. He is currently an associate editor for IEEE Transactions on Vehicular Technology, and IEEE Transactions on Consumer Electronics, and was the TPC co-chair for the PERSIST-IoT Workshop in conjunction with ACM MobiHoc 2019 and IEEE INFOCOM 2020.

Daxin Tian (Fellow, IEEE) received the Ph.D. degree in computer application technology from Jilin University, Changchun, China, in 2007. He is currently a Professor with the School of Transportation Science and Engineering, Beihang University, Beijing, China. His research interests include mobile computing, intelligent transportation systems, vehicular ad-hoc networks, and swarm intelligence. He is a member of the IEEE Intelligent Transportation Systems Society and the IEEE Vehicular Technology Society.