

Energy-Efficient Joint Task Offloading and 3D Trajectory Optimization for UAV-Assisted MEC Systems Over Uneven Terrain

Zhao Tong , Senior Member, IEEE, Shiyang Zhang , Jing Mei , Jiayi Sun , and Keqin Li , Fellow, IEEE

Abstract—With the rapid advancement and deep integration of the Internet of Things (IoT) and 5G technologies, mobile edge computing (MEC) has undertaken an increasingly important role in enhancing service quality. Leveraging their high mobility and flexible deployment, unmanned aerial vehicles (UAVs) extend MEC services to challenging environments such as mountainous areas. Nevertheless, UAVs have inherent limitations, including restricted onboard resources (e.g., energy and computing capacity) and the need for obstacle avoidance flight. In this work, which investigates a UAV-assisted MEC system with uneven terrain and dynamic service scenarios, these limitations bring additional challenges to system optimization. The incorporation of terrain information in high-dimensional state space, the continuous action space required for fine control, and the variable network demands under dynamic service scenarios complicate the non-convex optimization problem. By jointly designing UAV's trajectory and user equipments' (UEs) task allocation, we address the task offloading problem under safe flight conditions, aiming to maximize both service coverage ratio and UAV's propulsion energy efficiency. Then, we propose a phased hierarchical deep reinforcement learning (PH-DRL) algorithm, in which the network training is designed in phases and the network structure is organized hierarchically. Specifically, the phased method overcomes insufficient network experience in complex environments, while the hierarchical method decomposes the optimization variables, enabling independent solution. Experimental results demonstrate that the PH-DRL algorithm substantially improves service coverage ratio and propulsion energy efficiency, achieving system utility that significantly outperforms other comparative strategies.

Index Terms—Mobile edge computing, task offloading, trajectory optimization, uneven terrain, unmanned aerial vehicle.

I. INTRODUCTION

THE Internet of Things (IoT) connects billions of sensors, industrial terminals, and smart devices, enabling realtime perception and control of the physical environment. With the rollout of 5G and emerging beyond 5G technologies, the IoT is generating data traffic that grows exponentially. Many critical IoT services — such as smart manufacturing, intelligent transportation, and remote healthcare — demand ultralow latency, high reliability, and massive connectivity [1]. However, traditional cloud data centers are often located far from user equipment (UEs), resulting in long roundtrip delays that cannot satisfy these stringent quality of service (QoS) requirements. Mobile edge computing (MEC) addresses these challenges by pushing computation, storage, and networking functions from centralized clouds to edge nodes situated close to UEs (e.g., base stations (BS), routers, or dedicated edge servers) [2]. By processing tasks at the network edge, MEC can dramatically reduce end-to-end latency and improve system scalability. Nevertheless, fixed edge infrastructures are constrained by deployment cost, geographic coverage, and adaptability, which limit their effectiveness in remote regions or sudden demand surges such as disaster recovery.

Integrating unmanned aerial vehicles (UAVs) with MEC creates a UAV-assisted MEC paradigm that leverages the high mobility, rapid deployment, and wide coverage of aerial platforms [3]. UAV equipped with edge servers can hover over hotspots or cruise with predefined trajectories to provide on-demand computational resources, thereby filling coverage gaps and augmenting capacity where ground-based edge nodes are insufficient. The application of UAVs has shortened the ground-to-air (G2A), air-to-air (A2A), and air-to-ground (A2G) links, further reducing communication latency [4]. Moreover, UAV-assisted MEC is well suited for scenarios lacking permanent infrastructure—such as disaster zones, islands, or scarcely populated areas—offering temporary yet reliable services that sustain critical IoT operations. Coordinating UEs, BSs, and UAVs into a cohesive system can better utilize computing and communication resources to provide better services. Consequently, UAV-assisted MEC emerges as a key technology for enhancing the resilience, adaptability, and low-latency performance of next-generation IoT systems [5].

Received 27 January 2026; accepted 19 May 2026. Date of publication 22 May 2026; date of current version 2 September 2026. This work was supported by the Program of National Natural Science Foundation of China under Grant 62372172 and Grant 62072174, in part by the Distinguished Youth Science Foundation of Hunan Province, China under Grant 2023JJ10030, in part by the National Natural Science Foundation of Hunan Province, China under Grant 2022JJ40278, in part by the Research Foundation of Education Bureau of Hunan Province, China under Grant 23B0100, and in part by the Graduate Research Innovation Program of Hunan Province, China under Grant LXBZZ2024086. Recommended for acceptance by C.M. Pinotti. (Corresponding Author: Jing Mei.)

Zhao Tong, Shiyang Zhang, Jing Mei, and Jiayi Sun are with the College of Information Science and Engineering, Hunan Normal University, Hunan 410081, China (e-mail: tongzhao@hunnu.edu.cn; 202370294159@hunnu.edu.cn; jingmei1988@163.com; 2166871192@qq.com).

Keqin Li is with the College of Information Science and Engineering, Hunan University, Changsha 410082, China, also with the National Supercomputing Center in Changsha, Changsha 410082, China, and also with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Digital Object Identifier 10.1109/TMC.2026.3695882

In UAV-assisted MEC systems, key factors for performance improvement include, but are not limited to, optimizing: flight trajectories, task offloading, and energy control. Trajectory optimization focuses on how to plan the UAV's flight trajectory in a 2D or 3D environment. Researchers need to consider how to guide the UAV to avoid obstacles, ensure signal coverage, minimize service latency, or maximize UE's fairness [6]. Task offloading determines the computation distribution and communication resource scheduling of each UE's task across local, UAV, or BS. The offloading decision should account for task information, channel quality, and other factors to balance latency, energy consumption, fairness, and other objectives [7]. Energy control needs to address the limited onboard battery capacity of UAVs, where enhanced endurance allows UAVs to serve more UEs or for longer service durations. Therefore, effective trajectory and task scheduling strategies are needed to optimize UAV's energy efficiency [8]. Solving optimization problems with a single objective is often achievable in polynomial time, such as using convex optimization or graph search algorithms to find the flight trajectory that satisfies airspace constraints. However, when trajectory optimization, task offloading, and energy control are highly coupled, and the effects of terrain data as well as terrain-induced communication and flight blockages are taken into account, the solution space of the joint decision problem grows exponentially. This requires the design of a low-complexity, high-efficiency, and fast-implementing joint optimization framework.

To address these issues, we propose a phased hierarchical deep reinforcement learning (PH-DRL) algorithm that provides optimal offloading decisions and improves UAV energy efficiency while ensuring safe flight. The main contributions of this paper are as follows:

- We construct a UAV-assisted MEC system model that takes into account three practical aspects: UAV flight over uneven terrain, task offloading under dynamic signal coverage, and a refined UAV propulsion energy consumption model. These factors are interrelated in realistic scenarios—the terrain affects the UAV's flight and signal coverage, while both influence communication quality and energy consumption. This enhances the realism of the model, while also increasing the complexity of the optimization problem, thereby providing a more challenging and meaningful testbed for algorithm evaluation.
- We integrate hierarchical and phased methods into the DRL framework to solve the optimization problem. Specifically, the hierarchical method decomposes the optimization variables into different decision levels and reformulates the problem as a corresponding Markov decision process (MDP), effectively reducing dimensional redundancy. The phased training method divides the learning process into multiple stages, enabling the agent to accumulate experience quickly and stabilize convergence in a complex system.
- We design a dynamically invoked network architecture to tackle the challenge of neural network input/output dimensionality variation, which arises from the dynamically changing number of equipment within the UAV's

signal coverage. Specifically, the task offloading network is invoked according to the number of covered UEs at each time instant, improving the adaptability of the proposed algorithm.

- We incorporate real-world terrain data into our experiments. Numerical experiments demonstrate that the proposed algorithm outperforms other DRL baseline algorithms in terms of service coverage ratio and propulsive energy efficiency. Moreover, through extended experiments that validate the algorithm's robustness and applicability under varying UE numbers and changes in communication and computational resources.

The remainder of this paper is organized as follows: Section II reviews the related work in UAV-assisted MEC. Section III presents the system model and the problem formulation. In Section IV, the PH-DRL algorithm is introduced. Section V analyzes the performance of the proposed algorithm through experimental results. Lastly, Section VI concludes this paper.

II. RELATED WORK

In IoT sensor networks, utilizing UAVs to collect data from sensor nodes (SN) and transmit it to remote cloud centers can reduce system latency and energy consumption while ensuring sufficient data collection [9], [10], [11], [12]. Han et al. addressed the energy constraints of SNs during the data collection process. By employing block coordinate descent (BCD) and slack variable methods, they transformed the original non-convex problem into two subproblems and applied successive convex approximation (SCA) for iterative solutions, aiming to minimize the energy consumption of all SNs [9]. Li et al. proposed a novel data collection framework leveraging orthogonal frequency division multiple access (OFDMA) technology, enabling UAVs to simultaneously collect data from SNs. They modeled the problem as one of maximizing data collection, proved it to be NP-hard, and designed approximation and heuristic algorithms for its solution [10]. Liu et al. considered delay constraints of data, UAV energy consumption limitations, and interference mitigation for ground users. They achieved task completion time optimization through the joint optimization of BS and UAV association variables, UAV flight speed, and UAV trajectory [11]. Sun et al. studied UAVs with fixed coverage areas capable of receiving data from SNs within their range. By employing multiple subcarriers for data collection, they applied DRL to optimize flight trajectory control and coordination among multiple UAVs [12]. Zhao et al. focused on ensuring the connection quality between UAVs and BSs within cellular networks while minimizing flight energy consumption. They reconstructed an outage probability knowledge graph via the deep image prior (DIP) map reconstruction to perceive the environment and performed online optimization of UAV trajectories [13]. Xu et al. examined the joint optimization of task offloading decisions with task dependencies, UAV trajectory design, and resource allocation. They modeled the problem as a mixed integer programming problem and proposed an iterative solution method based on the BCD method [14]. Xie et al. investigated the problem of throughput

maximization in a UAV-assisted wireless powered communication network (WPCN). They proposed an efficient continuous hovering–flight trajectory design, and obtained a locally optimal solution through alternating optimization and successive convex programming (SCP) techniques [15]. However, these studies generally assume that UAVs operate within fixed altitudes or predefined grid points, without addressing 3D flight control or obstacle avoidance. In realistic mountainous environments with uneven terrain, UAVs must perform fine-grained altitude adjustments and path planning to ensure safe navigation and reliable connectivity, which significantly increases the complexity of trajectory design.

In the research of UAV-enabled computation offloading and resource management, Gupta et al. proposed a UAV replacement framework designed to ensure uninterrupted service in UAV-assisted wireless communication (UWC) systems, thereby addressing the limitation of UAV battery endurance. When a serving UAV's battery is depleted, a fully charged UAV departs from the charging station to replace it, ensuring continuous coverage for UEs [16]. Huang et al. employed UAVs as network relay nodes to assist data centers in information collection while considering obstacle-avoidance trajectories. They propose an improved shuffled frog-leaping algorithm (ISFLA) based on Dubins paths for solution [17]. Li et al. assumed UEs take heterogeneous task types, and incorporated uncertainties in communication and computation (such as imperfect channel state information (CSI) and inaccurate task complexity estimation) to achieve robust system design [18]. Wang et al. assumed UAVs are equipped with caching capability, enabling them to temporarily store incoming data in their queues. They jointly designed data scheduling and UAV trajectory planning strategies to minimize the maximum queue length among UEs [19]. Yuan et al. characterized the optimal SN assignment as a segment-based scheme and proved that the UAV trajectory follows a successive-hover-fly (SHF) structure, aiming to minimize data collection completion time [20]. Qin et al. investigated RIS-assisted UAV-enabled MEC systems, where reconfigurable intelligent surfaces (RIS) are deployed on surrounding building walls to establish virtual line-of-sight (LoS) links by adjusting reflection phases when direct IoT-to-UAV channels are blocked [21]. Zhu et al. focused on minimizing the overall age of information (AoI) of collected data by jointly optimizing UAV hovering point selection and visiting sequences. They designed a machine learning (ML) solution based on Transformer models and a weighted A* path search algorithm [22]. Nevertheless, these studies generally assume that UAVs can connect to all devices in the environment or provide only one-to-one service during a given period, without accounting for UAV coverage limitations or the potential performance improvements arising from parallelized services. Besides, uneven terrain can obstruct LoS links and lead to irregular signal coverage, which significantly affects communication quality and overall service performance provided by UAVs.

In terms of optimization objectives, Gao et al. considered a complex 3D urban environment, where both UE mobility and UAV obstacle avoidance were taken into account, with the goal of minimizing the total system time cost for the task

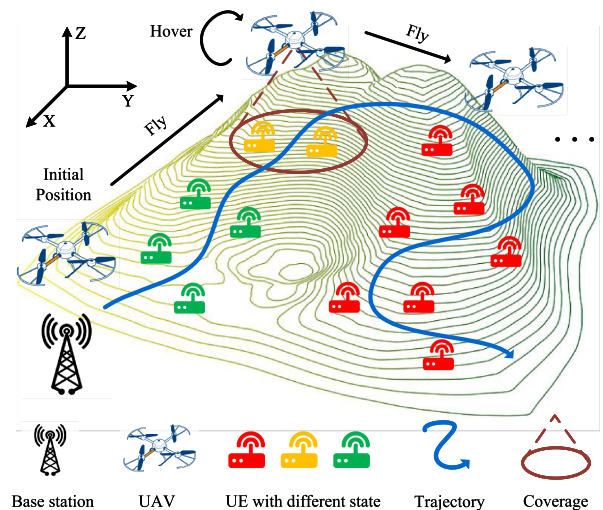


Fig. 1. UAV-assisted MEC system with uneven terrain.

offloading [23]. Chen et al. account for the issue of coverage holes in aerial cellular networks, reformulating the problem as a more tractable mixed integer convex optimization task based on graph theory, aiming to minimize the AoI for all points-of-interest (PoIs) [24]. Liu et al. modeled the system as an UAV-assisted edge caching network, incorporating Zipf-distributed file requests along with Rician distribution and Rayleigh fading channels, with the objective of maximizing overall system throughput [25]. Fu et al. assumed solar-powered wireless nodes (WNs) transmit data to multiple UAVs using harvested solar energy, and proposed an online convex-assisted reinforcement learning (CARL) approach to address energy efficiency and collection fairness among WNs [26]. Chang et al. focused on multi-UAV communication systems, aiming to maximize system utility for all served UEs within a given time constraint [27]. Gong et al. proposed a multi-UAV-assisted wireless network model, wherein UAVs form multi-hop relay networks to accelerate data transmission and optimize network topology, thereby minimizing both energy consumption and transmission delay [28]. While some works considered SN energy harvesting or energy consumption in computation and communication. However, UAVs are constrained by limited onboard battery capacity and cannot sustain prolonged operations, making it essential to account for energy consumption across computation, communication, and flight processes. When operating over uneven terrain, the UAV consumes additional propulsion energy due to frequent altitude variations and 3D maneuvers, which necessitate more precise energy management strategies.

III. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we present an overview of the UAV-assisted MEC system with uneven terrain, including the task offloading model and UAV propulsion model, and then formulate them as an optimization problem.

As illustrated in Fig. 1, we consider a UAV-assisted MEC system deployed in a remote mountainous area where communication infrastructure is limited. The system consists of three layers

TABLE I
SUMMARY OF KEY NOTATIONS

Notation	Description
$\mathbf{c}_m, \mathbf{c}_u, \mathbf{c}_s$	3D coordinate vector of UE m , UAV and BS
$\mathbf{o}_m, \mathbf{o}_u, \mathbf{o}_s$	2D coordinate vector of UE m , UAV and BS
$\lambda_m^{\text{unserved}}(t)$	Unserved indicator of UE m
$\lambda_m^{\text{cover}}(t)$	Cover indicator of UE m
$\lambda_m(t)$	Service indicator of UE m
$\varphi_m^u(t)$	Elevation angle between UE m and UAV
$M_u(t)$	Number of UEs served by UAV
$\xi_m(t)$	Proportion of tasks of UE m computed at BS
$g_m^u(t), g_u^s(t)$	Channel power gain of G2A link for UE m and A2G link for UAV
$B_m^S(t), F_m^U(t)$	The UAV's computing resources and BS's bandwidth allocated to UE m
$T_m^{\text{G2A}}(t), T_m^{\text{A2G}}(t)$	G2A and A2G transmission latency of UE m
$T_m^{\text{CA}}(t), T_m^{\text{CG}}(t)$	Computation latency of UE m on UAV and BS
$T_m^{\text{OL}}(t)$	Task offloading latency of UE m
$\Delta t_{\text{hover}}(t)$	Hover time of UAV
$\mathbf{v}_u(t), \mathbf{v}_u^d(t), v_u(t)$	Velocity vector, velocity direction vector, and velocity scalar of UAV
$Z(\mathbf{o})$	Terrain elevation at 2D coordinates $\mathbf{o}$
$T_u^r(t)$	Thrust of each UAV's rotor
$P_u^{\text{Prop}}(t)$	Propulsion power of UAV
$E_u(t)$	Energy consumption of UAV

of MEC architecture in a 3D environment, including M UEs, a UAV equipped with an onboard MEC server, and a base station connected to the core network. UEs are dispersed at different heights across different locations due to terrain-induced height variations. Therefore, the location of UE m is denoted by $\mathbf{c}_m = [x_m, y_m, z_m] \in \mathbb{R}^3$.¹ The UAV performs 3D mobility in the airspace to provide computation offloading services for ground UEs within its best communication coverage [29], and its instantaneous position defined as $\mathbf{c}_u(t) = [x_u(t), y_u(t), z_u(t)] \in \mathbb{R}^3$. Given the limited computational resources onboard the UAV, the UE's computational tasks need to be partitioned. A portion is offloaded via the UAV to the BS for processing, while the remaining tasks are computed at the UAV. Thus, we denote the location of BS by $\mathbf{c}_s = [x_s, y_s, z_s] \in \mathbb{R}^3$. Moreover, to ensure the reliability of the offloading service, the UAV remains in a hovering state during task offloading. The summary of the key notations used in this paper is listed in Table I.

A. Task Offloading Model

Each UE m possesses a task W_m characterized by its data size W_m^s and required CPU cycles W_m^c , denoted as $W_m = [W_m^s, W_m^c]$. The UAV cruises in the airspace while providing task offloading services. Upon receiving the tasks from UEs, the UAV performs partial offloading, where each task is partitioned and executed collaboratively by the UAV's onboard processor and the BS's server.

1) *UAV-UE Interaction*: This model categorizes UEs into three statuses: "unserved," "serving," and "served." To distinguish these three categories, two indicators were used, that is, the

¹In this paper, scalars are denoted by italic letters, and vectors are denoted by boldface letters.

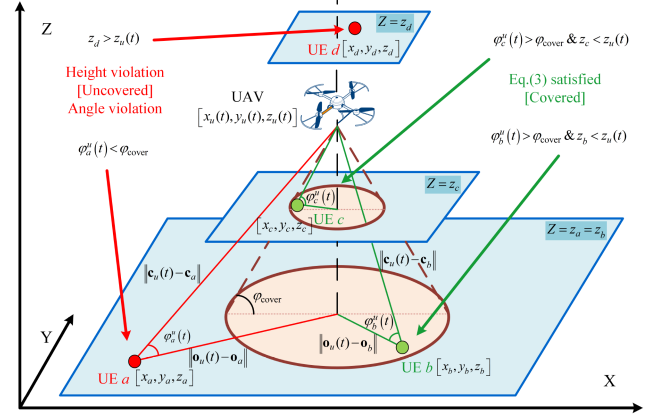


Fig. 2. Coverage region of the UAV-UE over uneven terrain.

unserved indicator $\lambda_m^{\text{unserved}}(t)$ and coverage indicator $\lambda_m^{\text{cover}}(t)$. If the UE has already been serviced, that is, the task it carries has been offloaded, the unserved indicator will be set to 0; otherwise, it will be 1, i.e.,

$$\lambda_m^{\text{unserved}}(t) = \begin{cases} 1, & \text{unserved,} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

The necessary condition for UEs to receive services is that they can establish a good transmission link with the UAV. At the same time, the coverage region of the UAV-UE is a circular disk. As shown in Fig. 2, in a 3D environment, the UAV signal covers the corresponding area at a certain coverage angle φ_{cover} [30]. Therefore, by calculating the elevation angle between the UE m and the UAV, it is possible to determine whether the UE m is within the service coverage area of the UAV, and the angle is defined as

$$\varphi_m^u(t) = \arccos\left(\frac{\|\mathbf{o}_u(t) - \mathbf{o}_m\|}{\|\mathbf{c}_u(t) - \mathbf{c}_m\|}\right), \quad (2)$$

where $\mathbf{o}_m = [x_m, y_m] \in \mathbb{R}^2$, $\mathbf{o}_u(t) = [x_u(t), y_u(t)] \in \mathbb{R}^2$.

In an uneven terrain environment, there may be UEs whose altitude is higher than that of the UAV, even when meeting the coverage angle, which is unreasonable. Therefore, the coverage indicator also needs to consider the height status of UE m and UAV, which can be expressed as

$$\lambda_m^{\text{cover}}(t) = \begin{cases} 1, & \varphi_m^u(t) > \varphi_{\text{cover}} \text{ and } z_u(t) > z_m, \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where $\lambda_m^{\text{cover}}(t) = 1$ means the UE m meet the coverage conditions at time t . As illustrated by the four examples in Fig. 2, UE a violates the angle condition, and UE d violates the height condition, so they are considered uncovered. In contrast, UEs b and c , although located at different altitudes, satisfy both the coverage angle and height conditions, and thus meet the coverage conditions at time t .

By using the unserved indicator and coverage indicator, the service indicator $\lambda_m(t)$ is defined to determine whether UE m is be serving at time t , that is,

$$\lambda_m(t) = \lambda_m^{\text{unserved}}(t) \lambda_m^{\text{cover}}(t). \quad (4)$$

Then, it is possible to count the UEs being served by the UAV at the current time t , which is denoted by $M_u(t)$, i.e.,

$$M_u(t) = \sum_{m=1}^M \lambda_m(t). \quad (5)$$

Considering the remote mountainous deployment scenario, it is assumed that the number of simultaneously served UEs remains within a normal and practical range, since the UAV's flight altitude and coverage area are naturally limited. Additionally, we maintain a list $L(t)$ to store the indices of served UEs in time t , where $M_u(t)$ equals the length of $L(t)$.

2) *Task Transmission*: After determining the UEs currently served by the UAV, a communication link will be established to upload tasks from the UEs to the UAV via the G2A channel. Since the UAV maintains a hovering state while providing offloading services, the channel state remains constant from the start until the completion of service, i.e., from the beginning to the end of the hover period. Considering the large-scale fading in UAV-UE communication links caused by terrain features and obstacles, this paper adopts a probabilistic LoS channel model. The LoS probability of UE m at time t , denoted as $\mathbb{P}(LoS, \varphi_m^u(t))$, can be approximated to be an elevation angle-related logistic function of the following form [31]

$$\mathbb{P}(LoS, \varphi_m^u(t)) = \frac{1}{1 + a \exp(-b(\frac{180}{\pi} \varphi_m^u(t) - a))}, \quad (6)$$

where a and b are parameters related to the environment. Meanwhile, we can obtain the probability of the non-line-of-sight (NLoS) channel, i.e.,

$$\mathbb{P}(NLoS, \varphi_m^u(t)) = 1 - \mathbb{P}(LoS, \varphi_m^u(t)). \quad (7)$$

Based on the occurrence probabilities of LoS and NLoS channel states, the expected channel power gain between the UAV and UE m becomes

$$\begin{aligned} \mathbb{E}[|g_m^u(t)|^2] &= \frac{(\mathbb{P}(LoS, \varphi_m^u(t)) + \kappa \mathbb{P}(NLoS, \varphi_m^u(t))) g_0}{[d_m^u(t)]^\iota} \\ &= \tilde{\mathbb{P}}(LoS, \varphi_m^u(t)) g_0 [d_m^u(t)]^{-\iota}, \end{aligned} \quad (8)$$

where $\tilde{\mathbb{P}}(LoS, \varphi_m^u(t)) = \kappa + (1 - \kappa) \mathbb{P}(LoS, \varphi_m^u(t))$ represents the regularized LoS probability, $\kappa < 1$ is the additional attenuation factor due to the NLoS condition, g_0 represents the channel gain at a distance of 1m, ι is the path loss exponent, and $d_m^u(t) = \|\mathbf{c}_u(t) - \mathbf{c}_m\|$ represents the current Euclidean distance between the UAV and UE m at time t .

In the task upload phase, we assume that the bandwidth of the UAV is distributed to all UEs in the "serving" status evenly. This bandwidth allocation strategy is designed based on practical considerations for this stage. First, this strategy helps reduce control complexity under the probabilistic LoS channel model. Second, this strategy can be seen as a redundant design that ensures fair uplink transmission between UEs in situations where task information cannot be accurately obtained before the UAV enters "hover mode". Then, the G2A transmission rate

between UE m and UAV is given by

$$R_m^u(t) = \frac{B_U}{M_u(t)} \log_2 \left(1 + \frac{P_M |g_m^u(t)|^2}{\sigma^2} \right), \quad (9)$$

where B_U is UAV's bandwidth, P_M denotes the UE's transmission power, and σ represents the noise power. Considering the UAV's cover condition and the probabilistic LoS channel model, the G2A transmission latency from UE m to UAV can be defined as

$$T_m^{G2A}(t) = \lambda_m(t) \frac{W_m^s}{R_m^u(t)}. \quad (10)$$

In this work, we consider a partial offloading model, where each task may be either partially or fully processed at the UAV, while the remaining portion is offloaded to the BS for computation. The proportion of tasks of UE m computed at BS is defined as $\xi_m(t)$. After receiving the task from the UE, the UAV needs to determine the allocated portion and transmit it to the BS via the A2G channel for calculation. This paper assumes the absence of obstacles between the UAV and the BS during the UAV's flight operation, thus modeling the A2G transmission link as a LoS channel. Then the A2G channel power gain between UAV and BS at time t is given by

$$g_u^s(t) = g_0 [d_u^s(t)]^{-2}, \quad (11)$$

where $d_u^s(t) = \|\mathbf{c}_u(t) - \mathbf{c}_s\|$ represents the Euclidean distance between the UAV and the BS. During the process of transmitting tasks from UAV to BS, the waste of bandwidth resources may occur due to the different allocation proportions of tasks for different UEs. In this stage, the A2G channel can be regarded as more deterministic, and the remaining task sizes to be transmitted to the BS are already known. Therefore, instead of evenly distributing the BS bandwidth (referred to as User-equal allocation), this paper allocates it according to the proportion of task volume (referred to as Load-equal allocation), i.e.,

$$B_m^S(t) = \frac{\xi_m(t) W_m^s B_S}{\sum_{i=1}^M \lambda_i(t) \xi_i(t) W_i^s}, \quad (12)$$

where B_S is BS's bandwidth. Then, the A2G transmission rate at time t for UE m is defined as

$$R_m^s(t) = B_m^S(t) \log_2 \left[1 + \frac{P_U g_u^s(t)}{\sigma^2} \right], \quad (13)$$

where P_U denotes the UAV's transmission power. Therefore, for the task of UE m , the A2G transmission latency from UAV to BS at time t can be expressed as

$$T_m^{A2G}(t) = \lambda_m(t) \frac{\xi_m(t) W_m^s}{R_m^s(t)}. \quad (14)$$

3) *Task Computation*: While the UAV transmits the offloaded tasks to the BS, it simultaneously processes the remaining tasks using its onboard computing resources to reduce the latency. To avoid UAV's computational resource wastage, we adopted the same method as BS's bandwidth allocation, which divides computing resources according to the proportion of task

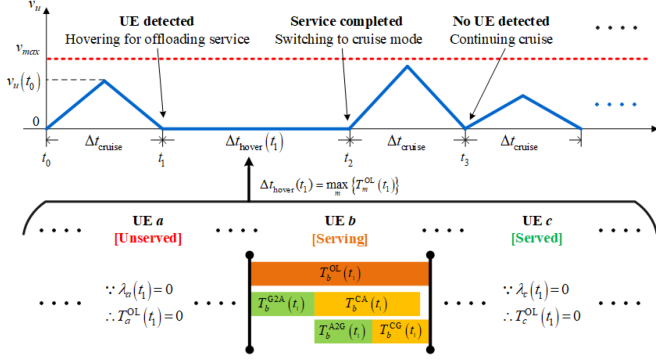


Fig. 3. The workflow for UAV.

volume, i.e.,

$$F_m^U(t) = \frac{(1 - \xi_m(t)) W_m^s F_U}{\sum_{i=1}^M \lambda_i(t) (1 - \xi_i(t)) W_i^s}, \quad (15)$$

where F_U represents the UAV's computational resources. Then, for task of UE m , the computation latency on the UAV at time t can be given by

$$T_m^{CA}(t) = \lambda_m(t) \frac{(1 - \xi_m(t)) W_m^s W_m^c}{F_m^U(t)}. \quad (16)$$

Similarly, the computation latency caused by BS computing the UE's task is defined as

$$T_m^{CG}(t) = \lambda_m(t) \frac{\xi_m(t) W_m^s W_m^c}{F_S}, \quad (17)$$

where F_S represents the BS's computational resources.

For a UE in the “serving” status, its task offloading process is illustrated by UE b in Fig. 3. First, the entire task of UE b is transmitted to the UAV via the G2A link (requiring $T_b^{G2A}(t)$). Then, partial computation is performed on the UAV (requiring $T_b^{CA}(t)$), while the remaining portion is simultaneously offloaded to the BS for computation via the A2G link (requiring $T_b^{A2G}(t) + T_m^{CG}$). The downlink delay for result feedback is neglected due to its relatively minor impact. Therefore, the task offloading latency for UE m can be expressed as

$$T_m^{OL}(t) = T_m^{G2A}(t) + \max \{T_m^{CA}(t), T_m^{A2G}(t) + T_m^{CG}(t)\}, \quad (18)$$

Similarly, for UEs in the “unserved” or “served” status (e.g., UEs a and c), since their service indicators $\lambda_a(t) = \lambda_c(t) = 0$, their corresponding task offloading latencies $T_a^{OL}(t)$ and $T_c^{OL}(t)$ are also zero. This ensures the completeness and consistency of equation at any time t .

B. Quad-Rotor UAV Propulsion Model

While providing task offloading services, the UAV must also ensure safe flight (including both cruising and hovering) in uneven terrain environments. In addition, since the UAV is constrained by its limited onboard energy, improving propulsion energy efficiency becomes another key consideration.

1) *Safe Mobility Model*: The UAV's movement model operates in two modes: “hover mode” and “cruise mode”. As shown in Fig. 3, the UAV switches between “hover mode” and “cruise mode” at the start or completion of task offloading—for example, entering “hover mode” at time t_1 and transitioning back to “cruise mode” at time t_2 . Otherwise, it maintains “cruise mode” to reach a new location for UE detection and service, as seen in the continued cruise mode at time t_3 .

The hover duration depends on the time of task offloading. Since the UAV serves UEs in parallel within its coverage, the hover time equals the longest task offloading time among Serving UEs (such as UE b). The service indicator $\lambda_m(t)$ sets the offloading time of unserved UEs (such as UE a) and served UEs (such as UE c) to zero. Therefore, the hover time $\Delta t_{\text{hover}}(t)$ can be obtained by calculating the maximum offloading time among all UEs, defined as

$$\Delta t_{\text{hover}}(t) = \max_m \{T_m^{OL}(t)\}. \quad (19)$$

During the “hover mode”, the coordinates of the UAV will remain unchanged, that is,

$$\mathbf{c}_u(t + \Delta t_{\text{hover}}(t)) = \mathbf{c}_u(t). \quad (20)$$

In “cruise mode”, the UAV's positional variation can be modeled through its velocity vector $\mathbf{v}_u(t) \in \mathbb{R}^3$. The velocity vector $\mathbf{v}_u(t)$ is composed of a velocity scalar $v_u(t)$ and heading direction vector $\mathbf{v}_u^d(t) \in \mathbb{R}^3$, expressed as:

$$\mathbf{v}_u(t) = v_u(t) \mathbf{v}_u^d(t). \quad (21)$$

Since at time t when the UAV enters the “cruise mode”, the service indicators of all UEs are set to 0, the hover time $\Delta t_{\text{hover}}(t)$ equals 0, implying that no hovering occurs. Then, this model assumed the “cruise mode” persists for duration Δt_{cruise} , comprising two symmetrical operational phases: acceleration and deceleration. During the acceleration phase, the UAV accelerates from zero velocity to $v_u(t)$, then decelerates back to zero velocity in the subsequent phase. Both phases maintain equal time duration ($\Delta t_{\text{cruise}}/2$) and identical cruise heading direction $\mathbf{v}_u^d(t)$. Then, we have

$$\begin{aligned} \mathbf{c}_u(t + \Delta t_{\text{cruise}}) &= \mathbf{c}_u(t) + \mathbf{v}_u^d(t) \int_t^{t+\Delta t_{\text{cruise}}} v_u(\tau) d\tau \\ &= \mathbf{c}_u(t) + \frac{1}{2} \mathbf{v}_u^d(t) v_u(t) \Delta t_{\text{cruise}}. \end{aligned} \quad (22)$$

Moreover, the system model defines a cubic service area to ensure UAV operations within the designated domain. The movement of the UAV cannot exceed the boundaries of the service area, that is, $0 \leq x_u(t) \leq X_{\max}$, $0 \leq y_u(t) \leq Y_{\max}$, $0 \leq z_u(t) \leq Z_{\max}$, where $X_{\max}$, $Y_{\max}$, and $Z_{\max}$ respectively represent the length, width, and altitude of the rectangle-shaped service area.

This paper accounts for real-world operational scenarios where the service area contains uneven terrain. The varying ground elevation necessitates dynamic adjustment of safe flight altitudes during path planning. In this model, the terrain elevation at 2D coordinates $\mathbf{o} = [x, y]$ is defined as $Z(\mathbf{o})$. Consequently, the UAV's minimum safe flight altitude constraint is

expressed as

$$H_{min} \leq z_u(t) - Z(\mathbf{o}_u(t)) \leq H_{max}, \quad (23)$$

where H_{min} , H_{max} denote the minimum and maximum flight altitude of the UAV, respectively. This constraint ensures a minimum safe clearance height above the terrain, which enables the UAV to effectively avoid terrain-induced obstacles during flight. The trajectory generation and optimization of the UAV in this work are grounded in this constraint, guaranteeing the development of safe and efficient 3D flight trajectories.

2) *Energy Consumption Model*: In the UAV-assisted MEC system, the energy carried by the UAV is primarily consumed by three processes: task computation, data transmission, and propulsion. We focus exclusively on propulsion energy because the energy expended during propulsion is much greater than the sum of the energy consumption for the UAV's computation and transmission. The Quad-Rotor UAV moves by the thrust generated by its rotor, which needs to overcome gravity and aerodynamic drag to propel the UAV to fly [32]. According to Newton's second law, the thrust of each rotor $T_u^r(t)$ can be expressed as a function of UAV's velocity $\mathbf{v}_u(t)$ and acceleration $\mathbf{acc}_u(t)$, i.e.,

$$T_u^r(t) = T_u^r(\mathbf{v}_u(t), \mathbf{acc}_u(t)) \\ = \frac{1}{n_r} \left\| \left(W_u \|\mathbf{acc}_u(t)\| + \frac{\rho v_u^2(t) S_{FP}}{2} \right) \mathbf{v}_u^d(t) - W_u \mathbf{g} \right\|, \quad (24)$$

where n_r is the number of rotors; W_u represents the UAV's weight; ρ denotes air density; S_{FP} means fuselage equivalent flat plate area; $\mathbf{g}$ is gravity acceleration vector.

The propulsion power of the UAV can be derived from the rotor thrust $T_u^r(t)$ and UAV's velocity scalar $v_u(t)$, defined as (25), shown at the bottom of this page, where δ is the profile drag coefficient, c_T represents the thrust coefficient based on disc area; A means disc area for each rotor; c_S and d_0 are the solidity and fuselage drag ratio for each rotor respectively; c_f is the incremental correction factor of induced power; g represents the gravity acceleration scalar; τ_c denotes the climb angle. Thus, the energy consumption of the UAV from start to time t can be defined as

$$E_u(t) = \int_0^t P_u^{\text{Prop}}(\tau) d\tau. \quad (26)$$

Based on the movement model of the UAV, we can calculate the total energy consumption by calculating the energy consumption under different modes. When the UAV is in "hover

mode", its rotors only needs to overcome gravity while maintaining its velocity $\mathbf{v}_u(t)$ and acceleration $\mathbf{acc}_u(t)$ at zero. Thus, the propulsion power remains constant, defined as P_u^{hover} . Therefore, the energy consumption of the UAV after "hover mode" can be converted into

$$E_u(t + \Delta t_{\text{hover}}(t)) = E_u(t) + \int_t^{t+\Delta t_{\text{hover}}(t)} P_u^{\text{Prop}}(\tau) d\tau \\ = E_u(t) + P_u^{\text{hover}} \Delta t_{\text{hover}}(t). \quad (27)$$

Under "cruise mode" with dynamic velocity $\mathbf{v}_u(t)$ and acceleration $\mathbf{acc}_u(t)$, the UAV's propulsion power becomes a continuously changing value. This paper employs discretization with the midpoint method to convert the energy consumption of the UAV after "cruise mode". Then, we have

$$E_u(t + \Delta t_{\text{cruise}}) = E_u(t) + \int_t^{t+\Delta t_{\text{cruise}}} P_u^{\text{Prop}}(\tau) d\tau \\ = E_u(t) + \sum_{n=0}^{N-1} P_u^{\text{Prop}} \left(t + \frac{(2n+1)\delta_t}{2} \right) \delta_t, \quad (28)$$

where N is the number of points segments, and $\delta_t = \frac{\Delta t_{\text{cruise}}}{N}$.

Through these integral transformations under different movement modes, it is possible to accurately model the energy consumption during UAV mobility, which significantly improves precision compared to constant-power assumptions.

C. Problem Formulation

The goal of this model is to serve the most UEs with the least energy consumption. Thus, we introduce two objectives, the service coverage ratio $SCR(t)$ and the UAV's propulsion energy efficiency $E_{\text{eff}}(t)$. The service coverage ratio is defined as

$$SCR(t) = 1 - \frac{\sum_{m=1}^M \lambda_m^{\text{unserved}}(t)}{M}. \quad (29)$$

In subsequent algorithmic performance comparisons, the communication region of the UAV may fail to cover any UEs, thereby resulting in a case where no UEs were served, and the service coverage ratio is zero. Therefore, instead of using the energy consumption per served UE (which may cause division-by-zero issues), we adopt the number of UEs served per unit energy consumption, i.e., energy efficiency [33], [34],

$$P_u^{\text{Prop}}(t) = P_u^{\text{Prop}}(v_u(t), T_u^r(t)) = n_r \left[\frac{\delta}{8} \left(\frac{T_u^r(t)}{c_T \rho A} + 3v_u^2(t) \right) \sqrt{\frac{T_u^r(t) \rho c_S^2 A}{c_T}} + \frac{1}{2} d_0 v_u^3(t) \rho c_S A \right. \\ \left. + (1 + c_f) T_u^r(t) \left(\sqrt{\frac{(T_u^r(t))^2}{4\rho^2 A^2} + \frac{v_u^4(t)}{4}} - \frac{v_u^2(t)}{2} \right)^{\frac{1}{2}} + \frac{W_u g v_u(t)}{n_r} \sin \tau_c \right] \quad (25)$$

defined as

$$E_{\text{eff}}(t) = \frac{M - \sum_{m=1}^M \lambda_m^{\text{unserved}}(t)}{E_u(t)}. \quad (30)$$

To further evaluate the overall system performance, we define the system utility as the product of the service coverage ratio and propulsion energy efficiency, that is,

$$Utility(t) = SCR(t)E_{\text{eff}}(t). \quad (31)$$

To ensure efficient system operation, we impose a maximum system runtime T . Simultaneously, to guarantee flight safety, any violation of constraints during UAV operation is considered a default step. Thus, the UAV must maximize system utility while maintaining safe flight operations within the prescribed time T . By optimizing the UAV's location $\mathbf{c}_u(t)$ and task allocation ratios $\xi_m(t)$, the formulated optimization problem is denoted as follows:

$$\max_{\{\mathbf{c}_u(t)\}, \xi_m(t)} Utility(t) \quad (32)$$

$$s.t. \ t \leq T, \quad (32a)$$

$$0 \leq \xi_m(t) \leq 1, \forall m, \quad (32b)$$

$$v_u(t) \leq v_{\max}, \forall t, \quad (32c)$$

$$0 \leq x_u(t) \leq X_{\max}, \quad (32d)$$

$$0 \leq y_u(t) \leq Y_{\max}, \quad (32e)$$

$$0 \leq z_u(t) \leq Z_{\max}, \quad (32f)$$

$$H_{\min} \leq z_u(t) - Z(\mathbf{o}_u(t)) \leq H_{\max}, \quad (32g)$$

where (32a) is the constraint about the system runtime, (32b) denotes the constraint of task allocation ratio, (32c) represents the maximum velocity constraint of the UAV, and (32d)–(32g) include the safety mobility constraints of the UAV.

In this system model, the UAV needs to perform 3D flight operations while simultaneously executing task offloading, where complex terrain and dynamic UE coverage present significant challenges. Furthermore, the combined effects of uncertain UE communication conditions and the UAV's high mobility induce dynamic environments. These challenges render the non-convex optimization problem (32) particularly difficult to solve using conventional methods. Consequently, in the following section, we design a DRL approach that decomposes the problem and employs multi-phase exploration to enable the agent to get the near-optimal policy.

IV. PH-DRL FOR TASK OFFLOADING AND TRAJECTORY OPTIMIZATION PROBLEM

In this section, we formulate the optimization problem as an MDP and subsequently present the PH-DRL algorithm for its solution.

A. MDP Formulation

In this system, the UAV must fly safely through uneven terrain while serving distributed ground UEs. Its movement not only alters its position but also affects the service statuses of UEs.

Since the joint task offloading and trajectory planning constitute a decision-making process, we formulate the optimization problem as an MDP that can be solved using the reinforcement learning (RL) method.

1) *State Space*: The system state at time t includes the UAV's positional coordinates $\mathbf{c}_u(t)$ and energy consumption $E_u(t)$, UEs' location coordinates $\mathbf{c}_m$, task information W_m , and unserved indicators $\lambda_m^{\text{unserved}}(t)$. Besides, considering the uneven terrain conditions, terrain elevation information must also be incorporated into the state space. However, including complete terrain elevation data $Z(\mathbf{o})$ would lead to a dimensionality explosion. Therefore, we consider the grid block containing the UAV's 2D coordinates as the central block, and extend outward to form a 5×5 square grid region as the terrain observation window. The elevation information within this window (defined as $\vartheta_u(t)$) is incorporated into the state space. Then, the state space is defined as

$$s(t) = \{\mathbf{c}_u(t), E_u(t), \vartheta_u(t), \mathbf{c}_m, W_m, \lambda_m^{\text{unserved}}(t), \forall m\}. \quad (33)$$

2) *Action Space*: The UAV conducts 3D flight operations to accommodate varying terrain elevations. The flight trajectory is governed by $\mathbf{v}_u(t)$, which we transform into spherical coordinates for more convenient parametric representation [35]. In this representation, $\varphi_v(t)$ denotes the vertical angle from the positive z-axis with $0 \leq \varphi_v(t) \leq \pi$, and $\varphi_h(t)$ denotes the horizontal angle in the xy-plane from the x-axis with $-\pi \leq \varphi_h(t) \leq \pi$. Additionally, the UAV's action needs include task allocation ratios for different UEs $\xi_m(t)$. Then, the action space is defined as

$$a(t) = \{v_u(t), \varphi_v(t), \varphi_h(t), \xi_m(t), \forall m\} \quad (34)$$

3) *Reward Function*: In this model, the UAV serves UEs within its signal coverage region during flight, then continues moving to reach and serve unserved UEs, repeating this process cyclically. This can be considered a reach sub-destination task, for which we assign positive rewards when UEs are served to guide the UAV's service behavior, i.e.,

$$r_1(t) = \begin{cases} \psi_1, & M_u(t) \neq 0, \\ 0, & \text{otherwise,} \end{cases} \quad (35)$$

where ψ_1 is a positive reward factor to adjust the reward of exploring UEs.

Moreover, a summative episodic reward is provided at the end of each episode to evaluate the UAV's overall service coverage performance during that episode. Additionally, we relocate the energy efficiency-related rewards from per-step assignments to the episode's end, thereby preventing the UAV from avoiding long-distance flights solely for energy conservation purposes. Then, the comprehensive reward is expressed as

$$r_2(t) = \begin{cases} \psi_2 SCR(t) + \psi_3 E_{\text{eff}}(t), & t = T \text{ or } SCR(t) = 1, \\ 0, & \text{otherwise,} \end{cases} \quad (36)$$

where ψ_2 and ψ_3 are positive reward factors like ψ_1 .

Furthermore, to ensure the UAV maintains safe flight operations in uneven terrain environments, we implement penalty

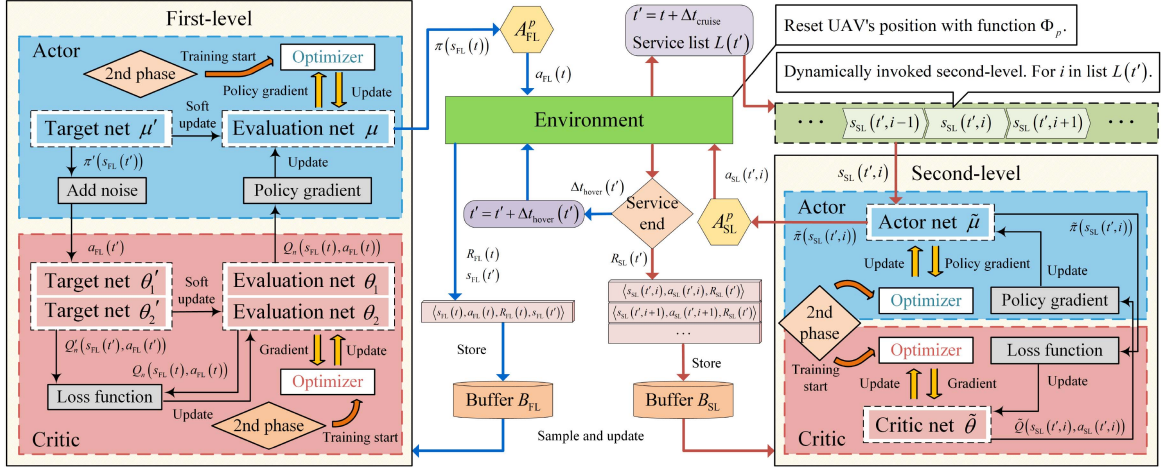


Fig. 4. The framework of the proposed PH-DRL algorithm.

rewards to punish violations of constraints (23):

$$r_3(t) = \psi_4 \varsigma(t), \quad (37)$$

where ψ_4 is a negative reward factor, and $\varsigma(t)$ is the constraint indicator with $\varsigma(t) = 1$ indicates constraint violation, while $\varsigma(t) = 0$ denotes constraint satisfaction. In summary, the reward function can be formulated as

$$R(t) = r_1(t) + r_2(t) + r_3(t). \quad (38)$$

B. Phased Hierarchical DRL Algorithm

This model's optimization problem poses two key challenges for DRL algorithms: First, the UAV operates predominantly in "cruise mode", and only serves UEs within its signal coverage during "hover mode". Consequently, in the agent's state space, UE's task information is only relevant at specific moments, causing redundancy and interference to the neural network at other times. Similarly, in the agent's action space, the task allocation ratio of each UE is only effective at particular moments, adversely affecting neural network training. Second, while the terrain observation window effectively addresses high-dimensional state space issues, the agent struggles to acquire global terrain information through limited exploration, which influences optimal flight path determination. Besides, the infrequent occurrence of "hover mode" results in insufficient task-offloading optimization experience, leading to poor optimization performance.

To address these challenges, we propose a phased hierarchical DRL algorithm, whose network architecture is illustrated in Fig. 4. In conventional DRL workflows, only a single-level network architecture is employed, where the environment and the network interact through state, action, and reward, and experiences are stored for optimization. In contrast, our algorithm adopts a two-level architecture (corresponding to the First-level and Second-level shown on the left and right in the figure), and incorporates a phased training method into the workflow. The following subsections first elaborate on the definitions of state, action, and reward in each level, and then describe the design of

the phased training method as well as the network composition within each level.

1) *Hierarchical Architecture*: To address the temporal redundancy issues in both state and action spaces, we propose a hierarchical network architecture that separates the optimization variables. Specifically, UAV's trajectory $\{\mathbf{c}_u(t)\}$ is optimized by the first-level while UEs' task allocation ratios $\xi_m(t)$ are handled by the second-level, achieving functional separation of the optimization objectives.

Optimizing variable separation also requires modifications to the MDP problem. For the first-level, which exclusively handles $\{\mathbf{c}_u(t)\}$ without considering $\xi_m(t)$, the agent no longer requires the task information W_m for all UE. Consequently, the state space can be substantially simplified from the original state space (33) and defined as

$$s_{FL}(t) = \{\mathbf{c}_u(t), E_u(t), \vartheta_u(t), \mathbf{c}_m, \lambda_m^{\text{unserved}}(t), \forall m\}, \quad (39)$$

achieving a dimensionality reduction of $2 * M$ parameters. The action space retains only UAV flight control variables while removing all task allocation ratios, that is,

$$a_{FL}(t) = \{v_u(t), \varphi_v(t), \varphi_h(t)\}. \quad (40)$$

The goal of this network is basically the same as the original goal, so the reward function remains the original design, which is $R_{FL}(t) = R(t)$.

The second-level's state space cannot simply incorporate all UE task information, as this would introduce redundancy. Moreover, the number of UEs in "serving" status varies between UAV's propulsion modes: zero during "cruise mode" and one or more during "hover mode". This variability makes it difficult to simply set the number of input/output neurons to a fixed value. To address both redundancy and input/output dimensionality variation challenges, we designed a dynamically invoked network structure that employs iterative calling mechanisms.

Specifically, we consider the number of UEs in "serving" status as the second-level's invocation count. Then, each invocation processes only a single UE's information and outputs the corresponding task allocation ratio for that UE. Considering

that the UAV provides parallel services and requires a reasonable allocation of system resources, this process is related to the UE's task information. We introduce a new variable $\zeta_m(t)$ representing each UE's task proportion relative to the total demand from all currently served UEs, defined as: $\zeta_m(t) = \frac{W_m^s}{\sum_{i=1}^M \lambda_i(t) W_i^s}$. Then, the state space of the second-level can be represented as

$$s_{SL}(t, m) = \{W_m, \zeta_m(t), d_m^u(t), d_u^s(t)\}, \quad (41)$$

where $d_m^u(t)$ and $d_u^s(t)$ take into account the relative position of the UE-UAV-BS to obtain the optimal task allocation. The corresponding action space is denoted as

$$a_{SL}(t, m) = \{\xi_m(t)\}. \quad (42)$$

For the dynamically-invoked network structure, we designed a novel reward function that incorporates the UAV's hover time as a negative penalty term, expressed as

$$R_{SL}(t) = \psi_5 \Delta t_{\text{hover}}(t), \quad (43)$$

where ψ_5 is a negative reward factor. This design stems from the direct correlation between task offloading times and hover mode duration (as established in (19), thereby guiding the network to minimize total service time through optimized decision-making.

2) *Phased Training Strategy*: To address the potential low training efficiency caused by insufficient environmental information, we introduce a phased training method. The first phase ($p = 1$) aims to obtain general knowledge about the environment and runs for e_p . In the reset function Φ_p , we randomized the agent's initial position to enable comprehensive perception of terrain elevation data $Z(\mathbf{o})$ across different regions while establishing correlations between the UE's information and rewards. Besides, we applied a random exploration strategy to the action functions A_{FL}^p and A_{SL}^p , allowing the agent to quickly gather substantial effective experience without requiring training.

The second phase ($p = 2$) focuses on refining the agent's action details and continues for a corresponding episodes e_p . We preserved experience buffers B_{FL} and B_{SL} from the first phase to accelerate network training, while modifying the reset function Φ_p to initialize the agent at predetermined positions. An ε -greedy strategy with Gaussian noise $\tilde{n} \sim \mathcal{N}(0, \hat{\sigma}^2)$ is implemented in the action functions A_{FL}^p and A_{SL}^p . This establishes a learning approach where the agent prioritizes exploration in the early stages of training and focuses on optimization in later stages within the specified environment.

Algorithm 1 presents the pseudocode for PH-DRL. The first-level addresses a more complex problem and therefore employs the TD3 algorithm, which consists of six networks: an evaluation actor network $\pi(s|\mu)$, two evaluation critic network $Q_1(s, a|\theta_1)$, $Q_2(s, a|\theta_2)$, and their corresponding target networks $\pi'(s|\mu')$, $Q'_1(s, a|\theta'_1)$, $Q'_2(s, a|\theta'_2)$. The second-level operates in a simpler solution space, for which we adopt the AC algorithm comprising two networks: an actor network $\tilde{\pi}(s|\tilde{\mu})$ and a critic network $\tilde{Q}(s, a|\tilde{\theta})$. Besides, we use the variable k to denote the step count of UAV "cruise mode" execution. The target networks in first-level updates every d steps, while the second-level updates according to its invocation frequency. The proposed

algorithm operates in two phases with a phase index $p \in \{1, 2\}$. In each phase, the first-level network is responsible for UAV flight control, while the second-level network is activated when serviceable UEs are detected. UAV flight and UE service both consume continuous time, and the total duration of an episode is constrained by a time budget T , rather than a fixed number of time steps. Within each episode, the UAV may perform multiple flight actions before detecting UEs. However, each UE can be served at most once, and the second-level network is therefore activated no more than M times per episode. Since the phase index and network architectures are fixed, the overall computational complexity of the proposed algorithm can be expressed as $O(E \times M)$.

V. EXPERIMENT RESULTS AND ANALYSIS

In this section, we first describe the system configuration parameters, followed by numerical experiments that evaluate the proposed PH-DRL algorithm in terms of convergence, performance, and extended analyses.

A. Simulation Settings

In our simulations, real elevation data from a paid-platform (<http://www.tuxingis.com/store.html>) digital elevation model (DEM) were utilized. The specific coordinates are (Left = 107.3612972521917, Top = 34.08013815795105, Right = 107.36568909492071, Bottom = 34.07574631522204). The area has dimensions of 500 m in both length and width, with the highest terrain elevation within the region being 94 m. Therefore, the system environment is considered as a cube service area of $500 \times 500 \times 194 \text{ m}^3$. The 15 UEs are randomly distributed across the uneven terrain, with the BS fixed at [6, 6, 56]. The UAV departs from [6, 6, 131] and attempts to serve as many UEs as possible. The detailed simulation parameters are summarized in Table II.

The network architecture of the first-level in the proposed PH-DRL algorithm comprises a fully connected neural network with three hidden layers containing 256, 128, and 64 neurons, respectively. The second-level employs a dual hidden-layer neural network with 64 neurons in each layer. Both level employs ReLU activation functions and maintain a consistent action noise standard deviation of 0.1. Besides, the first-level operates with a discount factor of 0.99, soft update rate of 0.005, update interval of 5 steps, and policy noise standard deviation of 0.2. Moreover, the actor and critic learning rates are set to 0.0003 and 0.001, respectively, for the first-level, compared to 0.003 and 0.01 for the second-level. The experience replay buffer sizes are configured as 2^{17} for the first level and 2^{12} for the second level, with corresponding batch sizes of 512 and 256 samples during training.

Apart from the introduced PH-DRL approach, we employed four other optimization algorithms to compare performance: degraded versions of PH-DRL without phased or hierarchical components, named hierarchical DRL (H-DRL) and phased TD3 (P-TD3); the original algorithms were also considered, with both TD3 and proximal policy optimization (PPO) serving as baseline algorithms, where PPO is a policy gradient method

Algorithm 1: PH-DRL Algorithm of Task Offloading Problem With Uneven Terrain.

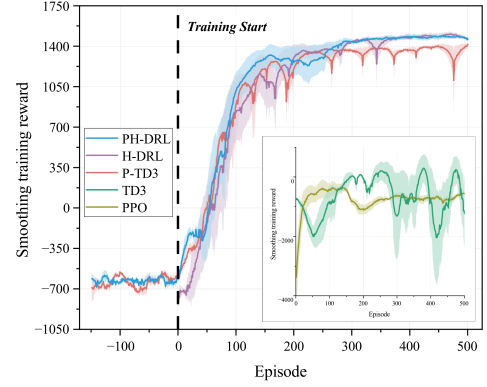
Input: Terrain's data $Z(\mathbf{o})$ and UEs' data $\mathbf{c}_m, W_m$.

Output: The optimal trajectory and task allocation ratios $\{\mathbf{c}_u(t)\}, \xi_m(t)$.

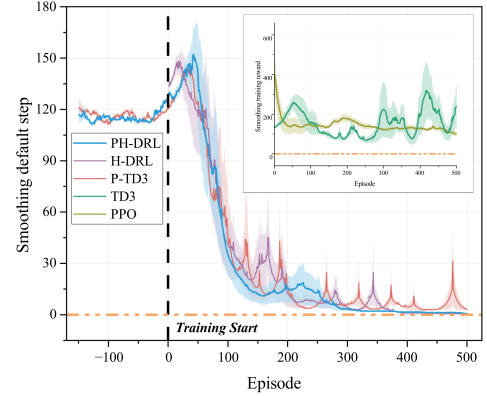
```

1 Initialize the first-level's networks  $\mu, \theta_1, \theta_2, \mu', \theta'_1, \theta'_2$ ;
2 Initialize the second-level's networks  $\tilde{\mu}, \tilde{\theta}$ ;
3 Initialize environment ENV, buffers  $B_{FL}$  and  $B_{SL}$ ;
4 for each phase  $p$  do
5   while episode  $e \leq E_p$  do
6     Set time  $t = 0$ , step  $k = 0$ ;
7     Reset ENV and get  $s_{FL}(t)$ ;
8     Reset UAV's position with function  $\Phi_p$ ;
9     while  $t \leq T$  do
10      Select  $a_{FL}(t) \leftarrow A_{FL}^p(s_{FL}(t))$ ;
11      Set  $t' = t + \Delta t_{cruise}$ ;
12      Get  $L(t')$  from ENV;
13      if length of  $L(t') \neq 0$  then
14        for  $i$  in list  $L(t')$  do
15          Select  $a_{SL}(t', i) \leftarrow A_{SL}^p(s_{SL}(t', i))$ ;
16          Get and store  $T_i^{OL}(t')$ ;
17        end
18        Get  $R_{SL}(t')$  from ENV;
19        for  $i$  in list  $L(t')$  do
20          Store experience
             $\langle s_{SL}(t', i), a_{SL}(t', i), R_{SL}(t') \rangle$ 
            to buffer  $B_{SL}$ ;
21        end
22        Calculate  $\Delta t_{hover}(t')$  by  $\{T_m^{OL}(t')\}$ ;
23        Set  $t' = t' + \Delta t_{hover}(t')$ ;
24        if  $p == 1$  then break;
25        Sample experience from buffer  $B_{SL}$ ;
26        Update second-level's networks  $\tilde{\mu}, \tilde{\theta}$ ;
27      end
28      Get  $s_{FL}(t'), R_{FL}(t)$  from ENV;
29      Store experience
         $\langle s_{FL}(t), a_{FL}(t), R_{FL}(t), s_{FL}(t') \rangle$ 
        to buffer  $B_{FL}$ ;
30      Set  $t = t', k = k + 1$ ;
31      if  $p == 1$  then continue;
32      Sample experience from buffer  $B_{FL}$ ;
33      Update first-level's evaluation networks  $\theta_1, \theta_2$ ;
34      if  $k \bmod d$  then
35        Update first-level's evaluation network  $\mu$ ;
36        Soft update first-level's target networks  $\mu', \theta'_1, \theta'_2$ ;
37      end
38    end
39  end
40 end
41 return

```



(a) Trend of the training reward



(b) Trend of the default step

Fig. 5. Convergence of different algorithms.

that uses clipped surrogate objectives to ensure stable policy updates while maintaining sample efficiency.

To ensure fairness in comparison, both H-DRL and PH-DRL maintain identical network architectures and hyperparameter configurations. Similarly, P-TD3, TD3, and PPO algorithms adopt the same network structure and hyperparameters as the first-level of PH-DRL. For the subsequent experiments, trajectory results were generated using an identical random seed, while all other experimental outcomes represent averaged values obtained under three consistent random seeds. The comparison of various algorithms includes both training and testing results. To ensure that each algorithm has sufficient exploration and learning iterations, the time limit T for agent training was set to 1800s.

B. Convergence Analysis

We first compared the convergence performance of the different algorithms. Fig. 5(a) presents the training reward curves for all algorithms, where the range from -150 to 0 episodes corresponds to the first phase ($p = 1$) of the PH-DRL and P-TD3 algorithms employing the staged method. The solid lines represent the smoothed curves obtained by applying the exponential moving average (EMA) method to the three results, and the shaded areas indicate the standard error ranges for

TABLE II
EXPERIMENTAL SIMULATION PARAMETERS

Parameters	Value	Parameters	Value
System runtime, T	150 s	UAV's maximum velocity, v_{max}	15 m/s
Size of area, $X_{max}, Y_{max}, Z_{max}$	500 m, 500 m, 194m	UAV's flight altitude, H_{min}, H_{max}	50 m, 100 m
Number of UEs, M	15	UAV's rotors, n_r	4
Size of task data, W_m^s	[1,5] Mbits	UAV's weight, W_u	2 kg
Number of CPU cycles, W_m^c	[100,200] cycles/bit	UAV's cruise duration, Δt_{cruise}	3 s
Environmental parameters, a, b	15, 0.5	Number of points segments, N	10
NLoS attenuation, κ	0.2	Air density, ρ	1.225 kg/m ³
Path loss exponent, ι	2.3	Fuselage equivalent flat plate area, S_{FP}	0.01
Noise power, σ	-80 dBm	Gravity acceleration, g	9.81 m/s ²
Channel gain, g_0	-50 dB	Profile drag coefficient, δ	0.012
UE's transmit power, P_M	0.1 W	Thrust coefficient based on disc area, c_T	0.302
BS's bandwidth, B_S	5 MHz	Disc area for each rotor, A	0.0314 m ²
BS's computation resource, F_S	1 GHz	Solidity for each rotor, c_S	0.0955
UAV's transmit power, P_U	5 W	Fuselage drag ratio for each rotor, d_0	0.834
UAV's bandwidth, B_U	1 MHz	Positive reward factor, ψ_1, ψ_2, ψ_3	100, 10, 10000
UAV's computation resource, F_U	1 GHz	Negative reward factor, ψ_4, ψ_5	-10, -1

each algorithm. PH-DRL and H-DRL reached convergence around the 300th episode, both achieving the highest reward values. This demonstrates the effectiveness of the hierarchical method, which reduces the redundancy in input dimensions by separating optimization variables. As a result, the input dimension is reduced from 119 to 94, thereby lowering the training difficulty of the neural network. PH-DRL is more stable than H-DRL, as it never exhibited a sudden drop in reward values. P-TD3, utilizing the staged method, gathered substantial effective information and was able to converge, but its reward value was lower than those of PH-DRL and H-DRL, with greater fluctuations. The TD3 and PPO algorithms did not achieve satisfactory convergence, indicating that the combination of uneven terrain and dynamic service coverage presents a significant challenge, making it difficult for conventional algorithms to solve. Fig. 5(b) illustrates the number of default steps during the training process, representing the number of violations of the constraint conditions. It is evident that PH-DRL, H-DRL, and P-TD3 all effectively perceive the terrain altitudes, optimizing the UAV flight actions accordingly. However, PH-DRL outperformed the others, reaching zero default steps the earliest and maintaining stability. While PPO showed some improvement compared to TD3, both have poor results.

C. Performance Comparison

The performance is evaluated in terms of trajectory planning, service coverage, and energy efficiency to demonstrate the effectiveness of the proposed algorithm. Fig. 6 shows the optimal trajectories produced by different algorithms in the testing phase. Notably, the UE distribution is predominantly positioned along the straight line from (0, 0) to (500, 500), and the trajectories of most algorithms also align with this path, indicating that the reward function effectively guides the network's learning direction. Among them, PH-DRL and H-DRL successfully serve

all UEs without violating any constraint conditions. P-TD3 and TD3 manage to maintain safe flight for most of the duration but breach constraints toward the end, likely due to the complexity of network inputs or insufficient information. The trajectory of PPO reveals issues in its network training, where outputting the extreme values in the action range results in movement restricted to only the X and Z dimensions, accompanied by unreasonable altitude fluctuations, indicating that the algorithm fails to achieve effective convergence.

Fig. 7 depicts the service coverage ratio, propulsive energy efficiency, and overall system utility of the different algorithms. PH-DRL achieves full UE coverage, serving each UE with an average energy expenditure of 2 kJ. H-DRL, P-TD3, and TD3 each attain certain improvements in either service coverage ratio or energy efficiency, yet their overall utility remains inferior to that of PH-DRL. To further explore the relationship between UAV's energy efficiency and task allocation ratio, propulsive energy efficiency is decomposed into cruise energy efficiency and hover energy efficiency, as illustrated in Fig. 8. It is evident that PH-DRL and H-DRL effectively leverage the second level to optimize task offloading, thereby enhancing energy efficiency in "hover mode". Moreover, under the given parameters, a UE's task allocation ratio of approximately 0.45 proves optimal. P-TD3 and TD3 exhibit relatively high average task allocation ratios, which in turn lead to elevated hover energy consumption. Combined with the observations in Fig. 6, the PPO algorithm undertakes only a very brief flight, serving solely UE 1, 4, 7, and 14, as its generated trajectory is restricted mainly to the X and Z dimensions and exhibits unreasonable altitude fluctuations. Nevertheless, it achieves a higher degree of parallelism in serving UEs. In contrast, other algorithms at times serve only a single UE, resulting in PPO's hovering efficiency surpassing that of P-TD3 and TD3. This confirms the superior resource utilization of parallel service over one-to-one service.

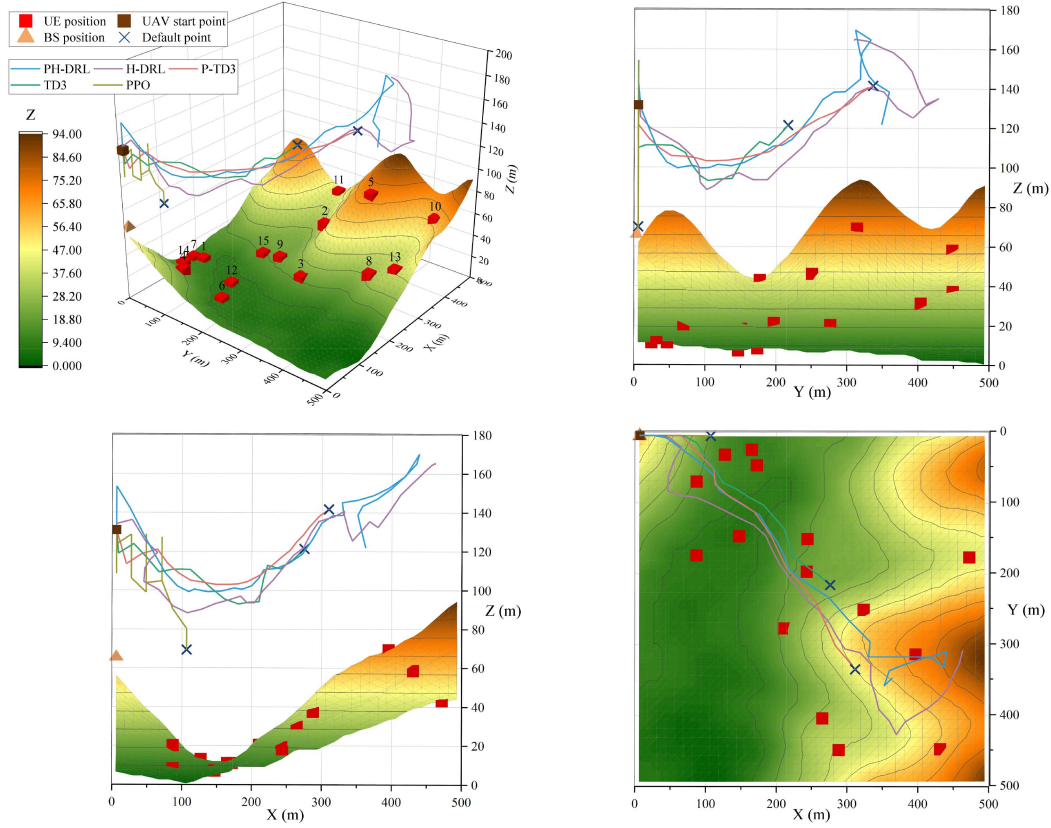


Fig. 6. UAV trajectories of different algorithms: including 3D and three-view drawing.

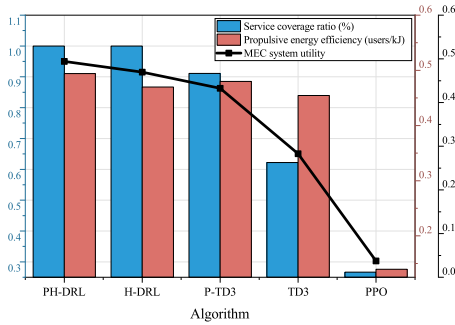


Fig. 7. Comparison of different algorithms: including service coverage ratio, propulsive energy efficiency, and system utility.

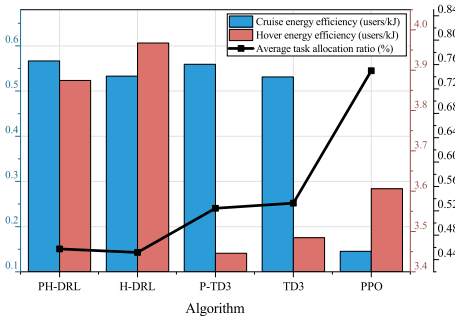
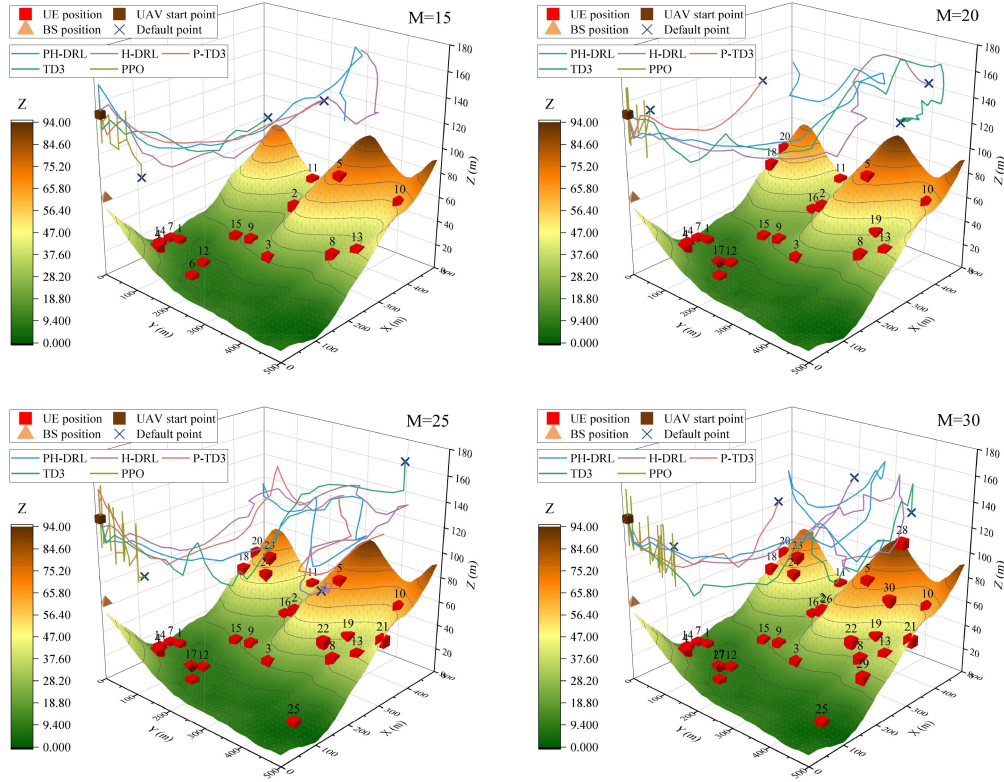
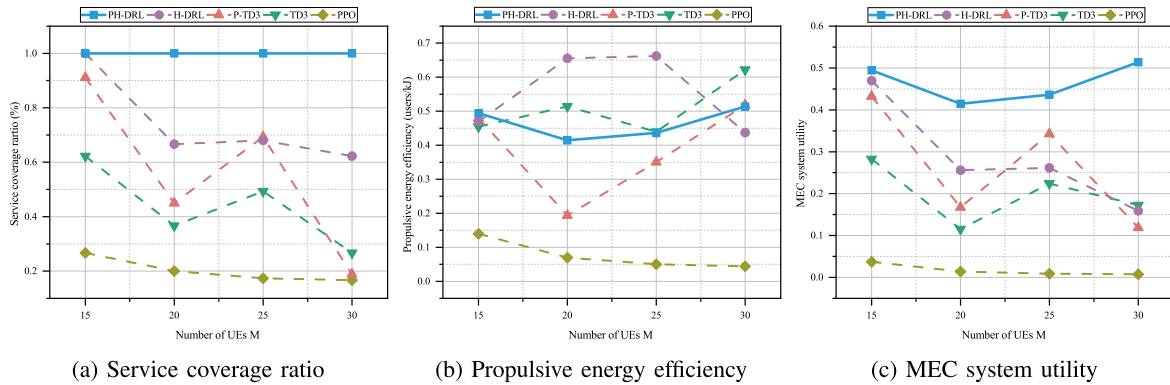


Fig. 8. Comparison of different algorithms: including cruise energy efficiency, hover energy efficiency, and average task allocation ratio.

D. Extended Evaluation

To further compare the performance improvements brought by hierarchical and phased methods, we conducted extended experiments involving three parameters: the number of UEs, the bandwidth of the BS, and the computational resources of the BS. Fig. 9 presents the optimal trajectories generated by different algorithms under varying UE counts. To ensure the validity of the experiment, as the number of UE increased, we also relaxed the time constraint T : 20 UE corresponds to a time of 300s, 25 UE to 450s, and 30 UE to 600s. Except for the proposed PH-DRL, all other algorithms encountered the default step. This occurred because, as the number of UEs increased, the UAV was required to reach more new areas, thereby elevating the probability of violating constraints. Moreover, the increase in the number of UEs leads to a rapid growth in the input dimension of the learning model. By adopting the hierarchical method, the input dimension is effectively reduced, from 119 to 94 with 15 UEs, from 149 to 114 with 20 UEs, from 179 to 134 with 25 UEs, and from 209 to 154 with 30 UEs. This dimensionality reduction significantly alleviates the training difficulty and contributes to the superior robustness of the proposed PH-DRL.

Fig. 10 illustrates the variation in algorithm performance as the number of UEs changes. As the number of UEs increases, the dimensionality of the network's state space expands dramatically, posing greater challenges for convergence and sample efficiency. The PPO algorithm shows a rapid decline in

Fig. 9. UAV 3D trajectories of different algorithms and UE counts M .Fig. 10. Comparison of different algorithms and UE counts M .

performance, as its on-policy nature prevents it from reusing past experiences and makes it difficult to handle experience sampling and gradient estimation under the increasingly complex state and action spaces. The other baseline methods, such as TD3 and P-TD3, perform better than PPO but still exhibit fluctuations as the environment dimension increases, largely due to the increased exploration difficulty and higher control complexity. In contrast, PH-DRL consistently maintains a 100% service coverage ratio, keeps efficient training and stable convergence even in high-dimensional settings. In terms of propulsive energy efficiency, PH-DRL did not retain the highest efficiency, which can be attributed to the presence of certain UEs—such as UE

18, 20, 23, 24, and 25—who are located in more remote areas. Serving these UEs necessitates greater cruising energy expenditure, thus reducing the average propulsive energy efficiency. To account for this, we incorporated a system utility metric that combines service coverage ratio and propulsive energy efficiency. From this, it is evident that PH-DRL consistently maintains the highest overall system utility. This demonstrates the effectiveness of the first level in the hierarchical method and the staged approach in optimizing UAV path planning.

Fig. 11 illustrates the impact of different bandwidth allocation strategies on hover energy efficiency under varying UE counts and algorithms. As shown in Fig. 11(a), under different

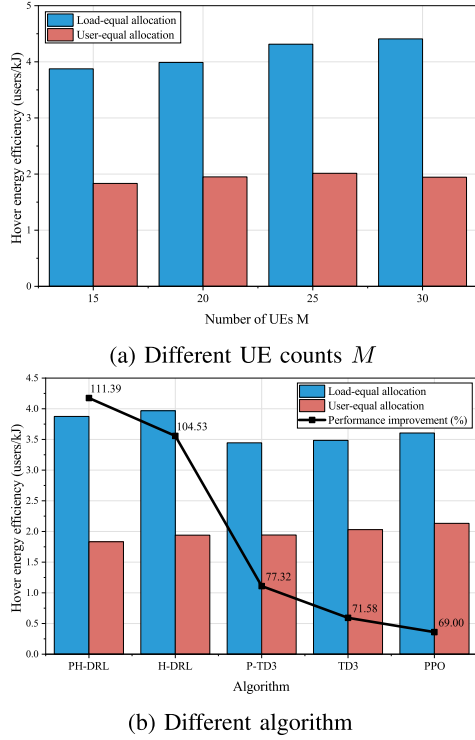
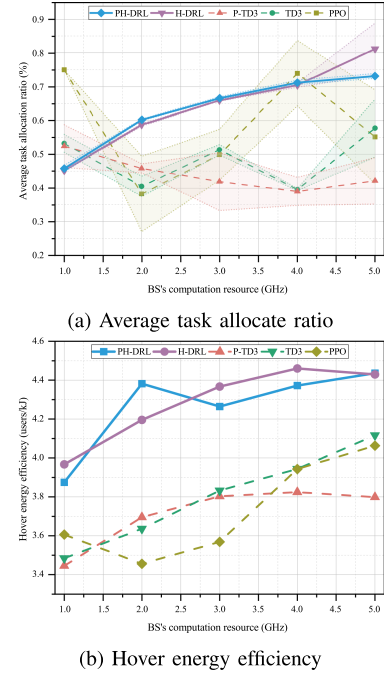
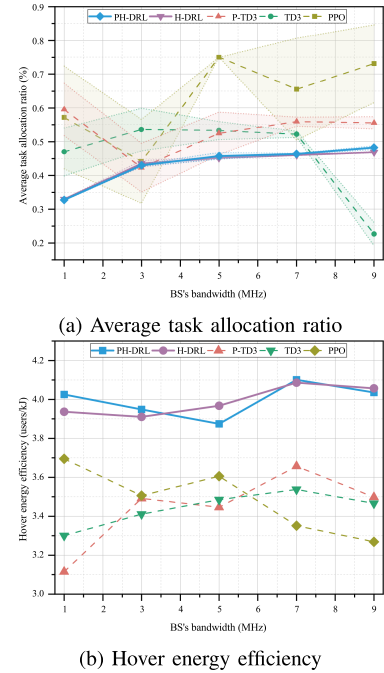


Fig. 11. Comparison of different bandwidth allocation methods.

numbers of UEs M , the load-equal bandwidth allocation scheme consistently outperforms the user-equal allocation scheme. The superiority of load-equal allocation can be attributed to the fact that, during the A2G transmission stage, the UAV operates under relatively stable LoS channel conditions and the remaining task loads are already known. This enables bandwidth to be proportionally allocated according to task load, thereby reducing transmission delay and improving overall energy efficiency. Fig. 11(b) further compares the two allocation schemes under different algorithms. It can be observed that load-equal allocation consistently achieves higher hover energy efficiency than user-equal allocation across all algorithms, demonstrating its robustness and algorithm-independent advantage. Moreover, when combined with the proposed PH-DRL framework, the load-equal strategy yields the most significant performance improvement. This is because PH-DRL can more effectively exploit the known task load information and stable channel conditions in the A2G stage, enabling more accurate decisions.

Fig. 12 depicts the variation in the average task allocation ratio under changing BS's computing resources, where the shaded areas indicate the standard error ranges for each algorithm. The results given by PH-DRL and H-DRL using a hierarchical approach show an increasing trend. This is because, as the BS's computational capacity increases, the latency of $T_m^{CG}(t)$ decreases prompting the UAV to allocate a greater proportion of tasks to the BS for processing. In contrast, the results from the remaining algorithms fluctuate irregularly without a stable trend, and their wide error margins indicate that their allocation ratios have not been effectively optimized. Regarding hover

Fig. 12. Comparison of different algorithms and BS's computation resource F_S .Fig. 13. Comparison of different algorithms and BS's bandwidth B_S .

energy efficiency, PH-DRL and H-DRL consistently outperform all other algorithms.

Fig. 13 shows the variation in the average task allocation ratio under changing BS's bandwidths. Similar to Fig. 12, as the BS's bandwidth increases, the communication latency of the A2G channel decreases, leading to a rising trend in the average allocation ratio. The figure shows that PH-DRL and H-DRL display a growth tendency, albeit not pronounced. This is due

to two reasons: first, A2G latency $T_m^{A2G}(t)$ exerts a smaller influence on total latency compared with $T_m^{CG}(t)$; second, with computational resources held constant, the task allocation ratio has a theoretical upper limit, beyond which assigning more tasks to the BS would instead increase total latency. Besides, the results of the remaining algorithms remain unstable. In terms of hover energy efficiency, PH-DRL and H-DRL continue to deliver the best performance. The outcomes in Figs. 12 and 13 both attest to the effectiveness of the second level in the hierarchical method for optimizing task allocation ratios.

VI. CONCLUSION

This paper establishes a UAV-assisted MEC system that accounts for UAV flight trajectory design over uneven terrain, dynamic UE servicing within the signal coverage region, and UAV propulsion energy consumption modeling. The system is then formulated as a multi-objective non-convex optimization problem, seeking to maximize service coverage ratio and propulsion energy efficiency within a given system runtime. A PH-DRL optimization algorithm is proposed, which integrates a hierarchical network architecture and phased training strategy with DRL. Simulation experiments leveraging real elevation data are conducted to evaluate the performance of the proposed algorithm. The results demonstrate that PH-DRL surpasses other comparative algorithms in terms of system utility, thereby validating its effectiveness in optimizing UAV trajectories and task allocation ratio, as well as its applicability to complex non-convex optimization problems. Future work will focus on extend to multi-UAV collaboration in uneven terrain environments and enhancing system performance under additional terrain-related factors and environmental uncertainties like user mobility.

REFERENCES

- [1] L. Dai, F. Zeng, H. Kong, J. Cai, H. Jiang, and K. Li, "Throughput-aware cooperative task offloading in dynamic mobile edge computing systems," *IEEE Trans. Mob. Comput.*, vol. 24, no. 12, pp. 13276–13292, Dec. 2025.
- [2] J. Mei, L. Dai, Z. Tong, X. Deng, and K. Li, "Throughput-aware dynamic task offloading under resource constant for MEC with energy harvesting devices," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 3, pp. 3460–3473, Sep. 2023.
- [3] H. Bayerlein, M. Theile, M. Caccamo, and D. Gesbert, "UAV path planning for wireless data harvesting: A deep reinforcement learning approach," in *Proc. IEEE Glob. Commun. Conf.*, Dec. 2020, pp. 1–6.
- [4] B. Li, Q. Li, Y. Zeng, Y. Rong, and R. Zhang, "3D trajectory optimization for energy-efficient UAV communication: A control design perspective," *IEEE Trans. Wireless Commun.*, vol. 21, no. 6, pp. 4579–4593, Jun. 2022.
- [5] Y. Choi, M. Chen, Y. Choi, S. Briceno, and D. Mavris, "Multi-UAV trajectory optimization utilizing a NURBS-based terrain model for an aerial imaging mission," *J. Intell. Robotic Syst.*, vol. 97, no. 1, pp. 141–154, Jan. 2020.
- [6] A. V. Savkin, "Autonomous UAV 3D trajectory optimization and transmission scheduling for sensor data collection on uneven terrains," *Def. Technol.*, vol. 30, pp. 154–160, Dec. 2023.
- [7] Y. Luo, Y. Wang, Y. Lei, C. Wang, D. Zhang, and W. Ding, "Decentralized user allocation and dynamic service for multi-UAV-enabled MEC system," *IEEE Trans. Veh. Technol.*, vol. 73, no. 1, pp. 1306–1321, Jan. 2024.
- [8] C. Zhan and Y. Zeng, "Energy minimization for cellular-connected UAV: From optimization to deep reinforcement learning," *IEEE Trans. Wireless Commun.*, vol. 21, no. 7, pp. 5541–5555, Jul. 2022.
- [9] D. Han, T. Shi, T. Han, and Z. Zhou, "Joint optimization of trajectory and node access in UAV-aided data collection system," *IEEE Syst. J.*, vol. 17, no. 2, pp. 2574–2585, Jun. 2023.
- [10] Y. Li et al., "Data collection maximization in IoT-sensor networks via an energy-constrained UAV," *IEEE Trans. Mob. Comput.*, vol. 22, no. 1, pp. 159–174, Jan. 2023.
- [11] K. Liu and J. Zheng, "UAV trajectory planning with interference awareness in UAV-enabled time-constrained data collection systems," *IEEE Trans. Veh. Technol.*, vol. 73, no. 2, pp. 2799–2815, Feb. 2024.
- [12] W. Sun, Z. Bai, J. Shi, and Z. Li, "DDPG-based multi-UAV trajectory optimization for WSN's data collection," in *Proc. IEEE 11th Int. Conf. Inf. Commun. Netw.*, Xi'an, China: IEEE, Aug. 2023, pp. 241–247.
- [13] H. Zhao et al., "Online trajectory optimization for energy-efficient cellular-connected UAVs with map reconstruction," *IEEE Trans. Veh. Technol.*, vol. 73, no. 3, pp. 3445–3456, Mar. 2024.
- [14] 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, Aug. 2023.
- [15] L. Xie, J. Xu, and R. Zhang, "Throughput maximization for UAV-enabled wireless powered communication networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1690–1703, Apr. 2019.
- [16] N. Gupta, S. Agarwal, D. Mishra, and B. Kumbhani, "Trajectory and resource allocation for UAV replacement to provide uninterrupted service," *IEEE Trans. Commun.*, vol. 71, no. 12, pp. 7288–7302, Dec. 2023.
- [17] X. Huang, Y. Luo, X. Yang, and Q. Chen, "Energy consumption optimization for UAV-assisted communication by trajectory design," in *Proc. IEEE 97th Veh. Technol. Conf.*, Florence, Italy: IEEE, Jun. 2023, pp. 1–5.
- [18] 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, Feb. 2024.
- [19] B. Wang, Q. Wang, N. Yang, and R. Chai, "Long-term optimization-based data scheduling and trajectory planning for UAV-assisted systems," in *Proc. IEEE 98th Veh. Technol. Conf.*, Hong Kong, Hong Kong: IEEE, Oct. 2023, pp. 1–5.
- [20] X. Yuan, Y. Hu, J. Zhang, and A. Schmeink, "Joint user scheduling and UAV trajectory design on completion time minimization for UAV-aided data collection," *IEEE Trans. Wireless Commun.*, vol. 22, no. 6, pp. 3884–3898, Jun. 2023.
- [21] X. Qin, Z. Song, T. Hou, W. Yu, J. Wang, and X. Sun, "Joint optimization of resource allocation, phase shift, and UAV trajectory for energy-efficient RIS-assisted UAV-enabled MEC systems," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 4, pp. 1778–1792, Dec. 2023.
- [22] B. Zhu, E. Bedeer, H. H. Nguyen, R. Barton, and Z. Gao, "UAV trajectory planning for AoI-minimal data collection in UAV-aided IoT networks by transformer," *IEEE Trans. Wireless Commun.*, vol. 22, no. 2, pp. 1343–1358, Feb. 2023.
- [23] Y. Gao, X. Yuan, D. Yang, Y. Hu, Y. Cao, and A. Schmeink, "UAV-assisted MEC system with mobile ground terminals: DRL-based joint terminal scheduling and UAV 3D trajectory design," *IEEE Trans. Veh. Technol.*, vol. 73, no. 7, pp. 10164–10180, Jul. 2024.
- [24] G. Chen, C. Cheng, X. Xu, and Y. Zeng, "Minimizing the age of information for data collection by cellular-connected UAV," *IEEE Trans. Veh. Technol.*, vol. 72, no. 7, pp. 9631–9635, Jul. 2023.
- [25] B. Liu, C. Liu, and M. Peng, "Dynamic cache placement and trajectory design for UAV-assisted networks: A two-timescale deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 73, no. 4, pp. 5516–5530, Nov. 2024.
- [26] C.-W. Fu, M.-L. Ku, Y.-J. Chen, and T. Q. S. Quek, "UAV trajectory, user association, and power control for multi-UAV-enabled energy-harvesting communications: Offline design and online reinforcement learning," *IEEE Internet Things J.*, vol. 11, no. 6, pp. 9781–9800, Mar. 2024.
- [27] Z. Chang, H. Deng, L. You, G. Min, S. Garg, and G. Kaddoum, "Trajectory design and resource allocation for multi-UAV networks: Deep reinforcement learning approaches," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 5, pp. 2940–2951, Sep./Oct. 2023.
- [28] S. Gong, M. Wang, B. Gu, W. Zhang, D. T. Hoang, and D. Niyato, "Bayesian optimization enhanced deep reinforcement learning for trajectory planning and network formation in multi-UAV networks," *IEEE Trans. Veh. Technol.*, vol. 72, no. 8, pp. 10933–10948, Aug. 2023.
- [29] M. Alzenad, A. El-Keyi, F. Lagum, and H. Yanikomeroglu, "3-D placement of an unmanned aerial vehicle base station (UAV-BS) for energy-efficient maximal coverage," *IEEE Wireless Commun. Lett.*, vol. 6, no. 4, pp. 434–437, Aug. 2017.
- [30] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam, and A. Nallanathan, "Deep reinforcement learning based dynamic trajectory control for UAV-assisted mobile edge computing," *IEEE Trans. Mob. Comput.*, vol. 21, no. 10, pp. 3536–3550, Oct. 2022.

- [31] Z. Yang, S. Bi, and Y.-J. A. Zhang, "Online trajectory and resource optimization for stochastic UAV-enabled MEC systems," *IEEE Trans. Wireless Commun.*, vol. 21, no. 7, pp. 5629–5643, Jul. 2022.
- [32] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, Apr. 2019.
- [33] Y. Ma, Y. Tang, Z. Mao, D. Zhang, C. Yang, and W. Li, "Energy-efficient 3D trajectory optimization for UAV-aided wireless sensor networks," in *Proc. IEEE Glob. Commun. Conf.*, Kuala Lumpur, Malaysia: IEEE, Dec. 2023, pp. 6591–6596.
- [34] Q. Wu, W. Chen, D. W. Kwan Ng, J. Li, and R. Schober, "User-centric energy efficiency maximization for wireless powered communications," *IEEE Trans. Wireless Commun.*, vol. 15, no. 10, pp. 6898–6912, Oct. 2016.
- [35] R. Ding, F. Gao, and X. S. Shen, "3D UAV trajectory design and frequency band allocation for energy-efficient and fair communication: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 7796–7809, Dec. 2020.



Zhao Tong (Senior Member, IEEE) received the PhD degree in computer science from Hunan University, Changsha, China, in 2014. From 2017 to 2018, he is currently an professor with Hunan Normal University. He was a visiting scholar with the Georgia State University during 2017–2018. He has authored or coauthored more than 50 papers in peer-reviewed international journals and conferences, such as *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Services Computing*, *IEEE Transactions on Network and Service Management*, *IEEE Transactions on Vehicular Technology*, etc. His research interests include AI computing, parallel and distributed computing systems, and resource management. He is a senior member of the CCF.



Shiyan Zhang received the bachelor's degree from Guangxi Minzu University, Nanning, China, in 2022. He is currently working toward the master's degree with the College of Information Science and Engineering, Hunan Normal University, Changsha, China. His research interests include revolve around the areas of UAV-assisted mobile edge computing and deep reinforcement learning.



Jing Mei received the PhD degree in computer science from Hunan University, China, in 2015. She is currently an associate professor with the Hunan Normal University, Changsha, China. Her research interests include parallel and distributed computing and cloud computing. She has published 40 research articles in international conference and journals, such as *IEEE-TC/TPDS/TSC/Cluster Comput./J. Grid Comput./J. Supercomput*, etc.



Jiayi Sun is currently working toward the bachelor's degree with the College of Information Science and Engineering, Hunan Normal University, Changsha, China. Her research interests include task offloading and scheduling in low Earth orbit (LEO) satellite-based mobile edge computing.



Keqin Li (Fellow, IEEE) received the BS degree in computer science from Tsinghua University, in 1985, and the PhD degree in computer science from the University of Houston, in 1990. He is a SUNY distinguished professor with the State University of New York and a national distinguished professor with Hunan University (China). He has authored or co-authored more than 1230 journal articles, book chapters, and refereed conference papers. He holds nearly 80 patents announced or authorized by the Chinese National Intellectual Property Administration. He is among the world's top few most influential scientists in parallel and distributed computing, regarding single-year impact (ranked #2) and career-long impact (ranked #3) based on a composite indicator of the Scopus citation database. He is listed in *Scilit Top Cited Scholars* (2023–2025) and is among the top 0.02% out of more than 20 million scholars worldwide based on top-cited publications in the last ten years. He is listed in *ScholarGPS Highly Ranked Scholars* (2022–2025) and is among the top 0.002% out of more than 30 million scholars worldwide based on a composite score of three ranking metrics for research productivity, impact, and quality in the recent five years. He received the IEEE TCCLD *Research Impact Award* from the IEEE CS Technical Committee on Cloud Computing in 2022 and the IEEE TCSVC *Research Innovation Award* from the IEEE CS Technical Community on Services Computing in 2023. He won the IEEE Region 1 *Technological Innovation Award* (Academic) in 2023. He was a recipient of the 2022–2023 *International Science and Technology Cooperation Award* and the 2023 *Xiaoxiang Friendship Award* of Hunan Province, China. He is a member of the SUNY Distinguished Academy. He is an AAAS fellow, an AAIA fellow, an ACIS fellow, and an AIIA Fellow. He is a member of the European Academy of Sciences and Arts. He is a member of Academia Europaea (Academician of the Academy of Europe).