# Cost-Efficient Deployment Optimization for Multi-UAV-Assisted Vehicular Edge Computing Networks

Yinan Liu, Chao Yang, Yanqun Tang, Haitao Zhao, Yi Liu, and Shengli Xie, *Fellow, IEEE*

***Abstract*—Taking into account the flexible deployment and Line-of-Sight (LoS) communication links of uncrewed aerial vehicles (UAVs), this article proposes a multi-UAV-assisted vehicular edge computing networks (VECNs) architecture to provide instantaneous computation support at multiple congestion road segments. Given that the computation resources of a single UAV are insufficient, and offloading tasks directly to the cloud computing center (CCC) in intelligent transportation systems (ITSs) introduces significant latency, multiple UAVs with precached service or content caching data are deployed optimally for the vehicle users. In order to address the tradeoff between system costs and service efficiency, we propose a novel cost-efficient layered optimization scheme, in which the number and deployment positions of UAVs are jointly optimized. According to the varying vehicular network environments and the dynamic requirements of vehicle users, we design a hierarchical reinforcement learning algorithm, combining double deep *Q* network (DDQN) and multiagent deep deterministic policy gradient (MADDPG), the former is used to optimize the number of UAVs, and the deployment of UAVs are optimized via the MADDPG. Simulation results demonstrate the effectiveness of the proposed scheme in lowering total task completed latency and increasing the system profits. The service efficiency in dealing with the vehicle users' requirements also be improved.**

***Index Terms*—Deployment, hierarchical reinforcement learning (HRL), multiagent reinforcement learning, uncrewed aerial vehicle (UAV), vehicular edge computing networks (VECNs).**

Received 15 June 2024; revised 4 October 2024 and 3 November 2024; accepted 25 November 2024. Date of publication 17 December 2024; date of current version 7 March 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62320106008 and Grant 62003094; in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515012745; and in part by the Shenzhen Science and Technology Major Project of China under Grant KJZD20240903300415. *(Corresponding author: Chao Yang.)*

Yinan Liu is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China, and also with Smart Products Division, Southern Power Grid Big Data Service Company Ltd., Guangzhou 510655, China (e-mail: drcamellia@qq.com).

Chao Yang, Yi Liu, and Shengli Xie are with the School of Automation and the Guangdong Provincial Key Laboratory of Intelligent Systems and Optimization Integration, Guangdong University of Technology, Guangzhou 510006, China (e-mail: chyang513@gdut.edu.cn; yi.liu@gdut.edu.cn; slxie@gdut.edu.cn).

Yanqun Tang is with the School of Electronics and Communications, Sun Yat-sen University, Shenzhen 518107, China (e-mail: tangyq8@mail.sysu.edu.cn).

Haitao Zhao is with the College of Electronic Science and Technology, National University of Defense Technology, Changsha 410073, China (e-mail: haitaozhao@nudu.edu.cn).

Digital Object Identifier 10.1109/JIOT.2024.3515107

## I. Introduction

In THE era of rapid technological advancement for the Internet of Things (IoT) and intelligent transportation systems (ITSs), the moving vehicles on the road become more intelligent, safer, and efficient, as well as offering a plethora of new application services to vehicle users [1], [2], [3]. The successful implementation of the emerging applications depends on the high-density data sharing and rapid task processing among the vehicle users, roadside units (RSUs), and cloud computing centers (CCC) in ITSs. To address the abundance of latency-sensitive and computation-intensive tasks, vehicular edge computing networks (VECNs) had garnered significant attention by providing powerful computing, communication, and storage services at the edge of vehicular networks. However, the traffic network has high-spatio-temporal variability, it is impractical to deploy a large number of RSUs or other fixed base stations (BSs) near the user side, due to the significant infrastructure costs and potential resource waste in low-utilization time periods. It is a big challenge to serve multiple vehicle users efficiently with limited resources at the appropriate times and locations.

Due to the high mobility and flexible hovering deployment, uncrewed aerial vehicles (UAVs) are integrated into ITSs to cover the vehicle users on road, especially in the congested road segments and intersections. UAVs plan the flying trajectory and hovering positions based on the dynamic vehicular network environments and requirements of users, thereby establishing Line-of-Sight (LoS) links with ground vehicles and passengers. In fact, UAVs perform as edge computing nodes or relays. For the limited computation and storage capacities, precaching part of service or content caching data on UAVs and deploying them to suitable locations become the efficient strategy to improve the utility of UAVs [4], [5]. The current caching schemes are divided into service caching and content caching [6], [7], [8]. For the former, the related program data are cached to support the complex tasks. Liu et al. [9] proposed a novel joint hybrid caching and replacement scheme in a scene where a single UAV assists RSU to cover a set of vehicle users with a large number of iterative calculation tasks, both the content caching and service caching are considered for the RSU and UAV. Particularly, the task offloading and service caching should be considered jointly [7], the covered vehicle users can only send tasks to UAVs that have cached the associated program data [8]. In dynamic topology networks, the offloading tasks in UAVs at different times and locations can be processed locally, relayed to RSU with the corresponding caching data, or further offloaded to CCC. For the UAV deployment, there exists a tradeoff between the UAV number and the quality of service (QoS) of users. Being closer to the user side enables faster communication rates, the service response latency is reduced, and the communication bandwidth consumption also be saved significantly. But a single UAV cannot meet the users' requirements in large areas when the number of operation UAVs increases and the system costs also increase. With caching suitable service and content data, multiple UAVs should be deployed under a collaboration scheme to cover vehicle users as much as possible.

An efficient multi-UAV collaborative scheme in VECNs can balance the system's overall communication and computation resource distribution [10]. The objectives are set as maximizing global revenue for UAVs or minimizing task completed latency for the covered users [11]. To enable UAVs to better handle tasks from vehicle users, it is crucial for UAVs to adjust their deployment positions based on the dynamic ground users' requirements and data transmission environments [12]. The current research often focuses on UAV network operation within a single service area or a fixed number of UAVs and deployment positions. The number of operation UAVs and their deployment positions affect the system costs and the QoS of the covered vehicle users. In addition, the computation resource collaboration and service range among UAVs also need to be optimized to maximize the network revenue with limited storage and computing capacities of UAVs. In details, three key issues should be addressed, as follows.

1) *How to select the number of precached UAVs based on the environment information and vehicle users' requirements?* Compared to RSUs, UAVs have limited caching and computing resources and operational time periods. With precaching part of service and content caching data, selecting a reasonable number of UAVs and then performing computing or relay operations can reduce the system costs.
2) *How to perform task offloading cooperatively between multiple UAVs and the covered users?* The offloading tasks can be processed on the current UAV, relayed to other UAVs, or further offloaded to CCC of ITSs. Users should select the optimal node among the currently communicable UAVs. To reduce task completed latency, the communication and computation resources should be considered based on the service and content caching conditions.
3) *How to optimize and adjust the deployment positions of multiple UAVs?* When the ground vehicle users' positions and requirements change, an efficient scheme should be proposed to coordinate the optimization of numerous UAVs' movements to better cope with the dynamic vehicle users' requirements.

In this article, we propose a cost-efficient multiple UAV deployment scheme in VECNs to reduce the system cost and the task completed latency. The UAVs precached part of service and content caching data, find the optimal deployment positions cover the vehicle users in multiple congested road segments or intersections. Actually, when more UAVs are deployed, the task completed latency will be reduced directly, however, the system costs also increase. And it is hard to satisfy the dynamic requirements of users. Thus, we optimize the number of UAVs and the deployment positions jointly. Specially, taking into account the various computation resources, deployment positions of UAVs, and user requirements, a hierarchical reinforcement learning (HRL) algorithm combines double deep $Q$ network (DDQN) and MADDPG is proposed, with each UAV acting as an independent agent to optimize its deployment position and state. The major contributions are described as follows.

1) We present a multi-UAV-assisted VECNs system model with multiple UAVs and multiple service areas. The UAVs precached part of suitable service caching and content caching data via hybrid caching. As temporary aerial BSs, UAVs provide computation resources to vehicle users on multiple congested road segments. The dynamic requirements of users and data transmission environments are mainly considered.
2) We design a layered optimization scheme for the deployment of multiple UAVs and a cost-efficient optimization of the number of UAVs, the system profit is maximized and the whole task completed latency is minimized.
3) We propose a HRL algorithm that combines DDQN and MADDPG. It optimizes the number and deployment positions of UAVs in a hierarchical manner, meeting the dynamic demands of vehicular users and achieving efficient resource utilization.

The remainder of this article is organized as follows. Section II reviews related works. Section III gives the system model. Section IV elucidates the hierarchical optimization scheme for the optimization of UAV numbers and deployment locations. Section V presents numerical examples to illustrate the performance evaluation of the proposed schemes. Conclusions are drawn in Section VI.

## II. Related Work

In this section, we introduce the basic mechanism of UAV-assisted VECNs and summarize the current research on deployment location optimization in multi-UAV-assisted VECNs.

### A. UAV-Assisted VECNs

Along with the rapid development of ITSs, VECNs leverage computation, communication and storage resources to enhance the data processing capabilities and the data sharing efficiency. To cope with the sudden increasing requirements due to traffic congestion, UAVs are introduced [13] to extend the coverage range of RSUs, the task completed latency and network resiliency are improved. However, for the limited storage and energy capacities, practical applications of UAVs still face numerous challenges, including energy management of UAVs, security of data transmission, collaborative optimization between UAVs and vehicles, and network management [14], [15], [16]. A novel multihop collaborative scheme among UAVs was proposed, the user association, task offloading and other resource allocation were jointly optimized [16]. Furthermore, efficient dynamic adjusting deployment and flying trajectory of UAVs to meet the charging requirements of ground vehicle users remains a hot research topic. Wang et al. [17] introduced a trajectory control algorithm based on multiagent reinforcement learning for managing the movement trajectories of multiple UAVs. Mukherjee et al. [18] and Mohammed et al. [19] presented the low-complexity approaches to optimize task offloading decisions among multiple UAVs. Zeng et al. [20] explored the potential of UAVs as caching devices to provide layered video services to users. Shen et al. [21] focused on minimizing completion time for UAV content delivery. Feng et al. [22] proposed a method to calculate the phased mission reliability of UAV swarms, addressing challenges in determining the optimal number of UAVs for maximizing mission reliability. To address the limited energy budget of UAVs, Liu et al. [23] proposed a set of UAV computation and communication offloading schemes that classify ground users into task collection nodes and other nodes based on data volume within the current coverage area, the UAVs can only fly to the former.

### B. Deployment of UAVs in VECNs

In VECNs, the deployment locations and the operation number of UAVs are crucial factors that affect the network's coverage range, service quality, and operational efficiency. Proper planning of the UAV number can ensure the full utilization of network resources, avoiding resource wastage or insufficient service. Meanwhile, precise UAV deployment schemes [24], [25] can optimize network coverage performance and enhance the reliability of data transmission. Existing works have adopted various algorithmic models, including dynamic adjustment strategies, energy consumption optimization, multi-UAV collaborative operations, and deployment adjustments based on real-time varying demands [26]. In addition, the UAVs with precaching data need to be deployed carefully, for the service caching or content caching should be considered differently. Chai and Lau [27] introduced a joint trajectory and communication scheduling scheme for multi-UAV supported wireless caching networks. Zheng et al. [28] developed a strategy for service caching, task offloading, communication, and computation resource allocation among multiple UAVs, as well as UAV layout optimization to ensure the energy budget of all devices and UAVs. A hybrid caching and UAV deployment scheme was proposed, the cooperation between the UAV and RSU was analyzed [9]. Moreover, the deployment of multiple UAVs or UAV swarms becomes complex. Yao et al. [29] studied a joint deployment, power control, and channel access optimization strategy within a unified UAV swarm. Bayessa et al. [30] proposed a joint optimization algorithm for user grouping, UAV deployment and multicast precoder optimization. Liu et al. [31] presented a UAV-distributed, location-aware task offloading scheme to provide low-cost and resource-constrained services. Wang et al. [32] proposed a CNN-based UAV deployment method that can quickly respond to users' communication demands and improve network throughput. Chen and Zheng [33] explored how to deploy UAVs at optimal locations to provide task offloading using the time division multiple access (TDMA) protocol. Zheng et al. [34] presented a UAV-assisted IoT system and proposed a joint optimization algorithm of UAV deployment and resource allocation to determine UAV positions, the relationships between resources and rewards are balanced.

TABLE I
MAIN NOTATIONS

| Notations | Description |
|---|---|
| $P_{u_x}$, $F_{u_x}$, $E_{u_x}$ | The cache capacity, comprehensive computational resources, and endurance capability of the UAV $u_x$ |
| $pos_{u_x}$ | The current deployment position of the UAV $u_x$ |
| $pop_{S_{1n}}, pop_{S_{2m}}$ | The popularity of the service/content data |
| $p_{S_{1n}}, p_{S_{2m}}$ | The necessary cache space |
| $s_{1n}, s_{2m}$ | The type of service/content data |
| $tr_{i,j}$ | The latency constraint |
| $\Theta^t_{u_x,\xi}$ | The elevation angle between UAV and either UAV $u_x$ or vehicle user $i$ |
| $N_{u_x,i}$ | The multi-channel transmission interference between UAV $u_x$ and vehicle user $i$ |
| $N_{u_{x'},u_{x'+1}}$ | The multi-channel transmission interference between UAVs $u_{x'}$ and $u_{x'+1}$ |
| $C^{u_x}_{s_{1n}}, C^{u_x}_{s_{2m}}$ | The caching parameter for the service and content caching data in UAV $u_x$ |
| $F^{remain}_{u_x}$ | The remaining computational resources on the UAV $u_x$ |
| $size^{result}_{task_{i,j}}$ | The data size of the task processing result $task_{i,j}$ |
| $T^{cloud}_{task_{i,j}}$ | The transmission delay of the UAV offloading the task to the CCC |
| $T^{cloudBack}_{task_{i,j}}$ | The transmission delay of the computation results back to the UAV |
| $\alpha_{i,j}$ | The offloading decision of the task $task_{i,j}$ |
| $\beta^{u_x}_1$, $\beta^{u_x}_2$, $\beta^{u_x}_3$, $\beta^{u_x}_4$ | The cost weights |
| $MT_{u_x}$ | The maintenance and additional costs due to the factors such as weather conditions |
| $\delta^{u_x}_0$, $\delta^{u_x}_1$, $\delta^{u_x}_2$, $\delta^{u_x}_3$ | The influence weight of the environment in which the UAV $u_x$ operates. |

However, the existing works only focus on optimizing either the number or the deployment locations of UAVs, failing to serve vehicle users from multiple congested road segments and intersections. Unlike the cited studies, we consider a more practical scenario that multiple UAVs with precached part of service and content caching data, cover the vehicle users in multiple areas. We examine a layered optimization scheme for both the UAV number and deployment locations, to maximize the overall system profit while decreasing the task completed latency. The collaborative deployment positions among multiple UAVs and the changing demands of vehicle users are considered primarily. The main notations used in this article is summarized in Table I.

## III. System Model

The system model of multi-UAVs assisted VECNs is illustrated in Fig. 1, where multiple ground vehicle users are connected with UAVs and CCC in ITS via advanced 5G/B5G networks. To meet the dynamic user demands from traffic congestion road segments, UAVs with precached service caching and content caching data [9] to cover the computation insensitive and latency-sensitive tasks of vehicle users. Moreover, the received tasks can be further offloaded to nearby

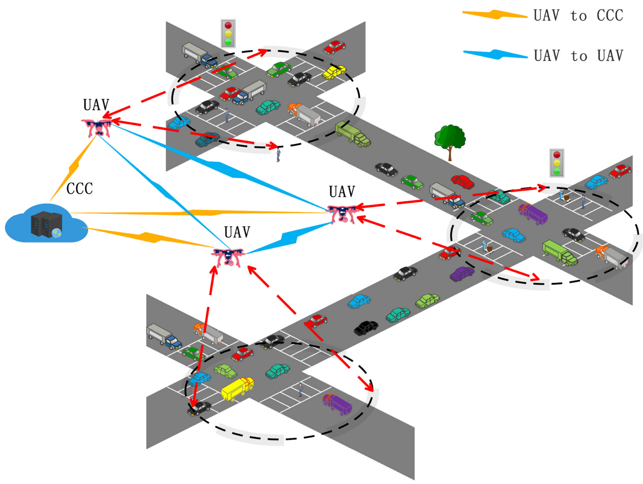



Fig. 1. System model.

UAVs or CCC for processing. In this article, under the various traffic flow and vehicle users' demands, the number and deployment positions of UAVs are jointly optimized to reduce the system cost and enhance the user experience.

Set UAV $u_x$ with a cache capacity denoted as $P_{u_x}$, comprehensive computational resources $F_{u_x}$, and an overall endurance capability as $E_{u_x}$, where $x \in \{1, 2, \ldots, X\}$, $\text{pos}_{u_x}$ denotes the current deployment position of the UAV, which cannot exceed the defined constraint area $\text{Pos}_{re}$, $\text{pos}_{u_x} \in \text{Pos}_{re}$. The required service and content data of users, denoted as $S_1 = \{S_{11}, \ldots, S_{1n}, \ldots, S_{1N}\}$, with $n \in \{1, 2, \ldots, N\}$. Each piece of service data is made of a triplet $S_{1n} = <pop_{S_{1n}}, ps_{S_{1n}}, s_{1n}>$, where $pop_{S_{1n}}$ represents the popularity of the service data, $ps_{S_{1n}}$ signifies the necessary cache space, and $s_{1n}$ identifies the type of service data. Similarly, content data is represented as $S_2 = \{S_{21}, \ldots, S_{2m}, \ldots, S_{2M}\}$, with $m \in \{1, 2, \ldots, M\}$. Each piece of content data consists of a triplet $S_{2m} = <pop_{S_{2m}}, ps_{S_{2m}}, s_{2m}>$, which is structured similarly. Each vehicle user generates multiple tasks, the task $j$ generated by vehicle user $i$ is denoted as $\text{task}_{i,j} = <\text{type}_{i,j}, \text{size}_{i,j}, cr_{i,j}, tr_{i,j}>$, $i \in \{1, 2, \ldots, I\}$, $j \in \{1, 2, \ldots, J\}$, where $\text{type}_{i,j}$ denotes the type of the task, $\text{size}_{i,j}$ represents the required storage space, $cr_{i,j}$ indicates the computational resources required when processing the task with service data, and $tr_{i,j}$ specifies the latency constraint.

### A. Multi-UAV Communication Model

Set that vehicles are equipped with multiple communication technologies. The dynamic TDMA is used to ensure the transmission among the vehicles and UAVs [35], [36]. Each vehicle user individually utilizes the entire main frequency band for a certain time period. Bandwidth utilization is managed collaboratively by the UAV swarm, which acts as the central coordinator. The main frequency band is allocated to vehicle users and UAVs to prevent channel interference from other users. The time range for UAV-to-multiple vehicle users is divided into $T$ equal time slots, indexed by $t \in \{1, 2, \ldots, T\}$ [37]. We consider the transmission links between UAVs and vehicle users [38] as both the LoS and Non-Line-of-Sight (NLoS) propagation models. The path loss can be represented as

$$PL_{\text{LoS}} = 20B \log\left(\frac{4\pi f_c d_u}{c}\right) + \eta_{\text{LoS}} \tag{1}$$

$$PL_{\text{NLoS}} = 20B \log\left(\frac{4\pi f_c d_u}{c}\right) + \eta_{\text{NLoS}} \tag{2}$$

where $d_u = \{d^t_{u_x,i}, d^t_{u_x,u_{x+y}}, d^t_{u_x,\text{cloud}}\}$ denotes the distance between UAV $u_x$ and vehicle user $i$ within time slot $t$, the distance between UAV $u_x$ and UAV $u_{x+y}$, and the distance from UAV $u_x$ to the CCC. $f_c$ represents the carrier frequency, $\eta_{\text{LoS}}$ and $\eta_{\text{NLoS}}$ correspond to the additional losses incurred by LoS and NLoS links, respectively. The probability of LoS links for UAVs and vehicle users depends on their elevation and environment, as

$$\text{PrLoS} = \frac{1}{1 + X_0 \exp\left(-Y_0(\Theta^t_{u_x,\xi} - X_0)\right)} \tag{3}$$

$$\text{PrNLoS} = 1 - Pr_{\text{LoS}} \tag{4}$$

where $\Theta^t_{u_x,\xi}$, $\xi \in \{u_{x+y}, i\}$ represents the elevation angle between UAV $u_x$ and either UAV $u_{x+y}$ or vehicle user $i$, and $X_0$ and $Y_0$ are environmental parameters. The average path loss is given as

$$\text{PL} = \text{Pr}_{\text{LoS}} \times PL_{\text{LoS}} + Pr_{\text{NLoS}} \times PL_{\text{NLoS}}. \tag{5}$$

The data transmission rate between UAV $u_x$ and vehicle user $i$ is

$$R^{\text{trans}}_{u_x,i} = B \log_2\left(1 + \frac{P_{u_x,i}}{10^{PL/10}\sigma^2 + N_{u_x,i}}\right). \tag{6}$$

When the task becomes complex or the corresponding UAV does not cache the related service caching data in advance, other UAVs can help to process the task via multihop communications between UAVs, the transmission rate between UAV $u_{x'}$ and $u_{x'+1}$ is

$$R^{\text{trans}}_{u_{x'},u_{x'+1}} = B \log_2\left(1 + \frac{P_{u_{x'},u_{x'+1}}}{10^{PL/10}\sigma^2 + N_{u_{x'},u_{x'+1}}}\right) \tag{7}$$

where $B$ is the bandwidth, $x' \in X'$. $N_{u_x,i}$ and $N_{u_{x'},u_{x'+1}}$ are the multichannel transmission interference between UAV $u_x$ and vehicle user $i$, UAVs $u_{x'}$ and $u_{x'+1}$. The data transmission rate for returning the task processed results is fixed at $R^{UV}_{\text{back}}$. The transmission rate between UAV $u_x$ and CCC is

$$R^{\text{trans}}_{u_x,\text{cloud}} = B_{u_x} \log_2\left(1 + \frac{P_O}{10^{PL/10}\sigma^2 + N_O}\right) \tag{8}$$

where $B_{u_x}$ represents the wireless channel bandwidth allocated to UAV $u_x$ by the CCC and $P_O$ denotes the transmission power. Given the CCC's abundant computational resources and cached service data, the computation delay for tasks within the CCC is minimal and negligible.

### B. UAV Deployment and Task Offloading

Suffering from limited energy and storage capacities of a single UAV and the time-varying requirements from multiple congested road segments, UAVs precached appropriate service and content caching data perform as computing or relay nodes for the vehicle users. Set $\mathcal{C}_{u_x} = \{C^{u_x}_{s_{1n}} \in \{0, 1\}, C^{u_x}_{s_{2m}} \in$

$\{0, 1\}\}$ as the caching parameter for the service and content caching data in UAV $u_x$, where $C^{u_x}_{s_{1n}}$ indicates whether the service $S_{1n}$ is cached in UAV $u_x$ and $C^{u_x}_{s_{2m}}$ indicates whether the content $S_{2m}$ is cached. The caching status of each UAV is independent. For content caching data, it can process offloaded tasks directly, whereas service data caching requires the computational resources of UAVs. Given that multiple UAVs can cache the same types of service or content data, but the same type of service and content data cached on a single UAV will be in conflict, thus

$$C^{u_x}_{s_{1n}} + C^{u_x}_{s_{2m}} \le \begin{cases} 1, \text{ if } n = m \\ 2, \text{ if } n \ne m. \end{cases} \quad (9)$$

Ensuring the safety of UAV swarm flights is critical, the position of UAVs changes over time. The distance between any two UAVs has to satisfy a minimum safe distance constraint to prevent collisions, as

$$D\Big(\text{pos}_{u_x}, \text{pos}_{u_{x+y}}\Big) \ge D_{\min} \quad (10)$$

where $D(\text{pos}_{u_x}, \text{pos}_{u_{x+y}})$ denotes the distance between UAVs $u_x$ and $u_{x+y}$, $\text{pos}_{u_x}$ denotes the position of UAV $u_x$, and $D_{\min}$ is the minimum safe distance to prevent UAVs from colliding. It is worth noting that the minimum safe distance $D_{\min}$ is much smaller than the communication coverage of any UAV.

In the multi-UAV-assisted VECNs, according to the caching situations, the task offloading schemes can be categorized into three scenarios.

1) *The UAVs cached the type of service or content data matching the offloading tasks*. For the processing latency of a single task on UAV, it consists of three components: a) the transmission latency of the vehicle to offload the task to UAV; b) the computation latency, which only applies to processing the task via service data; and c) content data processing does not involve computation latency. And the transmission latency of the task computation results back to the vehicle. Therefore, the total latency in processing task $\text{task}_{i,j}$ is

$$T^{1}_{\text{task}_{i,j}} = T^{\text{trans}}_{\text{task}_{\text{i,j}}} + C^{u_x}_{s_{1n}}\big(1 - C^{u_x}_{s_{2m}}\big)T^{\text{compute}}_{\text{task}_{i,j}} + T^{\text{back}}_{\text{task}_{i,j}} \quad (11)$$

where $T^{\text{trans}}_{\text{task}_{i,j}} = ([\text{size}_{i,j}]/[R^{\text{trans}}_{u_x,i}])$ is the transmission delay of the task $\text{task}_{i,j}$ offloaded from the vehicle user $i$ to UAV $u_x$. $T^{\text{compute}}_{\text{task}_{i,j}} = ([cr_{i,j}]/[F^{\text{remain}}_{u_x}])$ is the computation latency, $F^{\text{remain}}_{u_x}$ denotes the remaining computational resources on the UAV $u_x$, and $T^{\text{back}}_{\text{task}_{i,j}} = ([\text{size}^{\text{result}}_{\text{task}_{i,j}}]/[R^{UV}_{\text{back}}])$ is the return delay of the computational results of the task $\text{task}_{i,j}$, $\text{size}^{\text{result}}_{\text{task}_{i,j}}$ is the data size of the task processing result $\text{task}_{i,j}$.

2) *When a UAV within the communication range does not cache the service or content data for the corresponding type of task, but other UAVs do, the task can be processed by the target UAV through a multihop relay link*. The task processing in this case covers four stages: a) the transmission of the task offloaded from the vehicle user $i$ to UAV $u_x$; b) the transmission of the task to the target UAV $u_{x'}$ via the multihop relay; c) the task computation delay on the target UAV $u_{x'}$; and d) the delay of the task's computation result back to the vehicle user. Among them, we set that the task needs at least $X'$ UAVs relaying hops to the target UAV. The task completed latency is

$$T^{2}_{\text{task}_{i,j}} = T^{\text{trans}}_{\text{task}_{i,j}} + T^{\text{multihop}}_{\text{task}_{i,j}} + C^{u_x}_{s_{1n}}\big(1 - C^{u_x}_{s_{2m}}\big)T^{\text{compute}}_{\text{task}_{i,j}} + T^{\text{back}}_{\text{task}_{i,j}} \quad (12)$$

where $T^{\text{multihop}}_{\text{task}_{i,j}} = \sum_{x'}^{X'+1} T^{\text{trans}}_{u_{x'},x'+1}$ denotes the transmission delay of task $\text{task}_{i,j}$ over multiple hops in the UAV swarm, and $T^{\text{trans}}_{u_{x'},x'+1} = ([\text{size}_{i,j}]/[R^{\text{trans}}_{u_{x'},x'+1}])$ denotes the transmission delay of the task between UAV $u_{x'}$ and UAV $u_{x'+1}$.

3) *When none of the UAVs cached the service and content data of the task, it can only be further offloaded to the CCC for processing via the UAVs*. In this case, the task needs to be further offloaded to CCC for processing via the nearest UAV within communication range, the task completed latency consists of four parts: a) the task transmission latency from the vehicle user to the UAV; b) the transmission latency for the UAV to further offload the task to the CCC; c) the transmission latency for the task's computation results back to the vehicle user, as

$$T^{3}_{\text{task}_{i,j}} = T^{\text{trans}}_{\text{task}_{i,j}} + T^{\text{cloud}}_{\text{task}_{i,j}} + T^{\text{cloudBack}}_{\text{task}_{i,j}} + T^{\text{back}}_{\text{task}_{i,j}} \quad (13)$$

where $T^{\text{cloud}}_{\text{task}_{i,j}}$ denotes the transmission delay of the UAV offloading the task to the CCC; and d) $T^{\text{cloudBack}}_{\text{task}_{i,j}}$ denotes the transmission delay of the computation results back to the UAV.

In summary, the processing latency of task $\text{task}_{i,j}$ is

$$T_{\text{task}_{i,j}} = -\alpha_{i,j}T^{1}_{\text{task}_{i,j}} + \big(1 + \alpha_{i,j}\big)[\big(1 - \alpha_{i,j}\big)T^{2}_{\text{task}_{i,j}} + \frac{\alpha_{i,j}}{2}\Big(T^{3}_{\text{task}_{i,j}} + T^{1}_{\text{task}_{i,j}}\Big)] \quad (14)$$

where $\alpha_{i,j}$ denotes the offloading decision of the task $\text{task}_{i,j}$, $\alpha_{i,j} = -1$ means that the task is directly processed on the UAVs within the current communication range, $\alpha_{i,j} = 0$ means that the task is processed through the UAV swarm multihop links to the target UAVs, and $\alpha_{i,j} = 1$ means that the task is further offloaded to CCC. Therefore, the processing latency of task $\text{task}_{i,j}$ needs to satisfy the constraint as

$$T_{\text{task}_{i,j}} \le tr_{i,j}. \quad (15)$$

### C. Problem Formulation of Multi-UAV Deployment

The multi-UAV deployment and task offloading model focuses on how to dynamically adjust the locations of UAVs and handle the offloaded tasks according to the current network states and user demands. The service efficiency and user satisfaction are maximized. The model takes into account the UAV's communication capability, computation capability, and its relative positions to the covered users. In summary, the final task offloading latency is

$$T_{\text{task}} = \sum_{i=1}^{I}\sum_{j=1}^{J} T_{\text{task}_{i,j}}. \quad (16)$$

The total number of tasks handled by the UAV swarm is

$$\mathrm{Num}_{\mathrm{task}} = \sum_{x=1}^{X} \mathrm{Num}_{\mathrm{task}}^{u_x} \tag{17}$$

where $\mathrm{Num}_{\mathrm{task}}^{u_x}$ denotes the number of tasks handled by UAV $u_x$. Set $([\mathrm{Num}_{\mathrm{task}}]/[T_{\mathrm{task}}])$ as the user service efficiency. To maximize the user service efficiency, the joint optimization problem $P1$ for the multi-UAV deployment location and task offloading scheme is shown as

$$\begin{aligned} \mathbf{P1}:\quad & \max_{\{\mathrm{pos}_{u_{x\in X}}, \alpha_{i,j}\}} \frac{\mathrm{Num}_{\mathrm{task}}}{T_{\mathrm{task}}} \\ \text{s.t.}\quad & C1: \; C_{s_{1n}}^{u_x} + C_{s_{2m}}^{u_x} \le \begin{cases} 1, \text{ if } n = m \\ 2, \text{ if } n \ne m \end{cases} \\ & C2: \; T_{\mathrm{task}_{i,j}} \le tr_{i,j} \\ & C3: \; F_{u_x}^{\mathrm{remain}} \le F_{u_x} \\ & C4: \; \alpha_{i,j} \in \{-1, 0, 1\} \\ & C5: \; \mathrm{pos}_{u_x} \in \mathrm{Pos}_{re} \\ & C6: \; D\left(\mathrm{pos}_{u_x}, \mathrm{pos}_{u_{x+y}}\right) \ge D_{\min} \end{aligned}$$

where $C1$ represents constraints on the UAV hybrid caching parameters to ensure that the service and content data cached by the UAV conform to the desired policy, $C_{s_{1n}}^{u_x} \in \{0, 1\}$, and $C_{s_{2m}}^{u_x} \in \{0, 1\}$. $C2$ involves constraints on the task processing latency; $C3$ is a constraint on the computation resources of UAV; $C4$ determines where the task is processed to optimize the network performance. $C5$ is a constraint for the UAV's deployment position, and C6 specifies the minimum safe distance constraint to prevent collisions. These constraints form a comprehensive optimization framework that adjusts UAV behaviors and task offloading selection to optimize system performance.

### D. Cost-Efficient Optimization of UAV Number

For problem $P1$, we can find that under the constraint $C5$, as the number of UAVs increases, the system performance will be improved directly. However, the system costs also become huge. Moreover, the dynamic vehicular network environments and requirements of vehicle users in multiple congested road segments make that it is not efficient when increasing the number of UAVs directly and disorderly. Thus, in this section, we focus on how to optimize the number of UAVs to maximize network profit. Considering that the service provided by UAVs incurs certain costs, which are influenced by UAV endurance (i.e., battery capacity), caching capacity, and total computation resources. Hence, a UAV cost function is constructed based on not only the direct operational costs of UAVs, such as battery replacement or charging and hardware maintenance, but also the indirect costs, such as potential communication costs generated by UAV deployment, as well as the energy consumption costs when UAVs perform caching and processing for the tasks. Moreover, we consider various environmental factors UAVs may encounter during task execution, such as weather conditions and airspace restrictions, these factors can also affect the system's operational efficiency and cost of UAVs.

Calculating the marginal cost of services provided by UAVs, the optimal number of UAVs is determined to respond to urban road congestion. The cost function of UAV $u_x$ is

$$\mathrm{Cost}_{u_x} = \beta_1^{u_x} E_{u_x} + \beta_2^{u_x} P_{u_x} + \beta_3^{u_x} F_{u_x} + \beta_4^{u_x} MT_{u_x} \tag{18}$$

where $\beta_1^{u_x}$, $\beta_2^{u_x}$, $\beta_3^{u_x}$, and $\beta_4^{u_x}$ are the cost weights, $E_{u_x}$ is the endurance of the UAV $u_x$, $P_{u_x}$ and $F_{u_x}$ are the cache and computation resources, respectively, and $MT_{u_x}$ is the maintenance and additional costs due to the factors, such as weather conditions.

Given that task processing through service caching data consumes computation resources, the benefit function of the service caching-based task processing is expressed as

$$\begin{aligned} \mathrm{Earn}_{\mathrm{task}_{i,j}}^{S_1,u_x} = \delta_0^{u_x} \frac{tr_{i,j}}{T_{\mathrm{task}_{i,j}}} \Bigg( & \delta_1^{u_x} \frac{\mathrm{size}_{i,j}}{P_{u_x}^{\mathrm{remain}}} + \delta_2^{u_x} \frac{cr_{i,j}}{F_{u_x}^{\mathrm{remain}}} \\ & + \delta_3^{u_x} \frac{ec_{i,j}}{E_{u_x}^{\mathrm{remain}}} \Bigg) \end{aligned} \tag{19}$$

where $\delta_0^{u_x}$, $\delta_1^{u_x}$, $\delta_2^{u_x}$, and $\delta_3^{u_x}$ are the weights, which are affected by the environment where UAV $u_x$ is located. $ec_{i,j} = \gamma_0 T_{\mathrm{task}_{i,j}}$ denotes the required computation resources, $\gamma_0$ is the energy consumption per time unit for task processing, and $P_{u_x}^{\mathrm{remain}}$, $F_{u_x}^{\mathrm{remain}}$, and $E_{u_x}^{\mathrm{remain}}$ denote the residual storage space, the residual computation resources, and the residual endurance capacity of UAV $u_x$, respectively. Similarly, the gain function for the task processing with content caching data is

$$\mathrm{Earn}_{\mathrm{task}_{i,j}}^{S_2,u_x} = \delta_0^{u_x} \frac{tr_{i,j}}{T_{\mathrm{task}_{i,j}}} \left( \delta_1^{u_x} \frac{\mathrm{size}_{i,j}}{P_{u_x}^{\mathrm{remain}}} + \delta_3^{u_x} \frac{ec_{i,j}}{E_{u_x}^{\mathrm{remain}}} \right). \tag{20}$$

Above all, we obtain the gain function for processing task $\mathrm{task}_{i,j}$ on UAV $u_x$ as

$$\mathrm{Earn}_{\mathrm{task}_{i,j}}^{u_x} = C_{s_{1n}}^{u_x} \mathrm{Earn}_{\mathrm{task}_{i,j}}^{S_1,u_x} + C_{s_{2m}}^{u_x} \mathrm{Earn}_{\mathrm{task}_{i,j}}^{S_2,u_x}. \tag{21}$$

### E. Problem Formulation of Optimization of UAV Number

The cost efficient multi-UAV deployment model focuses on how to control and optimize overall system operation costs while meeting service demands of vehicle users on road. It determines the appropriate number of UAVs via calculating the marginal benefits and costs.

The total benefit of UAVs after processing the offloaded task is

$$\mathrm{Earn}_{\mathrm{task}} = \sum_{x=1}^{X} \sum_{i=1}^{I} \sum_{j=1}^{J} \mathrm{Earn}_{\mathrm{task}_{i,j}}^{u_x}. \tag{22}$$

Similarly, the total system cost of UAVs is

$$\mathrm{Cost}_u = \sum_{x}^{X} \mathrm{Cost}_{u_x}. \tag{23}$$

Then, the cost efficient UAV number optimization problem P2 is formulated as

$$\begin{aligned} \mathbf{P2}:\quad & \max_{X} \left(\mathrm{Earn}_{\mathrm{task}} - \mathrm{Cost}_{u_x}\right) \\ \text{s.t.}\quad & C7: \; \sum_{u_x} ec_{i,j} \le E_{u_x} \end{aligned}$$

$$C8: \quad C^{u_x}_{s_{1n}} + C^{u_x}_{s_{2m}} \leq 1$$
$$C9: \quad \text{size}_{i,j} \leq P^{\text{remain}}_{u_x} \leq P_{u_x}$$
$$C10: \quad cr_{i,j} \leq F^{\text{remain}}_{u_x} \leq F_{u_x}$$
$$C11: \quad ec_{i,j} \leq E^{\text{remain}}_{u_x} \leq E_{u_x}$$
$$C12: \quad T_{\text{task}_{i,j}} \leq tr_{i,j}$$

where C7 is a constraint on the energy consumption of UAV. C8 is the mixing parameters of UAV. C9 is the caching space required for the task offloaded to UAV cannot exceed the remaining storage space. C10 can ensure that the required computation resources cannot exceed the remaining resources. C11 ensures the energy consumption must be less than the remaining energy. C12 ensures the QoS of vehicle users.

## IV. Hierarchical Reinforcement Learning Algorithm Design

For the task offloading of the vehicle users, there are two issues that should be considered. First, the tasks are initially offloaded to UAVs within their communication range, regardless of whether the tasks are ultimately processed on the UAVs or not. Second, for the service caching, the tasks should be offloaded to the UAVs that had cached the corresponding service caching data. It is decided whether to process the tasks directly on the current UAV, forward them through multiple hops to other UAVs, or further offload them to CCC for processing.

Furthermore, UAVs can dynamically adjust their deployment positions based on the real time requirements of vehicle users, the movement locations of vehicles, and the cached service and content data on UAVs to ensure that the maximum number of tasks can be processed in the shortest possible time. To achieve the objective, we employ a HRL optimization strategy [39], [40], combined with deep reinforcement learning (DRL) techniques, to collaboratively optimize the deployment positions and the number of multiple UAVs. From the perspective of vehicle users, this model preemptively considers the current traffic congestion on multiple road segments and the access requests of vehicle users, to perform a reasonable collaborative optimization of the UAVs' deployment positions. Given the mutual coupling of positional movements among multiple UAVs and the dynamic vehicles' positions and demands, we utilize multiagent DDPG techniques to optimize the deployment positions of multiple UAVs. The original problem is decomposed into two subproblems to be solved sequentially, using HRL to optimize the number of UAVs in the outer layer first, and then optimize the deployment locations of the UAV swarm in the inner layer based on the output of the outer layer. The network environment is variable and the optimal solution can be obtained under such conditions using reinforcement learning methods.

The HRL network framework is shown in Fig. 2. In the outer layer, the DDQN algorithm is used to optimize the number of UAVs, and then the number of UAVs optimized in the outer layer is passed as input data to the inner layer. In the inner layer, the outer layer passes the input data and the MADDPG algorithm to adjust the deployment position of each UAV, and finally, the optimized UAV swarm positions are fed back to the outer layer as a way to optimize the overall network latency and profitability. Through this collaborative optimization scheme, efficient and low-latency services are provided to vehicle users, to maximize the overall service revenue and minimize task completion latency, Finally, the service efficiency and satisfaction of vehicle users are enhanced.

As shown in Fig. 2, the design of HRL in the multi-UAV deployment location cooperative optimization scheme is as follows: define a quintuple $\mathcal{M} = < \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \Pi >$, which represents a continuous Markov decision process for multi-UAV cooperation, where $\mathcal{S}$ is the state space of UAVs, $\mathcal{A}$ denotes the action space, $\mathcal{R}$ denotes the global reward function, $\mathcal{P}$ is the state transfer rule, and $\Pi$ is the policy of multi-UAV collaborative learning. The specific meanings of these elements are as follows.

*State*: The state space is divided into two levels: 1) internal and 2) external ones. The external state, labeled as $\mathcal{S}_{\text{outer}} = (\text{num}^{re}_{\text{task}}, \text{num}^{re}_{\text{uav}}, \text{cost}^{\text{total}}_{\text{uav}}, \text{profit}^{\text{total}}_{\text{uav}}, \text{time}^{\text{total}}_{\text{uav}}, \text{num}^{cd}_{\text{user}})$, consists of several components: $\text{num}^{re}_{\text{task}}$, which is the number of tasks that have been completed. $\text{num}^{re}_{\text{uav}}$ is the number of UAVs; $\text{cost}^{\text{total}}_{\text{uav}}$ is the total cost associated with deploying the UAVs. $profit^{\text{total}}_{\text{uav}}$ is the total revenue generated. $time^{\text{total}}_{\text{uav}}$ denotes the overall delay in task processing; and $\text{num}^{cd}_{user}$ is the number of users covered by the UAV network. $\mathcal{S}_{\text{inner}}$ represents the state space of the environment for multiple UAVs. Assume the joint state is $\mathcal{S}_{\text{inner}} = (s^{u_1}_t, \ldots, s^{u_x}_t, \ldots, s^{u_X}_t)$, where $s^{u_x}_t$ denotes the state of UAV $u_x$ at time slice $t$, which is composed of a quintuple denoted as $s^{u_x}_t = < cn^{u_x}_t, pn^{u_x}_t, \text{num}^{u_x}_t, pt^{u_x}_t, time^{u_x}_t >$, where $cn^{u_x}_t$ represents the number of vehicular users within the communication range of the UAV $u_x$ at time slice $t$, $pn^{u_x}_t$ denotes the deployment position, $\text{num}^{u_x}_t$ represents the number of tasks processed, $pt^{u_x}_t$ is the obtained service profit, and $time^{u_x}_t$ is the total latency for task processing. Moreover, it is evident from the overall model that the parameters within this tuple are intercoupled.

*Action*: The action space is also divided into two layers: 1) internal and 2) external. The external action is defined as $\mathcal{A}_{\text{outer}} = X$, which indicates that the external action involves deploying $X$ UAVs. $\mathcal{A}_{\text{inner}}$ denotes the joint action space of multiple UAVs. $\mathcal{A}_{\text{inner}} = (a^{u_1}_t, \ldots, a^{u_x}_t, \ldots, a^{u_X}_t)$, where $a^{u_x}_t$ denotes the action of UAV $u_x$ at time slot $t$. The control UAV selects a suitable angle to move a certain distance to optimize its deployment position.

*Reward*: It denotes the global reward of multiple UAVs. $\mathcal{R}^{u_x}(\mathcal{S}|\mathcal{A})$ denotes the reward function of UAV $u_x$. The immediate reward for each UAV is $r^{u_x}_t$, which depends on the current state $s^{u_x}_t$ and action $a^{u_x}_t$ of the UAV. The mechanism is designed based on a comprehensive consideration of the environmental state of the UAV swarm as a whole, which not only reflects the impact of the individual behavior of each UAV, but also promotes the synergy between UAVs and environmental adaptation.

*Policy*: Set $\Pi = (\pi^{u_1}, \ldots, \pi^{u_x}, \ldots, \pi^{u_X})$ as a joint policy for multiple UAVs, denoted as a mapping from the state space to the action space, i.e., $\pi^{u_x}$ is the policy representation $a^{u_x}_t = \pi^{u_x}(s^{u_x}_t|\theta^{u_x})$ for UAV $u_x$, where $\theta^{u_x}$ denotes a parameter of the policy network, which means that the policy always produce the same action under the same state information.

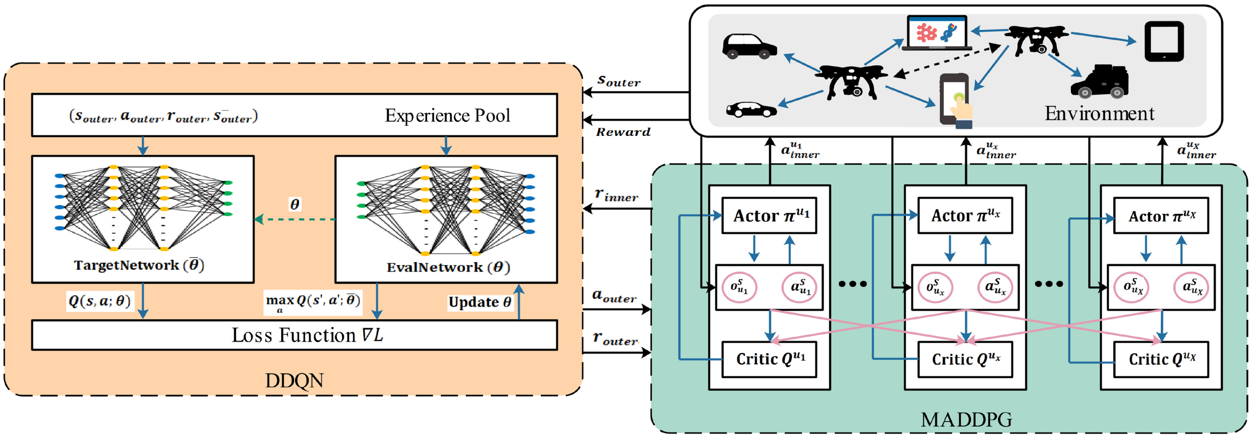



Fig. 2. HRL network framework diagram.

The optimization of the policy model is achieved by adjusting the network parameters $\theta^{u_x}$, with the objective of maximizing the agents' expected cumulative rewards. The cumulative rewards can be estimated through either a value function or an action-value function. In MADDPG, the critic network is typically used to estimate the action-value function $Q^{u_x}$. Policy gradient methods are employed to update the parameters of the policy model, the gradient ascent update rule is

$$\theta^{u_x} \leftarrow \theta^{u_x} + lr\nabla_{\theta^{u_x}} J\left(\pi^{u_x}\right) \tag{24}$$

where $J(\pi^{u_x}) = E_{s^{u_x}\sim p^{\pi}, a^{u_x}\sim\pi^{u_x}}[R^{u_x}]$ is the performance objective function of UAV $u_x$ and $lr$ is the learning rate. The step of the performance objective function is

$$\nabla_{\theta^{u_x}} J\left(\pi^{u_x}\right) = \mathbb{E}_{s^{u_x}\sim p^{\pi}, a^{u_x}\sim\pi^{u_x}} \left[\nabla_{\theta^{u_x}} \log \pi^{u_x}\left(s^{u_x}|\theta^{u_x}\right) Q^{u_x}(s,a)\right]. \tag{25}$$

In practice, the policy gradient is computed in conjunction with the dominance function $A^{u_x}(s,a) = Q^{u_x}(s,a) - V^{u_x}(s)$, to reduce variance and improve training stability. The gradient can be further simplified, by using the chain rule to compute the gradient of the actor network directly through the critic network, as

$$\nabla_{\theta^{u_x}} J\left(\pi^{u_x}\right) = \mathbb{E}_{s^{u_x}\sim p^{\pi}} [\nabla_{a^{u_x}} Q^{u_x}\left(s, a|\phi^{u_x}\right)|_{s=s^{u_x}, a=\pi^{u_x}(s^{u_x})} \nabla_{\theta^{u_x}} \pi^{u_x}\left(s^{u_x}|\theta^{u_x}\right)] \tag{26}$$

where $\phi^{u_x}$ denotes the parameters of the critic network. We optimize the overall benefit of the whole vehicular network by constructing a HRL scheme. In the outer layer, the DDQN algorithm first optimizes the number of UAVs to get the determined number of UAVs, and after that, the deployment locations of multiple UAVs are co-optimized by the inner layer MADDPG, and its specific strategy optimization flow is shown in Algorithm 1.

Both DDQN and MADDPG are the DRL algorithms, that the neural networks (NNs) are used to approximate the $Q$ values, and the exploration–exploitation tradeoff exists for the convergence performance. Similar as the analysis in [41] and [42], to determine the computational complexity, we denote $L_O$ and $L_I$ as the multiplication operations in the selected NNs, and the sampled numbers of outer and inner layers are set as $I_O$ and $I_I$. Then, the algorithm computational complexity is obtained that the one in the outer layer multiply by the inner layer, as $\mathcal{O}((ML_O I_O)(\xi_{th}\mathrm{Num}_{totalTask} L_I I_I))$.

## V. Simulation Results

### A. Parameter Setting

In this section, lots of experimental simulations are proposed to assess the effectiveness of the designed schemes. A simulated multiple congested road segment scenario is constructed with multiple UAVs and vehicles. In the peak periods, the sudden increases in vehicle flow leading to congestion, the vehicles move slowly in four directions with a speed in $[1.5 - -3]$ m/s. The communication range radius of UAVs is 1 km, with a caching capacity of $3 \times 10^4$ MB, allowing a single UAV serves multiple vehicles simultaneously. Assuming each UAV flies at an altitude of 50 m, with their horizontal deployment positions optimized subsequently, the movement range of the UAV swarm is set within a square area with a side length of 10 km, the minimum safety distance constraint is 500 meters between any two UAVs. Set that 100 vehicle users, and each randomly generates 6 tasks. The vehicles' positions will also be generated at the start time randomly. The types of tasks, services, and content generated by vehicles are 100 in variety, and the data sizes are randomly generated within a certain range [50, 100] Mbit. Python 3.9 and Pytorch 1.10.2 are used to construct the simulation model. The NN hidden layers of DDQN are set as two layers of 64 × 64, with a learning rate of $0.5 \times 10^{-3}$, discount factor of 0.95, an activation function using the relu function, an initial greed rate set to 0.1, and increasing with the number of learning times at a growth rate of 0.01. In MADDPG, the NNs of the critic and actor networks are set to two layers of 64 × 64 and three layers of 128 × 128 × 64, respectively, with a learning rate as 0.0005, and other parameter settings the same as DDQN. On the hardware side, the computer CPU model is AMD R7600. Other main parameters are detailed in Table II.

### B. Discussion of Simulation Results

Denote the proposed *Multi-UAV Cost-Efficient and Deployment Scheme through a HRL* algorithm as *MUCEDS*. Fig. 3 demonstrates the convergence of the proposed

**Algorithm 1** HRL Algorithm Based on DDQN and MADDPG

**Initialize:** A critic $Q(s, a; \theta^C)$ and an actor $u(s; \theta^A)$ with random parameters $\theta^C$ and $\theta^A$;
**Initialize:** $lr$, $\gamma$, $tau$, $pos_{u_x}$, $\xi_{th}$, $\xi^*_{th}$, number of users;
**for** episode $= 0$ to $M$ **do**
  Selects an action randomly with probability $\epsilon$, or take $a_{outer} = argmax_{a \in A} Q(s_{t+1}, \eta \max_a Q(s_{t+1}, a|\theta_t)\theta_t^-)$ with probability $1 - \epsilon$;
  $Earn_{task} = 0$, $\xi = 0$, $T_{task} = 0$, $Cost_u = 0$;
  **while** $\xi \leq \xi_{th}$ **do**
    Number of UAVs $X$ are obtained according to action $a_{outer}$, update $Cost_u$;
    Initialize $X$ UAVs, inner state $s_{inner}$, $Num_{task} = 0$;
    **while** $Num_{task} \leq Num_{totalTask}$ **do**
      Selects inner action $a_{inner} = [a^{inner}_{u_1}, \ldots, a^{inner}_{u_x}, \ldots, a^{inner}_{u_X}]$ to optimize the position of $X$ UAVs;
      Calculates $Earn_{task}$ and $T_{task}$ in UAVs;
      Obtains $r_{inner}$, the next observe $s^-_{inner} = [s^{inner}_{u_1}, \ldots, s^{inner}_{u_x}, \ldots, s^{inner}_{u_X}]$;
      Store $(s_{inner}, a_{inner}, r_{inner}, s^-_{inner})$ in replay buffer;
      Updating the network parameters to obtain $\nabla_{\theta^{u_x}} J(\pi^{u_x})$;
      Record the number of tasks handled by each UAV to $Num_{task}$, update $Cost_u$;
    **end while**
    Calculating $Earn_{task} - Cost_u$;
    Obtains $r_{outer}$, the next outer observe $s^-_{outer}$;
    Store $(s_{outer}, a_{outer}, r_{outer}, s^-_{outer})$ in experience pool;
    $y_t = R_{t+1} + \eta Q(s_{t+1}, \eta \max_{a_{outer}} Q(s_{t+1}, a|\theta_t)\theta_t^-)$;
    Performs a gradient descend step on $L(\theta_t)$ with respect to $\theta_t$;
    **if** $|Earn_{task} - Cost_u| \leq \xi^*_{th}$ **then**
      $\xi = \xi + 1$;
    **end if**
    Updating the location of vehicle users;
    Until convergence;
  **end while**
**end for**
**Output:** Optimal $Earn_{task}$, number of UAVs $X$, $T_{task}$, the position of UAVs.

TABLE II
DEFAULT PARAMETER SETUP

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $P_0$ | [2, 8]W | $\sigma^2$ | -96dBm |
| $P_{u_x,i}$ | [0.1, 1.0]W | $B$ | [2, 3]MHz |
| $P_{u_x,u_{x+1}}$ | [0.4, 1.0] | $B_{u_x}$ | [20, 25]MHz |
| $X_0, Y_0$ | 11.9, 0.13 | $R^{UV}_{back}$ | 50Mb/GHz |
| $\eta_{LoS}$ | 1.8dBm | $\eta_{NLoS}$ | 30dBm |
| $F_{u_x}$ | [2, 2.5]MHz | $P_{u_x}$ | [2, 3]GB |
| $E_{u_x}$ | [15000,20000]mAh | $D_{min}$ | 500m |

MUCEDS for the joint number and deployment locations optimization of multiple UAVs. Specifically, Fig. 3(a) shows the convergence of the total revenue obtained from multiple UAVs providing services to vehicular users, Fig. 3(b) illustrates the convergence of network rewards of HRL during the simulation process of optimizing profits for multiple UAVs. Fig. 3(c) displays the convergence of the optimal UAV number based on the vehicular users' requests and the costs of the multiple UAV swarm. Fig. 3(d) shows the changes of the loss function during the simulation process.

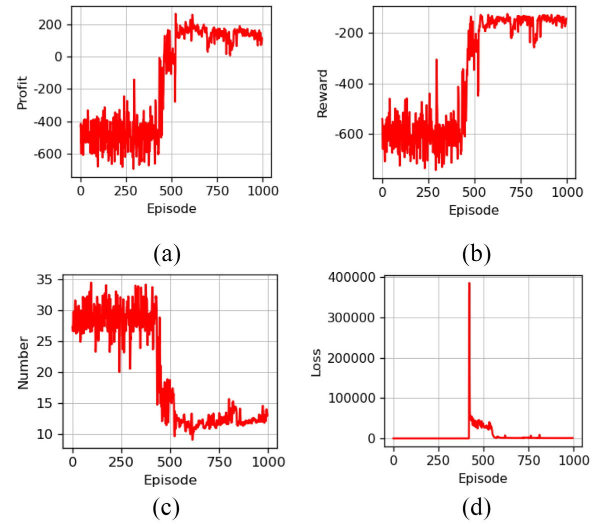



Fig. 3. Convergence of the proposed MUCEDS. (a) Convergence of profit. (b) Convergence of reward. (c) Convergence of UAV number. (d) Changes of loss function.

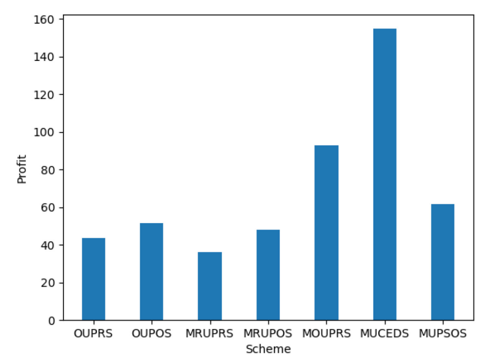



Fig. 4. Comparison of total profit of different schemes.

From Fig. 3(a), it is evident that the proposed MUCEDS can reach the anticipated revenue range after about 1000 iterations. The convergence trend shown in Fig. 3(c) also indicates that the number of UAVs gradually stabilizes with an increase in iterations. It suggests that under the varying multisegment user access conditions, the converged number of UAVs can bring the maximum profit to the overall multi-UAV-assisted VECNs. Although the increasing in the number of UAVs can provide more services, it also lead to the increased system costs and decrease the overall profit, even resulting in negative profit. These results reflect the efficiency and practicality of the proposed MUCEDS scheme for managing UAV networks in ITSs.

For the proposed MUCEDS scheme, both the number of UAVs and the deployment positions of UAVs are optimized to minimize the system task completed latency. Therefore, six benchmark schemes are designed to validate the system performance, as follows.

TABLE III
ALGORITHM TIME COMPLEXITY COMPARISON

| Algorithm | OUPRS | OUPOS | MRUPRS | MRUPOS |
|---|---|---|---|---|
| Time(s) | 0.659482 | 0.631054 | 8.49062 | 10.711149 |
| Algorithm | MOUPRS | MUPSOS | MUCEDS | / |
| Time(s) | 7.782711 | 3.319207 | 6.677222 | / |

1) *One-UAV Position Random Scheme* (*OUPRS*)*:* Only one UAV is dispatched to a designated area to serve all vehicular users, the UAV's deployment locations are randomly adjusted. Vehicle users within the UAV's communication range can receive services, otherwise, they must process tasks independently or further offload tasks to the CCC.
2) *One-UAV Position Optimize Scheme* (*OUPOS*)*:* Only one UAV serves vehicular users, but differently, the UAV's deployment location is optimized based on the positions and access conditions of vehicle users to better provide services.
3) *Multi-UAV Random-Position Scheme* (*MRUPRS*)*:* Multiple UAVs are deployed in a random number, and the deployment locations are randomly moved while only ensuring a minimum safety distance. Compared to the single UAV scheme, the scheme can serve more vehicles, but the cost also increases.
4) *Multi-UAV Random-Position Optimize Scheme* (*MRUPOS*)*:* Set the number of UAV randomly, and the deployment locations of each UAV are optimized based on the distribution and requests of vehicle users.
5) *Multi-UAV Optimize-Position Random Scheme* (*MOUPRS*)*:* The number of UAVs is optimized using the DDQN algorithm based on the distribution and request conditions of vehicular users within the current multisegment area, while the deployment locations of UAVs remain randomly moved.
6) *Multi-UAV Particle Swarm Optimize Scheme* (*MUPSOS*)*:* The number of UAVs is set to a fixed value, and the particle swarm optimization (PSO) algorithm [12] is utilized to optimize the deployment positions of each UAV based on the distribution and requests of vehicle users.

In the OUPRS, OUPOS, MRUPRS, MRUPOS, MOUPRS, and MUPSOS schemes, all schemes employ the same data transmission technology to ensure the consistency of communication links. Hence, the primary differences among these schemes lie in the optimization of UAV numbers and deployment locations. Performance comparisons show that appropriate UAV number and deployment optimization are crucial for enhancing network service efficiency and user QoS, and different strategies demonstrate the clear advantages and effectiveness in practical applications.

Fig. 4 shows a comparison of the total revenue obtained from serving vehicle users under different schemes. To assess the performance of the MUCEDS scheme, five other benchmark schemes were set with the same communication technology standards, allowing for a more accurate comparison of the MUCEDS scheme's performance advantages. Compared to the other benchmark schemes: OUPRS, OUPOS,

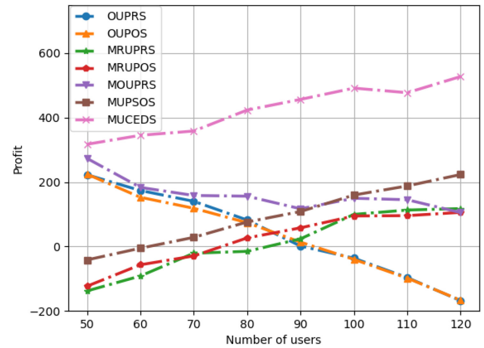



Fig. 5. Comparison of system profits under different schemes with varying numbers of users.

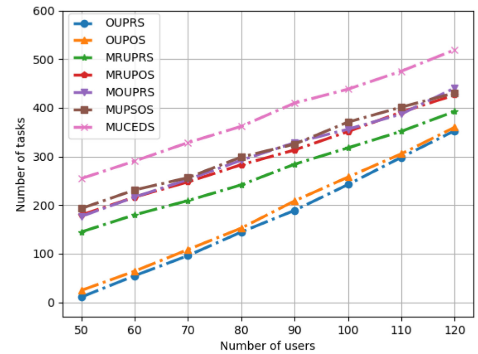



Fig. 6. Comparison of the number of processed tasks under different schemes with varying numbers of users.

MRUPRS, MRUPOS, MOUPRS, and MUPSOS, the proposed MUCEDS scheme reduces UAV costs and the task completed latency of the multi-UAV network, while also improving the QoS of vehicle users. Therefore, the overall network profit is maximized, and a win-win situation for service providers and vehicle users is achieved. In the proposed MUCEDS scheme, a reasonable optimal number of UAVs is deployed based on the road segment congestion conditions, positions, and request access situations of vehicular users, the deployment locations of UAVs are also optimized. For the benchmark schemes, when dealing with a large number and wide distribution of vehicle user tasks, their executability and optimization capacity are relatively weak and have difficulty in convergence, resulting in either lower or negative profit.

In addition, we analyze the average time consumption of each iteration during the iterative optimization process. As shown in Table III, the time complexity of the proposed MUCEDS is greater than that of several baseline algorithms. However, Figs. 5–9 demonstrate that MUCEDS exhibits the best performance. The reason is that MUCEDS optimizes both the number of UAVs and their deployment locations, resulting in reduced total completion delay for system tasks and maximizing profits.

Fig. 5 shows the comparison of system profits under different schemes with varying numbers of vehicle users. We can find that the proposed MUCEDS scheme achieves the highest revenue compared to the benchmark schemes under various conditions. The main reason is that in the proposed scheme, both the number of UAVs and the deployment positions are

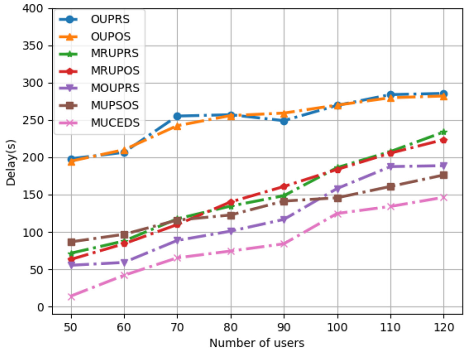



Fig. 7. Comparison of total tasks completed latency $T_{\text{task}}$ under different schemes with varying numbers of users.

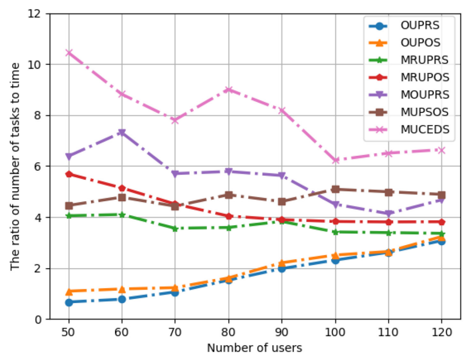



Fig. 8. Comparison of service efficiency under different schemes with varying numbers of users.

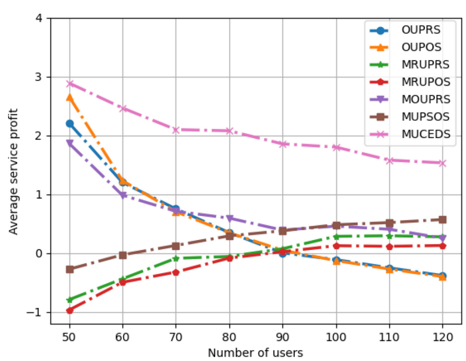



Fig. 9. Comparison of average service profit under different schemes with varying numbers of users.

optimized, so the tasks of users can be offloaded to the suitable UAVs. In single UAV configurations, such as OUPRS and OUPOS, the cost is influenced by overload rather than the number of UAVs. The total system profit primarily reduced for much more tasks should be offloaded to the CCC.

Simulation experiments conducted across various scenarios yielded results as presented in Figs. 6–9. Fig. 6 shows that under different vehicle user numbers, the proposed MUCEDS scheme processes the highest number of tasks compared to the other benchmark schemes, with the number of successfully processed tasks increasing as the vehicular user number increases. Fig. 7 displays the total task completed latency $T_{\text{task}}$ under different vehicle user numbers for each scheme. The proposed MUCEDS scheme exhibits shorter latency within a certain task range.

Fig. 8 further highlights the advantages of the proposed MUCEDS scheme through the ratio of the number of tasks processed to the total latency, $([\text{Num}_{\text{task}}]/[T_{\text{task}}])$, under different vehicular user quantities. We can find that it processes more tasks while maintaining overall shorter task completed latency, compared to other benchmark schemes. Fig. 9 presents the change in average profit per task served by UAVs as the number of vehicle users increases, i.e., the ratio of total profit to the number of successfully processed tasks. The MRUPRS and MRUPOS schemes might deploy many UAVs with fewer users, leading to costs exceeding benefits and negative network profits. The OUPRS and OUPOS schemes, using single UAV deployment, have lower costs and can handle a small number of user demands, but are prone to overload as the number of users increases, resulting in decreased net profit. These results prove that the proposed scheme can more effectively perform global optimization based on the current positions and request status information from vehicular users across multiple road segments; it can adjust the number and deployment locations of UAVs reasonably, thereby achieving the highest overall profit and relatively lower task completed latency. They provide users with higher service efficiency and QoS.

## VI. Conclusion

In this article, considering UAVs with precached parts of service and content caching data, serving as temporary BSs for multiple congested road segments, we propose a cost-efficient joint optimization scheme in which the number and deployment locations of UAVs are optimized in multi-UAV-assisted VECNs. As the varying vehicular network environments and the dynamic requirements of users, we design a novel HRL algorithm, employing DDQN for the number optimization and MADDPG for the deployment of UAVs, that aims to enhance service efficiency via maximizing revenue and minimizing task completed latency. Lots of simulations are proposed to demonstrate the superior performance of the designed scheme against five benchmark schemes, it offers significant improvements in revenue and service latency for vehicle users. The current model is limited by simplified assumptions that may not reflect real-world variability and does not fully consider UAV energy optimization, leading to suboptimal efficiency in long-duration missions. Additionally, UAV collaboration in dynamic environments remains underexplored. Future work can focus on two areas: 1) exploring more comprehensive channel conditions and 2) integrating advanced algorithms like edge AI to improve scalability and response time in dynamic vehicular environments.

## References

[1] D. Katare, D. Perino, J. Nurmi, M. Warnier, M. Janssen, and A. Y. Ding, “A survey on approximate edge AI for energy efficient autonomous driving services,” *IEEE Commun. Surveys Tuts.*, vol. 25, no. 4, pp. 2714–2754, 4th Quart., 2023.

[2] J. Liu et al., “RL/DRL meets vehicular task offloading using edge and vehicular cloudlet: A survey,” *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8315–8338, Jun. 2022.

[3] X. Huang and X. Fu, “Fresh data collection for UAV-assisted IoT based on aerial collaborative relay,” *IEEE Sensors J.*, vol. 23, no. 8, pp. 8810–8825, Apr. 2023.

[4] P. Zhang, C. Wang, C. Jiang, and A. Benslimane, "UAV-assisted multi-access edge computing: Technologies and challenges," *IEEE Internet Things Mag.*, vol. 4, no. 4, pp. 12–17, Dec. 2021.

[5] J. Hu, C. Chen, L. Cai, M. R. Khosravi, Q. Pei, and S. Wan, "UAV-assisted vehicular edge computing for the 6G Internet of Vehicles: Architecture, intelligence, and challenges," *IEEE Commun. Standards Mag.*, vol. 5, no. 2, pp. 12–18, Jun. 2021.

[6] K. Hongchang, W. Hui, S. Hongbin, and H. Yang, "Deep reinforcement learning-based task offloading and service migrating policies in service caching-assisted mobile edge computing," *China Commun.*, vol. 21, no. 4, pp. 88–103, Apr. 2024.

[7] Y. Gong, "Optimal edge server and service placement in mobile edge computing," in *Proc. IEEE 9th Joint Int. Inf. Technol. Artif. Intell. Conf. (ITAIC)*, 2020, pp. 688–691.

[8] Y. Liu, S. Xie, and Y. Zhang, "Cooperative offloading and resource management for UAV-enabled mobile edge computing in power IoT system," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12229–12239, Oct. 2020.

[9] Y. Liu, C. Yang, X. Chen, and F. Wu, "Joint hybrid caching and replacement scheme for UAV-assisted vehicular edge computing networks," *IEEE Trans. Intell. Vehicles*, vol. 9, no. 1, pp. 866–878, Jan. 2024.

[10] F.-Y. Wang et al., "Transportation 5.0: The DAO to safe, secure, and sustainable intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 10, pp. 10262–10278, Oct. 2023.

[11] Z. Chen, J. Zhang, X. Zheng, G. Min, J. Li, and C. Rong, "Profit-aware cooperative offloading in UAV-enabled MEC systems using lightweight deep reinforcement learning," *IEEE Internet Things J.*, vol. 11, no. 12, pp. 21325–21336, Jun. 2024.

[12] Z. Han, T. Zhou, T. Xu, and H. Hu, "Joint user association and deployment optimization for delay-minimized UAV-aided MEC networks," *IEEE Wireless Commun. Lett.*, vol. 12, no. 10, pp. 1791–1795, Oct. 2023.

[13] C. Yang, B. Liu, H. Li, B. Li, K. Xie, and S. Xie, "Learning based channel allocation and task offloading in temporary UAV-assisted vehicular edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 9884–9895, Sep. 2022.

[14] Z. Wang et al., "Learning to routing in UAV swarm network: A multi-agent reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 72, no. 5, pp. 6611–6624, May 2023.

[15] M. Hosseini and R. Ghazizadeh, "Stackelberg game-based deployment design and radio resource allocation in coordinated UAVs-assisted vehicular communication networks," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 1196–1210, Jan. 2023.

[16] S. Tong, Y. Liu, J. Mišić, X. Chang, Z. Zhang, and C. Wang, "Joint task offloading and resource allocation for fog-based intelligent transportation systems: A UAV-enabled multi-hop collaboration paradigm," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 12933–12948, Nov. 2023.

[17] 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, Mar. 2021.

[18] M. Mukherjee, V. Kumar, A. Lat, M. Guo, R. Matam, and Y. Lv, "Distributed deep learning-based task offloading for UAV-enabled mobile edge computing," in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2020, pp. 1208–1212.

[19] A. Mohammed, H. Nahom, A. Tewodros, Y. Habtamu, and G. Hayelom, "Deep reinforcement learning for computation offloading and resource allocation in blockchain-based multi-UAV-enabled mobile edge computing," in *Proc. 17th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process. (ICCWAMTIP)*, 2020, pp. 295–299.

[20] B. Zeng, C. Zhan, C. Xu, and J. Liao, "Caching and 3-D deployment strategy for scalable videos in cache-enabled multi-UAV networks," *IEEE Trans. Veh. Technol.*, vol. 72, no. 11, pp. 14875–14888, Nov. 2023.

[21] H. Shen, Y. Luo, Y. Xu, and T. Zhang, "Completion time minimization for content delivery in cache-enabled multi-UAV networks," in *Proc. 21st Int. Symp. Commun. Inf. Technol. (ISCIT)*, 2022, pp. 58–63.

[22] Q. Feng et al., "Importance measure-based phased mission reliability and UAV number optimization for swarm," *Rel. Eng. Syst. Saf.*, vol. 223, Jul. 2022, Art. no. 108478.

[23] R. Liu, A. Liu, Z. Qu, and N. N. Xiong, "An UAV-enabled intelligent connected transportation system with 6G communications for Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 2045–2059, Feb. 2023.

[24] 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. (VTC2)*, 2023, pp. 1–5.

[25] Z. Wu, Z. Yang, C. Yang, J. Lin, Y. Liu, and X. Chen, "Joint deployment and trajectory optimization in UAV-assisted vehicular edge computing networks," *J. Commun. Netw.*, vol. 24, no. 1, pp. 47–58, Feb. 2022.

[26] C. Sun, X. Xiong, W. Ni, and X. Wang, "Three-dimensional trajectory design for energy-efficient UAV-assisted data collection," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2022, pp. 3580–3585.

[27] S. Chai and V. K. N. Lau, "Multi-UAV trajectory and power optimization for cached UAV wireless networks with energy and content recharging-demand driven deep learning approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3208–3224, Oct. 2021.

[28] G. Zheng, C. Xu, M. Wen, and X. Zhao, "Service caching based aerial cooperative computing and resource allocation in multi-UAV enabled MEC systems," *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 10934–10947, Oct. 2022.

[29] K. Yao et al., "Distributed joint optimization of deployment, computation offloading and resource allocation in coalition-based UAV swarms," in *Proc. Int. Conf. Wireless Commun. Signal Process. (WCSP)*, 2020, pp. 207–212.

[30] G. A. Bayessa, R. Chai, and Q. Chen, "Joint user grouping, UAV deployment and multicast precoder optimization in UAV-enabled networks," in *Proc. SoutheastCon*, 2024, pp. 151–156.

[31] J. Liu, Z. Wu, J. Liu, and X. Tu, "Distributed location-aware task offloading in multi-UAVs enabled edge computing," *IEEE Access*, vol. 10, pp. 72416–72428, 2022.

[32] L. Wang, H. Zhang, S. Guo, D. Li, and D. Yuan, "Learning to deployment: Data-driven on-demand UAV placement for throughput maximization," *IEEE Trans. Veh. Technol.*, vol. 73, no. 6, pp. 8007–8012, Jun. 2024.

[33] Y. Chen and Z. Zheng, "Joint deployment and task computation of UAVs in UAV-assisted edge computing network," in *Proc. 21st Asia-Pac. Netw. Oper. Manag. Symp. (APNOMS)*, 2020, pp. 413–416.

[34] Y. Zheng, B. Yang, and C. Chen, "Joint optimization of the deployment and resource allocation of UAVs in vehicular edge computing and networks," in *Proc. IEEE 92nd Veh. Technol. Conf. (VTC)*, 2020, pp. 1–6.

[35] H. Touati, A. Chriki, H. Snoussi, and F. Kamoun, "Cognitive radio and dynamic TDMA for efficient UAVs swarm communications," *Comput. Netw.*, vol. 196, Sep. 2021, Art. no. 108264.

[36] S. Hayashi and Z.-Q. Luo, "Spectrum management for interference-limited multiuser communication systems," *IEEE Trans. Inf. Theory*, vol. 55, no. 3, pp. 1153–1175, Mar. 2009.

[37] S. Araf, A. S. Saha, S. H. Kazi, N. H. Tran, and M. G. R. Alam, "UAV assisted cooperative caching on network edge using multi-agent actor-critic reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 72, no. 2, pp. 2322–2337, Feb. 2023.

[38] J. Luo, J. Song, F.-C. Zheng, L. Gao, and T. Wang, "User-centric UAV deployment and content placement in cache-enabled multi-UAV networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 5656–5660, May 2022.

[39] T. Ren et al., "Enabling efficient scheduling in large-scale UAV-assisted mobile-edge computing via hierarchical reinforcement learning," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7095–7109, May 2022.

[40] Z. Liu, Y. Cao, J. Chen, and J. Li, "A hierarchical reinforcement learning algorithm based on attention mechanism for UAV autonomous navigation," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 13309–13320, Nov. 2023.

[41] M. Tang and V. W. Wong, "Deep reinforcement learning for task offloading in mobile edge computing systems," *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 1985–1997, Jun. 2022.

[42] D. Xu, F. Zhu, Q. Liu, and P. Zhao, "Improving exploration efficiency of deep reinforcement learning through samples produced by generative model," *Expert Syst. Appl.*, vol. 185, Dec. 2021, Art. no. 115680.

**Yinan Liu** received the B.E. degree from Nanchang Hangkong University, Nanchang, China, in 2021. He is currently pursuing the M.S. degree with the School of Automation, Guangdong University of Technology, Guangzhou, China.

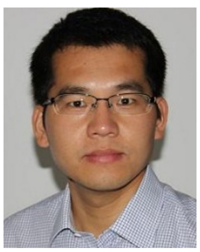
His research interests include wireless communication networks, cooperative communications, and intelligent edge computing.

**Chao Yang** received the Ph.D. degree in signal and information processing from South China University of Technology, Guangzhou, China, in 2013.

He currently work with the School of Automation, Guangdong University of Technology, Guangzhou, China. From July 2014 to July 2016, he was a Research Associate with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His research interest focuses on VANETs, edge computing, and smart grid.

**Yanqun Tang** received the B.Sc., M.Sc., and Ph.D. degrees from the School of Electronic Science and Engineering, National University of Defense Technology, Changsha, China, in 2007, 2009, and 2013, respectively.

He is currently an Associate Processor with the School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen, China. His research interests are integrated sensing and communication, full duplex communications, wireless physical-layer security, and machine learning techniques for wireless communications.



**Haitao Zhao** received the B.E., M.Sc., and Ph.D. degrees from National University of Defense Technology (NUDT), Changsha, China, in 2002, 2004, and 2009, respectively.

He is currently a Professor with the College of Electronic Science and Technology, NUDT. Prior to this, he visited the Institute of ECIT, Queen's University of Belfast, Belfast, U.K., and Hong Kong Baptist University, Hong Kong. His main research interests include wireless communications, cognitive radio networks, and intelligent self-organized networks.

Prof. Zhao has served as a TPC Member for IEEE ICC'2014–2024 and Globecom'2016–2024, and the Guest Editor for *Physical Communications* and *IEEE Communications Magazine*.

**Yi Liu** received the Ph.D. degree from South China University of Technology, Guangzhou, China, in 2011.

After that, he joined Singapore University of Technology and Design, Singapore, as a Postdoctoral Fellow. In 2014, he worked with the Institute of Intelligent Information Processing, Guangdong University of Technology, Guangzhou, where he is currently a Full Professor. His research interests include wireless communication networks, cooperative communications, smart grid, and intelligent edge computing.

**Shengli Xie** (Fellow, IEEE) received the M.S. degree in mathematics from Central China Normal University, Wuhan, China, in 1992, and the Ph.D. degree in control theory and applications from South China University of Technology, Guangzhou, China, in 1997.

He is currently a Full Professor and the Head of the Institute of Intelligent Information Processing, Guangdong University of Technology, Guangzhou, and the Director of the Laboratory for Intelligent Information Processing, the Director of the Guangdong–Hong Kong–Macao Joint Laboratory for Smart Discrete Manufacturing. He has authored or co-authored two books and over 150 scientific papers in journals and conference proceedings. His research interests include wireless networks, automatic control, and signal processing.