Wireless Networks (2025) 31:965–981
https://doi.org/10.1007/s11276-024-03804-3

Enhancing UAV-assisted vehicle edge computing networks through a digital twin-driven task offloading framework

Zhiyang Zhang[1] · Fengli Zhang[1] · Minsheng Cao[1] · Chaosheng Feng[2] · Dajiang Chen[1]

Accepted: 17 June 2024 / Published online: 2 July 2024

Abstract
Enhancing the task offload performance of UAV-assisted Vehicular Edge Computing Networks (VECNs) is complex, especially in vehicle-to-everything (V2X) applications. These networks rely on UAVs and roadside units (RSUs) to offload heavy computational tasks and reduce the load on the on-board systems. However, UAV-assisted VECNs face severe challenges from heterogeneous offload node resources and dynamic edge network environments in providing low-latency and high-response task offloading, especially during traffic congestion or infrastructure failures. In this paper, we propose a digital twin (DT)-driven task offloading framework for UAV-assisted VECNs. The aim of the proposed framework is to improve the global performance of VECN task offloading under limited computational and communication resource constraints. Firstly, we construct a decentralized offloading decision-centralized evaluation task offloading framework for UAV-assisted VECNs based on the asynchronous advantage actor-critic (A3C) algorithm. Secondly, we integrate the graph attention networks (GAT) into the framework to incorporate the dynamically changing DT network topology information into the state evaluation of VECNs. By simulating a DT-driven multi-UAV cooperative system and comprehensive evaluation of real-world task request datasets. The framework has a better task throughput rate and stability when performing task offloading in local resource overload and dynamic edge environment scenarios.

Keywords Internet of vehicles (IoV) · Digital twin (DT) · Task offloading · Edge intelligence · Graph attention network (GAT)

1 Introduction

The conflict between resource-limited in-vehicle terminals and computation-intensive applications has become a critical bottleneck. It affects user satisfaction and quality of service reliability enhancement in the Internet of Vehicles (IoV) [1]. Vehicular edge computing networks (VECNs) deploy servers at the edge of the radio access network, enabling on-board units (OBUs) to offload computational tasks to edge servers in nearby idle vehicles or roadside units (RSUs) [2]. By collecting large amounts of data from sensors to dynamically adjust task offloading policies, VECN not only ensures that vehicle systems remain efficient and reliable across different environments but also accelerates fault diagnosis and prediction models [3]. It enables VECNs to predict and identify potential faults promptly, which is critical for prognostic health management (PHM) and immediate troubleshooting in the IoV [4, 5]. As an enhancement to VECNs, unmanned aerial vehicles (UAV)-assisted VECNs provide more reliable network coverage and wireless transmission [6]. UAVs, equipped with

Z. Zhang, F. Zhang, M. Cao, C. Feng and D. Chen have contributed equally to this work.

✉ Dajiang Chen
djchen@uestc.edu.cn

Zhiyang Zhang
202011090904@std.uestc.edu.cn

Fengli Zhang
fzhang@uestc.edu.cn

Minsheng Cao
cms@uestc.edu.cn

Chaosheng Feng
csfenggy@sicnu.edu.cn

1 School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, Sichuan, China

2 School of Computer Science, Sichuan Normal University, Chengdu 610101, Sichuan, China

communication transceivers, facilitate ubiquitous broadband wireless communications, unaffected by road configurations and obstructions. Furthermore, UAVs' ability to adjust their altitude and position enables effective vehicle communication using line-of-sight (LoS) links [7]. Their high maneuverability allows them to be dispatched to optimal locations based on evolving edge offloading demands, ensuring sufficient communication resources for vehicle-to-everything (V2X) applications, especially those sensitive to latency.

However, in scenarios such as sudden road anomalies and traffic accidents, UAV-assisted VECNs inevitably need to perform complex collaborative task offloading [8, 9]. The tasks include high mobility, frequent topology changes, unstable radio links, and limited energy, which makes low latency and load balancing for UAV-assisted VECN task offloading important. In addition, few studies have focused on infrastructure failures or traffic congestion scenarios, which makes these approaches typically assume that the VECN structure is stable and well-resourced, and thus no longer reliable in emergency situations. We summarize the challenges of UAV-assisted VECNs in the face of sudden road anomalies or traffic accidents into two key points:

1. *How to prevent resource overload due to localized computational resource exhaustion?* In a distributed coordination or control scenario, any localized service disruption can reduce the task offload performance of VECNs and jeopardize road safety on a wide scale [10]. When a group of UAVs performs distributed collaboration, they encounter different RSU resource environments [11]. For the same arriving task, some UAVs can perform offloading normally due to sufficient RSU resources, while others may experience severe system interruptions during offloading due to RSU resource exhaustion. Therefore, it is unwise to perform similar evaluation and task offloading decisions for each UAV. A potential solution is to train a separate task offloading policy for each UAV and then optimize them using local task offload performance as a metric. However, this approach is costly and inefficient due to the limited resource allocation on the UAV side.
2. *How to avoid global performance degradation due to frequent changes in local edge network topology?* In the case of a UAV, both the sources and destinations of offloaded tasks are in motion. We consider two common scenarios, one in which RSUs providing computational power are disconnected from the UAV due to machine failures or other events; and the other is that a change in the UAV's hovering position causes it to disconnect its communication link with some RSUs and parked vehicles. Both scenarios lead to frequent changes in the network topology of the edge environment, thus state evaluations of the VECN that incorporate spatial characterization to be able to capture such dynamics and uncertainty. In addition, frequent network topology changes also accelerate the maintenance and updating of the task offloading policies deployed on UAVs, which can lead to an unstable optimization process of the task offloading policies, thus affecting the global performance [12].

In response to the above challenges, digital twin (DT) and deep reinforcement learning (DRL) techniques have been attempted to be introduced into UAV-assisted Vehicular Edge Networks to provide more accurate and timely state evaluation and task offloading policy optimization [13, 14]. Digital twin technology is a cutting-edge technology that uses digital data for high-fidelity digital representation of physical entities in a virtual world while monitoring the physical entities in real time. By combining deep reinforcement learning methods [15], such as deep Q network (DQN), actor-critic (AC), and deep deterministic policy gradient (DDPG) [16], the DT-driven VECNs can achieve trial-and-error learning and delayed-reward learning and optimize task offloading policies by using the results of state evaluation. More importantly, when combined with deep reinforcement learning algorithms, DT-driven task offloading does not need to always interact with the edge environment in real time or always query the operation status of individual RSUs and UAVs. It not only improves the efficiency of task offloading and ensures the user experience but also reduces the system energy consumption, saves system resources, and enhances prognostic health [17].

However, the use of DRL in DT-driven UAV-assisted VECNs still faces several limitations. Firstly, all UAVs cooperate to serve VECNs in a decentralized manner, i.e., environmental information and offloading decisions for each UAV cannot be shared between UAVs. As a result, the task offloading optimization is often modeled as a partially observable Markov decision process (POMDP) [18, 19], which leads to the fact that the evaluations and decisions made on a single UAV often lack a global view. Second, these iterative learning methods lack initial knowledge, require extensive interaction cycles with the DT, and typically require millions of steps to determine an effective task-offloading policy. Environmental fluctuations or disruptions, such as frequent changes in the topology of the DT network, may significantly reduce their efficiency [20]. This elongated learning curve is fraught with potential inaccuracies, posing a significant challenge to UAV-assisted VECNs, which require precision in their task offloading policies and cannot tolerate sub-optimal decisions.

In this paper, we propose a DT-driven task-offloading framework for UAV-assisted VECNs to support centralized evaluation and distributed optimization of task-offloading decisions. First, the non-convex and highly complex task offloading decision optimization problem is formulated as a POMDP under the constraints of limited computational and communication resources. Secondly, we explore an asynchronous advantage actor-critic (A3C)-based optimization algorithm for task offloading policies [21], called A3GAC, which performs a global evaluation using a centralized Critic at the base station side and independently updates the task at the UAV side with the offloading policy. To further improve performance and efficiency under adverse conditions, the graph attention network (GAT) [22] is integrated into the A3C framework for modeling the DT network topology. These enhancements are essential to significantly improve the adaptability of A3GAC to effectively handle unpredictable task volumes and varying resource availability. The superiority of the framework is demonstrated through a comprehensive evaluation using a simulated multi-UAV collaborative task offload scenario and a real-world task request dataset. The contributions of this paper are summarized as follows:

- We present a DT-driven framework designed for UAV-assisted VECN. UAVs and RSUs are designed as task offload nodes in the DT network topology, where UAVs can also act as relay nodes to forward tasks. Furthermore, a POMDP-based task offloading optimization problem for UAV-assisted VECN is proposed by considering task offloading throughput, load balancing, and transmission delay.
- A decentralized execution-centralized evaluation A3C framework is utilized for distributed task offloading in heterogeneous resource UAV-assisted VECNs, considering both the global state of the UAV-assisted VECN and the local heterogeneous compute/communication resource environments of each UAV. In this framework, we globally evaluate the state of the UAV-assisted VECN and independently update the task offloading policy network parameters for each UAV based on the policy gradient.
- GAT is introduced in the state evaluation module of A3GAC. With self-learning attention weights between task offloading nodes, A3GAC can capture elements such as communication load, resource capacity, network location, and task relevance of all nodes in a DT network more quickly and enable UAVs to prioritize more reliable offloading nodes.

This paper is organized as follows: Sect. 2 provides a comprehensive overview of the research progress in IoV task offloading policy evaluation and optimization. Section 3 designs the DT model of UAV-assisted VECNs and defines the task offloading decision optimization problem. Section 4 describes the algorithmic framework and algorithm design of A3GAC, and we focus on the redesign of the POMDP and actor-critic architecture. Section 5 analyzes and discusses the experimental evaluation, and discusses the performance of A3GAC in static network environments, dynamic network environments, and resource-overloaded environments, respectively. Finally, we will summarize and discuss the paper in Sect. 6.

2 Related work

With the aid of long-term evolution V2X (LTE)-V2X and 5th generation new radio (5 G NR)-V2X technologies [23, 24], IoV can provide a rich information environment for drivers and passengers. However, the service demands of IoV vary with time (peak and off-peak periods) and location (city centers and beyond), rendering uniform deployment policies both expensive and inefficient. Additionally, the complexity of modern road layouts, including overpasses, bridges, and tunnels, introduces the need for three-dimensional (3D) service coverage, which current cellular V2X (C-V2X) [25, 26], primarily designed for two-dimensional networking, struggle to provide.

UAVs equipped with edge servers have been proposed as a solution to provide edge computing services in IoV. In contrast to traditional cloud server-based vehicular networks, UAV-assisted VECNs depend on advanced wireless technologies to rapidly distribute computationally intensive tasks and corresponding sensor data from the vehicle to the edge of the network, thus enabling intelligent V2X applications for IoV [27]. These UAVs, equipped with transceivers, support ubiquitous broadband wireless communication and are not constrained by road layouts and obstacles. For instance, rotor or fixed-wing/rotor hybrid UAVs can hover in fixed positions to provide continuous cellular coverage, and their high maneuverability allows precise deployment of base stations as needed, or carrying base stations along designated flight paths. Their flying capabilities enable them to provide 3D multi-angle aerial interfaces for vehicles. Additionally, the flexibility of UAVs in adjusting their height and position enables highly successful communication with vehicles using LoS links. Furthermore, UAVs can be dynamically positioned according to the varying edge offloading demands over time and location, ensuring sufficient communication resources for vehicle applications, particularly latency-sensitive ones [28]. A lot of task offloading research has been conducted on IoV, but most of the task offloading analysis and optimization work focuses on enhancing security, trust management, and quality of service [29, 30], and the quantitative analysis lacks the use of unified task offload performance metrics. Hou et al. [31] designed a low-complexity fault-tolerant particle swarm optimization algorithm to improve task offload performance under delay constraints. Liu et al. [32] improve the service reliability of time-critical computer vision services in MEC systems by allowing imperfect transmissions. Task offload performance in most of these papers is modeled by data transmission and calculated failure probabilities. However, task offload performance may also be affected by other factors.

To address this problem, several studies have been conducted to improve the task offload performance of the entire network by optimizing a multi-constrained task offloading policy and incorporating factors such as system energy consumption, operating trajectory, and resource loading into the task offload analysis of IoV [33]. Initial studies on optimizing the multi-constrained reliable task offloading problem relied on the continuous convex approximation methods [34, 35]. Xu et al. [34] designed a method for solving the joint problem of trajectory planning and resource allocation as a mixed-integer nonlinear programming problem, using piecewise linear approximation and linear relaxation. To minimize the total mobile energy consumption while satisfying the quality of offloaded mobile applications, Liu et al. [35] proposed a UAV-assisted mobile cloud computing system, in which UAVs equipped with servers provide computational offloading services for locally constrained users. However, traditional convex approximation methods struggle in complex, dynamically evolving edge network environments, mainly because many problems in edge network environments (e.g., resource allocation, energy optimization) are usually nonlinear and nonconvex. In addition, traditional methods are highly susceptible to computational efficiency and scalability problems when dealing with large-scale problems.

DRL has been widely applied to learn and optimize various problems encountered in UAV-assisted edge computing, often considered the most effective method for these issues [36–38]. In the DRL framework, the reliable task-offloading problems in UAV-assisted MEC are modeled as MDPs, with task offloading algorithms based on MDP processes autonomously learning complex environments and making decisions according to environmental changes. Zhang et al. [39] proposed an integrated task offloading framework for optimizing UAV costs and developing autonomous deployment policies for integrated traffic situational awareness. Garg et al. [40] presented a data-driven traffic optimization model that minimizes service-related computation time and storage costs. [41] proposed an efficient data transmission policy and an overall decision co-optimization policy for channel allocation and task processing mode selection. Yan et al. [42] employed a long short-term memory (LSTM) network with an attention mechanism and a DDPG algorithm for task offloading policy optimization, and the advantages of this scheme are reflected in the processing of temporal tasks. Zhang et al. [43] reduced the computational complexity by transforming the computational task offloading problem of MEC into two sub-problems: finding the optimal solution of whether to offload each user device or not, and allocating computational and communication resources. Kang et al. [44] introduced a centrally trained and decentralized execution framework for vehicular networking to collaboratively decide on ground device associations, resource allocation, and task offloading. The aforementioned DRL-based task offloading policy optimization algorithms have made comprehensive progress in terms of simultaneous optimization with multiple constraints, reduced computational complexity, and decentralization. However, these algorithms only target scenarios where IoV runs smoothly, while the task offload performance of the DRL framework in traffic congestion or large RSU failure scenarios has not been widely discussed, and these algorithms do not consider the impact of edge network topology changes on the learning ability of the DRL framework, which may be amplified in emergency scenarios [45].

In this paper, we combine the advantages of A3C and GAT to design a decentralized execution-centralized state evaluation framework for UAV-assisted VECN task offloading, which is driven by a DT that uses a fine-tuned A3C algorithm for global state evaluation at the base station side and an independent task offloading policy update at the UAV side. We use the GAT-integrated evaluation network in the A3C framework for state evaluation, and use long-term task throughput rates to intuitively analyze task offloading optimization performance. A major advantage of our algorithm is that it fully considers the task offload performance of the task offloading framework in special cases such as traffic congestion.

3 System model and problem formulation

Figure 1 depicts a UAV-assisted-VECN consisting of a base station, multiple UAVs, OBUs, and RSUs; in particular, we consider parked vehicles, which temporarily provide computational resources, as movable RSUs in the following discussion.

3.1 Physical network architecture

In the physical network architecture, the base station acts as a central server that dispatches a group of UAVs to perform task offloading to different geographic areas, and each UAV establishes communication links with RSUs in its coverage area and provides services to OBUs. The OBUs and RSUs in different UAV overlapping areas can be associated with only one UAV.

Specifically, we denote the base station as $\mathcal{B}$, and all the nodes (including UAVs and RSUs) in the UAV-assisted-VECN as $\mathcal{N} = \mathcal{N}^{\mathrm{UAV}} \cup \mathcal{N}^{\mathrm{RSU}}$. where $\mathcal{N}^{\mathrm{UAV}}$ and $\mathcal{N}^{\mathrm{RSU}}$ denote the set of all UAVs and all RSUs, respectively. Since the set of nodes within the coverage of each UAV in a UAV-assisted VECN may change dynamically, we divide $\mathcal{N}$ by the coverage of different UAVs in time slot t, i.e., $\mathcal{N} = \cup_{u\in\mathcal{N}^{\mathrm{UAV}}}\mathcal{N}_{u,t}$, where $\mathcal{N}_{u,t}$ denotes the set of nodes associated with the UAV u coverage in time slot t.

In the above UAV-assisted VECN, delay-sensitive, computationally intensive service requests generated by OBUs arrive at each UAV randomly. At each time slot, UAV u, based on its task offloading policy π_u, chooses to either send the task offloading request to a node that has deployed the desired service entity in its coverage area, i.e., local offloading or to forward the current task to another UAV for offloading in the other UAV's coverage area, i.e., relay offloading.

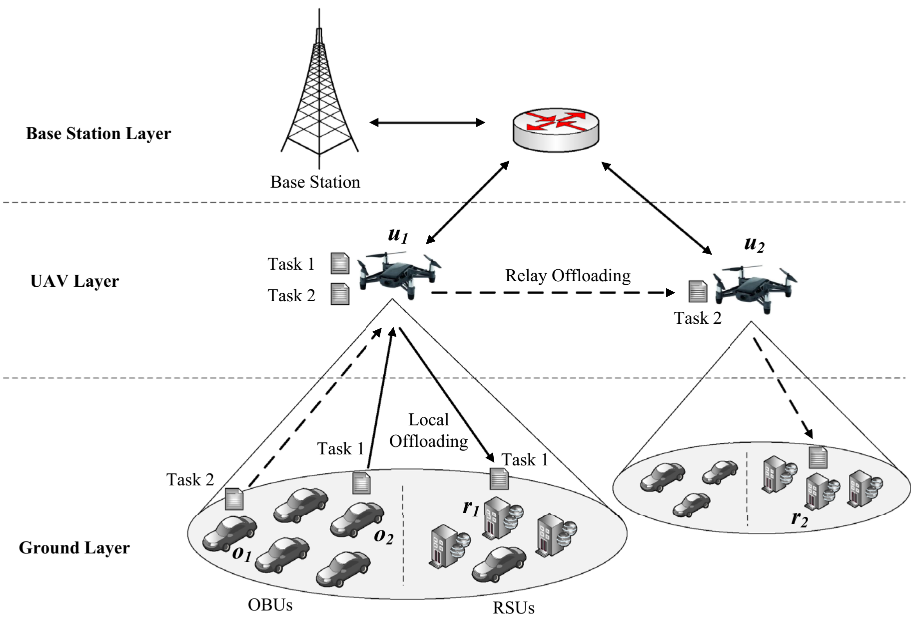


Fig. 1 Task offloading in a UAV-assisted-VECN. The task offloading mode of Task 1 (marked with a solid line) is local offloading, which is uploaded to UAV u_1 by OBU o_1 and offloaded to RSU r_1; the task offloading mode of Task 2 (marked with a dotted line) is relay offloading, which is uploaded to UAV u_1 by OBU o_2 and then first forwarded to relay UAV u_2, and then offloaded to RSU r_2

3.2 Dynamic DT model

In this paper, the virtual DT layer mainly consists of two major types of DTs, DT-UAVs, and DT-RSUs, which map the real-time traffic information, UAV statistics, and RSU information in the physical world through real-time interaction with the physical world to realize the computational/communication resource modeling of UAV and RSU nodes. In addition, DT is used to assist model training and parameter synchronization for deep reinforcement learning-based state evaluation and optimization methods.

We construct an undirected graph $\mathcal{G}_t = (\mathcal{N}, \mathcal{E}_t)$ in time slot t to represent the topological relationship among all nodes in the current DT layer, where $\mathcal{E}_t$ denotes the set of edges in time slot t. The construction of $\mathcal{G}_t$ is based on the following assumptions:

- No edges between DT-RSUs;
- An edge between two DT-UAVs exists if and only if the two DT-UAVs can communicate directly without a relay node.

It should be noted that the virtual replication of some information, such as the available resource loads of RSU and UAV nodes, may have a small deviation from the current actual values. We use $\{\widetilde{f}_{i,t} : i \in \mathcal{N}\}$ in time slot t to denote the estimated available resource load deviation of node i concerning its physical counterpart. At this point, the DT $\mathcal{D}^{\mathrm{UAV}}_{u,t}$ for UAV u and the DT $\mathcal{D}^{\mathrm{RSU}}_{u,t}$ for RSU r in time slot t can be given by Eq. 1 and Eq. 2, respectively:

$$\mathcal{D}^{\mathrm{UAV}}_{u,t} = \left\{ F_{u,t} \pm \widetilde{f}_{u,t}, O_{u,t}, D_{u,t}, \mathcal{E}_{u,t} \right\}, \tag{1}$$

$$\mathcal{D}^{\mathrm{RSU}}_{r,t} = \left\{ F_{r,t} \pm \widetilde{f}_{r,t} \right\}, \tag{2}$$

where $F_{i,t}$ is the resource load of node i in time slot t, $O_{u,t}$ and $D_{u,t}$ are the resource request sizes and delay requirements of the tasks that UAV u offloads in time slot t, respectively, $\mathcal{E}_{u,t}$ is the set of all edges whose edges originate from u, which describes the sub-topology of the entire DT network topology of the UAV-assisted VECN on the UAV u side.

3.3 Throughput rate model

Each task arriving at the UAV has its resource request and latency requirements, and if the offloading delay exceeds its latency requirements or the RSU performing the computation does not meet the resource request of the task, the task offloading fails. Prior studies [46] have typically measured the task offload performance of UAV-assisted VECNs using the long-term task throughput rate Φ for performing task offloading, which represents the ratio of the number of tasks successfully offloaded within the delay requirement to the total number of tasks arriving at the UAV-assisted VECN. The mathematical definition of Φ is given below:

$$\Phi = \frac{\sum_{t=1}^{T} \left(\sum_{i \in \mathcal{N}} \omega(i,t) \right)}{\sum_{t=1}^{T} \sum_{u \in \mathcal{N}^{\mathrm{UAV}}} \psi(u,t)}, \tag{3}$$

where $\psi(u,t)$ denotes the number of tasks arriving at UAV u within time slot t, and $\omega(i,t)$ denotes the number of tasks successfully executed by node i within time slot t, i.e., the task meets the task latency and the target node meets the resource requirement of the task. In this paper, each UAV is allowed to offload only one task request per time slot t, so $\psi(u,t)$ takes the value of 0 or 1. In addition, to accommodate the need for DT-driven distributed task offloading policy learning, we also provide the edge-side local task throughput rate $\Phi_{u,t} = \sum_{i\in\mathcal{N}_{u,t}} \omega(i,t)/\psi(u,t)$, which denotes the proportion of tasks that are successfully offloaded by UAV u within time slot t.

We do not advocate the use of Φ as the only metric for UAV-assisted VECN task offload performance because this metric ignores some of the deeper features that affect performance, including the spatial characteristics of the network topology versus the heterogeneous characteristics of local resources. For example, when one UAV in a UAV-assisted VECN experiences a system outage due to localized resource overload, other edge networks may still have good performance for a short period, thus masking the localized disaster performance. Therefore, we advocate the use of a GAT-integrated evaluation network for state evaluation later on, and the long-term task throughput rate will be used to visually analyze the task offload optimization performance.

3.4 Transmission delay model

Transmission delay is the time required to transfer data from the sending device to the communication medium. The transmission delay $T(i,j,t)$ from UAV i to UAV/RSU j is determined by the channel gain and bandwidth between them. Since the channel models are different for different offloading modes, we calculate the transmission delay for Local and Relay offloading separately.

3.4.1 Local offloading

In local offloading, we assume that the communication between the UAV and the offloading node (itself or any RSU in its coverage area) is based on LoS communication, and the channel is modeled as a free-space trajectory loss. We can then estimate the signal-to-noise ratio (SNR) in terms of the channel gain between the nodes:

$$\mathrm{SNR}(i,j,t) = \frac{P_{\mathrm{UAV}}}{P_{\mathrm{noise}}} \cdot \frac{|h_{i,j}(t)|}{(d_{i,j}(t))^2}, \tag{4}$$

where P_{noise} and P_{UAV} denote the noise power and the transmit power of UAV, respectively; $h_{i,j}(t)$ denotes the reference channel gain at a distance of 1 m and a transmission power of 1 watt-hour [47], $d_{i,j}(t)$ is the Euclidean distance between node i and j. Then, the transmission delay between UAV i and the offloading node j is given by Eq. 5:

$$T(i,j,t) = \frac{s}{B \cdot \log_2(1 + \mathrm{SNR}(i,j,t))}, \tag{5a}$$

$$\text{s.t.} \quad j \in \mathcal{N}_{i,t}, \tag{5b}$$

where s and B denote the task data size and channel bandwidth between UAV i and the offloading node j, respectively. Especially, when UAV i decides to offload the task to itself, then we have $T(i,i,t) = 0$.

3.4.2 Relay offloading

Relay offloading requires simultaneous consideration of transmission delays between the UAV and RSU and between the UAVs. The channel gain between UAV i and UAV k is $h_{i,k}(t) = h_{i,k}^{\mathrm{LoS(t)}}$. Thus, for task offloading from UAV i to the offloading node $j \in \mathcal{N}_{k,t}$, the transmission delay is:

$$T(i,j,t) = T(i,k,t) + T(k,j,t), \tag{6a}$$

$$\text{s.t.} \quad j \in \mathcal{N}_{k,t}. \tag{6b}$$

3.5 Resource load model

A necessary measure to improve the task offload performance of UAV-assisted VECNs is to encourage the task offloading model to evenly distribute computing tasks on each RSU to avoid overburdening certain nodes, thereby preventing potential performance bottlenecks and failures. We evaluate the degree of load balancing of UAV u by calculating the standard deviation of the resource load of all RSUs covered by UAV u:

$$\Omega_{u,t} = \sqrt{\frac{1}{|\mathcal{N}_{u,t}|} \sum_{i\in\mathcal{N}_{u,t}} ((F_{i,t} \pm \widetilde{f}_{i,t}) - \bar{F}'_{u,t})^2}, \tag{7}$$

$$\bar{F}'_{u,t} = \frac{\sum_{i\in\mathcal{N}_{u,t}} (F_{i,t} \pm \widetilde{f}_{i,t})}{|\mathcal{N}_{u,t}|}, \tag{8}$$

where $\bar{F}'_{u,t}$ is the average resource load of all nodes in time slot t, taking into account the deviation of resource load due to DT.

3.6 Problem formulation

According to the above description, we can express the problem of optimizing the task offloading policy of UAV-assisted VECN in time slot t in terms of (1) maximizing local task throughput rate, (2) minimizing transmission delay, (3) minimizing the standard deviation of the loads of all the resources, to maximize the task offload performance of the UAV-assisted VECN:

$$\max_{\Pi} \sum_{t=1}^{T} \sum_{u=1}^{|\mathcal{N}^{\mathrm{UAV}}|} \Phi_{u,t} - \Omega_{u,t} - T_{u,r_u,t} \tag{9a}$$

$$\text{s.t.} \quad F_{i,t} \pm \widetilde{f}_{i,t} \geq O_{i,t}, i \in \mathcal{N}, \forall t \tag{9b}$$

$$\psi(u,t) \in [0,1], u \in \mathcal{N}^{\text{UAV}}, \forall t \tag{9c}$$

where $\Pi = \{\pi_1, \pi_2, ..., \pi_{|\mathcal{N}^{\text{UAV}}|}\}$ is the set of task offloading policies on all UAVs, the first constraint indicates that there can be no resource overload (i.e., the available resources are less than 0) at any of the scheduling nodes, and the second constraint indicates that the UAV can schedule at most one task in a time slot.

4 A3C-based uav task offloading optimization

4.1 Algorithm framework

In UAV-assisted VECNs, UAVs must make task offloading decisions in a dynamically changing environment involving state (e.g., current load and connectivity status of the RSUs), action (choice of task offloading), and cost (success or failure based on the offloading action.) The MDP, through the state transfer and the cost function, can efficiently model this process, thus providing UAVs with a systematic way to evaluate and optimize their decision-making policies. In this paper, we represent the task offload performance optimization problem for UAV-assisted VECNs as a stochastic optimization problem $\mathcal{G} = (\hat{\mathcal{S}}, \hat{\mathcal{A}}, \mathcal{R}, \mathcal{P}r, \mathcal{Y})$.

4.1.1 State space

$\hat{\mathcal{S}}$ is the joint state space, i.e., the set of all node states in the UAV-assisted VECN. For each time slot t, we construct a local state $S_{u,t} = \{\boldsymbol{s}_{i,t} : i \in \mathcal{N}_u\}$, which aggregates the information of DT nodes in the coverage area of all UAVs u, including the resource load of UAV u, i.e., $F_{u,t} \pm \widetilde{f}_{u,t}$, the resource load of RSUs, i.e., $\{F_{r,t} \pm \widetilde{f}_{r,t} : r \in \mathcal{N}_{u,t}\}$, and resource requests $O_{u,t}$ for the arriving task with delay requirement $D_{u,t}$. We maintain the global state $S_t = \{\boldsymbol{s}_{i,t} : i \in \mathcal{N}\}$ at the base station, which contains the states of all nodes, and for the transmission delays between each node, we maintain a matrix $T_t = [T_{i,j,t}]_{i,j \in \mathcal{N}}$.

4.1.2 Action space

$\hat{\mathcal{A}}$ denotes the joint action space of all UAVs in time slot t. where the action space of the UAV u specifies the index of the node to which the current task request can be offloaded. At each time slot t, each UAV is allowed to perform task offloading only once. We use $\hat{\boldsymbol{a}}_t$ to denote the joint decision action of all UAVs in time slot t, i.e., $a_{u,t} \in \hat{\boldsymbol{a}}_t$. As shown in Fig. 2, the action space of UAV u contains $|\mathcal{N}_{u,t}^{\text{Double-hop}} - 1|$ discrete actions denoted as $a_{u,t} = \{i\}_0^{|\mathcal{N}_{u,t}^{\text{Double-hop}} - 1|}$, where $\mathcal{N}_{u,t}^{\text{Double-hop}}$ denotes all nodes in the coverage area of UAV u as well as all other nodes reachable through only one relay UAV, defined as second-order neighbors of node u in the UAV-assisted VECN network topology.

4.1.3 Reward function

We design the reward function $\mathcal{R}$ generated by UAV u in time slot t as a combination of local task throughput rate $\Phi_{u,t}$ and local resource load balancing $\boldsymbol{\Omega}_{u,t}$ and the transmission delay $T_{u,r_u,t}$. At this point, $\mathcal{R}$ is defined as:

$$\mathcal{R}_{u,t} = \mathrm{e}^{\Phi_{u,t} - \lambda \mathrm{Sigmoid}(\Omega_{u,t}) - \gamma \mathrm{Sigmoid}(T_{u,r_u,t})}, \tag{10}$$

where λ and γ are the weights of the resource load term and the transmission delay term, and Sigmoid$(\cdot)$ converts the standard deviation or transmission delay to a value between 0.5 and 1. Exponential functions provide a nonlinear way to map reward values. This nonlinear and smooth property allows the reward function to be more sensitive to changes in, for example, task success, load balancing, and transmission latency.

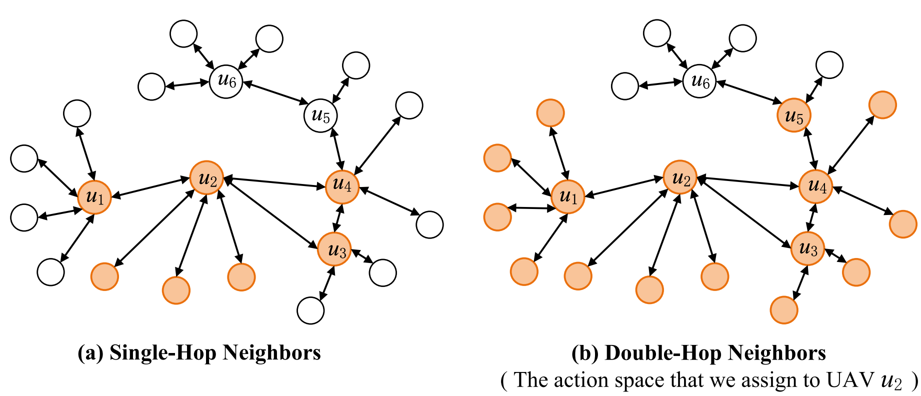


Fig. 2 Two types of neighbors in a DT topology network for UAV-assisted VECNs

4.1.4 Transition probability

$\mathcal{P}r(S_{t+1}|S_t, \hat{\boldsymbol{a}}_t)$ is the probability transition matrix, which outputs the probability distribution for transitioning to the subsequent state given the present state and task offloading action. $p(S_{u,t+1}|S_{u,t}, a_{u,t})$ is the probability that UAV u will transition from state $S_{u,t}$ to state $S_{u,t+1}$ in response to a deterministic task offloading action $a_{u,t}$.

4.1.5 Discount factor

The primary function of $\mathcal{Y}$ is to accelerate convergence by discounting the reward for the following state. We set $\mathcal{Y} = 0.9$, which helps each agent further consider future rewards when making decisions and pursue long-term common interests in collaborative VECN task scheduling.

4.2 Algorithm design

As discussed in Sect. 1, the main challenges in implementing low-latency, highly reliable task offloading policies in UAV-assisted VECNs include distributed coordination of heterogeneous RSU resource environments and modeling frequent changes in edge network topology. In this context, we combine A3C with GAT to provide a paradigm for addressing these challenges.

In the designed A3GAC architecture, we combine the advantages of the policy gradient approach and graph attention networks to address the task offload challenges of UAV-assisted VECNs. This architecture consists of multiple distributed UAV task offload Actors and a centralized state evaluation Critic, as shown in Fig. 3.

4.2.1 Distributed UAV task offload actor

In our Actor-Critic framework, each Actor represents a UAV that chooses a task offloading action based on the current state of the environment and its policy network. These deep neural network (DNN)–based policy networks are progressively optimized in interaction with the environment, to maximize long-term cumulative rewards. Through the policy gradient approach, UAV Actors can efficiently explore their action spaces and gradually learn how to make optimal decisions.

For UAV u, we deploy a policy network $\pi_{\theta_u^p}$ with parameter θ_u^p. UAV u will take action $a_{u,t}$ based on the local state $S_{u,t}$ and the policy $\pi_{\theta_u^p}$ to immediately receive the reward $\mathcal{R}_{u,t}$ and transition to the state $S_{u,t+1}$, where $\pi_{\theta_u^p}(a_{u,t}|S_{u,t})$ denotes the probability that the policy network outputs the unloading action $a_{u,t}$ when in state $S_{u,t}$. Then we employ the advantage function $A(S_{u,t}, a_{u,t})$ to appraise the merit of selecting the action $a_{u,t}$ in state $S_{u,t}$. The advantage function is articulated as

$$A(S_{u,t}, a_{u,t}) = \mathcal{R}_{u,t} + \mathcal{Y}v(S_{u,t+1};\theta^e) - v(S_{u,t};\theta^e), \tag{11}$$

where $v(S_{u,t};\theta^e)$ is the state evaluation function provided by Critic. Finally, we update the parameters θ^p of the policy network according to the advantage function:

$$\Delta\theta_u^p = \alpha^p \nabla_{\theta_u^p} \log \pi(a_{u,t}|S_{u,t};\theta_u^p) A(S_{u,t}, a_{u,t}), \tag{12}$$

where α^p is the learning rate of the policy network.

4.2.2 Centralized state evaluation critic

Critic uses GAT to evaluate the current global state and train by minimizing the difference between the expected global

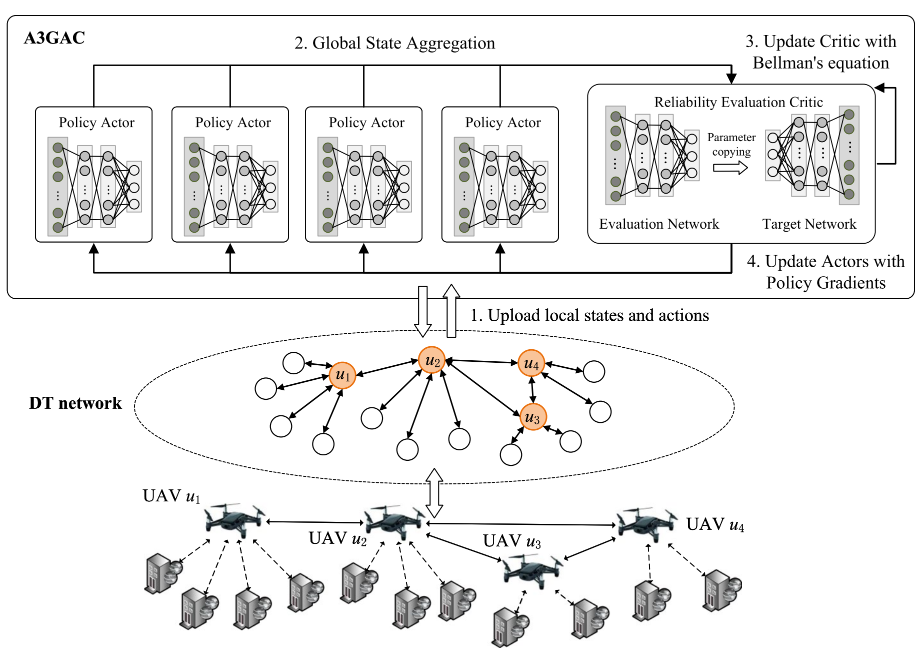


Fig. 3 A3GAC-based UAV-assisted VECN Task Offloading Framework

state and each actual local state. The introduction of GAT allows Critic to efficiently deal with the complex DT network topology and dynamic relationships between nodes in UAV-assisted VECNs.

For UAV u with time slot t, its node feature $\boldsymbol{s}_{u,t}$ can be expressed as a high-dimensional vector, i.e., $\boldsymbol{s}_{u,t} \in \mathbb{R}^{F^u}$, and for the node feature of RSU r, we express it as $\boldsymbol{s}_{r,t} \in \mathbb{R}^{F^r}$. To obtain the same expressiveness for the state evaluation network in Critic, we first use the learnable linear transformation $\boldsymbol{W}^{\mathrm{uav}} \in \mathbb{R}^{F' \times F^u}$ with $\boldsymbol{W}^{\mathrm{rsu}} \in \mathbb{R}^{F' \times F^r}$ maps the features of all UAV and RSU nodes into a unified F'-dimensional vector space, respectively, we denote the mapped set of node features as $\{\boldsymbol{s}'_{i,t} : i \in \mathcal{N}\}$. Then, for node $i \in \mathcal{N}_{u,t}$, we compute the concern coefficient between it and node j:

$$e_{ij} = \boldsymbol{a}([\boldsymbol{s}'_{i,t} || \boldsymbol{s}'_{j,t}]), \tag{13a}$$

$$\text{s.t.} \quad j \in \mathcal{N}_{u,t}^{\text{Double-hop}}, \tag{13b}$$

where $||$ denotes the stitching of the vectors, $\boldsymbol{a}$ is a trainable parameter, which aims to map high-dimensional node features to raw attention scores. Then, the similarity coefficients are then normalized using the softmax function with LeakyReLU nonlinearity:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(e_{ij}))}{\sum_{k \in \mathcal{N}_{u,t}^{\text{Double-hop}}} \exp(\text{LeakyReLU}(e_{ik}))}, \tag{14}$$

where α_{ij} quantifies the relative importance of node j when node i is deciding where to offload tasks. Higher values of α_{ij} indicate that node i views node j as more favorable for offloading tasks due to factors such as lower expected latency or better resource availability. Essentially, α_{ij} acts as a soft decision metric that guides the distribution of tasks across the network, aiming to optimize overall system performance while adhering to operational constraints.

Next, the calculated attention weights α_{ij} are used to perform a weighted summation to update the features of each node. Equation 15 incorporates information from neighboring nodes to generate new node features:

$$\boldsymbol{s}_{i,t}^{\text{att}} = \sigma^{\text{att}}\left(\sum_{j \in \mathcal{N}_{u,t}^{\text{Double-hop}}} \alpha_{ij} \boldsymbol{s}'_{j,t}\right), \tag{15}$$

where σ^{att} is the activation function of the GAT layer. In this way, GAT can construct an attention mechanism between different nodes, generating local state $\boldsymbol{s}_{i,t}^{\text{att}}$ for node i. We then use average pooling to aggregate the features of all nodes in the UAV-assisted VECN and generate s_t that represent the global state via a fully connected output layer:

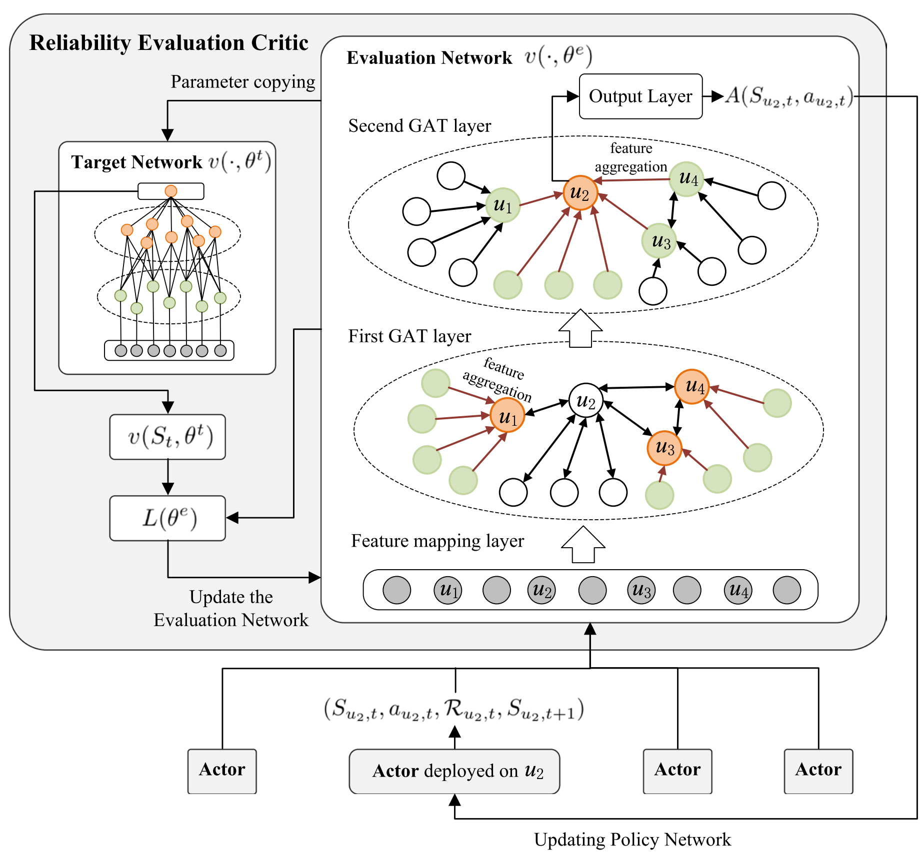


Fig. 4 Algorithmic structure of A3GAC. The learning process of the task offloading policy for a UAV Actor deployed on UAV u_2 is demonstrated: 1. UAV Actor u interacts with the local environment to obtain the experience $(S_{u_2,t}, a_{u_2,t}, \mathcal{R}_{u_2,t}, S_{u_2,t+1})$ and upload it to Critic; 2. Critic aggregates all local states to form a global state and updates the state evaluation network and target network parameters; 3. Each UAV Actor independently updates its policy network based on the advantage feedback from the critics. In the next time slot, all UAV Actors and Critic will repeat the above process

$$s_t = \sigma^{\text{out}}\left(\boldsymbol{W}^{\text{out}}\left(\frac{1}{|\mathcal{N}_t|} \sum_{i \in |\mathcal{N}_t|} \boldsymbol{s}_{i,t}^{\text{att}} \right) + b^{\text{out}} \right), \tag{16}$$

where σ^{out}, $\boldsymbol{W}^{\text{out}}$ and b^{out} are the activation functions, weight matrices, and bias terms of the output layer, respectively.

We denote the above state evaluation network as $v(\cdot;\theta^e)$, where θ^e is a joint parameter. In particular, if the input of the evaluation network is the global state S_t, the output of the evaluation network $s_t = v(S_{u,t};\theta^e)$ represents the global state of the UAV-assisted VECN, while if the input is the local state $S_{u,t}$, the output of the evaluation network $s_{u,t} = v(S_{u,t};\theta^e)$ represents the local state of UAV u. For UAV u and its local state $S_{u,t}$, we use Bellman's equation to compute the loss to train the state evaluation network:

$$L(\theta^e) = \left(v(S_{u,t};\theta^e) - (\mathcal{R}_{u,t} + \mathcal{Y} v(S_t;\theta^t)) \right)^2, \tag{17}$$

where $v(\cdot;\theta^t)$ is an additional target network introduced with a parameter θ^t that is periodically updated from the state evaluation network θ^e to stabilize the learning process. Figure 4 illustrates the interaction and parameter update process of a UAV Actor deployed on UAV u_2 with a Critic containing a two-layer GAT network, and the A3GAC algorithm is illustrated in Algorithm 1.

Algorithm 1 A3GAC

Initialization: Initialize state evaluation network parameters θ^e, target network parameters θ^e and policy network parameters $\{\theta_u^p : u \in \mathcal{N}^{\text{UAV}}\}$.
for each time slot t **do**
  for each **Actor** asynchronously **do**
    Observe local state $S_{u,t}$
    Select action $a_{u,t}$ using $\pi_{\theta_u^p}$
    Execute action $a_{u,t}$, observe $\mathcal{R}_{u_t}$ and $S_{u,t+1}$
    Upload experience $(S_{u,t}, a_{u,t}, \mathcal{R}_{u,t}, S_{u,t+1})$ to Critic
  end for
  Observe the global DT network topology $\mathcal{G}_t$ and the global state S_t
  Aggregate node features based on attention weights and compute global state s_t using (10)-(14)
  Update the state evaluation network parameters θ^e centrally:

$$L(\theta^e) = \left(v(S_{u,t};\theta^e) - (\mathcal{R}_{u,t} + \mathcal{Y} v(S_t;\theta^t)) \right)^2$$

  for each **Actor** asynchronously **do**
    Compute advantage function separately:

$$A(S_{u,t}, a_{u,t}) = \mathcal{R}_{u,t} + \mathcal{Y} v(S_{u,t+1};\theta^e) - v(S_{u,t};\theta^e)$$

    Update the parameters of the policy networks separately:

$$\Delta\theta_u^p = \alpha_p \nabla_{\theta_u^p} \log \pi(a_{u,t}|S_{u,t};\theta^p) A(S_{u,t}, a_{u,t})$$

  end for
  Periodically update the target network:

$$\theta^t = \theta^e$$

end for

---

5 Experiments and evaluations

In this section, we conduct simulation experiments to evaluate the performance of A3GAC concerning task offloading. We first deploy an edge cluster network on Google Cloud Platform (GCP) [48] to simulate the DT layer of the UAV-assisted VECN, and second, simulate task arrivals using real workload traces. We evaluate and compare the performance of A3GAC with other algorithmic policies and analyze the experimental results.

5.1 Experiment setup

5.1.1 DT-driven task offloading system setup

We deployed 1 base station server and 10 edge clusters on GCP, the base station cluster consists of 16 virtual machines (VMs) configured with "4 vCPU, 16 GB RAM and 4 vGPU". Each edge cluster consists of 1 master node and 8 edge nodes, where each master node, representing DT-UAV, consists of 4 VMs configured with "4 vCPU, 16 GB RAM, and 4 vGPU" and each edge node, representing DT-RSU, consists of 8 VMs configured with " 4vCPU, 32 GB RAM and 8 vGPU".

To model the dynamically changing edge DT network topology at the DT layer, we maintain a binary mask vector for each UAV at each time slot to mask nodes that are failing (malfunctioning or out of coverage of that UAV). We then let the output of the UAV Actor policy network be multiplied element-by-element with this vector such that the final task offloading policy satisfies the current DT network topology.

5.1.2 Task requests

We simulated task requests in the cluster environment using the real compute-intensive task dataset 'cluster-trace-gpu-v2023' [49], which contains 8152 tasks submitted to the GPU cluster and lists their resource specifications for CPU, RAM & GPU. The input we use is sampled and clipped from the default file, highlighting the workloads of GPU-shared tasks. GPU-sharing refers to the ability of multiple machine learning tasks to run safely on a single GPU that is guaranteed to be isolated, where each task is allocated a portion of the GPU's resources through virtualization. Figure 5 illustrates the distribution of CPU, memory, and GPU requirements for the sampled dataset.

5.1.3 A3GAC setup

A3GAC runs a UAV Actor on each master node, which periodically observes and collects the current resource load information and network latency of the edge clusters through the state monitors deployed on the master nodes and edge nodes, uses the policy network to make task offloading decisions, and uploads the current state to the cloud cluster; correspondingly, A3GAC deploys a Critic on the cloud cluster, which summarizes the state of the edge clusters, updates the state evaluation network and target network parameters, and returns the advantage function estimation based on the current state of each UAV Actor to each UAV Actor respectively, and finally, each Actor updates the policy network parameters based on the advantage function.

5.2 Training settings

The detailed parameter settings for the experiment are shown in Table 1. We implement A3GAC based on TensorFlow 2.0. The policy network $\pi(\cdot;\theta^p)$ in each Actor is implemented independently using a three-layer ReLU neural network (NN) with 256, 128, and 32 hidden units per layer, and the output layer uses ReLU+1 as an activation function to ensure that the output elements are positive; The state evaluation network $v(\cdot;\theta^e)$ is implemented using a two-layer GAT containing 128 units, and the output layer uses a fully-connected layer containing one neuron. All networks were trained using the Adam optimizer.

5.3 Baseline methods

To evaluate the effectiveness of A3GAC, we selected the following four task offloading strategies for comparison experiments. We classify the baseline algorithms into centralized task offloading and distributed task offloading algorithms, for the centralized task offloading algorithm, is completely deployed on the base station side, and the UAV needs to upload the local observation state $S_{u,t}$ to the base station server firstly, and wait for the offloading decision to be made

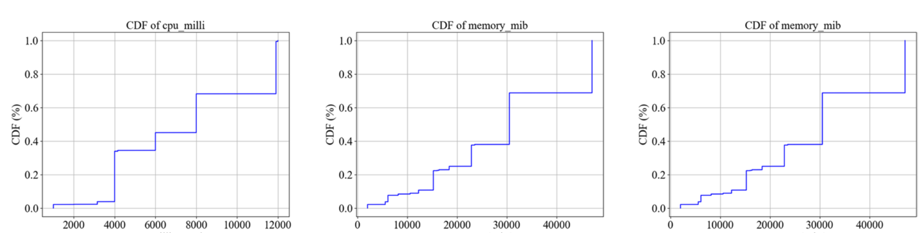


Fig. 5 Alibaba cluster trace v2023. **a** Number of CPUs requested (in milli). **b** Main memory requested (in MiB). **c** Number of GPUs requested (in milli)

in the base station server, i.e., the UAV on the edge side cannot make the task offloading decision independently; whereas for the Distributed task offloading algorithm, it will be deployed both on the base station side and the UAV side, and the communication between the base station and each UAV is only used for the training of the state evaluation network $v(\cdot;\theta^e)$ and policy networks, so the UAVs can independently execute the offloading decision after the task arrives at each UAV.

5.3.1 Centralized task offloading strategy

- **Random**. This strategy randomly offloads each task request to any node within the UAV-assisted VECN.
- **Greedy**. This strategy prefers to offload each task request to the edge node with the smallest resource load.
- **Deep Q-Learning (DQN)**. We appropriately modify the underlying DQN algorithm to make it meet the requirements of UAV-assisted VECN. At each time slot, the DQN selects actions based on the ϵ-greedy policy, generates the returns, and then trains the network using empirical playback techniques.

5.3.2 Distributed task offloading strategy

- **A3C**. We modify Critic's state evaluation network and target network in A3GAC to a DNN with 3 hidden layers and keep the number of neurons in each network constant, thus reproducing the distributed A3C algorithm.
- **MAPPO** [44]. Multi-Intelligent Proximal Policy Optimization (MAPPO) is an extension of the Proximal Policy Optimization (PPO) algorithm that focuses on the interaction and coordination among multiple intelligences. Same as A3GAC, we adopt a multi-intelligentsia framework with centralized training and decentralized execution. An Actor is deployed with a Critic on each UAV, and a DNN deployed at the base station side guides each Critic to perform local state evaluation by summarizing the global state.

5.4 Experimental evaluation

5.4.1 Global task offload performance analysis

Figure 6 shows the long-term task throughput rate of the entire UAV-assisted VECN. It is worth mentioning that the comparison experiments are carried out in a constant DT network topology. We can observe: **(O1) Global task throughput rate:** with the increase of time slot, all algorithms except Random can gradually improve the global task throughput rate. After 200 time slots, A3GAC achieves leading performance, indicating its greater effectiveness in enhancing the global performance of UAV-assisted VECNs compared to other baseline algorithms. **(O2) Stability:** In terms of stability, MAPPO is slightly better than A3GAC because MAPPO uses a truncated policy gradient approach to avoid excessive policy changes by limiting the magnitude of the policy updates; meanwhile, the stability of A3GAC is better than that of A3C and DQN, which is because these two algorithms use the experience of different UAVs to train the same policy network, which complicates finding a policy network parameter that stably matches various edge environments. **(O3) Convergence speed:** In terms of convergence speed, A3GAC significantly outperforms the other algorithms, and A3GAC reaches its optimal performance about 70-time slots earlier than MAPPO. This gap comes from two key designs: first, GAT has outstanding task offload performance advantages in global information aggregation compared to the underlying NNs, and second, MAPPO needs to train multiple Critic networks at the same time, which dramatically increases the UAV's computational burden and makes the UAV has to invest more computational resources

Table 1 Simulation parameters

Parameters	Meaning	Value
P_{Noise}	Power of noise	−100 dBm
P_{UAV}	Transmit power of UAV	10 W
$h_{i,j}$	Reference channel gain	−50 dB
$d_{u,r}$	Distance between UAV and RSU	50 m
d_{u_1,u_2}	Distance between 2 UAVs	1000 m
B	Bandwidth	5 MHz
D	Transmission data size of the task	100 Mib
η	A3GAC learning rate	10^{-3}
λ	Resource load weight	1
$\widetilde{f}_{i,t}$	DT deviation of normalized resource load	[0.01 − 0.05]

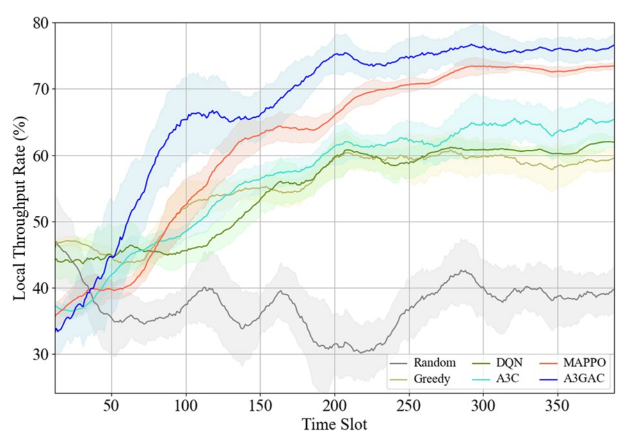


Fig. 6 Global long-term throughput rate comparison

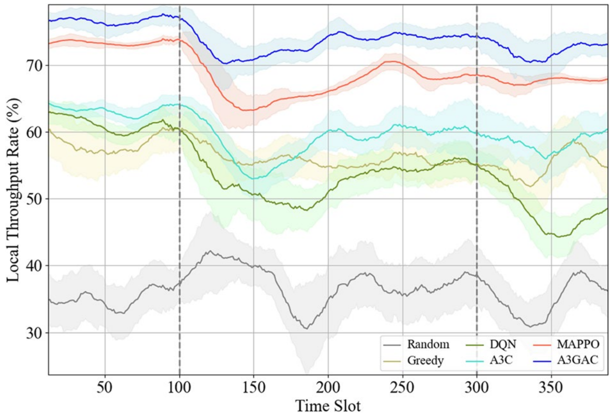


Fig. 7 Task offload performance analysis under dynamic DT network topology (updated at time slots 100 and 300)

to train the state evaluation network instead of performing local task offloading.

5.4.2 Task offload performance analysis under dynamic DT network topology

We update the DT network topology at time slots 100 and 300 and then show in Fig. 7 the change in task throughput rate of A3GAC versus other baseline algorithms in scenarios where the DT network topology dynamically changes on the edge side. We can observe that the distributed task offloading algorithm represented by A3GAC has better recovery capability compared to the centralized task offloading algorithm, which reflects the necessity of a distributed task offloading decision system. Specifically, the distributed algorithm allows each UAV to make decisions based on its localized information, and this localized decision-making process can quickly adapt to changes in the surrounding environment because it does not need to wait for the central node to collect and process global information before responding; in addition, the distributed task offloading system has better resilience and robustness, and even if some edge nodes become disengaged, other nodes can continue to work. This decentralized feature makes the whole system more robust in the face of an unstable environment.

5.4.3 Task offload performance analysis under localized resource overload

The local resource overload of UAV-assisted VECN is very likely to occur in traffic-congested sections of the city, to simulate this kind of scenario, we make the normalized resource load of all edge nodes of the edge cluster u_1 constant to 0.8 at time slot 100 after the convergence of the A3GAC and other baseline algorithms. This setup makes

Fig. 8 Task offload performance analysis under localized resource overload

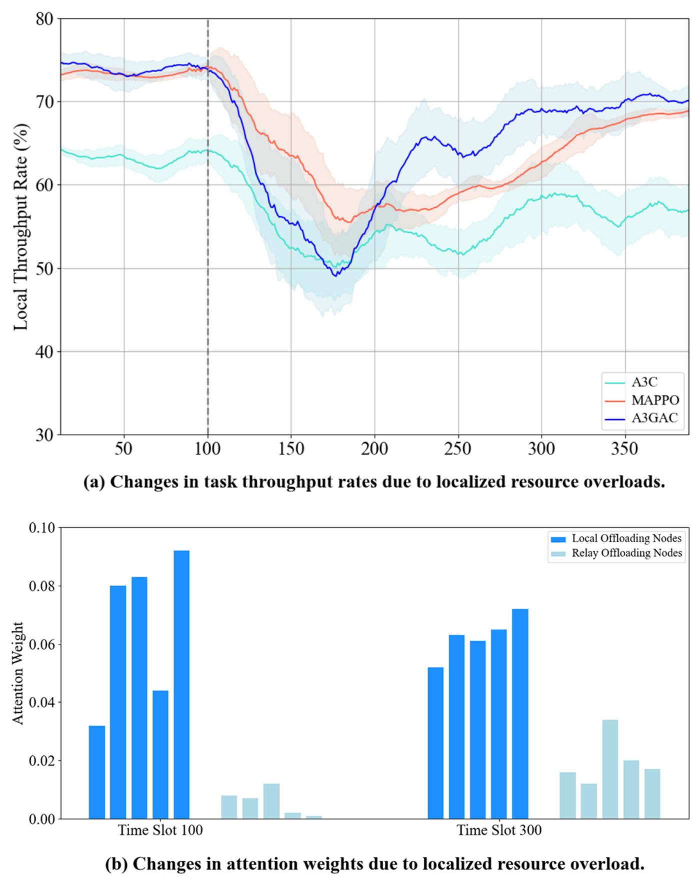
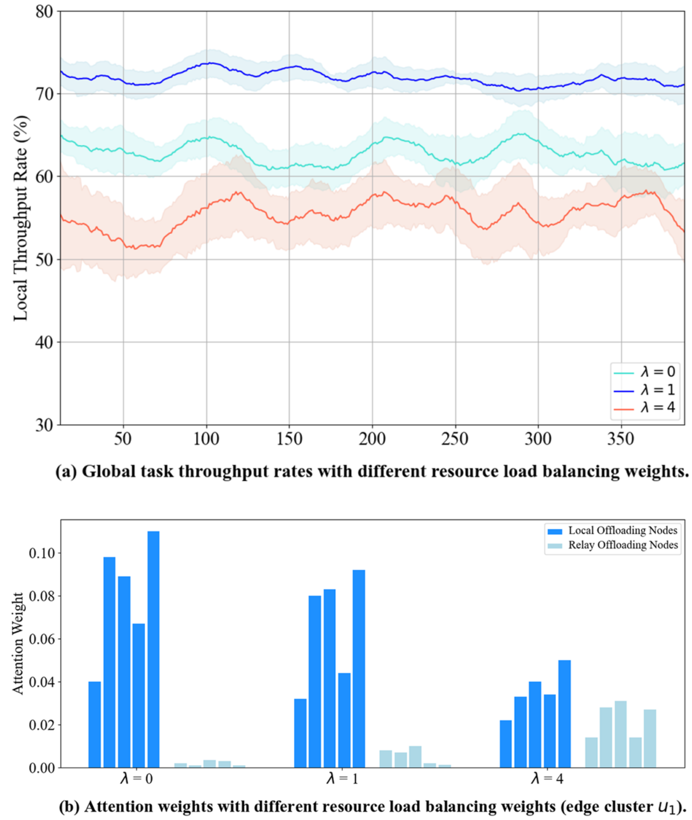


Fig. 9 Task offload performance analysis under different resource load weights λ



the variation of the local task throughput rate of edge cluster u_1 is given in Fig. 8a, we can observe **(O1)** Both A3GAC and MAPPO can handle the special case of local resource overload effectively, and A3GAC is slightly better in avoiding task failure and recovery efficiency. **(O2)** The distributed A3C algorithm does not perform well in this scenario because A3C tries to use a policy network to match all edge environments, but the huge resource heterogeneity between the edge cluster u_1 and the other edge clusters hinders the convergence of such a pervasive policy network.

Figure 8b gives the change in the attention weight of node $u1$ relative to the local offloading nodes (nodes inside the edge cluster u_1) and relay offloading nodes (second-order neighboring nodes of $u1$) in the Critic evaluation network. The attention weights represent the relative importance assigned to different nodes when making offloading decisions. This metric is calculated in the Critic evaluation network based on the expected contribution of each node to reducing latency and balancing load within the network, and it directly influences the task offloading strategy. Higher attention weights indicate a greater preference for offloading tasks to these nodes, highlighting their strategic importance in the network configuration. After the initial convergence of A3GAC, the local offloading node obtains a higher attention weight relative to the relaying offloading node, which is due to the advantage of local offloading node with low transmission delay in the case of relative load balancing of resources inside and outside the cluster, which makes A3GAC prefer local offloading node. And when the resources of the edge cluster u_1 are overloaded, A3GAC significantly increases the attention weight of relay offloading nodes, which is because offloading tasks to these nodes may help to equalize the load of the whole network, thus optimizing the task offload performance of the overall system.

5.4.4 Task offload performance analysis under different resource load weights λ

We assign different values to the weight λ for resource load balancing in the reward function of A3GAC, and at the same time observe its impact on the global task throughput rate and attention weight. As shown in Fig. 9, when $\lambda = 1$, the global task throughput rate is optimal, and the attention weight of the local offload node is slightly higher than that of the relay offload node, which indicates that A3GAC still prefers the local offload node when offloading tasks, but does not ignore the load balancing rewards from the relay offload node. When $\lambda = 0$, it means that A3GAC does not consider load balancing, at which time the attention weight of relay offloading nodes is significantly higher than that of relay offloading nodes, which is a potential behavior leading to local resource overloading discussed in Sect. 5.4.3. When $\lambda = 4$, the global task throughput rate shows a significant decrease, at which time A3GAC is overly concerned with load balancing, and the attention weights are largely determined by the resource load of each node, while the basic considerations of low latency and high response are abandoned. This indicates that A3GAC needs to find a balance between transmission delay and load balancing to simultaneously satisfy the needs of UAV-assisted VECN for both.

6 Conclusion

In this paper, we have explored the A3GAC algorithm, a distributed collaborative task offloading solution based on A3C and GAT. The approach addresses the non-convex and highly complex task offloading challenges in UAV-assisted VECNs, and significantly improves the efficiency of identifying and resolving performance and safety issues, thus enabling proactive maintenance and fast network recovery. A3GAC comprehensively considers both global and local state information. It utilizes A3C and GAT algorithms for evaluating global state at the base station and independently updates the task offloading strategies. It also realizes the algorithm's integrated consideration of global state information and local state information during the task offloading process. The superior performance of A3GAC across multiple simulation scenarios demonstrates its effectiveness. Future work will focus on integrating UAV navigation and service deployment states into the A3GAC framework. This will enable joint optimization of UAV trajectory and task offloading.

Acknowledgements This work is supported in part by the National Key R &D Program of China under No.2022YFB3304303; The National Science Foundation of China (62271128, U2333207, 62002047); Sichuan Science and Technology Plan "Unveiling and Leading" Project under Grants 23JBGOV0005; Sichuan Science and Technology Plan Key R &D project (No. 2022ZDZX004, 2023YFG0029, 2023YFG0150, 2022YFG0212, 2021YFS0391); Chengdu Key R &D Support Plan "Un-veiling and Leading" Project under Grants 2022-JB00-00013-GX; Sichuan Science and Technology Program (No. 2024NSFJQ0030, 2022NSFSC0552).

References

1. Hu, J., Chen, C., Qiu, T., & Pei, Q. (2020). Regional-centralized content dissemination for ev2x services in 5g mmwave-enabled iov. *IEEE Internet of Things Journal, 7*, 7234–7249.
2. Tang, F., Mao, B., Kato, N., & Gui, G. (2021). Comprehensive survey on machine learning in vehicular network: Technology, applications and challenges. *IEEE Communications Surveys & Tutorials, 23*, 2027–2057.
3. Chen, J., Liu, G., & Liu, Y. (2020). Lightweight privacy-preserving raw data publishing scheme. *IEEE Transactions on Emerging Topics in Computing, 9*, 2170–2174.
4. Hu, Y., Miao, X., Si, Y., Pan, E., & Zio, E. (2022). Prognostics and health management: A review from the perspectives of design, development and decision. *Reliability Engineering & System Safety, 217*, 108063.
5. Chen, J., Liu, Y., Xiang, Y., & Sood, K. (2021). Rpptd: Robust privacy-preserving truth discovery scheme. *IEEE Systems Journal, 16*, 4525–4531.
6. Hu, J., et al. (2021). Uav-assisted vehicular edge computing for the 6g internet of vehicles: Architecture, intelligence, and challenges. *IEEE Communications Standards Magazine, 5*, 12–18.
7. Khawaja, W., Guvenc, I., Matolak, D. W., Fiebig, U.-C., & Schneckenburger, N. (2019). A survey of air-to-ground propagation channel modeling for unmanned aerial vehicles. *IEEE Communications Surveys & Tutorials, 21*, 2361–2391.
8. Li, P., et al. (2023). Eptask: Deep reinforcement learning based energy-efficient and priority-aware task scheduling for dynamic vehicular edge computing. *IEEE Transactions on Intelligent Vehicles., 23*, 201.
9. Gao, H. et al. (2023). Tbdb: Token bucket-based dynamic batching for resource scheduling supporting neural network inference in intelligent consumer electronics. *IEEE Transactions on Consumer Electronics*.
10. Apostolopoulos, P. A., Fragkos, G., Tsiropoulou, E. E., & Papavassiliou, S. (2021). Data offloading in UAV-assisted multi-access edge computing systems under resource uncertainty. *IEEE Transactions on Mobile Computing, 22*, 175–190.
11. Xu, Y. et al. (2023). Heterogeneous data-driven failure diagnosis for microservice-based industrial clouds towards consumer digital ecosystems. *IEEE Transactions on Consumer Electronics*.
12. Khan, M. F., Yau, K.-L.A., Ling, M. H., Imran, M. A., & Chong, Y.-W. (2022). An intelligent cluster-based routing scheme in 5g flying ad hoc networks. *Applied Sciences, 12*, 3665.
13. Wang, S.-H., Tu, C.-H., & Juang, J.-C. (2022). Automatic traffic modelling for creating digital twins to facilitate autonomous vehicle development. *Connection Science, 34*, 1018–1037.
14. Yi, M., Yang, P., Chen, M., & Loc, N. T. (2022). A DRL-driven intelligent joint optimization strategy for computation offloading and resource allocation in ubiquitous edge iot systems. *IEEE Transactions on Emerging Topics in Computational Intelligence, 7*, 39–54.
15. Li, Y. (2017). Deep reinforcement learning: An overview. arXiv preprint arXiv:1701.07274.
16. Gao, H., Wang, X., Wei, W., Al-Dulaimi, A., & Xu, Y. (2023). Com-ddpg: task offloading based on multiagent reinforcement learning for information-communication-enhanced mobile edge computing in the internet of vehicles. *IEEE Transactions on Vehicular Technology*.
17. Lei, L., Tan, Y., Dahlenburg, G., Xiang, W., & Zheng, K. (2020). Dynamic energy dispatch based on deep reinforcement learning in iot-driven smart isolated microgrids. *IEEE Internet of Things Journal, 8*, 7938–7953.
18. Doshi, A., Yerramalli, S., Ferrari, L., Yoo, T., & Andrews, J. G. (2021). A deep reinforcement learning framework for contention-based spectrum sharing. *IEEE Journal on Selected Areas in Communications, 39*, 2526–2540.
19. Yi, M., Yang, P., Du, M., & Ma, R. (2022). Dmadrl: A distributed multi-agent deep reinforcement learning algorithm for cognitive offloading in dynamic mec networks. *Neural Processing Letters, 54*, 4341–4373.

20. Peng, H., & Shen, X. (2020). Multi-agent reinforcement learning based resource management in mec-and uav-assisted vehicular networks. *IEEE Journal on Selected Areas in Communications, 39*, 131–141.
21. Babaeizadeh, M., Frosio, I., Tyree, S., Clemons, J., & Kautz, J. (2016). Reinforcement learning through asynchronous advantage actor-critic on a gpu. arXiv preprint arXiv:1611.06256.
22. Veličković, P. et al. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903.
23. Jiang, X., Yu, F. R., Song, T., & Leung, V. C. (2021). Resource allocation of video streaming over vehicular networks: A survey, some research issues and challenges. *IEEE Transactions on Intelligent Transportation Systems, 23*, 5955–5975.
24. Chen, J., et al. (2024). Industrial blockchain threshold signatures in federated learning for unified space-air-ground-sea model training. *Journal of Industrial Information Integration, 39*, 100593.
25. Gyawali, S., Xu, S., Qian, Y., & Hu, R. Q. (2020). Challenges and solutions for cellular based v2x communications. *IEEE Communications Surveys & Tutorials, 23*, 222–255.
26. Zhang, Z., Hu, Q., Hou, G., & Zhang, S. (2023). A real-time discovery method for vehicle companion via service collaboration. *International Journal of Web Information Systems, 19*, 263–279.
27. Shi, W., et al. (2018). Drone assisted vehicular networks: Architecture, challenges and opportunities. *IEEE Network, 32*, 130–137.
28. Shafin, R., et al. (2020). Artificial intelligence-enabled cellular networks: A critical path to beyond-5g and 6g. *IEEE Wireless Communications, 27*, 212–217.
29. Ale, L., Zhang, N., Wu, H., Chen, D., & Han, T. (2019). Online proactive caching in mobile edge computing using bidirectional deep recurrent neural network. *IEEE Internet of Things Journal, 6*, 5520–5530.
30. Chen, D., et al. (2020). Magleak: A learning-based side-channel attack for password recognition with multiple sensors in iiot environment. *IEEE Transactions on Industrial Informatics, 18*, 467–476.
31. Hou, X., et al. (2020). Reliable computation offloading for edge-computing-enabled software-defined iov. *IEEE Internet of Things Journal, 7*, 7097–7111.
32. Liu, J., & Zhang, Q. (2020). Using imperfect transmission in mec offloading to improve service reliability of time-critical computer vision applications. *Ieee Access, 8*, 107364–107372.
33. Liu, J., Wu, Z., Liu, J., & Zou, Y. (2022). Cost research of internet of things service architecture for random mobile users based on edge computing. *International Journal of Web Information Systems, 18*, 217–235.
34. Xu, Y., Zhang, T., Loo, J., Yang, D., & Xiao, L. (2021). Completion time minimization for UAV-assisted mobile-edge computing systems. *IEEE Transactions on Vehicular Technology, 70*, 12253–12259.
35. Liu, Y., Li, Y., Niu, Y., & Jin, D. (2019). Joint optimization of path planning and resource allocation in mobile edge computing. *IEEE Transactions on Mobile Computing, 19*, 2129–2144.
36. Xue, H., Chen, D., Zhang, N., Dai, H.-N., & Yu, K. (2023). Integration of blockchain and edge computing in internet of things: A survey. *Future Generation Computer Systems, 144*, 307–326.
37. Shengdong, M., Fengyu, W., Zhengxian, X., Xiao, Z., & Lunfeng, Z. (2020). Optimal path strategy for the web computing under deep reinforcement learning. *International Journal of Web Information Systems, 16*, 529–544.
38. Gao, H., Wu, Y., Xu, Y., Li, R. & Jiang, Z. (2023). Neural collaborative learning for user preference discovery from biased behavior sequences. *IEEE Transactions on Computational Social Systems*.
39. Zhang, L., et al. (2018). Energy-aware dynamic resource allocation in UAV assisted mobile edge computing over social internet of vehicles. *IEEE Access, 6*, 56700–56715.
40. Garg, S., Singh, A., Batra, S., Kumar, N., & Yang, L. T. (2018). Uav-empowered edge computing environment for cyber-threat detection in smart vehicles. *IEEE Network, 32*, 42–51.
41. Yang, C., et al. (2022). Learning based channel allocation and task offloading in temporary UAV-assisted vehicular edge computing networks. *IEEE Transactions on Vehicular Technology, 71*, 9884–9895.
42. Yan, M., Xiong, R., Wang, Y. & Li, C. (2023). Edge computing task offloading optimization for a uav-assisted internet of vehicles via deep reinforcement learning. *IEEE Transactions on Vehicular Technology*.
43. Zhang, P., et al. (2023). Deep reinforcement learning based computation offloading in uav-assisted edge computing. *Drones, 7*, 213.
44. Kang, H. et al. (2023). Cooperative uav resource allocation and task offloading in hierarchical aerial computing systems: A mappo based approach. *IEEE Internet of Things Journal*.
45. Coutinho, R. W., & Boukerche, A. (2022). Uav-mounted cloudlet systems for emergency response in industrial areas. *IEEE Transactions on Industrial Informatics, 18*, 8007–8016.
46. Almeida, J., Ferreira, J., & Oliveira, A. S. (2018). A medium guardian for enhanced dependability in safety-critical wireless systems. *IEEE Transactions on Intelligent Transportation Systems, 19*, 965–976.
47. Hoa, N. T., Luong, N. C., Van Le, D., & Niyato, D. et al. (2023). Deep reinforcement learning for multi-hop offloading in uav-assisted edge computing. *IEEE Transactions on Vehicular Technology*.
48. Bisong, E. & Bisong, E. (2019). An overview of google cloud platform services. *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners* 7–10.
49. Siavashi, A. & Momtazpour, M. (2023). Gpu cluster dynamics: Insights from alibaba's 2023 trace release.

Zhiyang Zhang received the B.E. degree from the School of Information and Software, University of Electronic Science and Technology of China in 2018.He is currently pursuing the Ph.D. degree in the School of Information and Software, University of Electronic Science and Technology of China.His current research interests include reinforcement learning and edge computing.

Fengli Zhang received the M.S. and Ph.D. degrees from University of Electronic Science and Technology of China in 2009 and 2013, respectively. He is currently an assistant professor with University of Electronic Science and Technology of China. His research interests include quantum computation, security of network, and big data.

Minsheng Cao is currently an associate researcher at the School of Information and Software Engineering at the University of Electronic Science and Technology of China, where his research approach includes artificial intelligence, information security and privacy protection.

Chaosheng Feng is currently a professor and doctoral supervisor at Sichuan Normal University, and a distinguished member of the Chinese Computer Society. He has published more than 80 papers in international and domestic influential journals and conferences, such as IEEE Network, IEEE Transactions on Industrial Informatics, Journal of Computing, Journal of Software and so on. His main research interests include data security and privacy computing, cloud computing and big data, federated learning, and smart education.

Dajiang Chen is currently an Associate Professor in the School of Information and Software Engineering at University of Electronic Science and Technology of China (UESTC). He was a postdoc research fellow with the BBCR group, Department of Electrical and Computer Engineering, University of Waterloo, Canada, from 2015 to 2017. He received the Ph.D. degree in information and communication engineering from UESTC in 2014. Dr. Chen served as the workshop chair for BDEC-SmartCity'19(in conjunction with IEEE WiMob 2019). He also served as a Technical Program Committee Member for IEEE Globecom, IEEE ICC, and IEEE VTC. His current research interests include Physical Layer Security, Secure Channel Coding, and Machine Learning and its applications in Wireless Network Security and Wireless Communications.