Federated Reinforcement Learning Enabled Joint Communication, Sensing and Computing Resources Allocation in Connected Automated Vehicles Networks

Qixun Zhang, *Member, IEEE,* Hao Wen, Ying Liu, Shuo Chang, Zhu Han, *Fellow, IEEE*

***Abstract*—For future Connected Automated Vehicles (CAVs) networks, the joint optimization of communication, sensing and computing resources is crucial to guarantee the performance of cooperative automated driving's safety, which is attracting more and more attention. However, the existing works have not considered the low latency requirement for the raw perception data sharing with both wireless communication link capability and computing efficiency constraints, causing a serious threat to the cooperative automated driving's safety in CAVs networks. In this paper, a vehicle-road-base station cooperation architecture is designed, and a federated reinforcement learning based task offloading and resource allocation algorithm in the CAVs network is proposed to reduce the task execution delay with different communication and computing constraints. The problem of execution delay minimization is theoretically formulated and analyzed under three task practical offloading modes. To adapt to the dynamic topology of the CAVs network, we design a deep reinforcement learning algorithm to achieve the optimal task offloading and resource allocation. To further reduce the data transmission overhead of centralized reinforcement learning algorithm, the federated reinforcement learning enabled algorithm is proposed to minimize the execution delay of the optimal task offloading and resource allocation among multiple CAVs. Both the simulation and hardware testbed results verify that the proposed algorithms can not only reduce the execution delay and the communication overhead, but also improve the system throughput.**

***Index Terms*—Connected Automated Vehicles, Deep Reinforcement Learning, Federated Reinforcement Learning, Intelligent Transportation Systems, Mobile Edge Computing, Vehicular Networks.**

I. Introduction

Driven by the artificial intelligence (AI) technology, automated vehicles (AVs) have attracted extensive attention from both industry and academia worldwide recently [1]. In order to improve the driving safety of AVs, multiple sensors have been deployed extensively and accumulatively, including cameras, radar and LiDAR [2]. However, when the sensor failure occurs due to the bad weather and obstacle blockage issues, single AV will face the severe safety challenge that is beyond the capabilities of accumulating multiple sensors on-board.

With the fast development of beyond the fifth-generation (5G) and sixth-generation (6G) wireless communication technologies, the concept of connected automated vehicles (CAVs) has been proposed to improve the driving safety of AVs by enhancing the environment awareness capability with B5G and 6G wireless communication technologies [3]. Through the cooperation between multiples AVs and infrastructures, the sensing and computing capabilities of CAVs can be improved substantially. The processing delay of environment sensing tasks can be reduced, and the driving safety of CAVs can be improved as well. As a popular topic in the ITU-R standards, the use cases, spectrum requirements and radiocommunication requirements of CAVs have been defined and discussed [4], where both the Sub-6GHz and millimeter wave (mmWave) bands were considered as the candidate spectrum bands for CAVs [5]. Besides, the radiocommunication requirements of CAV including sensor, radio interfaces and reliability were defined and discussed. And the deployed connected vehicle pilot zones in some areas of China basically covered scenarios such as urban roads and rural roads, with intelligent networked infrastructures [6]. In response to the large bandwidth and low latency requirements of CAV sensing data sharing, China expounded the advantages of the sensing and communication integrated design system in the CAV system, which can reduce the delay of information interaction between sensing and communication systems, and improve communication performance by assisting the communication process with sensing information [6].In order to guarantee the driving safety and environment sensing data sharing among CAVs, the integration of communication, sensing and computing is important and considered as a key enabling technology for the 6G end-to-end information processing [7]-[8]. However, there are still many problems unsolved yet for the joint design of communication, sensing and computing functions for CAVs.

In the literature, there are many research works on the fusion of communication and sensing functions. In terms of the sensing assisted communication performance analysis, an intelligent service-oriented edge management architecture was proposed in [9]. By using a large number of sensing data extracted from vehicular networks, the coordination between edge resources and auxiliary information services

Corresponding author is Q. Zhang.

Q. Zhang, H. Wen, Y. Liu, and S. Chang are with the Key Laboratory of Universal Wireless Communications, Ministry of Education, School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: zhangqixun@bupt.edu.cn, wenhao98@bupt.edu.cn, liuying0504@bupt.edu.cn, changshuo@bupt.edu.cn).

Z. Han is with the Department of Electrical and Computer Engineering at the University of Houston, Houston, TX 77004 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul, South Korea, 446-701. (email: hanzhu22@gmail.com).

The work of Q. Zhang, H. Wen, Y. Liu, and S. Chang was supported in part by the National Natural Science Foundation of China (NSFC) (Grants No. 62022020, 61941102), the National Key Research and Development Project (Grant No. 2020YFB1807600). The work of Z. Han was partially supported by NSF CNS-2107216 and CNS-2128368.

was achieved. By using the joint communication and sensing functions deployed on the roadside unit (RSU), a beamforming method using the sensing information was proposed in [10] based on the dual function radar communication (DFRC). Furthermore, a beamforming algorithm based on factor graph and message passing among vehicles was proposed in [11] to realize the DFRC predictive beamforming in the vehicular network. However, the existing works have not considered the computing time delay problem in terms of the environment sensing tasks, which will lead to an unacceptable task execution delay.

In terms of the computing task requirement, the local computing ability of the vehicle is often insufficient to process these real-time and computation-intensive tasks [12]-[15], leading to a serious threat to the safety of CAVs [16]. Therefore, the time-sensitive computing tasks need to be offloaded efficiently to other infrastructures to meet the low latency requirements. The cloud computing is considered as a candidate method to manage the task offloading in order to reduce the execution delay of offloading tasks. Existing research works in [17]-[19] considered the computing task offloading to the cloud server. Both the single-task offloading and the distributed task offloading schemes were considered in [17] to reduce the power consumption of the centralized cloud. The theoretical and empirical analysis of vehicle cloud task scheduling problem was also studied in [18]. In [19], a task offloading problem was formulated to offload tasks to cloud for processing, which can ensure the minimum end-to-end delay of task processing. However, in the delay-sensitive and computation intensive CAVs scenario, the cloud computing technology for sensing task offloading will cause the impractical execution delay. As a promising technology, the multiaccess edge computing (MEC) technology [20]-[21] showed the capability of solving the time-sensitive computing task offloading problem by deploying servers at the network edge.

There are a considerable amount of research works that focus on the task offloading problem by using the MEC technology. In smart grid, the data distribution among various devices by using vehicles as the MEC servers was studied in [22]. Furthermore, the joint cloud and wireless resource allocation algorithm in the MEC enabled cellular network was proposed in [23] based on the evolutionary game theory. A cloud-based MEC offloading framework was proposed in [24] and the utility of MEC service providers can be maximized in the vehicular networks by using a contract-based offloading and computing resource allocation scheme. In addition, the MEC-based task offloading and channel resource allocation scheme was studied in [25] for the 5G ultra-dense network. An overall strategy of task offloading and resource allocation in a multi-cell MEC network was proposed in [26]. However, these studies only considered the channel resources or computing resources separately, without the joint optimization of communication and computing resources. For example, in [22]-[23], all the computing resources of the MEC server were allocated to users, lacking the flexible computing resources allocation. Besides, the task offloading path selection and channel allocation problems were not considered in [24]. The joint optimization problem of computing and communication resources allocation have not been considered in [25]-[26].

Besides, the existing research works usually divide the joint task offloading and resource allocation optimization problems into several subproblems by using the Lyapunov optimization method or linear programming relaxation method. In [27], the joint load balancing and task offloading problem was transformed into a mixed integer nonlinear programming problem, which was decoupled into two subproblems using a low-complexity method. And another joint computing offloading and radio resource allocation algorithm based on Lyapunov optimization method was proposed in [28]. By minimizing the upper bound of the Lyapunov drift plus penalty function, the main problem was divided into several subproblems which were solved accordingly. In [29], the resource management problem was divided into three subproblems and solved by using the linear programming relaxation and the first-order Taylor approximation. However, the method of splitting the optimization problem into several subproblems is complex and inefficient, which can not meet the timeliness task offloading requirement in the CAVs scenario. In addition, in the autonomous driving scenario, due to factors such as vehicle driving conditions, weather conditions, etc., the resource status presents obvious dynamic fluctuations. The dynamic fluctuation of resources will lead to the instability of computing task offloading, which will affect the adaptability of computing tasks to the offloading environment. Therefore, to adapt to the time-varying environment of CAVs, a deep reinforcement learning (DRL) algorithm based on the adaptive exploration approach has been applied in this paper according to [30]-[32]. The author in [33] provided a comprehensive literature review on the application of DRL in communication and networking. The authors in [34] have described the supervised learning, unsupervised learning, deep learning, reinforcement learning and their applications in wireless networks in detail, aiming to outline the motivations and methods of various machine learning algorithms and scenarios for future wireless networks. A deep reinforcement learning method using a deep Q-network to approximate the Q-value was proposed in [35] to obtain the optimal interference alignment (IA) user selection strategy in IA wireless networks. Simulation results showed that the proposed method can significantly improve the network's sum rate and energy efficiency. The author in [36] formulated resource allocation strategies as a joint optimization problem, which is solved by using a novel big data deep reinforcement learning approach. Simulation results with different system parameters were given to demonstrate the effectiveness of the proposed scheme. But the centralized reinforcement learning will lead to a huge communication overhead problem, which is a great burden on the CAVs network with limited channel resources.

On this basis, a distributed learning method namely federated learning [37]-[38] has been used to perform the weights update of the machine learning model. Because it is difficult to achieve stable and real-time interaction between edge devices and edge servers, the authors in [39] proposed an intelligent ultra-dense edge computing framework. To achieve a real-time and low-overhead computation offloading decision and the resource allocation strategies, a novel two-timescale deep reinforcement learning (2Ts-DRL) method was proposed, and federated learning was used to train the 2Ts-DRL model in a distributed manner to protect the privacy of edge devices. The authors in [40] studied the security incentive mechanism of multi-agent federated reinforcement learning in intelligent cyber-physical systems with heterogeneous devices. A multi-agent federated reinforcement learning algorithm that reduced the variance of policy evaluation was proposed. The experimental results showed that the proposed method can effectively reduce the training cost. To solve the complex dynamic control problem in edge caching, a federated deep reinforcement learning-based cooperative edge caching framework was proposed in [41]. The proposed framework federated all local users to jointly train the parameters and fed them back to the BS to speed up overall convergence rate. The simulation results showed that the proposed framework can reduce the average delay and improve the caching hit rate. The authors in [42] provided an extensive overview of existing research works on federated learning, implementation challenges and issues when applying federated learning to Internet of things environment. However, these works have not considered the joint task offloading and radio resource allocation problem with the low latency and the low overhead constraints in the CAVs scenario.

Therefore, a federated reinforcement learning (FRL) algorithm for task offloading and resource allocation in CAVs network is designed in this paper to reduce the task execution delay. First, we design a vehicle-road-base station (BS) cooperative task offloading architecture with different wireless access technologies. Three task offloading modes are analyzed, namely the local offloading, the RSU offloading via the PC5 interface, and the BS offloading via the Uu interface [43]. We theoretically formulate the minimization problem of average execution delay with the specific channel and computing resources constraints. To adapt to the time-varying topology of the CAVs network and solve the problem of large transmission overhead, we propose the FRL algorithm to solve the task offloading and resource allocation problem and minimize the execution delay in the CAVs network. The main contributions of this paper are summarized as follows.

- A vehicle-road-BS cooperative task offloading architecture with different wireless access technologies is proposed, where the transmission methods of Sub-6GHz and 28GHz frequency bands have been considered. The execution delay minimization problem is theoretically formulated with various communication and computing resources constrains in the CAVs network.
- The task offloading problem is modeled as a Markov decision process based on the definition of state and action in the CAVs scenario, and the DRL enabled joint task offloading and computing resource allocation scheme is proposed to minimize the execution delay of CAVs.
- To solve the transmission overhead problem in the centralized reinforcement learning model training process, the FRL algorithm is proposed to train the optimal task offloading and resource allocation selection model by sharing only the key model parameters among RSU and CAVs. And the convergence trend of the proposed FRL algorithm is analyzed.
- To verify the performance of the proposed vehicle-road-BS cooperation architecture and algorithms, we design and develop the hardware testbed by using the USRP platform, the mmWave communication equipments, and the MEC servers.

The rest of this paper is organized as follows. Section II proposes the vehicle-road-BS cooperation architecture and the execution delay minimization problem is formulated with both communication and computing resources constraints. In Section III, the joint task offloading and resource allocation problem is modeled as a Markov decision process, and the DRL algorithm is used to solve this problem. In Section IV, the FRL enabled task offloading and resource allocation algorithm is proposed to solve the transmission overhead problem in the model training process. Numerical results from both simulation and hardware testbed are discussed in Sections V and VI to verify the performance of the proposed algorithm. Finally, we summarized this paper in Section VII.

II. System Model

In this section, the vehicle-road-BS cooperation architecture is proposed, where both the Sub-6GHz and the mmWave spectrum bands are considered. Then, we described the communication model and the computation model in Section II-B, respectively. The execution delay minimization problem is formulated with the communication and computing resources constraints in Section II-C. The key parameters and notations are summarized in Table I.

A. Vehicle-Road-BS Cooperation Model

As shown in Fig. 1, a vehicular network consists of two BSs, N RSUs, and M vehicle user equipments (VUEs). The set of VUEs is defined by $\mathcal{M} = \{1, 2, ..., M\}$. RSUs are deployed on the roadside, which are denoted by $\mathcal{N} = \{1, 2, ..., N\}$. Each RSU or BS is connected to a MEC server. Server 1, Server 2, ..., Server N represent the servers connected to RSUs. Both Sub-6GHz and mmWave spectrum bands are considered to guarantee both the coverage and capacity demands in the CAVs scenario. According to the ITS communication network in the ITU handbook, we consider that RSU can access to the cloud server through the dedicated data network, and can receive instructions from the central cloud through the dedicated data network [43].

Each task is expressed as $\varphi_i = \{c_i, g_i, t_i^{\max}\}$ by three indicators, where c_i is the size of the task, g_i is the resource required for task computing, and $t_i^{\max}$ is the maximum tolerable delay of task φ_i. To formulate the following optimization problem, the time domain is divided into different time slots, and each time slot cannot exceed $t_i^{\max}$. For each task φ_i, three offloading decisions can be used, namely the local processing, the offloading to the RSU, and the offloading to the BS. We also define $L = \{l_i | l_i \in \{l_i^{loc}, l_i^{RSU}, l_i^{BS}\}, l_i^j \in \{0, 1\}, i \in \mathcal{M}, j \in \{loc, RSU, BS\}\}$ as the offloading decision set. $\mathcal{A} = \{loc, RSU, BS\}$ depicts the offloading path set, including the

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3188434

TABLE I
KEY PARAMETERS AND NOTATIONS.

Symbol	**Definition**
$\mathcal{M}$	VUEs set
$\mathcal{N}$	RSUs set
c_i	Size of task φ_i
g_i	Computing resources required to process task φ_i
$t_i^{\max}$	Maximum tolerable delay of task φ_i
P_t	Transmit power of VUE
$\gamma_{i,RSU}$	SINR of VUE i at RSU
$\gamma_{i,BS}$	SNR of VUE i at BS
$\theta_{i,n}^m$	Spectrum allocated by RSU n to VUE i in time slot m
$\eta_{i,n}^m$	Computing resources allocated by RSU n to VUE i in time slot m
$\theta_{i,BS}^m$	Spectrum allocated by the BS to VUE i in time slot m
$\eta_{i,BS}^m$	Computing resources allocated by the BS to VUE i in time slot m
f_{loc}, f_R, f_B	Computing ability of VUE, RSU and BS
B	Bandwidth between VUEs and RSU
B_0	Bandwidth between VUEs and BS
τ	Transmission delay required per unit result size between BS and VUE
$p_{\varphi_i,loc}$, $p_{\varphi_i,RSU}$, $p_{\varphi_i,BS}$	Offloading decisions of task φ_i for local, RSU, and BS processing

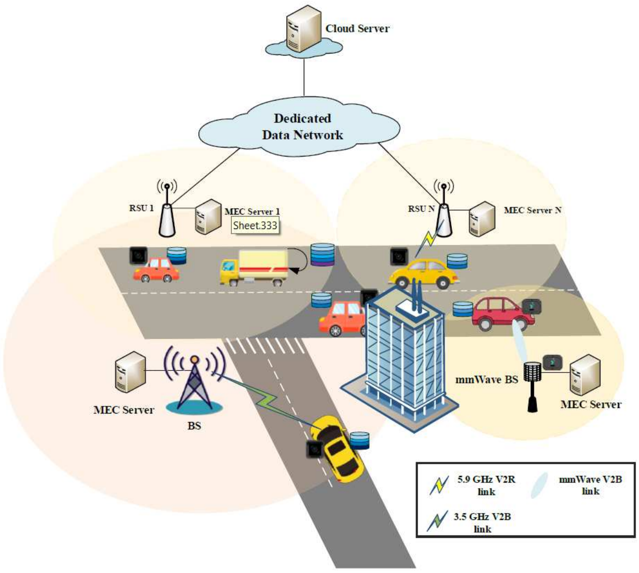


Fig. 1. Vehicle-road-BS cooperation architecture.

local processing, the offloading to the RSU, and the offloading to the BS, respectively. In addition, $l_i^j = 1$ means that the task φ_i is offloaded according to the decision j, otherwise $l_i^j = 0$.

Next, we consider the signal to interference plus noise ratio (SINR) of CAVs in three offloading modes. We assume that all VUEs have the same transmit power P_t, and the number of VUEs within the RSU coverage of VUE i is N_i. In the mode of offloading to RSU, the SINR of VUE i at RSU is defined by

$$\gamma_{i,RSU} = \frac{P_t g_{i,RSU}}{\sum\limits_{l=1,l\neq i}^{N_i} P_t g_{l,RSU} + \sigma^2}, \tag{1}$$

$$\begin{aligned} g_{i,RSU} =& 20\log_{10}(\frac{40\pi d_{i,RSU} f_{comm}}{3}) \\ &+ \min(0.03h_{RSU}^{1.72}, 10)\log_{10}(d_{i,RSU}) \\ &- \min(0.044h_{RSU}^{1.72}, 14.77) \\ &+ 0.002\log_{10}(h_{RSU})d_{i,RSU} \end{aligned} \tag{2}$$

where $g_{i,RSU}$ is the received signal gain at the RSU according to [44], $d_{i,RSU}$ is the distance from the VUE i to the RSU, h_{RSU} is the height of the RSU, and f_{comm} is the communication frequency between the VUE and the RSU, σ^2 is the white Gaussian noise power, and the interference within the coverage of RSU is considered based on [43].

In the mode of offloading to the BS, the spectrum allocated to each VUE is orthogonal [43]. The signal to noise ratio (SNR) of VUE i at the BS is defined by

$$\gamma_{i,BS} = \frac{P_t g_{i,BS}}{\sigma^2}, \tag{3}$$

where $g_{i,BS}$ is the received signal gain at the BS.

B. Communication Model and Computation Model

Both the communication transmission delay and calculation delay are defined according to three offloading modes, which are the local processing, the offloading to the RSU, and the offloading to the BS. The perception information collected by vehicle sensors can be offloaded to the BS through the 3.5GHz and 28GHz frequency bands for processing, or can be offloaded to the RSU through the 5.9GHz V2R link for processing.

1) Mode 1. Local Processing: When the VUE processes the task φ_i locally, the execution delay mainly depends on the calculation delay. Therefore, the execution delay $t_{i,loc}$ in the local processing mode is defined by

$$t_{i,loc} = \frac{g_i}{f_{loc}}, \tag{4}$$

where f_{loc} is the computation capability of VUEs.

2) Mode 2. Offloading to RSU: When the VUE decides to offload the task to the RSU, both the communication transmission delay and the calculation delay need to be considered. According to the SINR $\gamma_{i,RSU}$ of VUE i at the RSU, the communication transmission delay $t_{i,RSU}^{comm}$ can be obtained by

$$t_{i,RSU}^{comm} = \frac{c_i}{\theta_{i,n}^{m} B \log_2(1+\gamma_{i,RSU})}, \tag{5}$$

where B is the bandwidth of the communication link for VUEs and RSU, and $\theta_{i,n}^{m}$ is the percentage of the spectrum allocated to VUE i by RSU n in time slot m .

The calculation delay $t_{i,RSU}^{comp}$ can be expressed as

$$t_{i,RSU}^{comp} = \frac{g_i}{\eta_{i,n}^{m} f_R}, \tag{6}$$

where f_R represents the computing ability of the RSU, and $\eta_{i,n}^{m}$ is the percentage of computing resources allocated to VUE i by RSU n in time slot m.

In mode 2, the size of environment sensing task's computation results is much small than the size of sensing task's input from the MEC server. Thus, the time of receiving the computation results can be ignored based on [45]. Therefore, the execution delay $t_{i,RSU}$ is

$$t_{i,RSU} = t_{i,RSU}^{comm} + t_{i,RSU}^{comp}. \tag{7}$$

3) Mode 3. Offloading to BS: When the VUE decides to offload tasks to the BS, we need to consider both the communication transmission delay and the calculation delay. According to SNR $\gamma_{i,BS}$ of the VUE i at the BS, we can obtain the transmission delay $t_{i,BS}^{comm}$ by

$$t_{i,BS}^{comm} = \frac{c_i}{\theta_{i,BS}^{m} B_0 \log_2(1+\gamma_{i,BS})}, \tag{8}$$

where B_0 is the bandwidth of the communication link for VUEs and the BS, and $\theta_{i,BS}^{m}$ is the percentage of spectrum allocated by the BS to VUE i in time slot m.

The calculation delay $t_{i,BS}^{comp}$ can be expressed as

$$t_{i,BS}^{comp} = \frac{g_i}{\eta_{i,BS}^{m} f_B}, \tag{9}$$

where f_B represents the computing ability of the BS, and $\eta_{i,BS}^{m}$ is the percentage of computing resources allocated by the BS to VUE i in time slot m.

In mode 3, due to the greater distance between the VUE and BS, the time of receiving the environment sensing task's computation results needs to be considered. Therefore, the execution delay $t_{i,BS}$ is

$$t_{i,BS} = t_{i,BS}^{comm} + t_{i,BS}^{comp} + c_o \tau, \tag{10}$$

where c_o is the size of computation results and τ is the transmission delay required per unit of computation result between the BS and VUE.

C. Problem Formulation

We use $p_{\varphi_i,loc} = 1$ to denote that task φ_i is handled locally. Similarly, we use $p_{\varphi_i,RSU} = 1$ and $p_{\varphi_i,BS} = 1$ to denote that task φ_i is processed by offloading to the RSU and BS, respectively. Otherwise, they are all set to 0. Different task offloading methods correspond to different task execution delays. When the VUE offloads the task to the RSU through the 5.9GHz link, the transmission delay needs to be considered. When the VUE processing the task on their own, only the local processing delay needs to be considered. The task execution delay under different offloading modes is the optimization problem proposed in this paper. We define the time t_i^m required to process task φ_i in the mth time slot as

$$t_i^m = p_{\varphi_i,loc} t_{i.loc}^{m} + p_{\varphi_i,RSU} t_{i,RSU}^{m} + p_{\varphi_i,BS} t_{i,BS}^{m}. \tag{11}$$

To minimize the average execution delay with both the constraints in computing and communication resources, the optimization problem is defined by

$$\mathbb{P}_1 : \min_{\substack{p_{\varphi_i,loc}, p_{\varphi_i,RSU}, p_{\varphi_i,BS}, \\ \theta_{i,n}^{m}, \theta_{i,BS}^{m}, \eta_{i,n}^{m}, \eta_{i,BS}^{m}}} T_i = \sum_{m=1}^{\infty} t_i^m \tag{12a}$$

$$\text{s.t.}\mathbb{C}_1 : p_{\varphi_i,loc} = \{0,1\}, p_{\varphi_i,RSU} = \{0,1\}, p_{\varphi_i,BS} = \{0,1\}, \tag{12b}$$

$$\mathbb{C}_2 : p_{\varphi_i,loc} + p_{\varphi_i,RSU} + p_{\varphi_i,BS} = 1, \tag{12c}$$

$$\mathbb{C}_3 : \sum_{i\in M} p_{\varphi_i,RSU} \theta_{i,n}^{m} \leq 1, \sum_{i\in M} p_{\varphi_i,BS} \theta_{i,BS}^{m} \leq 1, \tag{12d}$$

$$\mathbb{C}_4 : \sum_{i\in M} p_{\varphi_i,RSU} \eta_{i,n}^{m} \leq 1, \sum_{i\in M} p_{\varphi_i,BS} \eta_{i,BS}^{m} \leq 1, \tag{12e}$$

$$\mathbb{C}_5 : t_{i,j}^{m} \leq t_i^{\max}, i \in M, j = [p_{\varphi_i,loc}, p_{\varphi_i,RSU}, p_{\varphi_i,BS}], \tag{12f}$$

where T_i represents the task execution cumulative delay of VUE i. Constraints $\mathbb{C}_1$ and $\mathbb{C}_2$ in (12b) and (12c) indicate that the VUE can only choose one of the three offloading modes. Constraint $\mathbb{C}_3$ in (12d) indicates that the sum of channel resources allocated to VUEs by the RSU and BS shall not exceed the maximum value that RSU and BS can provide. Similarly, constraint $\mathbb{C}_4$ in (12e) indicates that the sum of the computing resources allocated to VUEs by the RSU and BS must not exceed the maximum value. Constraint $\mathbb{C}_5$ in (12f) denotes that the task delay cannot exceed the maximum delay that can be tolerated. Traditional convex optimization method of splitting the mixed-integer nonlinear programming problem $\mathbb{P}_1$ into several subproblems is complex and inefficient, which can not meet the timeliness task offloading requirements in the CAVs scenario.

III. DRL Based Task Offloading and Resource Allocation

To solve the mixed-integer nonlinear programming problem $\mathbb{P}_1$ in the time-varying CAVs scenario, a deep reinforcement learning (DRL) based task offloading and resource allocation algorithm is proposed in this section, where the state, action, and reward function are defined.

A. Definition of State, Action and Reward Function

1) Space State: The state $s_i(m)$ provided by RSU $n \in \{1, 2, ..., N\}$ and the BS to VUE i in the time slot m is defined as

$$\begin{aligned} s_i(m) =&[R^1_{i,BS}(m), R^2_{i,BS}(m), R^1_{i,RSU}(m), ..., R^N_{\mathrm{i},RSU}(m), \\ & f^1_{i,BS}(m), f^2_{i,BS}(m), f^1_{i,RSU}(m), ..., f^N_{i,RSU}(m)]. \end{aligned} \tag{13}$$

In the time slot m, we define

$$R^l_{i,BS}(m) = \theta^{m-1}_{i,BS_l} B_0 \log_2(1 + \gamma_{i,BS}), l \in \{1, 2\}, \tag{14a}$$

$$R^n_{i,RSU}(m) = \theta^{m-1}_{i,n} B \log_2(1 + \gamma_{i,RSU}), n \in \{1, 2, ..., N\}, \tag{14b}$$

$$f^l_{i,BS}(m) = \eta^{m-1}_{i,BS_l} f_B, l \in \{1, 2\}, \tag{14c}$$

$$f^n_{i,RSU}(m) = \eta^{m-1}_{i,n} f_R, n \in \{1, 2, ..., N\}, \tag{14d}$$

where $R^l_{i,BS}(m)$ represents the transmission rate from VUE i to the BS l, $R^n_{i,RSU}(m)$ denotes the transmission rate from VUE i to RSU n, $f^l_{i,BS}(m)$ is the computing resources allocated by the BS l to VUE i, and $f^n_{i,RSU}(m)$ depicts the computing resources allocated by RSU n to VUE i.

2) Action Space: In the proposed vehicle-road-BS cooperation task offloading architecture, by deploying a global model of task offloading and resource allocation based on DRL algorithm on the RSU, the optimal task offloading and resource allocation decision can be achieved by using a global task offloading model. Therefore, the action space $a_i(m)$ in the time slot m can be defined by

$$\begin{aligned} a_i(m) =&\{p_{\varphi_i,loc}(m), p_{\varphi_i,RSU}(m), p_{\varphi_i,BS}(m), \\ & \theta^m_{i,BS_1}, \theta^m_{i,BS_2}, \theta^m_{i,1}, ..., \theta^m_{i,N}, \\ & \eta^m_{i,BS_1}, \eta^m_{i,BS_2}, \eta^m_{i,1}, ..., \eta^m_{i,N}\}. \end{aligned} \tag{15}$$

3) Reward Function: To minimize the average execution delay of the entire offloading system while satisfying the offloading decision and resource constraints, we define the reward function $r_i(m)$ of VUE i in the time slot m as the reciprocal of the proposed problem $\mathbb{P}_1$

$$r_i(m) = \frac{1}{t^m_i}. \tag{16}$$

where $r_i(m)$ represents the reward obtained when the VUE i takes an action $a_i(m)$ in the state $s_i(m)$. To maximize the long-term utility of the VUE, we define the cumulative reward function of VUE i by $R_i(m)$

$$R_i(m) = \mathbb{E}\left[\sum_{u=1}^{T} r_i(m+u)\right]. \tag{17}$$

B. Markov Decision Process

The proposed task offloading and resource allocation problem can be modeled as the Markov decision process. In the Markov decision process, the state at a certain moment is s, the behavior taken by the VUE in state s is a, and the reward for taking this behavior is r^a_s. We also define the state of the next stage as s', and the state transition probability from s to s' is $P_{ss'}$. The state value function $v_\pi(s)$ of taking the strategy π in state s is depicted by

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) q_\pi(s, a), \tag{18}$$

where $\pi(a|s)$ represents the probability of taking action a in state s, and $\mathcal{A}$ is the action set.

The action value function $q_\pi(s, a)$ can be expressed as

$$q_\pi(s, a) = \sum_{s' \in S} P_{ss'}[r^a_s + \gamma v_\pi(s')], \tag{19}$$

where γ is the attenuation factor.

And the Bellman equation of the action value function in (19) is

$$q_\pi(s, a) = \sum_{s' \in \mathcal{S}} P_{ss'}[r^a_s + \gamma \sum_{a' \in \mathcal{A}} \pi(a'|s') q_\pi(s', a')]. \tag{20}$$

Given the current strategy π, the value function v_π, and the action value function q_π, the new strategy is constructed as

$$\pi_*(s) = \arg \max_a q_\pi(s, a). \tag{21}$$

To prove the convergence of (18), we substitute the action value function $q_\pi(s, a)$ in (19) into the state value function $v_\pi(s)$ in (18) to obtain the Bellman operator which represents the change of the state value function. Therefore, we define the Bellman operator $\mathcal{B}_\pi$ for policy π as

$$\mathcal{B}_\pi v(s) := \sum_{a \in \mathcal{A}} \pi(a|s) \sum_{s' \in \mathcal{S}} P_{ss'}[r^a_s + \gamma v(s')]. \tag{22}$$

The Bellman operator in the above formula is an operation on $v(s)$. The Bellman operator $\mathcal{B}_\pi$ is a contracting map as proved below. According to the definition of Bellman operator, we have

$$\begin{aligned} & |\mathcal{B}_\pi v_1(s) - \mathcal{B}_\pi v_2(s)| \\ & = \left| \sum_{a \in \mathcal{A}} \pi(a|s) \sum_{s' \in \mathcal{S}} P_{ss'}[\gamma(v_1(s') - v_2(s'))] \right| \\ & \leq \gamma \sum_{a \in \mathcal{A}} \pi(a|s) \sum_{s' \in \mathcal{S}} P_{ss'} |(v_1(s') - v_2(s'))| \\ & \leq \gamma \sum_{a \in \mathcal{A}} \pi(a|s) \sum_{s' \in \mathcal{S}} P_{ss'} \left(\max_{s'' \in \mathcal{S}} |v_1(s'') - v_2(s'')| \right) \\ & = \gamma \max_{s'' \in \mathcal{S}} |v_1(s'') - v_2(s'')| \\ & = \gamma ||v_1 - v_2||_\infty. \end{aligned} \tag{23}$$

The above formula holds for any s, so (23) can be written as

$$||\mathcal{B}_\pi v_1 - \mathcal{B}_\pi v_2||_\infty \leq \gamma ||v_1 - v_2||_\infty. \tag{24}$$

The Bellman operator is a strictly contracted map when $\gamma < 1$, and the sequence $\{v, \mathcal{B}_\pi v, \mathcal{B}^2_\pi v, ...\}$ is shown to be convergent as follows

$$\begin{aligned} ||\mathcal{B}_\pi^{m+1}v - \mathcal{B}_\pi^{m}v||_\infty &\leq \gamma||\mathcal{B}_\pi^{m}v - \mathcal{B}_\pi^{m-1}v||_\infty \\ &\leq \gamma^2||\mathcal{B}_\pi^{m-1}v - \mathcal{B}_\pi^{m-2}v||_\infty \\ &\dots \\ &\leq \gamma^m||\mathcal{B}_\pi v - v||_\infty. \end{aligned} \tag{25}$$

According to (25), when m approaches infinity, the difference between v_{m+1} and v_m will approach 0. The sequence $\{v, \mathcal{B}_\pi v, \mathcal{B}_\pi^2 v, ...\}$ will converge to a fixed point. Next, we will prove the uniqueness of this fixed point.

Suppose $\mathcal{B}_\pi$ has two fixed points U and V, and $U \neq V$. Then there must be $||U - V||_\infty > 0$. Since both U and V are fixed points, so $||\mathcal{B}_\pi U - \mathcal{B}_\pi V||_\infty = ||U - V||_\infty$. However, since $\mathcal{B}_\pi$ is a shrinking map, there is $||\mathcal{B}_\pi U - \mathcal{B}_\pi V||_\infty \leq \gamma||U - V||_\infty < ||U - V||_\infty$. The two contradict each other. Therefore, the fixed point must be unique, and the value function obtained according to the value iteration must be optimal, so the optimal strategy can be obtained.

C. Deep Reinforcement Learning Algorithm

When the dimension of state and action is relatively low, the Q learning algorithm can solve the proposed $\mathbb{P}_1$ problem. However, in terms of a high-dimensional state and action space, the effectiveness of Q learning algorithm will decrease. Therefore, the DRL algorithm is used to increase the efficiency of the Q value estimation by the deep neural network (DNN) instead of using each state action pair to estimate the Q value.

Specifically, there are two neural networks in the DRL algorithm. One is the main network to estimate the Q value, which is defined by $Q(s, a; \omega)$, where ω represents the parameters in the main network. The other is the target network, which is used to generate the target value of Q by

$$Q_{tar} = r_i + \gamma \max_a Q(s', a'; \omega), \tag{26}$$

where s' and a' denote the state and reward of the next stage, respectively. The square error function of the difference between these two values is defined by L_{loss} as

$$L_{loss} = E[\frac{1}{2}(Q(s, a; \omega) - Q_{tar})^2]. \tag{27}$$

And the gradient descent algorithm is used to update ω

$$\omega \leftarrow \omega - \varsigma \frac{\partial L_{loss}}{\partial \omega}, \tag{28}$$

where $\frac{\partial L_{loss}}{\partial \omega} = E[\frac{\partial Q(s,a;\omega)}{\partial \omega}(Q(s, a; \omega) - Q_{tar})]$, and ς represents the learning rate. In the proposed DRL algorithm, each RSU is deployed with a central server. We use an experience replay memory D to store the data tuple (s^m, a^m, r^m, s^{m+1}). $s^m = [R_{i,BS}^m, R_{i,RSU}^m, f_{i,BS}^m, f_{i,RSU}^m]$ represents the channel resource and computing resource allocation status of the current environment. $a^m = [p, \eta_{i,RSU}^m, \theta_{i,RSU}^m, \eta_{i,BS}^m, \theta_{i,BS}^m]$ is the task offloading and resource allocation decision according to the current environment status. And the reward obtained by taking the action a^m in the current state s^m is r^m, which is the reciprocal of the proposed optimization problem. s^{m+1} denotes the state of the environment at the next moment. We use rectified linear unit (ReLU) function as the activation function between layers in the proposed DRL algorithm. The DRL agent randomly selects data from the experience replay memory each time to train the parameters of the DNN. In addition, to avoid the local optima while ensuring a balance between exploration and utilization, we adopt a greedy policy. The agent randomly selects an action a from the action space with a probability of ξ each time, otherwise it selects an action with the largest Q value with a probability of $1-\xi$. The update process of the proposed DRL algorithm is shown in Fig. 2 . The task offloading and resource allocation based DRL algorithm is also proposed in **Algorithm 1**.

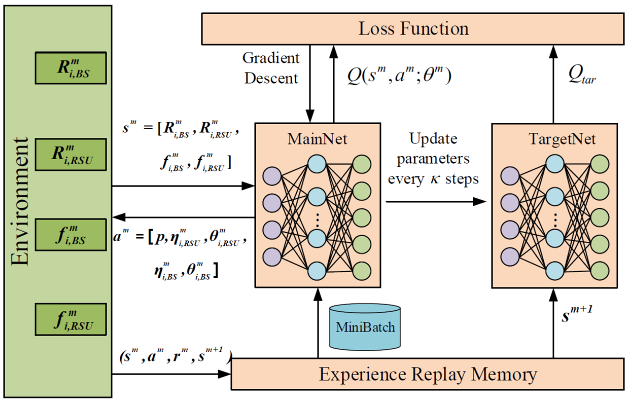

Environment
$R_{i,BS}^m$
$R_{i,RSU}^m$
$f_{i,BS}^m$
$f_{i,RSU}^m$
Loss Function
Gradient Descent
$Q(s^m, a^m; \theta^m)$
Q_{tar}
$s^m = [R_{i,BS}^m, R_{i,RSU}^m, f_{i,BS}^m, f_{i,RSU}^m]$
$a^m = [p, \eta_{i,RSU}^m, \theta_{i,RSU}^m, \eta_{i,BS}^m, \theta_{i,BS}^m]$
MainNet
Update parameters every κ steps
TargetNet
MiniBatch
s^{m+1}
(s^m, a^m, r^m, s^{m+1})
Experience Replay Memory

Fig. 2. The update process of the proposed DRL algorithm.

In **Algorithm 1**, we need to initialize the DNN parameters and the experience replay memory. In the DNN parameters training stage, we employ DRL to calculate the Q value. First, we obtain the initial state s^0 of the system by observing the stable traffic topology. To achieve a balance between exploration and utilization, we use a greedy policy. For each time slot, we will randomly choose an action with a probability of ξ, otherwise choose an action with the largest Q value. Then, the obtained data tuples are stored in the experience replay memory. For each time, C samples are taken from the experience replay memory to train the DNN network. And the square error function between the target network and the main network is constructed, where the network parameters are updated using a gradient descent method. The parameters of the main network are updated in each step, and the parameters of the target network are updated by every κ steps. Since all data comes from the previous environment, and the VUEs does not update the global model online based on local data, it is called an offline DRL training process. And then the RSU sends the offloading decision information and resource allocation results to the VUE and the corresponding infrastructure, respectively. The VUE can perform the task offloading processing of the image or video stream according to the corresponding offloading decision.

IV. FRL Based Task Offloading and Resource Allocation

To further improve the efficiency of the centralized DRL algorithm, the FRL based task offloading and resource allo-

Algorithm 1 DRL Based Task Offloading and Resource Allocation Algorithm.

```
**Initialization**: Initialize the parameters of DNN with $\omega$ and initialize the experience replay memory.
**Input**: Channel resource and computing resource information for each time slot $m$, including $R_{i,BS}^{m}$, $R_{i,RSU}^{m}$, $f_{i,BS}^{m}$ and $f_{i,RSU}^{m}$.
**Output**: Task offloading decision and resource allocation results, including $p$, $\eta_{i,RSU}^{m}$, $\theta_{i,RSU}^{m}$, $\eta_{i,BS}^{m}$ and $\theta_{i,BS}^{m}$.
**for** each stable traffic topology **do**
    Observe and get the status $s^0$ of the system.
    **for** each time slot $m = 1, 2, \ldots, t_{\max}$ **do**
        Choose action $a^m = [p, \eta_{i,RSU}^{m}, \theta_{i,RSU}^{m}, \eta_{i,BS}^{m}, \theta_{i,BS}^{m}]$ according to $s^m = [R_{i,BS}^{m}, R_{i,RSU}^{m}, f_{i,BS}^{m}, f_{i,RSU}^{m}]$ with probability $\xi$.
        Otherwise choose action
        $a^m = \arg\max_{\mathcal{A}} Q(s^m, a^m; \omega^m)$.
        Store the obtained data in the experience replay memory in the form of tuple $(s^m, a^m, r^m, s^{m+1})$.
        Calculate the value $Q_{tar}$ of the target network: $Q_{tar} = r_i + \gamma \max_{a^{m+1}} Q(s^m, a^m; \omega)$.
        Construct the error function according to (27).
        Use data tuple$(s^m, a^m, r^m, s^{m+1})$ in the experience replay memory to train the $Q$ network.
        Update the parameters of the main network according to (28) based on the gradient descent algorithm.
    **end for**
    Update the parameters of the target network every $\kappa$ steps.
    Obtain task offloading decision and resource allocation results.
**end for**
```

cation algorithm is designed by using the federated reinforcement learning between multiple VUEs and RSU. First, we design the framework of the proposed FRL algorithm. Then, the convergence of the proposed FRL algorithm is analyzed.

A. Framework Design

The centralized DRL algorithm requires users to upload all the data to the RSU for training, which causes the overhead, and privacy problems. First, offloading a large amount of data to the RSU for training will take a lot of time and it will lead to an excessive overhead, which will pose a challenge to the scarce spectrum resources of the existing communication system. In addition, offloading a large amount of data will threaten the privacy of users. However, the training of DRL agents in a distributed manner consumes a lot of time and battery energy resources. Therefore, we propose to train the DRL model via FRL algorithm among multiple vehicle users. By aggregating the updated parameters of multiple vehicle users at the RSU, the data upload overhead can be reduced and the training efficiency can be improved. The framework of the proposed FRL based task offloading and resource allocation algorithm is shown in Fig. 3 .

The initial parameters are obtained by training the DRL model on the RSU, which will deliver these parameters to the VUEs for model training. The VUEs update the model parameters based on the local data, and then the updated model parameters of multiple VUEs are aggregated at the RSU to obtain the global parameters. The RSU continues to deliver the global parameters to the VUEs, thus starting a new round of an iterative process. The iterative process will be repeated continuously until convergence.

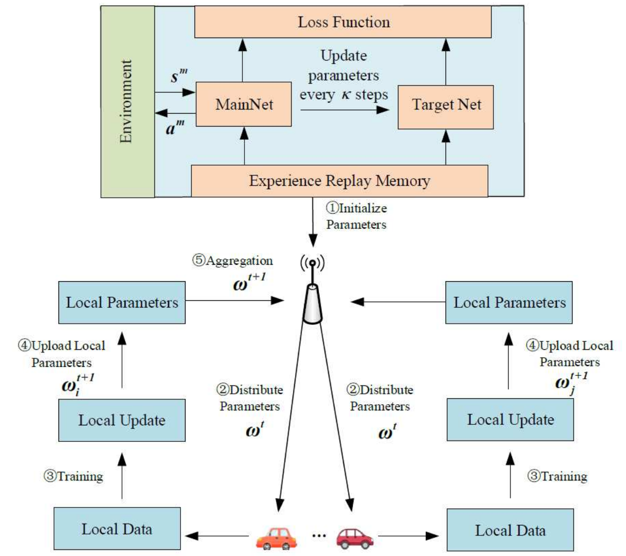


Fig. 3. The framework of the proposed FRL based task offloading and resource allocation algorithm.

After receiving the issued global parameters, VUEs will update the model parameters locally. VUE j will construct the square error function $f_j^{(i)}(\omega)$ between the theoretical output $\omega^T s^{(i)}$ and the real output $a^{(i)}$ based on local data

$$f_j^{(i)}(\omega) = \frac{1}{2}\left(w^T s^{(i)} - a^{(i)}\right)^2, \tag{29}$$

where i represents the ith sample data.

Therefore, the error function $f_j(\omega)$ of VUE j can be expressed as

$$f_j(\omega) = \frac{1}{N_j}\sum_{i=1}^{N_j} f_j^{(i)}(\omega), \tag{30}$$

where N_j is the total number of samples of VUE j.

The local optimization problem can be expressed as

$$\omega_j^t = \arg\min_{\omega_j} f_j(\omega). \tag{31}$$

The optimization goal can be expressed as

$$\omega^t = \arg\min_{\omega} f(\omega), \tag{32}$$

where $f(\omega) = \sum_{j=1}^{M} f_j(\omega)$.

However, it is difficult to solve (32) due to the high complexity. Therefore, we design a distributed method to solve

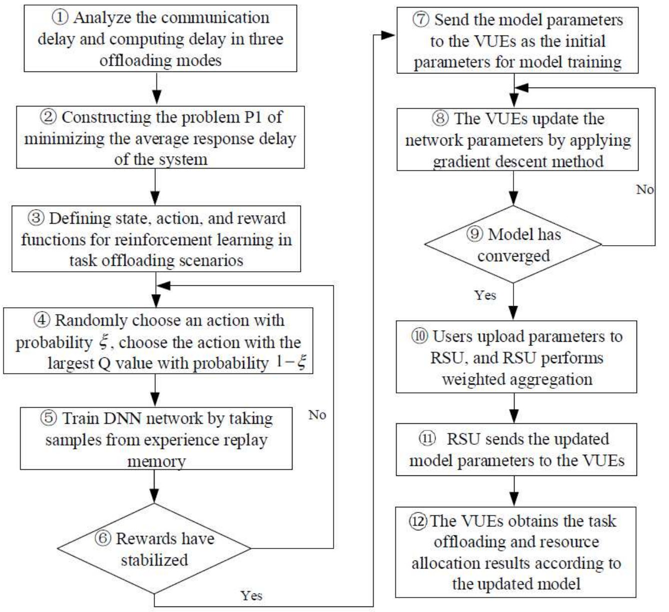

① Analyze the communication delay and computing delay in three offloading modes
② Constructing the problem P1 of minimizing the average response delay of the system
③ Defining state, action, and reward functions for reinforcement learning in task offloading scenarios
④ Randomly choose an action with probability ξ, choose the action with the largest Q value with probability 1−ξ
⑤ Train DNN network by taking samples from experience replay memory
No
⑥ Rewards have stabilized
Yes
⑦ Send the model parameters to the VUEs as the initial parameters for model training
⑧ The VUEs update the network parameters by applying gradient descent method
No
⑨ Model has converged
Yes
⑩ Users upload parameters to RSU, and RSU performs weighted aggregation
⑪ RSU sends the updated model parameters to the VUEs
⑫ The VUEs obtains the task offloading and resource allocation results according to the updated model

Fig. 4. The flowchart of the combination of **Algorithm 1** and **Algorithm 2.**

this problem. In each time slot, we use the gradient descent method to update the network parameters ${\omega_j}^{t+1}$

$$ {\omega_j}^{t+1} \leftarrow {\omega_j}^{t} - \varsigma \nabla f_j(\omega^t), \tag{33} $$

where ς is the gradient descent rate.

After completing multiple partial updates on each VUE side, the parameter ${\omega_j}^{t+1}$ is obtained. Each VUE uploads the updated parameters to the RSU and the RSU obtains the new global parameters ω^{t+1} by aggregating these updated parameters globally

$$ \omega^{t+1} = \frac{\sum\limits_{j=1}^{M} N_j \omega_j^{t+1}}{N}, \tag{34} $$

where $N = \sum\limits_{j=1}^{M} N_j$.

Then, the RSU will send the global parameters ω^{t+1} to the VUEs for a new round of global update. The FRL based task offloading and resource allocation algorithm is shown in **Algorithm 2**.

In each time slot, there is a global iterative process and a task offloading process. During the task offloading process, the VUE can perform task offloading processing according to the global aggregated result. And the task can be computed locally or offloaded to RSU and BS for processing. The two processes have a time sequence relationship, and there is no conflict between the RSU handling the offloading task and the federated learning process. Fig. 4 shows the flowchart of the combination of **Algorithm 1** and **Algorithm 2**. Steps 3 to 6 are the operation flow of **Algorithm 1**, and steps 7 to 12 are the operation flow of **Algorithm 2**. The training result of **Algorithm 1** is used as the initial training model of **Algorithm 2.**

Algorithm 2 FRL Based Task Offloading and Resource Allocation Algorithm.

```
Input:
  The initial parameters $\omega^0$ of the model is obtained according to Algorithm 1.
  The number of iterations $T$.
  Gradient descent rate $\varsigma$.
Output:
  Optimal parameters $\omega^*$ of task offloading and resource allocation models.
  Task offloading decision and resource allocation results, including $p$, $\eta_{i,RSU}^t$, $\theta_{i,RSU}^t$, $\eta_{i,BS}^t$ and $\theta_{i,BS}^t$.
for each iteration $t = 1, 2, \ldots, T$ do
    for each VUE $i$ do
        Receive the initial model parameters from the RSU.
        Update its model parameters ${\omega_j}^{t+1}$ based on (31) and (33).
        Upload the updated parameters to the RSU for weighted aggregation.
    end for
    for each RSU $n$ do
        The local updates of each VUE are weighted and aggregated to obtain the global update $\omega^{t+1}$ according to (34).
        Deliver the updated global parameters to VUEs.
    end for
    Obtain task offloading decision and resource allocation results based on the FRL model.
end for
```

B. Convergence Analysis

We define ω^* as the optimal solution with the following assumptions.

Assumption 1: For all i,

- $f_i(\omega)$ is convex,
- $f_i(\omega)$ is β-smooth, that is, $f_i(\omega') \leq f_i(\omega) + \nabla f_i(\omega) \cdot (\omega' - \omega) + \frac{\beta}{2}||\omega' - \omega||^2$, for $\forall$ ω' and ω.

The feasibility of linear regression and FRL update rules are guaranteed. Therefore, we have the following **Lemma 1**.

Lemma 1. *$f(\omega)$ is convex and β-smooth.*

*Proof: Please see **Appendix A**.* ■

Theorem 1. *Considering that $f(\omega)$ is β-smooth and a convex function, $\omega^* = \arg\min_\omega f(\omega)$, so we have*

$$ ||\omega^{t+1} - \omega^*||^2 \leq ||\omega^t - \omega^*||^2 - \varsigma(\frac{1}{\beta} - \varsigma)||\nabla f(\omega^t)||^2. \tag{35} $$

If the learning rate $\varsigma < \frac{1}{\beta}$, the result $||\omega^{t+1} - \omega^*||^2 \leq ||\omega^t - \omega^*||^2$ is obtained.

*Proof: Please see **Appendix B**.* ■

Theorem 2. *The cost sequence $f(\omega^t)$ is convergent, and the convergence rate is $O(\frac{1}{t})$.*

*Proof: Please see **Appendix C**.* ■

C. Computational Complexity Analysis

In this paper, we design a task offloading and resource allocation algorithm based on deep reinforcement learning and a federated reinforcement learning algorithm for vehicle-road coordination. Next, we analyze the computational complexity of these two algorithms. According to [46], the computational complexity of each step in the deep neural network training process can be expressed as $\mathcal{O}(S_{in}N_l + \sum_{l=1}^{H-1} N_l N_{l+1})$, where S_{in}, N_l and H represent the size of the input layer, the number of neurons in the l-th layer and the number of training layers, respectively. Therefore, for a neural network that requires $t_{\max}$ steps to converge, the computational complexity of the training process is defined as $\mathcal{O}(t_{\max}(S_{in}N_l + \sum_{l=1}^{H-1} N_l N_{l+1}))$. In addition, with a replay memory buffer size E, the computational complexity can be expressed as $\mathcal{O}(2|\mathcal{S}|^2 \times |\mathcal{A}| + \log_2 E)$ [47], where $\mathcal{S}$ and $\mathcal{A}$ represent the state space and action space, respectively. Therefore, the total computational complexity of **Algorithm 1** is $\mathcal{O}(t_{\max}(S_{in}N_l + \sum_{l=1}^{H-1} N_l N_{l+1}) + 2|\mathcal{S}|^2 \times |\mathcal{A}| + \log_2 E)$. For each iteration round in **Algorithm 2**, the computational complexity on the user side is $\mathcal{O}(|D_i|(S_{in}N_l + \sum_{l=1}^{H-1} N_l N_{l+1}) + 2|D_i|^2 + \log_2 E)$, where D_i represents the local data of user i. Therefore, the total computational complexity of **Algorithm 2** is $\mathcal{O}(T(M|\omega_i|(|D_i|(S_{in}N_l + \sum_{l=1}^{H-1} N_l N_{l+1}) + 2|D_i|^2 + \log_2 E) + |\omega|))$, where M, ω_i and ω represent the number of VUEs, local parameters of user i and global parameters, respectively.

The authors in [48] proposed a method for network compression that can reduce the CPU and storage requirements of neural networks by a factor of 10. The simulation results show that the deep learning algorithm can be efficiently deployed on the standard platforms of Intel and Qualcomm, which also provides the potential feasibility for the deployment of our proposed algorithms.

V. Simulation Results

In this section, the performance of the proposed FRL based task offloading and resource allocation algorithm is analyzed by comparing with the existing task offloading algorithms. Simulation results of the task execution delay, the transmission overhead, the convergence trend, and the reward are analyzed in detail.

A. Simulation Setup

We simulate and analyze the proposed algorithm based on Python [49]. We import the MXNET module in Python for DRL algorithm. In the simulation, we set the bandwidth of RSU and BS to 20MHz and 40MHz [50], respectively. The transmit power of user is 200mW [50]. And the transmit power of RSUs and medium range BS is 200mW and 1W [51]. The distance between BSs is 500m [44] and the distance between RSUs is 100m [52]. We consider the pathloss model as the model in 3GPP TR 38.901 RMa scenario [44]. In the proposed DRL algorithm, the attenuation factor is set to 0.99. In addition, we consider the reward under different discrete degrees of action. The CPU rates of the BS and RSU are set to 2×10^{10}cycle/s and 10^{10}cycle/s [12]. And the period of replacing the target network is set to 250 according to [41]. Table II denotes the key simulation parameters in detail.

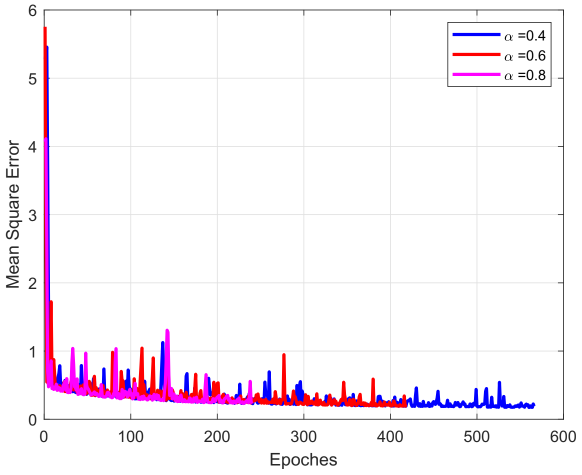


Fig. 5. Convergence of the proposed FRL algorithm.

TABLE II
Key simulation parameters.

Parameter	Value
Distance between BSs	500 m
Distance between RSUs	100 m
Bandwidth of RSU	20 MHz
Bandwidth of BS	40 MHz
Transmit power of medium range BS	1 W
Transmit power of RSU	200 mW
Transmit power of UE	200 mW
Pathloss Model	RMa scenario pathloss model
Attenuation factor of DRL algorithm	0.99
Thermal noise power spectral density	-174 dBm/Hz
CPU rate of BS	2×10^{10}cycle/s
CPU rate of RSU	10^{10}cycle/s
Maximum tolerant transmission delay	100 ms
The period of replacing the target network	250

B. Model Training

Fig. 5 shows the loss function convergence trend of the proposed FRL algorithm. The learning rates of the gradient descent method are set to 0.4, 0.6, and 0.8, respectively. As proved in **Appendix B**, as long as the learning rate is less than a certain threshold, the convergence rate will accelerate with the increase of learning rate. When the learning rate is set to 0.8, the number of convergence rounds can be reduced by 57.6% and 42.7%, respectively, compared with the learning rates of 0.4 and 0.6.

The relationship between the reward and the episode under different action dispersion conditions is show in Fig. 6. The computing power of MEC is set to 21GHz [53]. By using different resource allocation degrees, we consider 21-dimensional, 11-dimensional, and 6-dimensional action s-

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3188434

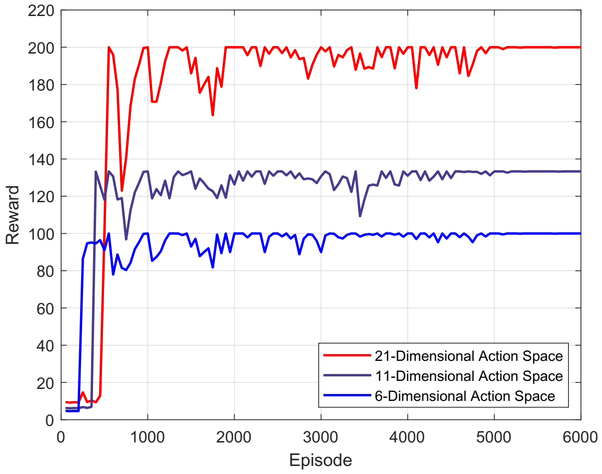


Fig. 6. Reward under different action dispersion conditions.

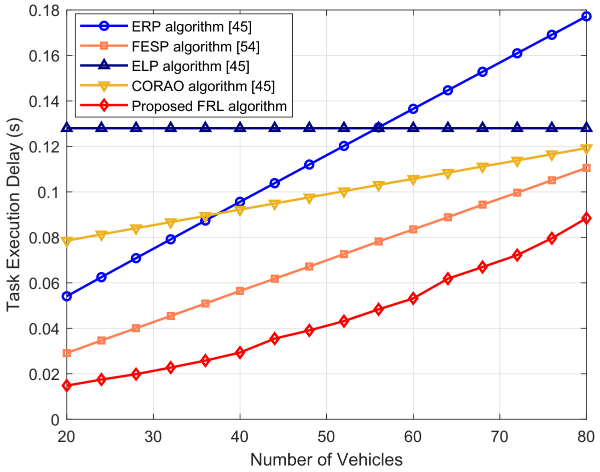


Fig. 7. Task execution delay of different offloading algorithms.

paces, respectively. Simulation results show that with the increase of the degree of action dispersion, the reward value increases and the number of iteration episode needed to achieve a stable reward state becomes larger. The reward function is defined as the inverse of the task execution delay, which is related to the allocation of computing resources. When the size of the computing task is unknown, it is difficult for the rough computing resource selection space to adapt to the size of the computing task, resulting in a high task execution delay and a low reward in the DRL model. The exhaustive selection space of computing resources can better adapt to the size of computing tasks, which not only avoids the problem of high latency caused by too small computing resources, but also avoids the problem of insufficient resources for other users. Therefore, a higher dimension selection space of action space leads to a higher reward. Compared with the 6-dimensional action space, the rewards of 11-dimensional and 21-dimensional action spaces can be improved by 33% and 100%, respectively. However, due to the high dimensions of action space, the number of iterations required to achieve a stable reward state will also increase in contrast to the low dimension of action space.

C. Results and Analysis

Based on the FRL training results in Section V-B, we set the learning rate and the action dispersion conditions of proposed FRL algorithm to 0.8 and 21-dimensional action spaces, respectively. Figs. 7 and 8 evaluate the delay performance of the proposed FRL algorithm in comparison with the following benchmark algorithms.

- The entire RSU processing (ERP) algorithm denotes that all the vehicles offload their computation tasks to the RSU.
- The fixed edge server processing (FESP) algorithm denotes that all the vehicles offload their computation tasks to the MEC server deployed on the BS.
- The entire local processing (ELP) algorithm denotes that all the vehicles execute the computation tasks on themselves.
- The computing offloading and resource allocation optimization (CORAO) algorithm denotes that the computing tasks are completed by computing locally or offloading to RSU.

Fig. 7 illustrates the relation between the task execution delay and the number of vehicles. It can be seen that with the increase of the number of vehicles, the task execution delay of ERP, FESP, CORAO and the proposed FRL algorithms are increasing. Because the RSU does not have enough computing resources as the BS, the interference caused by the shared spectrum is greater in the ERP algorithm, leading to the larger execution delay of the ERP algorithm. The CORAO algorithm can dynamically choose between local offloading and offloading to RSU, which not only reduces the need for offloading to RSU when the task is too small, but also can offload the task to more abundant computing resources when the task is too large. When the interference increases due to the increase of vehicles, the delay performance of CORAO is gradually better than that of ERP. The proposed FRL algorithm can allocate the spectrum and computing resources dynamically according to the environment of vehicle users, which has the lowest task execution delay of less than 100 ms. And there is no significant growth trend for the proposed FRL algorithm with the increase of users. In addition, the ELP algorithm only uses its own computing resources, resulting in the task execution delay of the local offload mode independent of the number of users.

Fig. 8 shows the relationship between the task execution delay and the number of CPU cycles required for a computation task. Results indicate that with the increase of CPU cycles requirement, the task execution delay of all algorithms are increasing. Due to the distance from users to the RSU and the interference caused by the shared spectrum in the ERP algorithm, the task execution delay is larger than that of the ELP algorithm. The BS has more sufficient computing resources, and so the task execution delay of FESP algorithm is lower than the ELP algorithm. Compared with the RSU

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3188434

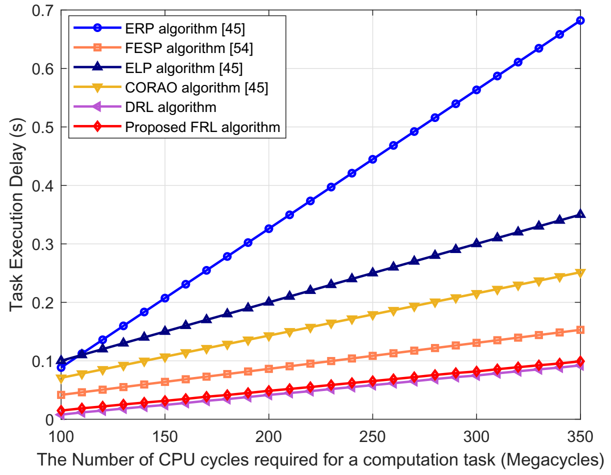


Fig. 8. Task execution delay under different task's CPU requirements.

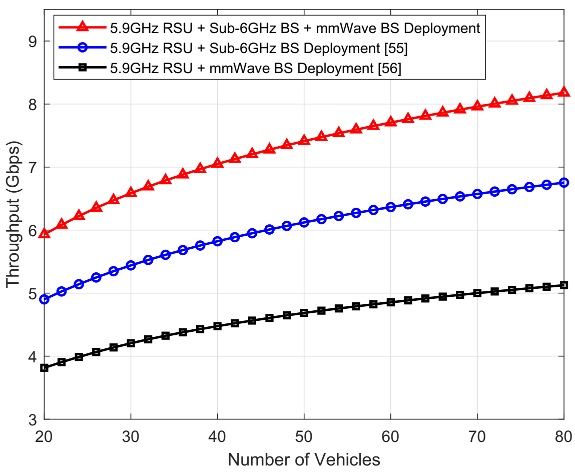


Fig. 10. Throughput comparison for three BS deployment scenarios.

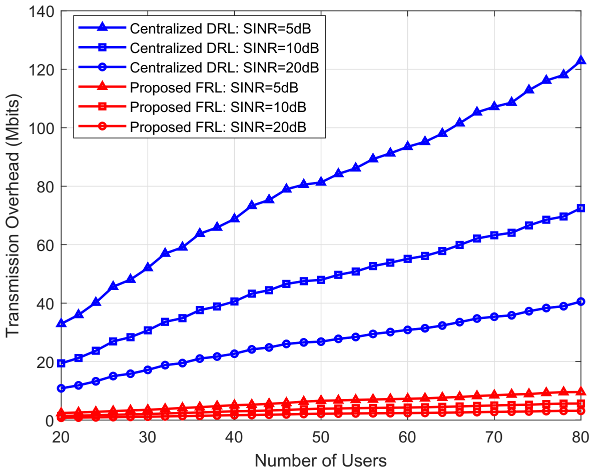


Fig. 9. Transmission overhead under different SINR.

offloading method with a relatively long offload distance and the local offloading method with limited computing resources, the CORAO algorithm can dynamically choose between the local offloading and the offloading to the RSU. Therefore, the latency performance of CORAO is better than that of ELP and ERP. Through the incentive feedback mechanism deployed in the DRL, the delay of the proposed FRL algorithm can be minimized. Compared with the other four algorithms, the task execution delay of the proposed FRL algorithm can be reduced by 84.6%, 70%, 60.6% and 30.9%, respectively. The proposed FRL scheme also performs local model weighted update on the basis of offline reinforcement learning. Although the task execution delay of the proposed FRL algorithm has increased compared with the centralized DRL algorithm, it is still close to the delay of the centralized scheme.

Fig. 9 depicts the relationship between the transmission overhead and the SINR of the centralized DRL algorithm and the proposed algorithm. It can be seen that with the increase of the number of users, the transmission overhead of both algorithms are increasing. The transmission overhead of the centralized DRL algorithm is significantly higher than that of the proposed algorithm. At the same time, with the increase of SINR, the transmission environment of uses is more stable, and the success rate of parameter upload is also improved with the reduced the communication overhead. By transmitting only the model parameters during the model training, the proposed FRL algorithm can reduce the transmission overhead significantly. When the SINR is set to 5dB, 10dB, and 20dB, the transmission overhead of the proposed FRL algorithm is reduced by 90.2%, 90%, and 91% compared with the centralized DRL algorithm.

Fig. 10 describes the relationship between the throughput and the number of users under different BS deployment scenarios. It can be seen that with the increase of the number of users, the throughput increases for all three deployment scenarios. Taking the advantages of both the Sub-6GHz BS with a large coverage and the mmWave BS with a large bandwidth, the proposed hybrid BS deployment architecture as shown in Fig. 1 can achieve the largest throughput of 8.2Gbps. Compared with the deployment scenarios of only using either mmWave BS or Sub-6GHz BS, the throughput of the proposed vehicle-road-BS cooperation architecture can be improved by 60.8% and 22.4%, respectively.

Fig. 11 shows the relationship between the throughput and the time slot of three deployment BS deployment scenarios with 30 VUEs. In each time slot, we generate the location distribution map of VUEs following a Poisson distribution. Due to the large coverage of 3.5GHz BS and the large bandwidth of mmWave BS, the proposed vehicle-road-BS cooperation architecture can achieve the highest throughput of 6.2Gbps, which is increased by 40.9% and 19.2% in contrast to the deployment scenarios of only using either mmWave BS or Sub-6GHz BS.

VI. Hardware Testbed Results

To verify the performance of the proposed vehicle-road-BS cooperation architecture, we design and develop the hardware testbed by using the USRP platform, the mmWave communication equipments, and the MEC servers. Fig. 12 shows the

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3188434

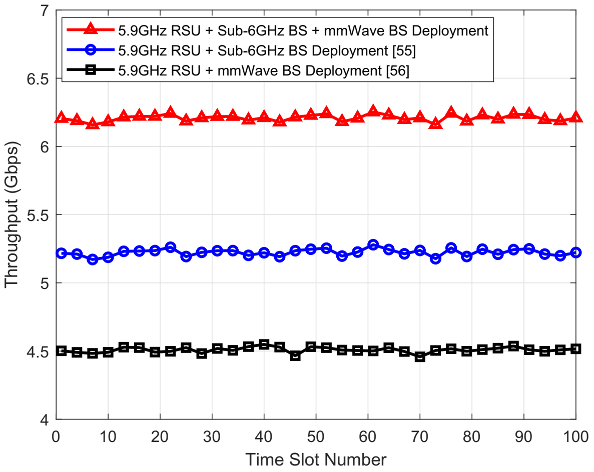


Fig. 11. Throughput comparison for three BS deployment scenarios under different time slots.

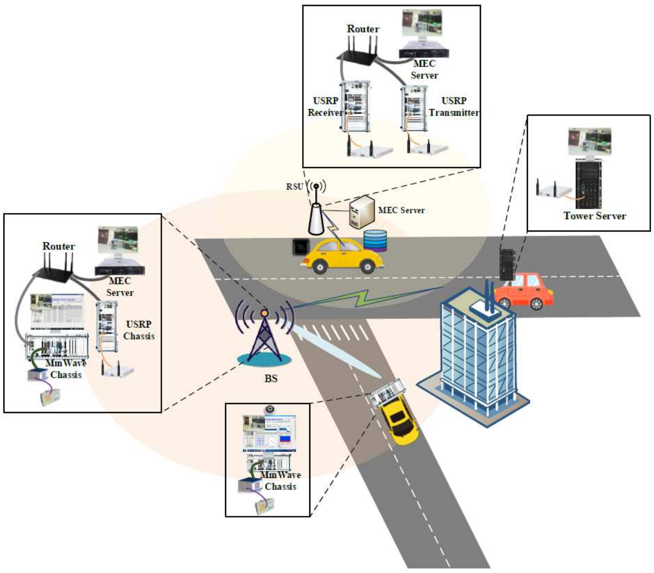
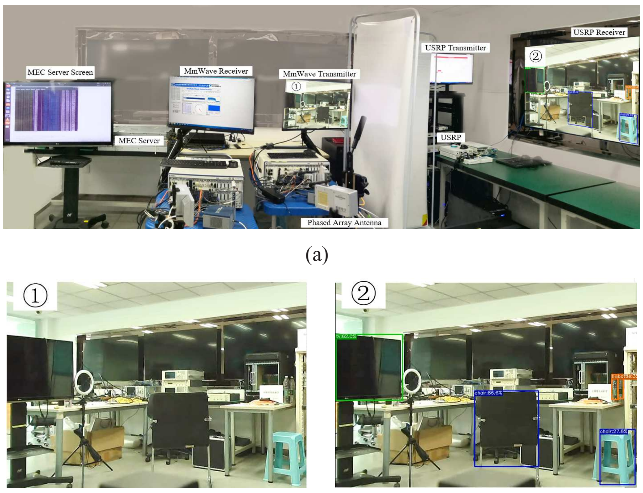


Fig. 12. Relationship between scenario and the hardware testbed.

hardware testbed layout according to the proposed vehicle-road-BS cooperation architecture in Fig. 1, including Sub-6GHz and 28GHz mmWave spectrum bands for a typical CAV scenario based on the ITU-R standard report in [57]. The mmWave communication equipments include two chasses as the CAV transmitter and the BS receiver, respectively. The BS consists of the 28GHz mmWave receiver chassis, MEC server, and the USRP transmitter in the hardware testbed. The RSU consists of the Sub-6GHz USRP equipment and the MEC server. Besides, the USRP equipment is deployed on the CAV side. The environment sensing information is transmitted between the CAV and BS by using both the 3.5GHz and 28GHz mmWave spectrum bands. And the RSU and CAV can utilize the 5.9GHz spectrum band for environment sensing information transmission based on [43]. The equipments of the hardware testbed are shown in Table III.



(a)

(b)

(c)

Fig. 13. Hardware testbed. (a) Testbed layout. (b) Video at the 28GHz mmWave transmitter. (c) Video at the Sub-6GHz USRP receiver.

TABLE III
HARDWARE TESTBED EQUIPMENTS.

Equipment	Model and Parameters	Number
MmWave Chassis	NI PXIe-3610,3620	2
MmWave RF Head	NI mmRH3602 RF Head, 24-33 GHz	2
MmWave Antenna	HDTX270280-64CH Antenna	2
MEC Server	Dell PowerEdge R730	1
USRP	NI USRP-2943R, 1.2-6GHz	4
USRP Chassis	NI PXIe-1085	3
Tower Server	Dell PowerEdge T630	1
SDN Router	Pica8 P-3922	1

The layout of the developed hardware testbed is shown in Fig. 13 (a). The real-time video of the environment sensing information is collected by a mmWave chassis as the CAV in Fig. 13 (b), which is transmitted to another mmWave chassis as the BS via the 28GHz mmWave mobile communication link using two phased array antennas. And the beam alignment and beam tracking algorithms are used based on our previous work in [58]. Then, the real-time video at the mmWave receiver is transmitted to the MEC server via the router. At the MEC server, the yolox algorithm [59] is used to classify and recognize the objects, and the detection results are shown by bounding boxes with different colors. Then, the processed video can be transmitted to the CAV receiver by using the USRP equipment in the Sub-6GHz spectrum band in Fig. 13 (c). Therefore, the environment sensing ability at the CAV receiver can be improved by using the processed video from the 28GHz mmWave transmitter CAV.

In addition, based on the hardware testbed, the performances of throughput and delay of the proposed algorithm are evaluated. Fig. 14 depicts the execution delay by using different radio access technologies (RATs) under various spectrum bands. It can be seen that with the increase of video size, the task execution delay of three RATs is increasing, where the task execution delay of only using the 5.9GHz RSU deployment is the largest. Taking the advantage of large bandwidth in the mmWave spectrum band, the task

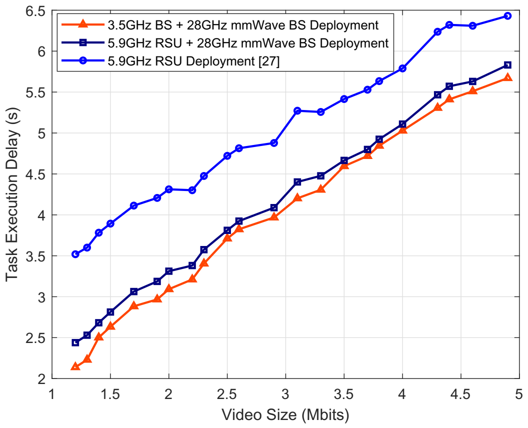


Fig. 14. Task execution delay by using different RATs under various spectrum bands.

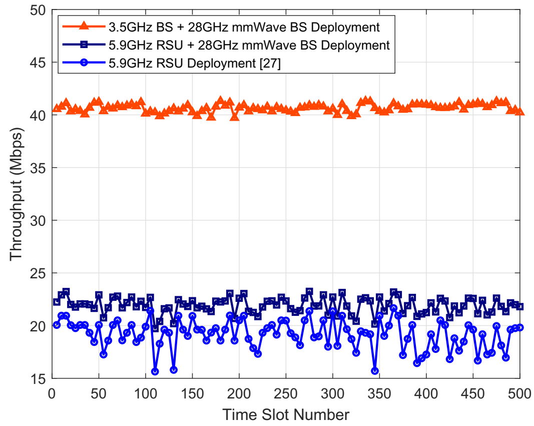


Fig. 15. Throughput comparison under different RATs scenario.

execution delay of using both 5.9GHz RSU and mmWave BS deployment is reduced. By using both the 3.5GHz BS and 28GHz mmWave BS deployment, the task execution delay is the smallest. Compared with the scenario without mmWave deployment, the task execution delay of the proposed vehicle-road-BS cooperation architecture can be reduced by 11.8% in Fig. 14 .

Fig. 15 describes the performance of throughput under different RATs scenario. Due to the large bandwidth of mmWave, the throughput in the scenario of using both 5.9GHz RSU and 28GHz mmWave BS is higher and more stable than that in the scenario of only using the 5.9GHz RSU deployment. The throughput of using both the 3.5GHz BS and 28GHz mmWave BS deployment is the largest, which is 88.3% and 122.9% higher than the other two deployment schemes, respectively.

In summary, the hardware testbed results show that the proposed vehicle-road-BS cooperation architecture can not only decrease the task execution delay, but also enhance the system throughput.

VII. Conclusion

In this paper, we design the vehicle-road-BS cooperation architecture and propose the FRL based task offloading and resource allocation algorithm in the CAVs network, in order to reduce the task execution delay with different communication and computing constraints. The execution delay problem is theoretically formulated and analyzed under three task offloading modes. To adapt to the dynamic topology of CAVs network, we design a DRL algorithm to achieve the optimal task offloading and resource allocation result. To further reduce the data transmission overhead in the centralized reinforcement learning model training process, the FRL algorithm is proposed to minimize the execution delay of the optimal task offloading and resource allocation decisions among CAVs. Simulation results show that compared with the existing benchmark schemes, the proposed algorithm can reduce the task execution delay over 30% and the transmission overhead over 90%. The hardware testbed based results verify that the proposed architecture can improve the system throughput over 88%.

Appendix

A. Proof of *Lemma 1*

Proof: According to the **Assumption 1**, from the definition of convexity and the finite sum structure of $f_i(\omega)$ is $f(\omega)$, we can get that $f(\omega)$ is convex and β-smooth.

B. Proof of *Theorem 1*

Proof:

$$\begin{aligned}||\omega^{t+1}-\omega^*||^2 =&||\omega^t-\varsigma\nabla f(\omega^t)-\omega^*||^2\\ =&||\omega^t-\omega^*||^2-2\varsigma\nabla f(\omega^t)(\omega^t-\omega^*)\\ &+\varsigma^2||\nabla f(\omega^t)||^2.\end{aligned} \tag{36}$$

We consider a new auxiliary point as

$$\omega^M=\omega^*-\frac{1}{\beta}(\nabla f(\omega^*)-\nabla f(\omega^t)). \tag{37}$$

Furthermore, we divide (37) as follows

$$f(\omega^t)-f(\omega^*)=f(\omega^t)-f(\omega^M)+f(\omega^M)-f(\omega^*). \tag{38}$$

According to the convexity of function f, we have

$$\begin{aligned}f(\omega^t)-f(\omega^M)\leq&\nabla f(\omega^t)(\omega^t-\omega^M)\\ =&\nabla f(\omega^t)(\omega^t-\omega^*)\\ &+\nabla f(\omega^t)(\omega^*-\omega^M).\end{aligned} \tag{39}$$

Using the properties of β-smooth, we have

$$\begin{aligned}f(\omega^M)-f(\omega^*)\leq&\nabla f(\omega^*)(\omega^M-\omega^*)+\frac{\beta}{2}||\omega^M-\omega^*||^2\\ =&-\nabla f(\omega^*)(\omega^*-\omega^M)\\ &+\frac{\beta}{2}||\omega^M-\omega^*||^2.\end{aligned} \tag{40}$$

By adding (39) and (40), we have

$$\begin{aligned} f(\omega^t) - f(\omega^*) \leq & \nabla f(\omega^t)(\omega^t - \omega^*) \\ & + (\nabla f(\omega^t) - \nabla f(\omega^*))(\omega^* - \omega^M) \\ & + \frac{\beta}{2}||\omega^M - \omega^*||^2. \end{aligned} \tag{41}$$

Substitute (37) into (41)

$$\begin{aligned} f(\omega^t) - f(\omega^*) \leq & \nabla f(\omega^t)(\omega^t - \omega^*) \\ & + \frac{1}{\beta}(\nabla f(\omega^t) - \nabla f(\omega^*))(\nabla f(\omega^*) - \nabla f(\omega^t)) \\ & + \frac{\beta}{2}\frac{1}{\beta^2}||\nabla f(\omega^t) - \nabla f(\omega^*)||^2 \\ = & \nabla f(\omega^t)(\omega^t - \omega^*) \\ & - \frac{1}{2\beta}||\nabla f(\omega^t) - \nabla f(\omega^*)||^2. \end{aligned} \tag{42}$$

Since ω^* is the final solution, $\nabla f(\omega^*) = 0$, $f(\omega^t) > f(\omega^*)$, we have

$$\nabla f(\omega^t)(\omega^t - \omega^*) - \frac{1}{2\beta}||\nabla f(\omega^t)||^2 \geq 0. \tag{43}$$

Substitute (43) into (36)

$$\begin{aligned} ||\omega^{t+1} - \omega^*||^2 \leq & ||\omega^t - \omega^*||^2 - \frac{\varsigma}{\beta}||\nabla f(\omega^t)||^2 \\ & + \varsigma^2||\nabla f(\omega^t)||^2 \\ = & ||\omega^t - \omega^*||^2 - \varsigma(\frac{1}{\beta} - \varsigma)||\nabla f(\omega^t)||^2. \end{aligned} \tag{44}$$

As long as the learning rate $\varsigma < \frac{1}{\beta}$, the distance between the current solution and the optimal solution will gradually decrease until the optimal solution is obtained

$$||\omega^{t+1} - \omega^*||^2 \leq ||\omega^t - \omega^*||^2. \tag{45}$$

C. Proof of **Theorem 2**

Proof:

$$\begin{aligned} f(\omega^{t+1}) - f(\omega^t) & \leq \nabla f(\omega^t)(\omega^{t+1} - \omega^t) + \frac{\beta}{2}||\omega^{t+1} - \omega^t||^2 \\ & = -\varsigma||\nabla f(\omega^t)||^2 + \frac{\beta}{2}\varsigma^2||\nabla f(\omega^t)||^2 \\ & = -\varsigma(1 - \frac{\beta\varsigma}{2})||\nabla f(\omega^t)||^2. \end{aligned} \tag{46}$$

Then, we insert the optimal solution and compare the distance between two iterations from the limit

$$[f(\omega^{t+1}) - f(\omega^*)] \leq [f(\omega^t) - f(\omega^*)] - \varsigma(1 - \frac{\beta\varsigma}{2})||\nabla f(\omega^t)||^2. \tag{47}$$

According to the properties of the convex function and the Cauchy-Schwarz inequality, we get

$$f(\omega^t) - f(\omega^*) \leq \nabla f(\omega^t)(\omega^t - \omega^*) \leq ||\nabla f(\omega^t)|| \cdot ||\omega^t - \omega^*||. \tag{48}$$

Substitute (48) into (47)

$$\begin{aligned} f(\omega^{t+1}) - f(\omega^*) \leq & [f(\omega^t) - f(\omega^*)] \\ & - \varsigma(1 - \frac{\beta\varsigma}{2})\frac{[f(\omega^t) - f(\omega^*)]^2}{||\omega^t - \omega^*||^2} \\ \leq & [f(\omega^t) - f(\omega^*)] \\ & - \varsigma(1 - \frac{\beta\varsigma}{2})\frac{[f(\omega^t) - f(\omega^*)]^2}{||\omega^0 - \omega^*||^2} \end{aligned} \tag{49}$$

Divide both sides by $[f(\omega^{t+1}) - f(\omega^*)][f(\omega^t) - f(\omega^*)]$

$$\begin{aligned} \frac{1}{f(\omega^t) - f(\omega^*)} \leq & \frac{1}{f(\omega^{t+1}) - f(\omega^*)} \\ & - \frac{\varsigma(1 - \frac{\beta\varsigma}{2})}{||\omega^0 - \omega^*||^2}\frac{f(\omega^t) - f(\omega^*)}{f(\omega^{t+1}) - f(\omega^*)} \\ \leq & \frac{1}{f(\omega^{t+1}) - f(\omega^*)} - \frac{\varsigma(1 - \frac{\beta\varsigma}{2})}{||\omega^0 - \omega^*||^2}. \end{aligned} \tag{50}$$

Accumulate the above formula from 0 to $t-1$

$$\frac{1}{f(\omega^{t+1}) - f(\omega^*)} - \frac{1}{f(\omega^0) - f(\omega^*)} \geq \frac{1}{||\omega^0 - \omega^*||^2}t\varsigma(1 - \frac{\beta\varsigma}{2}). \tag{51}$$

By enlarging the left side of (51), we have

$$\frac{1}{f(\omega^{t+1}) - f(\omega^*)} \geq \frac{1}{||\omega^0 - \omega^*||^2}t\varsigma(1 - \frac{\beta\varsigma}{2}), \tag{52}$$

$$f(\omega^t) - f(\omega^*) \leq ||\omega^0 - \omega^*||^2 \cdot \frac{1}{\varsigma(1 - \frac{\beta\varsigma}{2})} \cdot \frac{1}{t}. \tag{53}$$

The difference between the cost sequence $f(\omega^t)$ and the optimal cost is less than a constant multiple of the sequence $\frac{1}{t}$. Therefore, the convergence rate of the cost sequence is $O(\frac{1}{t})$.

References

[1] A. Eskandarian, C. Wu and C. Sun, “Research Advances and Challenges of Autonomous and Connected Ground Vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 683-711, Feb. 2021.

[2] University of Oulu, “Key Drivers and Research Challenges for 6G Ubiquitous Wireless Intelligence,” Sept. 2019.

[3] Y. Wang, N. Masoud and A. Khojandi, “Real-Time Sensor Anomaly Detection and Recovery in Connected Automated Vehicle Sensors,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1411-1421, Mar. 2021.

[4] ITU-R WP5A Document 5A/106-E, “Annex 16 to Working Party 5A Chairman Report Working Document Towards a Preliminary Draft New Report ITU-R M.[CAV] Connected Automated Vehicles (CAV),” May 2021.

[5] ITU-R WP5A Document 5A/440-E, “Proposed Modification to Working Document towards a Preliminary Draft New Report ITU-R M.[CAV] Connected Automated Vehicles (CAV),” Nov. 2021.

[6] ITU-R WP5A Document 5A/491-E, “Annex 23 to Working Party 5A Chairman Report Working Document Towards a Preliminary Draft New Report ITU-R M.[CAV] Connected Automated Vehicles (CAV),” Nov. 2021.

[7] Z. Zhang, Y. Xiao, Z. Ma, and et al., “6G Wireless Networks: Vision, Requirements, Architecture, and Key Technologies,” *IEEE Vehicular Technology Magazine*, vol. 14, no. 3, pp. 28–41, Sept. 2019.

[8] X. You, C. Wang, J. Huang, and et al., “Towards 6G Wireless Communication Networks: Vision, Enabling Technologies, and New Paradigm Shifts,” *Science China (Information Sciences)*, vol. 64, no. 1, pp. 5–78, Jan. 2021.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3188434

[9] W. Qi, Q. Li, Q. Song, and et al., “Extensive Edge Intelligence for Future Vehicular Networks in 6G,” *IEEE Wireless Communications*, vol. 28, no. 4, pp. 128-135, Aug. 2021.

[10] F. Liu, W. Yuan, C. Masouros, and et al., “Radar-Assisted Predictive Beamforming for Vehicular Links: Communication Served by Sensing,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 11, pp. 7704–7719, Nov. 2020.

[11] W. Yuan, F. Liu, C. Masouros, and et al., “Bayesian Predictive Beamforming for Vehicular Networks: A Low-Overhead Joint Radar-Communication Approach,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1442–1456, Mar. 2021.

[12] K. Zhang, X. Gui, and D. Ren, “Joint Optimization on Computation Offloading and Resource Allocation in Mobile Edge Computing,” *IEEE WCNC*, Marrakesh, Morocco, Apr. 2019.

[13] C. Liu, M. Bennis, M. Debbah, and et al., “Dynamic Task Offloading and Resource Allocation for Ultra-Reliable Low-Latency Edge Computing,” *IEEE Transactions on Communications*, vol. 67, no. 6, pp. 4132–4150, Jun. 2019.

[14] X. Li, C. You, S. Andreev, and et al., “Wirelessly Powered Crowd Sensing: Joint Power Transfer, Sensing, Compression, and Transmission,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 2, pp. 391–406, Feb. 2019.

[15] 5GCAR, “Deliverable D2.1 5GCAR Scenarios, Use Cases, Requirements and KPIs,” Feb. 2019.

[16] 5GAA, “C-V2X Use Cases: Methodology, Examples and Service Level Requirements,” Jun. 2019.

[17] A. A. Alahmadi, A. Q. Lawey, T. E. H. El-Gorashi, and et al., “Distributed Processing in Vehicular Cloud Networks,” *8th International Conference on the Network of the Future (NOF)*, London, U.K., Nov. 2017.

[18] M. Nabi, R. Benkoczi, S. Abdelhamid, and et al,, “Resource Assignment in Vehicular Clouds,” *IEEE ICC*, Paris, France, May 2017.

[19] M. Mukherjee, S. Kumar, M. Shojafar, and et al., “Joint Task Offloading and Resource Allocation for Delay-Sensitive Fog Networks,” *IEEE ICC*, Shanghai, China, May 2019.

[20] K. Zhang, S. Leng, Y. He, and et al., “Mobile Edge Computing and Networking for Green and Low-Latency Internet of Things,” *IEEE Communications Magazine*, vol. 56, no. 5, pp. 39–45, May 2018.

[21] K. Zhang, S. Leng, Y. He, and et al., “Cooperative Content Caching in 5G Networks with Mobile Edge Computing,” *IEEE Wireless Communications*, vol. 25, no. 3, pp. 80–87, Jun. 2018.

[22] N. Kumar, S. Zeadally, and J. J. P. C. Rodrigues, “Vehicular Delay-Tolerant Networks for Smart Grid Data Management Using Mobile Edge Computing,” *IEEE Communications Magazine*, vol. 54, no. 10, pp. 60–66, Oct. 2016.

[23] J. Zhang, W. Xia, Z. Cheng, and et al., “An Evolutionary Game for Joint Wireless and Cloud Resource Allocation in Mobile Edge Computing,” *9th International Conference on Wireless Communications and Signal Processing (WCSP)*, Nanjing, China, Dec. 2017.

[24] K. Zhang, Y. Mao, S. Leng, and et al., “Delay Constrained Offloading for Mobile Edge Computing in Cloud-Enabled Vehicular Networks,” *8th International Workshop on Resilient Networks Design and Modeling (RNDM)*, Halmstad, Sweden, Oct. 2016.

[25] X. Chen, Z. Liu, Y. Chen, and et al., “Mobile Edge Computing Based Task Offloading and Resource Allocation in 5G Ultra-Dense Networks,” *IEEE Access*, vol.7, pp. 184172–82, Dec. 2019.

[26] T. X. Tran and D. Pompili, “Joint Task Offloading and Resource Allocation for Multi-Server Mobile-Edge Computing Networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 856–868, Jan. 2019.

[27] Y. Dai, D. Xu, S. Maharjan, and et al., “Joint Load Balancing and Offloading in Vehicular Edge Computing and Networks,” *IEEE Internet of Things Journa*, vol. 6, no. 3, pp. 4377–4387, Jun. 2019.

[28] Z. Chang, L. Liu, X. Guo, and et al., “Dynamic Resource Allocation and Computation Offloading for IoT Fog Computing System,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3348-3357, May 2021.

[29] H. Peng, Q. Ye,and X. Shen, “Spectrum Management for Multi-Access Edge Computing in Autonomous Vehicular Networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 7, pp. 3001–3012, Jul. 2020.

[30] F. Tang, Y. Kawamoto, N. Kato, and et al., “Future Intelligent and Secure Vehicular Network Toward 6G: Machine-Learning Approaches,” *Proceedings of the IEEE*, vol. 108, no. 2, pp. 292–307, Feb. 2020.

[31] H. Ye, G. Y. Li, and B. F. Juang, “Deep Reinforcement Learning Based Resource Allocation for V2V Communications,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3163–3173, Apr. 2019.

[32] H. Ye and G. Y. Li, “Deep Reinforcement Learning for Resource Allocation in V2V Communications,” *IEEE ICC*, Kansas City, MO, May 2018.

[33] N. C. Luong, D. T. Hoang, S. Gong, and et al., “Applications of Deep Reinforcement Learning in Communications and Networking: A Survey,” *IEEE Communications Surveys and Tutorials*, vol. 21, no. 4, pp. 3133-3174, May 2019.

[34] J. Wang, C. Jiang, H. Zhang, and et al., “Thirty Years of Machine Learning: The Road to Pareto-Optimal Wireless Networks,” *IEEE Communications Surveys and Tutorials*, vol. 22, no. 3, pp. 1472-1514, Jan. 2020.

[35] Y. He, Z. Zhang, F. R. Yu, and et al., “Deep-Reinforcement-Learning-Based Optimization for Cache-Enabled Opportunistic Interference Alignment Wireless Networks,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10433-10445, Nov. 2017.

[36] Y. He, F. R. Yu, N. Zhao, and et al., “Software-Defined Networks with Mobile Edge Computing and Caching for Smart Cities: A Big Data Deep Reinforcement Learning Approach,” *IEEE Communications Magazine*, vol. 55, no. 12, pp. 31-37, Dec. 2017.

[37] S. Samarakoon, M. Bennis, W. Saad, and et al., “Federated Learning for Ultra-Reliable Low-Latency V2V Communications,” *IEEE GLOBECOM*, Abu Dhabi, United Arab Emirates, Dec. 2018.

[38] N. H. Tran, W. Bao, A. Zomaya, and et al., “Federated Learning over Wireless Networks: Optimization Model Design and Analysis,” *IEEE INFOCOM*, Paris, France, Apr. 2019, pp. 1387–1395.

[39] S. Yu, X. Chen, Z. Zhou, and et al, “When Deep Reinforcement Learning Meets Federated Learning: Intelligent Multitimescale Resource Management for Multiaccess Edge Computing in 5G Ultradense Network,” *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2238-2251, Feb. 2021.

[40] M. Xu, J. Peng, B. B. Gupta, and et al., “Multi-Agent Federated Reinforcement Learning for Secure Incentive Mechanism in Intelligent Cyber-Physical Systems,” *IEEE Internet of Things Journal (Early Access)*, May 2021.

[41] X. Wang, C. Wang, X. Li, and et al., “Federated Deep Reinforcement Learning for Internet of Things With Decentralized Cooperative Edge Caching,” *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9441-9455, Oct. 2020.

[42] A. Imteaj, U. Thakker, S. Wang, and et al., “A Survey on Federated Learning for Resource-Constrained IoT Devices,” *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 1-24, Jan. 2022.

[43] ITU-R WP5A, “Handbook on Land Mobile (Including Wireless Access) Volume 4 - Intelligent Transport Systems,” Mar. 2021.

[44] 3GPP TR 38.901, “Study on channel model for frequencies from 0.5 to 100 GHz,” Release 16, v16.1.0, Dec. 2019.

[45] J. Zhao, Q. Li, Y. Gong, and et al., “Computation Offloading and Resource Allocation For Cloud Assisted Mobile Edge Computing in Vehicular Networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7944–56, Aug. 2019.

[46] H. Yang, Z. Xiong, J. Zhao, and et al., “Deep Reinforcement Learning-Based Intelligent Reflecting Surface for Secure Wireless Communications,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 375-388, Jan. 2021.

[47] T. G. Nguyen, T. V. Phan, D. T. Hoang, and et al., “Federated Deep Reinforcement Learning for Traffic Monitoring in SDN-Based IoT Networks,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 4, pp. 1048-1065, Dec. 2021.

[48] S. Han, J. Pool, J. Tran, and et al., “Learning both weights and connections for efficient neural networks,” *Advances in Neural Information Processing Systems*, 2015.

[49] https://www.python.org/downloads/

[50] 3GPP TR 38.886, “User Equipment (UE) ratio transmission and reception,” Release 16, v16.3.0, Mar. 2021.

[51] 3GPP TS 38.104, “Base Station (BS) Radio Transmission and Reception,” Release 17, v17.3.0, Sept. 2021.

[52] 3GPP TR 38.913, “Study on Scenarios and Requirements for Next Generation Access Technologies,” Release 16, v16.0.0, Jul. 2019.

[53] J. Ren, G. Yu, Y. He, and et al., “Collaborative Cloud and Edge Computing for Latency Minimization,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 5031-5044, May 2019.

[54] Y. Liu, H. Yu, S. Xie, and et al., “Deep Reinforcement Learning for Offloading and Resource Allocation in Vehicle Edge Computing and Networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11158–11168, Nov. 2019.

[55] X. Zhang, M. Peng, S. Yan, and et al., “Deep-Reinforcement-Learning-Based Mode Selection and Resource Allocation for Cellular V2X Communications,” *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6380-6391, Jul. 2020.

[56] F. Liu, W. Yuan, C. Masouros, and et al., “Radar-Assisted Predictive Beamforming for Vehicular Links: Communication Served by Sensing,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 11, pp. 7704-7719, Nov. 2020.

[57] ITU-R WP5A Document 5A/TEMP/171-E, “Working Document towards a Preliminary Draft New Report ITU-R M.[CAV] Connected Automated Vehicles (CAV),” Nov. 2021.

[58] Q. Zhang, H. Sun, Z. Wei, and et al., “Sensing and Communication Integrated System for Autonomous Driving Vehicles,” *IEEE INFOCOM WKSHPS*, Toronto, ON, Canada, Jul. 2020, pp. 1278-1279.

[59] Z. Ge, S. Liu, F. Wang, and et al., “Yolox: Exceeding yolo series in 2021,” arXiv preprint arXiv:2107.08430, 2021.

Qixun Zhang (M’12) received the B.Eng. degree in communication engineering and the Ph.D. degree in circuit and system from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2006 and 2011, respectively.

From Mar. to Jun. 2006, he was a Visiting Scholar at the University of Maryland, College Park, Maryland. From Nov. 2018 to Nov. 2019, he was a Visiting Scholar in the Electrical and Computer Engineering Department at the University of Houston, Texas. He is a Professor with the Key Laboratory of Universal Wireless Communications, Ministry of Education, and the School of Information and Communication Engineering, BUPT. His research interests include B5G/6G mobile communication system, spectrum sharing access, joint communication and sensing system for autonomous driving vehicle, mmWave communication system, cognitive radio and heterogeneous networks, game theory, and unmanned aerial vehicles (UAVs) communication. He is a member of IEEE and active in ITU-R WP5A/5C/5D standards.

Hao Wen received the Master degree in Information and Communication Engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2022. His current research interests include deep reinforcement learning, federated reinforcement learning, multiaccess edge computing and vehicular networks.

Ying Liu is currently pursuing the Master degree with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications (BUPT), Beijing, China. Her current research interests include mobile edge computing algorithm and testbed design, and vehicular networks.

Shuo Chang received the B.S. degree in communication engineering from Shenyang Jianzhu University, Shenyang, China, in 2015. And he received Ph.D. degree from Beijing University of Posts and Telecommunications (BUPT), China, in 2020. He is currently a postdoc in the Beijing University of Posts and Telecommunications, Beijing, China. He is also a member of the Key Laboratory of Universal Wireless Communications, Ministry of Education, China. His research interests include signal processing, visual object tracking, visual detection, and sensor fusion.

Zhu Han (S’01-M’04-SM’09-F’14) received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1999 and 2003, respectively.

From 2000 to 2002, he was an R&D Engineer of JDSU, Germantown, Maryland. From 2003 to 2006, he was a Research Associate at the University of Maryland. From 2006 to 2008, he was an assistant professor at Boise State University, Idaho. Currently, he is a John and Rebecca Moores Professor in the Electrical and Computer Engineering Department as well as in the Computer Science Department at the University of Houston, Texas. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. Dr. Han received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the Journal on Advances in Signal Processing in 2015, IEEE Leonard G. Abraham Prize in the field of Communications Systems (best paper award in IEEE JSAC) in 2016, and several best paper awards in IEEE conferences. Dr. Han was an IEEE Communications Society Distinguished Lecturer from 2015-2018, AAAS fellow since 2019, and ACM distinguished Member since 2019. Dr. Han is a 1% highly cited researcher since 2017 according to Web of Science. Dr. Han is also the winner of the 2021 IEEE Kiyo Tomiyasu Award, for outstanding early to mid-career contributions to technologies holding the promise of innovative applications, with the following citation: “for contributions to game theory and distributed management of autonomous communication networks.”