Chinese Journal of Electronics
Vol.32, No.6, Nov. 2023

Collaborative Caching in Vehicular Edge Network Assisted by Cell-Free Massive MIMO

WANG Chaowei[1,2], WANG Ziye[1], XU Lexi[3], YU Xiaofei[1], ZHANG Zhi[4], and WANG Weidong[1]

(1. *School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China*)

(2. *Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China*)

(3. *Research Institute, China United Network Communications Corporation, Beijing 100048, China*)

(4. *School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China*)

Abstract — The 6G mobile communications demand lower content delivery latency and higher quality of service for vehicular edge network. With the popularity of content-centric networks, mobile users are paying more and more attention to the delay and reliability of fetching cached content. For reducing communication costs, increasing network capacity and improving the content delivery, we propose a collaborative caching scheme based on deep reinforcement learning for vehicular edge network assisted by cell-free massive multiple-input multiple-output (MIMO) system, in which the macro base station is considered as the central processor unit, and the roadside units are treated as roadside access points (RSAPs). The proposed scheme can effectively cache contents in edge nodes, i.e., RSAPs and vehicles with caching capability. We jointly consider the mobility of vehicles and the content request preferences of users, then we use deep Q-networks algorithm to optimize the caching decisions. Simulation results show that the proposed scheme can significantly reduce the content delivery average latency and increase the content cache hit ratio.

Key words — Vehicular edge network, Cell-free massive MIMO, Collaborative caching, Content delivery latency, Deep reinforcement learning, Cache hit ratio.

I. Introduction

With the development of smart vehicles and mobile communication, artificial intelligence (AI)-based Internet of vehicles (IoV) technologies are rapidly evolving. Meanwhile, the advancement of smart cities and intelligent transportation systems (ITSs) [1]–[3] also brings various new applications for the Internet of everything (IoE). However, IoV faces the challenges of high dynamic topology, computation resource limitation, extended network scale [4], and quality of service (QoS) provisions. The AI technology, especially deep learning (DL) in 6G wireless networks, is necessary for facilitating the IoV systems to coordinate communication and computation resources [5].

The standardization of vehicle-to-everything (V2X) technologies, such as IEEE 802.11 V2X and Cellular V2X (C-V2X) [6]–[8], has been supporting transportation services like road safety, traffic efficiency, and passenger entertainment, etc. [4], [9]. There is no doubt that 6G will generate more ultra-reliable and low-latency application scenarios, which makes the mobile users want to get content rapidly and accurately [10], [11]. Mobile edge computing (MEC) is considered as a novel computing model [12], [13] that provides quasi-cloud services at the edge of the wireless access network and can effectively relieve the burden on backhaul links by reducing communication overhead caused by content transmission [14]–[16]. Hence, vehicular edge computing (VEC) networks deploy a large number of services on edge nodes, such as roadside units (RSUs) equipped with MEC servers [17], [18] and vehicles equipped with onboard units (OBUs). There are two fundamental communication models in VEC: vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I). Vehicles within the coverage of an RSU can request content from the RSU, i.e., V2I, and the RSU can respond the request through

Manuscript Received Aug. 30, 2022; Accepted Feb. 7, 2023. This work was supported by the National Key R&D Program of China (2020YFB1807204).

 DOI:10.23919/cje.2022.00.294

infrastructure-to-vehicle (I2V) [19]. Vehicles can also request content from other neighboring vehicles via V2V links.

Content caching and delivery in vehicular edge network need to provide passengers with a comfortable and safe driving environment, abundant and various entertainment, information, and social network services [20], [21]. Meanwhile, it is also expected to meet the scalability and high coverage of communication brought by vehicular mobility. Thus we introduce cell-free massive multiple-input multiple-output (MIMO) to solve these problems more easily. Cell-free massive MIMO combines the concepts of distributed MIMO and massive MIMO by replacing the complicated macro base station (MBS) with a large number of simplified access points (APs) [22], [23]. In the vehicular communication, we consider multiple RSUs equipped with simple antenna configuration as roadside access points (RSAPs). Besides, we assume MBS connecting to the cloud servers via backhaul links can cache all contents and act as the central processor unit (CPU) [24], [25]. A set of distributed RSAPs simultaneously provides collaborative caching services to all the vehicles within the coverage. RSAPs receive content delivery requests that are collected and processed by the CPU. The integration of vehicular edge network and cell-free massive MIMO system can not only effectively improve the stability and scalability of the network in the case of high-speed vehicular movement, but also reduce the deployment cost of MBS and improve the performance of the content transmission.

To evaluate the caching placement scheme in vehicular edge network assisted by cell-free massive MIMO, cache delivery latency and cache hit ratio are the two metrics that are commonly used [20]. Mobile users are sensitive to the delay and reliability of content delivery, while the content providers focus on the content cache hit ratio, which means the faster retrieval of content brings fewer expenses. Caching contents in edge APs including RSAPs, and vehicles with caching capability (CVs), makes content closer to the requesting vehicles (RVs). When RVs enter the coverage of RSAPs, they can request their interested content according to their preference, and if RSAPs or CVs have cached the requested content [26], the corresponding content can be sent to the RVs via V2I or V2V links. Besides, the vehicular mobility can also expand content delivery. In contrast to content retrieval from the CPU, RSAPs and CVs collaboratively cache the content at the edge, which can effectively reduce the communication cost, increase the network capacity and content delivery ratio. However, considering the limited caching capacity of RSAPs and CVs, it is necessary to investigate a collaborative caching scheme that can both reduce content delivery latency and make full use of cache storage.

In this paper, we study the collaborative caching placement problem in urban intersections scenario. The main contributions of this work can be summarized as follows:

1) We consider traffic congestion, vehicular mobility, mobile users' content preferences and limited cache storage to establish an efficient collaborative caching model between RSAPs and CVs at intersections, which gathers various popular contents.

2) To cache popular contents preferentially in the limited storages of edge nodes and minimize the content delivery average latency, we propose a deep reinforcement learning (DRL)-based caching algorithm. The vehicular trajectory, speed, and user preferences, requested contents are deliberatively arranged for CVs and RSAPs.

3) We conduct extensive simulations to evaluate the proposed scheme. The results demonstrate that the proposed collaborative caching scheme can effectively reduce the content delivery average latency and improve the content cache hit ratio than other comparison schemes.

The rest of this paper is organized as follows. Section II reviews the related work. Section III describes the system model of vehicular edge network assisted by cell-free massive MIMO. Then Section IV illustrates DRL-based collaborative caching scheme in detail. Simulation results are presented and discussed in Section V. Finally, Section VI concludes this paper.

II. Related Work

With the increasing number of vehicles and the new demands of multimedia services, caching schemes in vehicular edge network have been extensively studied [27]–[33]. The VEC-based content caching strategy should consider not only the QoS of content delivery but also the vehicular mobility, the content popularity, the capacity of caching nodes, etc.

Su *et al.* [27] proposed a vehicular edge caching strategy based on cross-entropy to adapt to dynamic changes of the content popularity and considered the cooperative caching among RSUs. In [28], the authors presented the content dissemination framework, in which they took an auction game between vehicles into account, and incentivized vehicles to participate in the content distribution through V2X communication and earn more profits from the content providers. Akhavan Bitaghsir *et al.* [29] designed a content distribution network so that information can be shared among RSUs. They also proposed a caching algorithm that assigns an appropriate caching subset of content to each RSU and used a strategic resource allocation game to improve the caching hit ratio. Zhou *et al.* [30] focused on alleviating the consumption of backhaul links. They studied the proactive caching problem in multi-access edge networks and used game theory to minimize the communication cost by considering content preferences and social relationships between users. In [31], the authors used Markov processes to model the V2X communication and proposed a proactive caching algorithm involving vehicle mobility and content chunking to minimize the transmission delays while improving cache hit ratio. AlNagar *et al.* [32] investigated caching algorithm based on the knapsack problem and suboptimal relaxations and verified it in collaborative RSUs and non-collaborative ways. Yao *et al.* [33] introduced the concepts of social similarity and bridging centrality of vehicles. Furthermore, they used hidden Markov model (HMM) to predict the probability of vehicular destinations through historical trajectories, and select suitable vehicles as caching relay nodes to provide better quality of experience (QoE) to users.

Although the above studies have already take into account part of the vehicular mobility, content popularity, caching storage, and vehicles-RSUs collaboration in caching schemes, these factors are not jointly con sidered for higher caching efficiency. The introduction of AI technologies can effectively meet these requirements. To obtain the optimal caching strategy, caching schemes were optimized by federated learning (FL) and DRL approaches in [21], [34]–[38].

Yu *et al.* [34] presented a mobility-aware proactive edge caching scheme based on FL, which used context-aware adversarial autoencoder to predict highly dynamic content popularity. In [21], the authors designed a dynamic content delivery policy that minimizes the vehicular cost through the double deep Q-network algorithm, which overcomes the large-scale state space and reduces Q-value overestimation. Qiao *et al.* in [35] jointly optimize the content placement and delivery problem through three-way cooperation among macrocell station, RSUs, and intelligent vehicles, and they proposed a collaborative edge caching scheme based on the deep deterministic policy gradient (DDPG) framework. The aim in [36] is minimizing the latency of caching services. And the authors conceived a heuristic Q-learning scheme and used the long short-term memory (LSTM) networks to predict vehicular mobility and achieved an effective proactive caching policy. Ning *et al.* [37] developed an intent-based traffic flow control system using DRL to dynamically coordinate edge computing and content caching. It helps mobile network operators to maximize their profits. Zhang *et al.* [38] applied DDPG model and proposed an edge caching algorithm in a high-speed free-flow scenario to improve QoS.

Based on the above survey, we observe that the collaborative caching schemes need to be improved, and there is still a lack of joint consideration of communication, computation and caching resources, as well as a need for more efficient algorithms with better QoS.

III. System Model

In this section, we construct the collaborative caching framework, including the network model, vehicular mobility model, communication model, and content caching and delivery model.

1. Network model

In the urban area, shopping malls, supermarkets, office buildings, and well-known companies can all be potential content providers. The content providers hope that their contents can accurately target nearby vehicles and create greater cost-effectiveness. Different vehicular users have their own interests or preferences, and some popular contents may be requested by several vehicles simultaneously. To increase the profits for content providers and to help vehicular users obtain their interesting contents more quickly and accurately when passing a specific area, it makes sense to consider content caching strategies in popular urban areas.

Fig.1 illustrates an urban vehicular edge caching network assisted by cell-free massive MIMO, which consists of the cloud server, a CPU, RSAPs, and moving vehicles (CVs and RVs). The RSAPs with computing and caching capabilities that are equipped with MEC servers are denoted as $\{\mathrm{rsap}_1, \dots, \mathrm{rsap}_m, \dots, \mathrm{rsap}_M\}$. Besides, CVs equipped with OBUs that have caching capability are represented by the set of $\{\mathrm{cv}_1, \dots, \mathrm{cv}_k, \dots, \mathrm{cv}_K\}$. All these nodes mentioned above can serve as content providers. RVs with wireless transmission ability can be denoted by $\{\mathrm{rv}_1, \dots, \mathrm{rv}_n, \dots, \mathrm{rv}_N\}$. CPU and RSAPs have certain communication coverage. After RVs request the content, if these content providers have already cached the requested content, RVs can get the content from RSAPs via V2I links or CVs via V2V links. If there is no cached content, the content can only be obtained from the CPU. The CPU can access the cloud servers as they have enough storage to store all the contents. RSAPs and CVs have limited cache storage and can only selectively cache contents.

Taking into account the mobility of vehicles, the content request time of RVs is divided into T discrete time slots that can be expressed as $\mathcal{T}=\{1, \dots, t, \dots, T\}$. Considering the signaling overhead caused by frequent communication link switching, each RV can only choose one link from V2I or V2V in each time slot to deliver

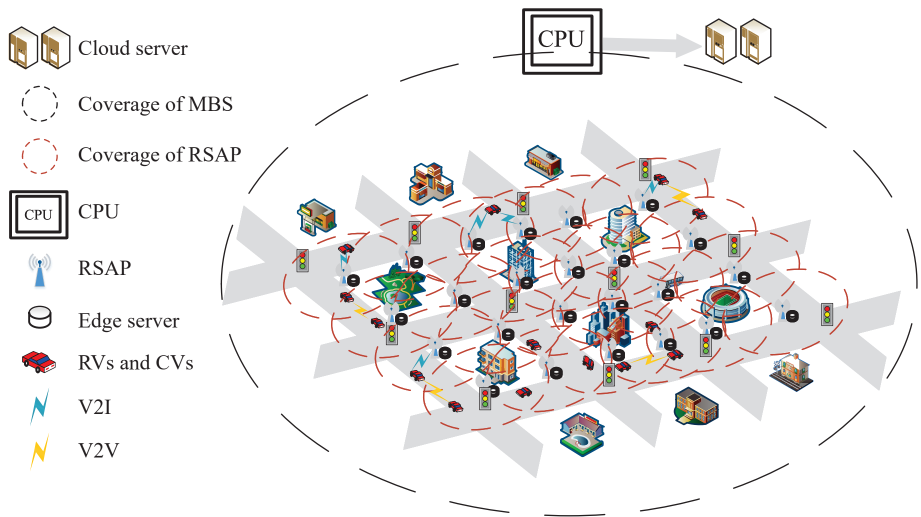


Fig. 1. Vehicular edge network assisted by cell-free massive MIMO.

the content. We focus on the caching policy that minimizes the content delivery latency for all RVs within T time slots.

2. Vehicular mobility model

The locations of the CPU and RSAPs are fixed and the positions of the vehicles are changing over time. The antenna height of RSAPs is denoted as $h_{\rm s}$, and the antenna height of vehicles (CVs and RVs) is denoted as $h_{\rm v}$. Assuming that each vehicle moves at a constant speed on the road, the sets of speeds of RVs and CVs are denoted as $\{v_1, \ldots, v_n, \ldots, v_N\}$ and $\{u_1, \ldots, u_k, \ldots, u_K\}$ respectively. The coverage radiuses of CPU and RSAPs are $R_{\rm cpu}$ and $R_{\rm s}$ respectively.

In V2V communication as shown in Fig.2, if rv_n and cv_k are moving in the same lane, d_0 is the initial opposite distance, the distance between them at time slot t can be expressed as

$$d_{k,n}(t) = |d_0 + (u_k - v_n)t| \tag{1}$$

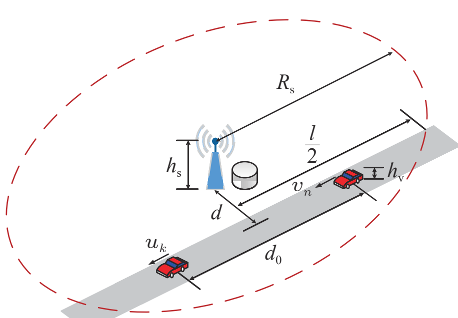


Fig. 2. The position model between RSAP and vehicles.

In V2I communication as shown in Fig.2, d is the vertical distance of rsap_m from the lane, l is the maximum length of the road covered by rsap_m, which can be calculated as $\sqrt{(\frac{l}{2})^2 + d^2 + (h_{\rm s} - h_{\rm v})^2} = R_{\rm s}$, the distance between rv_n and rsap_m at time slot t can be expressed as

$$d_{m,n}(t) = \sqrt{\left(\frac{l}{2} - v_n t\right)^2 + d^2 + (h_{\rm s} - h_{\rm v})^2} \tag{2}$$

3. Communication model

It is assumed that the V2I and V2V links occupy different frequency bands and there is no interference between each other. To simplify the problem, we consider that the channel environment is constant within one time slot of each content request. The bandwidths allocated by the system for the V2I and V2V links are $B_{\rm s}$ and $B_{\rm v}$ respectively. Due to the high mobility of vehicles, the V2V link is unstable. And we want to ensure that the RV can acquire one whole content from the CV during the period of V2V link connection. Therefore, for simplicity, we only consider V2V communication between RVs and CVs moving in the same lane and same direction. At time slot t, the communication models of V2I and V2V links are shown as follows.

We consider that RSAPs and CVs cache content with equal size C_f, and all communication channels are additive white Gaussian noise (AWGN) channels. The transmission rate of rsap_m to deliver cached contents via V2I link to rv_n can be expressed as

$$R_{m,n}(t) = B_{\rm s} \log_2 \left(1 + \frac{P_{\rm s} G_{m,n}(t)}{\sigma^2}\right) \tag{3}$$

where $P_{\rm s}$ is the constant transmission power of V2I links, $G_{m,n}(t)$ is the channel gain between rsap_m and rv_n, σ^2 is the noise power.

The content delivery latency of V2I link can be expressed as

$$D_{m,n}(t) = \frac{C_f}{R_{m,n}(t)} \tag{4}$$

The transmission rate of cv_k to deliver cached content via V2V link to rv_n can be expressed as

$$R_{k,n}(t) = B_{\mathrm{v}} \log_2 \left(1 + \frac{P_{\mathrm{v}} G_{k,n}(t)}{\sigma^2}\right) \tag{5}$$

where P_{v} is the constant transmission power of V2V links, $G_{k,n}(t)$ is the channel gain between cv_k and rv_n.

Similarly, the content delivery latency of V2V link can be expressed as

$$D_{k,n}(t) = \frac{C_f}{R_{k,n}(t)} \tag{6}$$

4. Content caching and delivery model

To improve the caching efficiency, RSAPs and CVs should cache not only the content of their own regions but also the content of the adjacent regions that may be requested by the vehicles passing by. S contents with the same size C_f are denoted as the set of $\{c_1, \dots, c_s, \dots, c_S\}$.

The Zipf distribution is often used to model the popularity of content according to [39], in which λ is the skewness parameter for the Zipf distribution. For different RVs, there are contents that they are not interested in at all or the contents that they are very interested in. We assume that rv_n has interest in $x\,(x \leq S)$ contents, the probability of ith content being requested by rv_n can be expressed as

$$P_{n,i} = \frac{i^{-\lambda}}{\sum_{j=1}^{x} j^{-\lambda}} \tag{7}$$

$\mathcal{Q} = [Q_1, \dots, Q_n, \dots, Q_N]^{\mathrm{T}}$ represents the content request sequences of RVs, and is generated according to formula (7). $Q_n = [q_n(1), \dots, q_n(t), \dots, q_n(T)]\,(1 \leq n \leq N)$, where $q_n(t)$ represents the index of content that is requested by rv_n. For example, $q_n(t) = s$ means the content c_s is requested by rv_n at time slot t.

The caching policy of CVs and RSAPs needs to take into account the preferences of different RVs and cache as much content as possible that are most requested by RVs. Due to the limited storage, we denote that the cache capacity of RSAPs and CVs are C_{s} and C_{v} respectively.

We define two binary cache indicators, i.e., $\alpha_{m,s} \in \{0,1\}$ and $\beta_{k,s} \in \{0,1\}$. If $\alpha_{m,s} = 1$, rsap_m have cached the content c_s, otherwise $\alpha_{m,s} = 0$. If $\beta_{k,s} = 1$, the content c_s has been cached by cv_k, otherwise $\beta_{k,s} = 0$. For the same content c_s, if $\alpha_{m,s} = 0$ and $\beta_{k,s} = 0$, the RVs can only fetch the content from the CPU which brings in higher latency. Because the content providers need to pay the caching incentives to RSAPs and CVs, the caching policy of RSAPs and CVs should satisfy as many user preferences as possible during the whole period.

At each time slot, rv_n requests only one content at most, $\eta_{m,n}(t)$ is the binary indicator that indicates which RSAP is used to deliver content to rv_n at time slot t. If $\eta_{m,n}(t) = 1$, the content is delivered by rsap_m, otherwise $\eta_{m,n}(t) = 0$. $\sum_{m=1}^{M} \eta_{m,n}(t) \leq 1$ indicates that only one of the RSAPs at most can deliver content to rv_n per time slot. Similarly, $\lambda_{k,n}(t)$ is the binary indicator of content delivery condition from cv_1 to cv_K. $\sum_{k=1}^{K} \lambda_{k,n}(t) \leq 1$ indicates that only one of the CVs at most can deliver content to rv_n per time slot. And $\sum_{m=1}^{M} \eta_{m,n}(t) + \sum_{k=1}^{K} \lambda_{k,n}(t) \leq 1$ means rv_n can choose at most one link between V2I and V2V links to fetch content per time slot.

We use a Markov process [31], [40] to model the mobility of vehicles. When vehicles approach the crossroads, their mobility can be seen as discrete random processes. For each RV, the probability of moving to the next position only depends on its position at present. As shown in Fig.1, one RV may connect with a RSAP or a CV to fetch content. At current time slot t of rv_n connecting with rasp_m, the state transition probability of rv_n connecting with cv_k in the next time slot $t+1$ can be expressed as $\Pr(\lambda_{k,n}(t+1) = 1 | \eta_{m,n}(t) = 1)$.

According to the previous discussions, at each content requesting period, for rv_n requests content c_s, the transmission delay of content delivery can be expressed as

$$\begin{aligned} D_{n,s}(t) = & \sum_{k=1}^{K} \lambda_{k,n}(t) \beta_{k,s} D_{k,n}(t) \\ & + \sum_{m=1}^{M} \eta_{m,n}(t) \alpha_{m,s} D_{m,n}(t) \\ & + (1 - \beta_{k,s})(1 - \alpha_{m,s}) \tau_{\mathrm{cpu}} \end{aligned} \tag{8}$$

where τ_{cpu} means the latency of content delivery from the CPU. $\lambda_{k,n}(t)$ and $\eta_{m,n}(t)$ cannot be both 1 at same time slot. When $\alpha_{m,s} = 1$, if $\lambda_{k,n}(t) = 0$ and $\eta_{m,n}(t) = 1$, rv_n can fetch content c_s from rsap_m at time slot t. When $\beta_{k,s} = 1$, if $\lambda_{k,n}(t) = 1$ and $\eta_{m,n}(t) = 0$, rv_n can fetch content c_s from cv_k at time slot t. Otherwise, rv_n can only fetch content c_s from the CPU.

5. Problem formulation

In our proposed vehicular edge caching network assisted by cell-free massive MIMO, RVs can be delivered their requested content from neighboring CVs, RSAPs, or CPU. To reduce the backhaul link pressure and improve QoS, we aim to minimize the content delivery average latency, we model the content delivery average latency optimization problem for collaborative caching scheme as follows:

$$\min_{\mathcal{Q},\alpha_{m,s},\beta_{k,s}} \frac{\sum_{t=1}^{T}\sum_{n=1}^{N} D_{n,s}(t)}{NT}$$

s.t.

$$\begin{aligned}
&\text{C1: } \sum_{s=1}^{S} \alpha_{m,s} \le \frac{C_s}{C_f}, \forall m \in [1,M], m \in \mathbb{N}^+ \\
&\text{C2: } \sum_{s=1}^{S} \beta_{k,s} \le \frac{C_v}{C_f}, \forall k \in [1,K], k \in \mathbb{N}^+ \\
&\text{C3: } \sum_{m=1}^{M} \eta_{m,n}(t) \le 1, \forall t \in \mathcal{T}, n \in [1,N], n \in \mathbb{N}^+ \\
&\text{C4: } \sum_{k=1}^{K} \lambda_{k,n}(t) \le 1, \forall t \in \mathcal{T}, n \in [1,N], n \in \mathbb{N}^+ \\
&\text{C5: } \sum_{m=1}^{M} \eta_{m,n}(t) + \sum_{k=1}^{K} \lambda_{k,n}(t) \le 1 \\
&\text{C6: } \alpha_{m,s} \in \{0,1\}, \beta_{k,s} \in \{0,1\} \\
&\text{C7: } \eta_{m,n}(t) \in \{0,1\}, \lambda_{k,n}(t) \in \{0,1\}
\end{aligned} \tag{9}$$

where C1–C2 guarantee that the capacity of RSAPs and CVs are not exceeded. C3–C5 limit an RV to select at most one communication link (V2V or V2I) per time slot to obtain content. C6 denotes the content caching indicators for RSAPs and CVs. C7 denotes the content delivery communication links indicators for RSAPs and CVs.

IV. Collaborative Caching Scheme Based on DRL

In this section, we propose a DRL-based collaborative caching placement and content delivery latency optimization algorithm to solve formula (9). We focus on the caching policy for multiple RSAPs and CVs deployed around the popular intersections in Fig.1. First, we are concerned with the mobility and preferences of vehicles. Then the deep Q-networks (DQN) algorithm is proposed to optimize the proactive caching policy of RSAPs. Furthermore, the CVs are introduced to help achieve optimized collaborative caching decisions.

1. System state space

Because of the limited budget of content providers, to improve the stability of V2V communication links, for more stable content delivery communication between CVs and RVs, we consider the position and speed of vehicles, calculate the Euclidean distance between vehicles, and choose the CV that is nearest and moves in the same lane to the RV, which is the most suitable for content delivery.

At the beginning of each time slot, RVs request content from RSAPs or CVs through V2I or V2V links, which vehicular mobility, communication transmission, and caching capacity have to be considered. The state space is denoted by $S_{t,n}$, it can be expressed as

$$\begin{aligned}
S_{t,n} = \big\{&t, L_{\text{rv}}(t), L_{\text{cv}}(t), C_{\text{s}}, C_{\text{v}}, \mathcal{Q}, \alpha_{m,s}, \beta_{k,s}, \\
&\eta_{m,n}(t), \lambda_{k,n}(t), R_{m,n}(t), R_{k,n}(t), \tau_{\text{cpu}}\big\}
\end{aligned} \tag{10}$$

The system state includes: content request time slot t, location sets of RVs and CVs are denoted as $L_{\text{rv}}(t)$ and $L_{\text{cv}}(t)$ respectively, RSAPs cache capacity C_{s}, CVs cache capacity C_{v}, content request sequences of RVs $\mathcal{Q}$, RSAPs cache status $\alpha_{m,s}$, CVs cache status $\beta_{k,s}$, content delivery status of V2I link $\eta_{m,n}(t)$, content delivery status of V2V link $\lambda_{k,n}(t)$, V2I transmission rate $R_{m,n}(t)$ and V2V transmission rate $R_{k,n}(t)$, latency of content delivery from the CPU τ_{cpu}.

2. System action space and state transition probability

At each time slot, vehicles can get contents from RSAPs, CVs, or the CPU. To reduce the communication loads, the CPU can decide which content should be precached in RSAPs and CVs based on the vehicles' locations and preferences to help improve the content delivery average latency. The action space $A_{t,n}$ can be expressed as

$$A_{t,n} = \{\alpha_{m,s}, \beta_{k,s}, \eta_{m,n}(t), \lambda_{k,n}(t)\} \tag{11}$$

where $\alpha_{m,s}$ represents the content caching indicator of RSAPs, $\beta_{k,s}$ represents the content caching indicator of CVs. $\eta_{m,n}(t)$ and $\lambda_{k,n}(t)$ indicates which communication links is used for content delivery.

When the current state $S_{t,n}$ and action $A_{t,n}$ are determined, the state transition probability from state $S_{t,n}$ to the next state $S_{t+1,n}$ can be expressed as $\Pr = (S_{t+1,n}|S_{t,n}, A_{t,n})$.

3. Reward

We denote the feedback of an action taken by the agent as $R_{t,n}$. It is a mapping from state space to action space, i.e., $S_{t,n} \times A_{t,n}$. Depending on the content requests of RVs, the system can take different actions, which result in different content delivery latency. For one RV:

1) If the nearest CV cache the content, the content is delivered to RV via V2V link.

2) If the nearest CV do not cache the content but the RSAP cache the content, the content is delivered via V2I link.

3) If neither CV nor RSAP cache the content, the content is delivered by the CPU, resulting in a higher delivery latency.

Therefore, we consider the reduction of content delivery latency as our reward, which can be expressed as

$$\gamma_{t,n} = \begin{cases} -D_{k,n}(t), & \text{if } \lambda_{k,n}(t) = 1 \\ -D_{m,n}(t), & \text{if } \eta_{m,n}(t) = 1 \\ -\tau_{\text{cpu}}, & \text{otherwise} \end{cases} \tag{12}$$

where τ_{cpu} is a constant.

When the caching policies of RSAPs and CVs are determined, the reward of completing the content delivery of all RVs under all content request time slots can be expressed as

$$R_{t,n} = \frac{\sum_{t=1}^{T}\sum_{n=1}^{N}\gamma_{t,n}}{NT} \tag{13}$$

4. Collaborative caching algorithm based on DQN

At time slot t, the current state is $S_{t,n}$, if action $A_{t,n}$ is taken, the state transfers into the next state $S_{t+1,n}$. The agent will be rewarded with $R_{t,n}$ accordingly. The collaborative caching policy can be expressed as a mapping from $S_{t,n}$ to $A_{t,n}$, i.e., $\pi : S \to A$.

We use the cumulative expected discount reward to represent the value of the current state, which can evaluate the long-term impact of the caching policy. In the initial state S_0, the cumulative expected discount rewards can be formulated as

$$V_\pi(S_{t,n}) = E\left[\sum_{t=1}^{\infty}\varphi R(S_{t,n}, A_{t,n})|S_{t,n} = S_0\right] \tag{14}$$

where $E[\cdot]$ is the expectation for a long time, and $\varphi \in \{0,1\}$ is the discount factor. It indicates the importance of predicting the expected payoff.

Our goal is to find an optimal policy π^*, which maximizes the total content delivery rewards. Then we obtain the optimal action of $S_{t,n}$ under the optimal policy $V_{\pi^*}(S_{t,n})$. It can be expressed as

$$\begin{aligned} V_{\pi^*}(S_{t,n}) = \max_{A_{t,n}}\Big[&R(S_{t,n}, A_{t,n}) \\ &+ \varphi\sum_{S_{t+1,n}}\Pr(S_{t+1,n}|S_{t,n}, A_{t,n})V_{\pi^*}(S_{t+1,n})\Big] \end{aligned} \tag{15}$$

We decompose a policy into multiple actions and obtain the Q-value function by an action $Q(S_{t,n}, A_{t,n})$, which is stored in the Q-table. Q-value is defined as

$$\begin{aligned} &Q_\pi(S_{t,n}, A_{t,n}) \\ &\quad= R(S_{t,n}, A_{t,n}) \\ &\quad\quad+ \varphi\sum_{S_{t+1,n}}\Pr(S_{t+1,n}|S_{t,n}, A_{t,n}t)V_{\pi^*}(S_{t+1,n}) \end{aligned} \tag{16}$$

We can express the relationship between $V_{\pi^*}(S_{t,n})$ and $Q_\pi(S_{t,n},A_{t,n})$ as $V_{\pi^*}(S_{t,n}) = \max\limits_{A_{t,n}} Q_\pi(S_{t,n}, A_{t,n})$. To obtain the maximum Q-value, an ε-greedy strategy is used in [41]. During the exploration phase, the agent executes random action with probability $\varepsilon\in\{0,1\}$. In the exploitation phase, the agent executes the action with probability $1-\varepsilon$, which has the highest estimated Q-value as

$$\begin{aligned} &Q_\pi(S_{t,n}, A_{t,n}) \\ &\quad= Q(S_{t,n}, A_{t,n}) + \varepsilon\Big[R(S_{t,n}, A_{t,n}) \\ &\quad\quad+ \varphi\max_{A_{t+1,n}} Q_\pi(S_{t+1,n}, A_{t+1,n}) - Q(S_{t,n}, A_{t,n})\Big] \end{aligned} \tag{17}$$

However, there is correlation among Q-values, which is likely to cause overestimation problem. Therefore, we propose a collaborative caching algorithm based on DQN. DQN combines Q-learning method with DL by adding deep neural networks (DNN), experience replay memory, and target Q-network. The weights of the DNN θ are used to approximate the optimal Q-value. The experience replay is used for learning, and the target network is used to eliminate the correlation between data and avoid divergence. To reduce the overestimation of Q-values, the agent should find the action that correspond to the maximum estimated Q-value, which can be expressed as

$$A_{t+1,n}^{\max} = \arg\max_{A_{t+1,n}} Q(S_{t+1}, A_{t+1,n}, \theta) \tag{18}$$

Then the action is taken to calculate the target Q-value in the target DQN. The Q-value of the target Q-network is expressed as

$$Y_{t,n} = R(S_{t,n}, A_{t,n}) + \xi Q(S_{t+1,n}, A_{t+1,n}^{\max}, \overline{\theta}) \tag{19}$$

where ξ is the learning rate of DQN.

Our goal is to make the estimated network approximate the target network gradually, and the loss function between them is formulated as

$$\text{Loss}(\theta_{t,n}) = E[Y_{t,n} - Q(S_{t,n}, A_{t,n}, \theta)]^2 \tag{20}$$

The stochastic gradient descent method is used to solve this problem [41], [42]. Our proposed collaborative caching algorithm based on DQN is shown in Algorithm 1.

Algorithm 1 can be divided into three phases. The first phase is the initialization of parameters. In the second phase, we use DQN algorithm to make the proactive caching decisions of RSAPs to achieve content delivery average latency minimization, which can effectively solve the problem of various user preferences and help improve the caching efficiency. In the third phase, we coordinate the CVs and RSAPs to implement collaborative caching. As we have already known the caching decisions of RSAPs through the second phase, which are different from RSAPs, CVs will precache their own preferred contents so that they can get the required content from themselves. Hence, caching decisions of CVs depend on their own preferences. Thereafter, RVs send their content requests to the CPU, which controls the V2X links and decides which RSAP or CV, who has cached the requested content, delivers the content to RVs to get the minimum latency.

Algorithm 1 DQN-based collaborative caching algorithm

Phrase 1 Initialization

1: Generate RVs and CVs (including number, trajectory and speed);
2: Initialize cache state and cache capacity of RSAPs and CVs;
3: Initialize content request sequences $\mathcal{Q}$ according to equation (7);
4: Initialize channel gains of V2I and V2V links;
5: Initialize the DQN network;

Phrase 2 RSAPs proactive caching based on DQN

6: **for** each episode **do**
7: Initialize cache state S_0 randomly;
8: **for** $t=\{1,2,\ldots,T\}$ **do**
9: Select an action A_t randomly with probability ε;
10: Otherwise choose A_t according to equation (18);
11: Execute action A_t and observe reward R_t and S_{t+1};
12: Store (S_t, A_t, R_t, S_{t+1}) in the replay memory;
13: Update the Q-network to reduce the loss function in equation (20);
14: **end for**
15: **end for**
16: Output the optimal caching action $A_{\max}$ of RSAPs;

Phrase 3 Collaborative caching scheme

17: Obtain the caching actions of RSAPs from Phase 2;
18: Obtain the caching decisions of CVs according to their own content preferences;
19: **for** each RV **do**
20: **for** $t=\{1,2,\ldots,T\}$ **do**
21: RVs request content from the CPU;
22: CPU arrange a communication link (V2V or V2I) which have the shortest content delivery latency;
23: **end for**
24: **end for**

V. Performance Evaluation

In this section, we evaluate the performance of our proposed algorithm by comparing with other algorithms. It is demonstrated that our proposed algorithm can effectively reduce the content delivery average latency and improve the content cache hit ratio.

1. Simulation parameters and settings

We consider an urban intersections scenario with 24 RSAPs, one CPU, and the cloud servers distributed in a $600\ \text{m} \times 800\ \text{m}$ area as shown in Fig.1. RSAPs are evenly distributed in the scenario. Each road has two lanes and RSAPs' communication coverage is 150 m. There is some overlapping coverage at the center of the intersection. In addition, the area is covered by one CPU, which is connected to the cloud server. We generate vehicular movements and trajectories using SUMO [43], which can add traffic lights to simulate vehicular waiting and steering situations, and it conforms to the Markovian movement model. The channel gain model is set according to [44]. We use Python to write algorithm for performance evaluation. The cloud server stores 10 contents, each content is 2 MB in size. We list the important simulation parameters in Table 1.

Table 1. The simulation parameters

Notation	Definition	Values
$h_{\rm s}$	The antenna heights of RSAPs	5 m
$h_{\rm v}$	The antenna heights of RVs and CVs	1.5 m
$R_{\rm cpu}$	The coverage radius of CPU	500 m
$R_{\rm s}$	The coverage radius of RSAPs	150 m
$B_{\rm s}, B_{\rm v}$	The bandwidth of V2I and V2V links	1 MHz
$P_{\rm s}$	The transmission power of V2I links	23 dBm
$P_{\rm v}$	The transmission power of V2V links	15 dBm
v_n, v_k	The speed of RVs and CVs	[0, 40] km/h
C_f	The size of each content	2 MB
S	The number of contents	10
λ	Zipf exponent	0.6
$C_{\rm s}$	Cache capacity of RSAPs	10 MB
$C_{\rm v}$	Cache capacity of CVs	6 MB
$\tau_{\rm cpu}$	The delivery latency from the CPU	80 ms
Episode	Training steps of DQN	12000
ξ	Learning rate of DQN	0.0001
$G_{m,n}, G_{k,n}$	The channel gain of V2I and V2V links	Ref.[38]

We compare our proposed algorithm with the following algorithm:

1) Popularity-based caching scheme [45]: RSAPs and CVs cache the most popular content at first, and there are a situation that different RSAPs cache the same contents.

2) Random caching scheme: RSAPs and CVs utilize their storage space to cache contents randomly.

3) Non-collaborative caching scheme: only RSAPs and the CPU can cache and deliver content without vehicular collaborative caching.

2. Simulation results

The relationship between the number of episodes and the loss function during the training and testing phases of the DQN is shown in Fig.3. It can be noticed that at the beginning, the loss function increases with the number of episodes, which is due to the random selection of actions by DQN for training. After about 4000 training steps, the loss function converges to 0 and the system can obtain the caching policy with the smallest average delay.

We evaluate the performance of the proposed caching scheme using content delivery average latency and content cache hit ratio. Figs.4 and 5 show the content

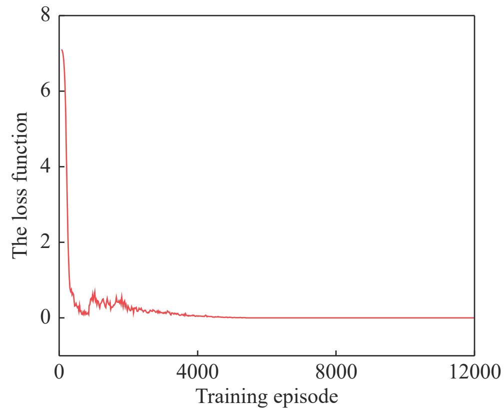


Fig. 3. The loss function vs. training episode.

delivery average latency and cache hit ratio for different caching schemes at different simulation times. The simulation results show that our proposed scheme greatly reduces the content delivery average latency and outperforms other schemes in terms of content caching hit ratio.

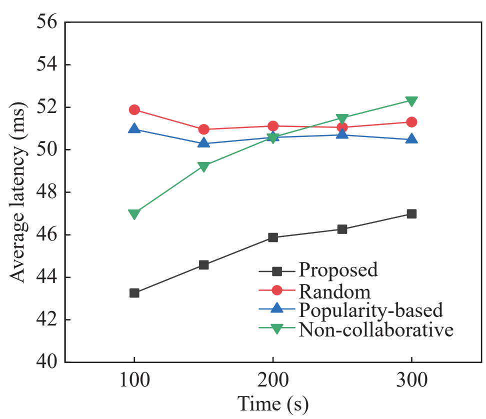


Fig. 4. Content delivery average latency vs. service time.

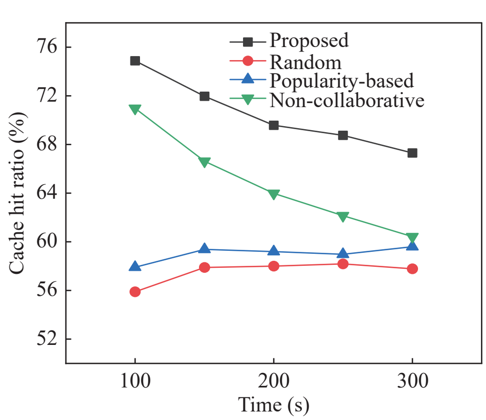


Fig. 5. Content cache hit ratio vs. service time.

In Figs.6 and 7, the performance comparison with different numbers of RVs in terms of average latency and cache hit ratio is shown. The increasing of RVs brings some pressure to the communication links of V2V and V2I, and there is a slight increase in content delivery latency and some decrease in the content cache hit ratio. However, our proposed optimization scheme still has the lowest latency and highest hit ratio than other three schemes.

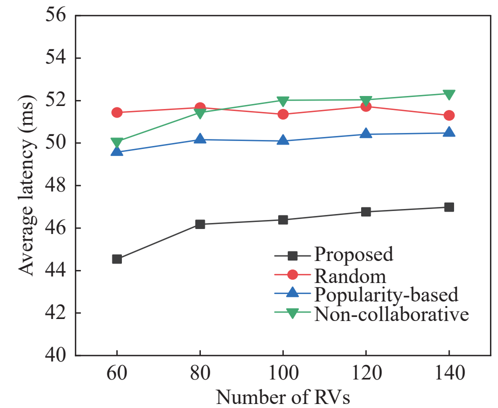


Fig. 6. Content delivery average latency vs. number of RVs.

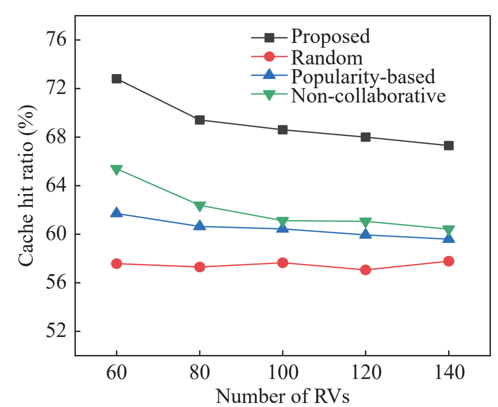


Fig. 7. Content cache hit ratio vs. number of RVs.

Figs.8 and 9 depict the relationship between the number of RSAPs and the content delivery average latency and cache hit ratio. We can see that as the number of RSAPs participating in caching increases, the content delivery average latency decreases and the cache hit ratio increases. The optimal content cache hit ratio of our proposed scheme indicates that vehicles can

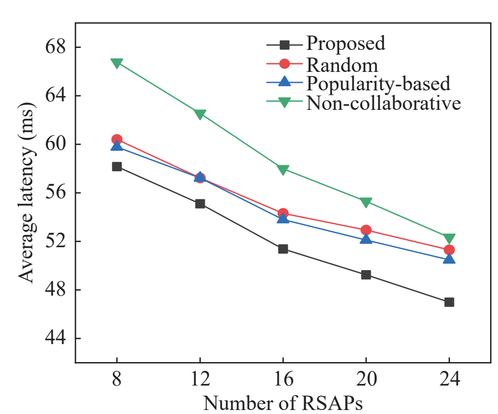


Fig. 8. Content delivery average latency vs. number of RSAPs.

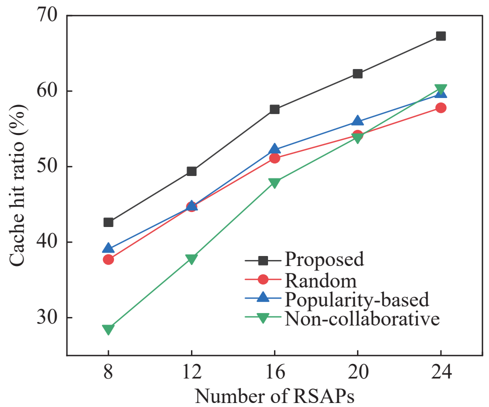


Fig. 9. Content cache hit ratio vs. number of RSAPs.

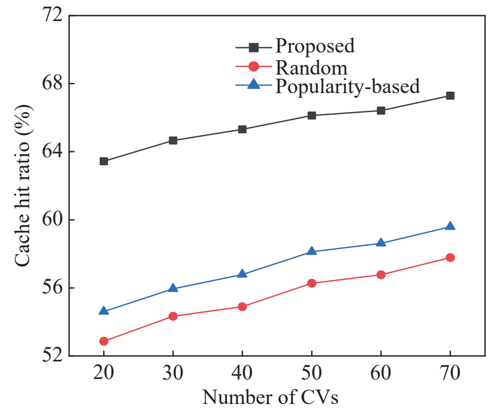


Fig. 11. Content cache hit ratio vs. number of CVs.

directly obtain their preferred contents with low latency through V2V and V2I communication without fetching content from the CPU with high latency.

Figs.10 and 11 show the effect of CVs' number on the content delivery average latency and cache hit ratio. Because the non-collaborative scheme only conducts RSAPs proactive caching, which is not affected by the number of CVs, we only compare the other three schemes. It can be seen that as the number of CVs increases, the content delivery average latency gradually decreases and the cache hit ratio becomes progressively larger. It proves that increasing CVs for collaborative caching is very effective and can help improve QoS.

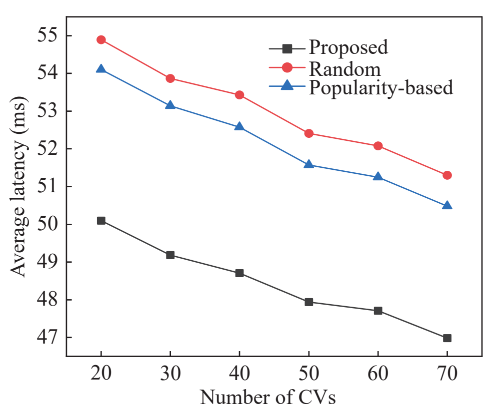


Fig. 10. Content delivery average latency vs. number of CVs.

In summary, the simulation results show that the proposed collaborative caching scheme can effectively improve the performance of the vehicular edge network assisted by cell-free massive MIMO, especially when considering vehicular mobility and preferences.

VI. Conclusions

In this paper, we proposed a collaborative caching scheme based on DQN algorithm in vehicular edge network assisted by cell-free massive MIMO system. We jointly consider vehicular mobility and content preferences to achieve minimum content delivery latency. Besides, our proposed collaborative caching scheme also provides a significant advantage in content cache hit ratio. In future work, we are going to improve the content request prediction to enhance the caching policy, we will also design a comprehensive V2V communication strategy between vehicles with more complicated trajectory.

References

[1] W. Bao, C. Wu, S. Guleng, *et al.*, "Edge computing-based joint client selection and networking scheme for federated learning in vehicular IoT," *China Communications*, vol.18, no.6, pp.39–52, 2021.

[2] Z. Y. Du, C. Wu, T. Yoshinaga, *et al.*, "Federated learning for vehicular internet of things: recent advances and open issues," *IEEE Open Journal of the Computer Society*, vol.1, pp.45–61, 2020.

[3] C. Wu, Z. Liu, D. Zhang, *et al.*, "Spatial intelligence toward trustworthy vehicular IoT," *IEEE Communications Magazine*, vol.56, no.10, pp.22–27, 2018.

[4] H. B. Zhou, W. C. Xu, J. C. Chen, *et al.*, "Evolutionary V2X technologies toward the internet of vehicles: challenges and opportunities," *Proceedings of the IEEE*, vol.108, no.2, pp.308–323, 2020.

[5] J. H. Zhao, Q. P. Li, Y. Gong, *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.7944–7956, 2019.

[6] J. Y. Feng, Z. Liu, C. Wu, *et al.*, "Mobile edge computing for the internet of vehicles: offloading framework and job scheduling," *IEEE Vehicular Technology Magazine*, vol.14, no.1, pp.28–36, 2019.

[7] S. Z. Chen, Q. Li, Y. Wang, *et al.*, "C-V2X equipment identification management and authentication mechanism," *China Communications*, vol.18, no.8, pp.297–306, 2021.

[8] S. Z. Chen, J. J. Hu, Y. Shi, *et al.*, "A vision of C-V2X: technologies, field testing, and challenges with Chinese development," *IEEE Internet of Things Journal*, vol.7, no.5, pp.3872–3881, 2020.

[9] S. Gyawali, S. J. Xu, Y. Qian, *et al.*, "Challenges and solutions for cellular based V2X communications," *IEEE Communications Surveys & Tutorials*, vol.23, no.1, pp.222–255, 2021.

[10] J. H. Zhao, S. J. Ni, L. H. Yang, *et al.*, "Multiband cooperation for 5G HetNets: A promising network paradigm," *IEEE Vehicular Technology Magazine*, vol.14, no.4, pp.85–93, 2019.

[11] S. Z. Chen, S. H. Sun, and S. L. Kang, "System integration of terrestrial mobile communication and satellite communication—the trends, challenges and key technologies in B5G and 6G," *China Communications*, vol.17, no.12, pp.156–171, 2020.

[12] Y. Shi, S. Z. Chen, and X. Xu, "MAGA: A mobility-aware computation offloading decision for distributed mobile cloud computing," *IEEE Internet of Things Journal*, vol.5, no.1, pp.164–174, 2018.

[13] P. C. Zhu, J. Xu, J. M. Li, *et al.*, "Learning-empowered privacy preservation in beyond 5G edge intelligence networks," *IEEE Wireless Communications*, vol.28, no.2, pp.12–18, 2021.

[14] H. Feng, S. T. Guo, L. Yang, *et al.*, "Collaborative data caching and computation offloading for multi-service mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol.70, no.9, pp.9408–9422, 2021.

[15] X. T. Ma, J. H. Zhao, and Y. Gong, "Joint scheduling and resource allocation for efficiency-oriented distributed learning over vehicle platooning networks," *IEEE Transactions on Vehicular Technology*, vol.70, no.10, pp.10894–10908, 2021.

[16] J. H. Zhao, X. K. Sun, Q. P. Li, *et al.*, "Edge caching and computation management for real-time internet of vehicles: An online and distributed approach," *IEEE Transactions on Intelligent Transportation Systems*, vol.22, no.4, pp.2183–2197, 2021.

[17] W. Gao, C. Wu, L. Zhong, *et al.*, "Communication resources management based on spectrum sensing for vehicle platooning," *IEEE Transactions on Intelligent Transportation Systems*, vol.24, no.2, pp.2251–2264, 2023.

[18] Y. J. Liu, S. G. Wang, Q. L. Zhao, *et al.*, "Dependency-aware task scheduling in vehicular edge computing," *IEEE Internet of Things Journal*, vol.7, no.6, pp.4961–4971, 2020.

[19] R. A. Dziyauddin, D. Niyato, N. C. Luong, *et al.*, "Computation offloading and content caching and delivery in vehicular edge network: A survey," *Computer Networks*, vol.197, article no.108228, 2021.

[20] F. Khandaker, S. Oteafy, H. S. Hassanein, *et al.*, "A functional taxonomy of caching schemes: towards guided designs in information-centric networks," *Computer Networks*, vol.165, article no.106937, 2019.

[21] Z. J. Nan, Y. J. Jia, Z. Ren, *et al.*, "Delay-aware content delivery with deep reinforcement learning in internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol.23, no.7, pp.8918–8929, 2022.

[22] X. J. Xia, P. C. Zhu, J. M. Li, *et al.*, "Joint user selection and transceiver design for cell-free with network-assisted full duplexing," *IEEE Transactions on Wireless Communications*, vol.20, no.12, pp.7856–7870, 2021.

[23] J. M. Li, M. M. Liu, P. C. Zhu, *et al.*, "Impacts of asynchronous reception on cell-free distributed massive MIMO systems," *IEEE Transactions on Vehicular Technology*, vol.70, no.10, pp.11106–11110, 2021.

[24] C. W. Wang, D. H. Deng, W. D. Wang, *et al.*, "UAV assisted communication and resource scheduling in cell-free massive MIMO based on deep reinforcement learning approach," *Journal of Electronics & Information Technology*, vol.44, no.3, pp.835–843, 2022. (in Chinese)

[25] F. Ye, J. M. Li, P. C. Zhu, *et al.*, "Fingerprint-based covariance matrix estimation for cell-free distributed massive MIMO systems," *IEEE Wireless Communications Letters*, vol.11, no.2, pp.416–420, 2022.

[26] Z. Zhang, C. H. Lung, M. St-Hilaire, *et al.*, "Smart proactive caching: empower the video delivery for autonomous vehicles in ICN-based networks," *IEEE Transactions on Vehicular Technology*, vol.69, no.7, pp.7955–7965, 2020.

[27] Z. Su, Y. L. Hui, Q. C. Xu, *et al.*, "An edge caching scheme to distribute content in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol.67, no.6, pp.5346–5356, 2018.

[28] Y. L. Hui, Z. Su, T. H. Luan, *et al.*, "Content in motion: An edge computing based relay scheme for content dissemination in urban vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol.20, no.8, pp.3115–3128, 2019.

[29] S. Akhavan Bitaghsir and A. Khonsari, "Cooperative caching for content dissemination in vehicular networks," *International Journal of Communication Systems*, vol.31, no.8, article no.e3534, 2018.

[30] J. Z. Zhou, X. Zhang, and W. B. Wang, "Social-aware proactive content caching and sharing in multi-access edge networks," *IEEE Transactions on Cognitive Communications and Networking*, vol.6, no.4, pp.1308–1319, 2020.

[31] S. S. Musa, M. Zennaro, M. Libsie, *et al.*, "Mobility-aware proactive edge caching optimization scheme in information-centric IoV networks," *Sensors*, vol.22, no.4, article no.1387, 2022.

[32] Y. AlNagar, R. H. Gohary, S. Hosny, *et al.*, "Mobility-aware edge caching for minimizing latency in vehicular networks," *IEEE Open Journal of Vehicular Technology*, vol.3, pp.68–84, 2022.

[33] L. Yao, Y. Q. Wang, X. Wang, *et al.*, "Cooperative caching in vehicular content centric network based on social attributes and mobility," *IEEE Transactions on Mobile Computing*, vol.20, no.2, pp.391–402, 2021.

[34] Z. X. Yu, J. Hu, G. Y. Min, *et al.*, "Mobility-aware proactive edge caching for connected vehicles using federated learning," *IEEE Transactions on Intelligent Transportation Systems*, vol.22, no.8, pp.5341–5351, 2021.

[35] G. H. Qiao, S. P. Leng, S. Maharjan, *et al.*, "Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol.7, no.1, pp.247–257, 2020.

[36] L. Hou, L. Lei, K. Zheng, *et al.*, "A Q-learning-based proactive caching strategy for non-safety related services in vehicular networks," *IEEE Internet of Things Journal*, vol.6, no.3, pp.4512–4520, 2019.

[37] Z. L. Ning, K. Y. Zhang, X. J. Wang, *et al.*, "Joint computing and caching in 5G-envisioned internet of vehicles: A deep reinforcement learning-based traffic control system," *IEEE Transactions on Intelligent Transportation Systems*, vol.22, no.8, pp.5201–5212, 2021.

[38] M. Zhang, S. Wang, and Q. Gao, "A joint optimization scheme of content caching and resource allocation for internet of vehicles in mobile edge computing," *Journal of Cloud Computing*, vol.9, no.1, article no.33, 2020.

[39] L. Breslau, P. Cao, L. Fan, *et al.*, "Web caching and Zipf-

like distributions: evidence and implications," in *Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320)*, New York, NY, USA, pp.126–134, 1999.

[40] X. W. Wu, X. H. Li, J. Li, *et al.*, "Caching transient content for IoT sensing: multi-agent soft actor-critic," *IEEE Transactions on Communications*, vol.69, no.9, pp.5886–5901, 2021.

[41] Q. Y. Chen, W. Z. Zhao, L. Li, *et al.*, "ES-DQN: A learning method for vehicle intelligent speed control strategy under uncertain cut-in scenario," *IEEE Transactions on Vehicular Technology*, vol.71, no.3, pp.2472–2484, 2022.

[42] H. Zhou, T. Wu, H. J. Zhang, *et al.*, "Incentive-driven deep reinforcement learning for content caching and D2D offloading," *IEEE Journal on Selected Areas in Communications*, vol.39, no.8, pp.2445–2460, 2021.

[43] P. A. Lopez, M. Behrisch, L. Bieker-Walz, *et al.*, "Microscopic traffic simulation using SUMO," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI, USA, pp.2575–2582, 2018.

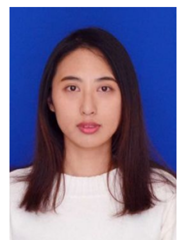
[44] L. Liang, H. Ye, and G. Y. Li, "Spectrum sharing in vehicular networks based on multi-agent reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol.37, no.10, pp.2282–2292, 2019.

[45] S. Hassan, I. U. Din, A. Habbal, *et al.*, "A popularity based caching strategy for the future Internet," in *2016 ITU Kaleidoscope: ICTs for a Sustainable World (ITU WT)*, Bangkok, Thailand, pp.1–8, 2016.

WANG Chaowei was born in Sichuan Province in 1982. He received the Ph.D. degree from Beijing University of Posts and Telecommunications in 2010. He is currently an Associate Professor at the School of Electronic Engineering, Beijing University of Posts and Telecommunications. His research interests include wireless communications and IoT applications. (Email: wangchaowei@bupt.edu.cn)

WANG Ziye (corresponding author) received the B.S. degree in optoelectronic information science and engineering from Beijing University of Posts and Telecommunications in 2021, and she is currently pursuing the M.S. degree in electronics science and technology from Beijing University of Posts and Telecommunications. Her research interests include Internet of vehicles and mobile edge computing. (Email: wzy11140@bupt.edu.cn)

XU Lexi received the M.S. and Ph.D. degrees from Beijing University of Posts and Telecommunications, Beijing, China, and Queen Mary University of London, London, United Kingdom, in 2009 and 2013, respectively. He is a Senior Engineer at China Unicom and a China Unicom Delegate in ITU, ETSI, CCSA. His research interests include big data, self-organizing networks, satellite systems, radio resource management in communication systems. (Email: xulx29@chinaunicom.cn)



YU Xiaofei received the B.S. degree in electronics science and technology from Beijing University of Posts and Telecommunications, China, in 2018, and she is currently pursuing the Ph.D. degrees in electronics science and technology from Beijing University of Posts and Telecommunications. Her research interests include mobile edge computing and wireless communication. (Email: yuxiaofei@bupt.edu.cn)

ZHANG Zhi received the B.S. and Ph.D. degrees from Beijing University of Posts and Telecommunications, Beijing, China, in 1999 and 2004, respectively. He is currently a Professor and the Ph.D. Supervisor with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications. His research interests include key technology of mobile communications, digital signal processing, and communication system design. (Email:zhangzhi@bupt.edu.cn)

WANG Weidong was born in Inner Mongolia Autonomous Region in 1967. He received the Ph.D. degree from Beijing University of Posts and Telecommunications in 2002. He is currently a Full Professor of School of Electronic Engineering at Beijing University of Posts and Telecommunications. His research interests include communication system, radio resource management, IOT, and signal processing. (Email: wangweidong@bupt.edu.cn)