Joint Content Update and Transmission Resource Allocation for Energy-Efficient Edge Caching of High Definition Map

Gaofeng Hong, Bin Yang, Wei Su, Haoru Li, Zekai Huang, Tarik Taleb

***Abstract*—Caching the high definition map (HDM) on the edge network can significantly alleviate energy consumption of the roadside sensors frequently conducting the operators of the traffic content updating and transmission, and such operators have also an important impact on the freshness of the received content at each vehicle. This paper aims to minimize the energy consumption of the roadside sensors and satisfy the requirement of vehicles for the HDM content freshness by jointly scheduling the edge content updating and the downlink transmission resource allocation of the Road Side Unit (RSU). To this end, we propose a deep reinforcement learning based algorithm, namely the prioritized double deep R-Learning Networking (PRD-DRN). Under this PRD-DRN algorithm, the content update and transmission resource allocation are modeled as a Markov Decision Process (MDP). We take full advantage of deep R-learning and prioritized experience sampling to obtain the optimal decision, which achieves the minimization of the long-term average cost related to the content freshness and energy consumption. Extensive simulation results are conducted to verify the effectiveness of our proposed PRD-DRN algorithm, and also to illustrate the advantage of our algorithm on improving the content freshness and energy consumption compared with the baseline policies.**

***Index Terms*—Vehicular Networks, High Definition Map, Edge Caching, Deep Reinforcement Learning, Content Update, Transmission Resource Allocation, Age of Information.**

I. Introduction

THE High Definition Map (HDM) is an essential tool to help autonomous vehicles make path planning and relative driving decision [1]- [3]. Generally, the HDM can be roughly divided into two layers, namely the static layer and the dynamic layer [9]. The static layer contains the road topology information stored in the remote cloud platform or pre-cached onboard, while the dynamic layer contains the real-time traffic condition of the specific road section requiring frequent information update from the roadside sensors. To reduce the file response latency and sensor energy consumption of the remote data transmission, a promising solution is to cache the dynamic layer files of the HDM at different network edges [4]- [8]. However, the frequent file update still brings high energy consumption of the roadside sensors due to the traffic condition perception and the data transmission. Actually, a large proportion of updates is unnecessary if the HDM files arriving the destined vehicles can meet the vehicular requirement on the freshness. Therefore, a new and dedicated research is deserved to explore how to relieve the energy consumption of the roadside sensors while keeping a relatively high file freshness.

The age of Information (AoI) is a promising metric to quantify the freshness of the dynamic contents, and is defined as the time elapsed beginning from the time when the content is generated from the source [10] [11]. The available works on AoI aim to improve system performances, which include the average/(peak) AoI minimization, trade-off between AoI and request latency, by optimizing various parameters such as content's AoI, energy consumption by information sources, and transmission bandwidth occupation. These works either use different mathematical optimization theories to deduce an optimal scheduling policies [12]- [27], or use learning-based methods to make real-time optimal scheduling decision [28]- [36] (see Related Works of Section II). The traditional mathematical optimization theories can efficiently deduce the feasible solution with relatively low algorithm complexity [37]- [40].

Note that most of the above mathematical optimization theories rely on the extra information about the models and the environment, which are difficult to implement in a practical scenario. The learning-based methods can overcome this disadvantage, they consider a more realistic case that the environment information is unknown, and obtains the optimal decision through the interaction with the environment. Most of the existing learning-based methods transform the proposed problem to a Markov decision process (MDP), and utilize model-based reinforcement learning (RL) (e.g., value iteration algorithm), or model-free RL methods (e.g., Deep Q-learning) to obtain optimal policy [41] [42]. The traditional Q-learning is an effective method to solve MDP problems [43], which utilizes a two-dimensional Q-table to evaluate

Manuscript received 2 March 2023; revised 6 July 2023 and 17 October 2023; accepted 15 November 2023. This work was also carried out in ICTFICIAL Oy. This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 2023JBZD001;in part by the Ministry of Education Innovation Group Joint Fund under Grant 8091B042222;in part by the National Natural Science Foundation of China under Grant 62372076;in part by the European Union's Horizon Europe Program for Research and Innovation through the aerOS Project under Grant 101069732; and in part by the Natural Science Project of Anhui under Grants KJ2021ZD0128 and 2022XJZD12. The review of this article was coordinated by Prof. Bin Lin. (Corresponding authors: Wei Su; Bin Yang.)

G. Hong, W. Su, H. Li and Z. Huang are with the School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing, China. E-mail: honggf@bjtu.edu.cn, wsu@bjtu.edu.cn, 21120074@bjtu.edu.cn, 21125029@bjtu.edu.cn.

B. Yang is with the School of Computer and Information Engineering, Chuzhou University, Chuzhou, Anhui, China. E-mail: yangbinchi@gmail.com.

Tarik Taleb is with the Department of Information Technology and Electrical Engineering, University of Oulu, 90570 Oulu, Finland(e-mail: tarik.taleb@oulu.fi)

the system performance of actions taken in each state. However, when applying to a large-scale reinforcement learning (RL) problem, it will face the curse of dimensionality due to the large state or action space and becomes ineffective. Therefore, the combination of Q-learning and the deep neural network (DNN), which is called DQN, has been proposed to approximate the Q-function of state-action pairs and execute automatic learning under the large system state [44]. The goal of the natural DQN and its variants is to maximize the long-term discount reward by utilizing the deep neural network (DNN) as an approximation function to learn policy and state value. Their execution usually contains three processes, namely agent exploration and exploitation, offline training and online decision [45]. During the exploration and exploitation process, the agent interacts with the environment and performs the optimal actions according to the greedy policy. It will cache the experience in the replay buffer once it performs the relevant action. When the agent obtains enough training experiences from the environment, it starts to train the network model with the cached experiences. Once the network performance has met the certain requirements, the agent can run the trained DQN model online to make optimal decision based on the given MDP. The learning-based methods on AoI optimization mainly put an eye on minimizing AoI and the relative file update cost (such as energy consumption, transmission latency, etc.) by finding out optimal status update policies [28]- [30] [33]- [34] or transmission related optimizations [35] [36].

However, the status update policy and the relevant transmission optimization are considered separately in the existing works. In general, reasonable transmission resource allocation can reduce the number of the instant updating when the real-time AoI of the request cannot meet the user's requirement. This means that more transmission resource can be allocated to the user whose requested file's AoI is approaching to its AoI requirement threshold. Therefore, jointly scheduling the HDM content update and the transmission resource allocation in the dynamic edge caching system is a promising solution in reducing the file update times while keeping a relative high file freshness. In this paper, we investigate how to satisfy the vehicular AoI requirements while maintaining relatively low energy consumption of the battery-powered roadside sensors in the edge HDM caching scenario by jointly schedule the content update and the downlink transmission resource allocation. We propose a PRD-DRN algorithm, which combines the superiority of prioritized double deep Q-learning [47] [61] and R-learning [46]. In the proposed algorithm, the agent can interact with environment and execute the optimal scheduling action for maximizing the long-term average system reward without adjusting the discount factor.

The main contributions of this paper can be summarized as follows.

- We first model the joint scheduling problem of the content update and the downlink transmission resource allocation in the HDM edge-cached scenario as an MDP, which depicts the real-time AoI of the edge-cached content and the AoI difference of vehicles' requested files in non-uniform decision epochs. During each decision epochs, the system cost is derived as the sum of each vehicle's AoI-related cost, that is, AoI difference and sensor energy consumption brought by the content updating.
- We further propose a PRD-DRN algorithm to adaptively solve the scheduling problem when the vehicular request patterns and the dynamics of environment information are unknown. The PRD-DRN algorithm has the properties of the R-learning, which can obtain the maximal long-term average reward without adjusting the discount factor in the traditional DQN-based algorithm.
- Extensive simulation results are conducted to verify the PRD-DRN algorithm, and also to illustrate the improvement of the content freshness and energy consumption under the PRD-DRN algorithm compared with the baseline policies like the heuristics and the traditional DQN-based policies.

The rest of the paper is organized as follows. Section II summarizes the related works. Section III introduces our concerned network model and formulates the problem. In Section IV, we transform the scheduling problem into an MDP and propose the PRD-DRN to solve it. Extensive simulation results are provided in section V. Finally, Section VI concludes this paper.

II. Related Works

The available works on AoI can be classified into two categories: 1) deducing an optimal scheduling policy under a specific system by utilizing different mathematical optimization theories [12]- [27], and 2) utilizing learning-based methods to make real-time optimal scheduling decision [28]- [36]. The former category usually makes the assumption that the environment information (e.g., the request patterns of users) is the pre-defined mathematical models or statistics. The latter category considers a more realistic case that the environment information is unknown, and obtains the optimal decision through the interaction with the environment.

AoI has been initially investigated in cache updating systems to ensure the file freshness when the requested files arrive at users [12]- [18]. The works in [12] [13] [14] [18] [24] aim at exploring optimal file update policies to minimize the average/(peak) AoI of the edge cache system by considering other factors such as content popularity etc.. The works [15] [16] [17] optimize the transmission resource allocation in the edge caching system to realize a trade-off between the file freshness and the request latency.

Later, researchers find out that the pursuit of AoI minimization in environmental monitoring systems will inevitably increase the energy consumption of the sensors [34]. Relevant works focus on improving the AoI performance with lower energy consumption by utilizing different optimization theories [19]- [23] [25]- [27]. In [19], the authors optimize the update rate to avoid unnecessary updates and reduce the energy consumption of the sensors in a monitoring system. The authors of [20] explore the optimal online status update policies in the finite battery scenario and the infinite batter scenario, respectively. A renewal structure is also proposed in the finite battery scenario to give the order of the sensor charging and the status update. In [21], the authors prove that

the erasure status feedback is good for online timely updating when the available energy of the sensors is limited. The authors of [22] solve the AoI-Energy optimal problem from a communication perspective, where optimal transmission policies for two-hop networks have been investigated. In [23], the authors investigate optimal state update policies under different battery recharge models. In [25], the authors investigate the age-energy tradeoff of IoT monitoring systems and adopts a Truncated Automatic Repeat reQuest (TARQ) scheme. The authors of [26] focus on the average AoI and energy cost for Low Density Parity Check Code (LDPC) coded status update over Additive White Gaussian Noise (AWGN) and Rayleigh fading channels. By utilizing the renewal processes theory, the expressions of the average AoI and energy cost can be derived. In [27], the authors investigate how to realize a tradeoff between AoI and energy consumption over an error-prone channel by taking sleep and retransmission mode into consideration.

Recently, learning-based methods [28]- [36] have also been applied to optimize the AoI-related caching problems. These works usually transform the proposed problem to a Markov decision process (MDP), and utilize model-based reinforcement learning (RL) (e.g., value iteration algorithm), or model-free RL methods (e.g., Deep Q-learning) to obtain optimal policy. The authors of [31] and [32] investigate the achievable optimal information sampling and updating strategies which can minimize the AoI in the environment monitoring system. The authors in [28]- [30] [33]- [34] study the status update control problems under different scenarios where for the unknown energy-related information of the sensors, they propose various file update policies to optimize the AoI performance with low energy cost. Authors of [35] propose an optimal transmission mode selection scheme to realize a trade-off between AoI and energy consumption. In [36], the authors aim to minimize the AoI by controlling the network's actions on an unknown network topology and delay distribution. In [40], the authors investigate the age-energy tradeoff in fading channels with packet-based transmissions, and solve the specific problem by using Bellman optimal equations.

We summarize the characteristics of the aforementioned works in Table I.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. Network Model

As illustrated in Fig. 1, we consider a HDM dynamic layer edge caching scenario consisting of a single Road Side Unit (RSU), F traffic information acquisition roadside sensors and several vehicles in the RSU's coverage range. The RSU is equipped with H_b transmission resource blocks for downlink data transmission. Each roadside sensor is responsible for refreshing the relevant HDM file with the same size cached on the RSU. The vehicle and HDM dynamic file sets are denoted by $\mathcal{N} = \{1, 2, \ldots, N\}$ and $\mathcal{F} = \{1, 2, \ldots, F\}$, respectively. To deal with the real-time network changes brought by the vehicular requests, we consider a time-slotted system, where each time step t is slotted into equal-sized time slots based on the practical demand. At each time step, the RSU may receive the vehicular HDM file request, and then it will decide whether to pull the up-to-date states of HDM files from the relevant sensors based on the file AoI demands of vehicles. If the demands can be satisfied, the RSU will respond to the vehicular HDM file requests with its local cached files. Otherwise, it will provide the updated HDM files from these sensors for the vehicle. It is notable that the proposed model with a single edge node in this paper is also suitable for the multiple edge nodes with non-overlapping scenario, which is widely used in the previous works [9] [15] [17].

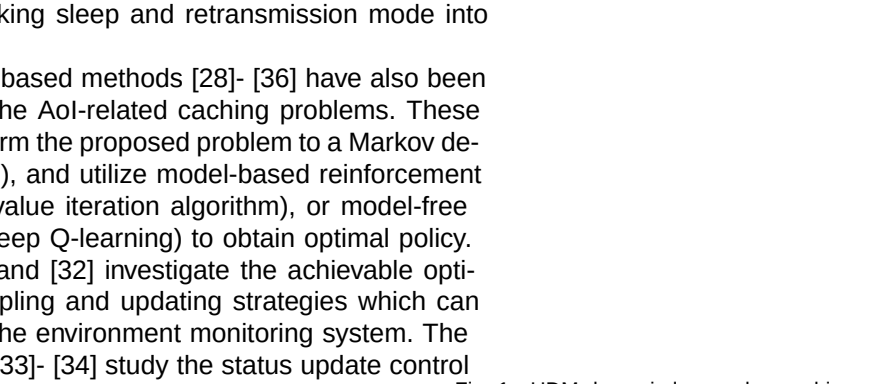

Fig. 1. HDM dynamic layer edge caching scenario.

In any time step t, the query detail for the request of vehicle n can be represented as a query profile $d_n(t) = \{d_n^1(t), d_n^2(t), \ldots, d_n^F(t)\}$, where $d_n^f(t) = \{0, 1\}$, and $f \in \{1, 2, \ldots, F\}$. $d_n^f(t) = 1$ represents the HDM file f has been requested by the vehicle n in the time step t, and $d_n^f(t) = 0$ otherwise. Meanwhile, we use an indicator $\theta_n(t)$ to represent whether vehicle n raises a request in the time step t. Then,

$$\theta_n(t) = \begin{cases} 0, & \sum_{f=1}^{F} d_n^f(t) = 0; \\ 1, & \text{Otherwise} \end{cases} \tag{1}$$

The RSU can obtain the query profiles $\mathcal{D}(t) = \{d_1(t), d_2(t), \ldots, d_N(t)\}$ of all the vehicles in the time step t, but it has no prior knowledge of the vehicular request arrival rates and the popularity of each cached HDM file.

Once if the RSU receives the query profiles $\mathcal{D}(t)$, it will make the HDM dynamic file update decision based on the requested HDM files and the relevant AoI demands. We use $\mathcal{U}(t) = \{u_1(t), u_2(t), \ldots, u_F(t)\}$ to represent the HDM file update decision in the time step t, where $u_f(t) \in \{0, 1\}$, $f \in \mathcal{F}$. $u_f(t) = 1$ represents that the RSU decides to refresh file f and pull the up-to-date states from the relevant sensor in the time step t, otherwise $u_f(t) = 0$. The RSU will select file f to refresh in the time step t based on the comparison between its real-time AoI on the RSU and the AoI demand in the query profile $\mathcal{D}(t)$. Notice that, when there is no query of file f in the query profiles $\mathcal{D}(t)$, file f may also be updated to reduce the transmission delay caused by the temporary request update only if there is available uplink transmission resources. Then,

TABLE I
THE SUMMARY OF AOI MINIMIZATION IN ENVIRONMENTAL MONITORING SYSTEMS

Optimization Objective	Reference	Status Update Policy	Optimal Transmission Schedule	Solution Taxonomy	
				Optimization Theory	Learning-based
AoI only	[12] [13] [14] [18]	X		X	
	[24]		X	X	
AoI & Latency	[15] [16] [17]		X	X	
AoI & Energy Consumption	[19]- [21] [23]	X		X	
	[22] [25]- [27]		X	X	
	[28]- [34]	X			X
	[35] [36] [40]		X		X

TABLE II
PARAMETERS

Parameters	Description
$d_n^f(t)$	Vehicle n request identifier for HDM file f.
$u_f(t)$	Update identifier for HDM file f.
B	The available bandwidth of each transmission resource block.
$H_n(t)$	The number of the consecutive resource blocks allocated to the vehicle n.
Y	The maximum number of updated HDM files in each time slot.
$_n(t)$	The max transmission rate from the vehicle n to the RSU.
$_n(t)$	The spectrum efficiency of vehicle n associated with the RSU.
$_{max}$	The maximum system AoI a file cached on the RSU can reach.
$_0^f(t)$	The real-time AoI value of HDM file f in the RSU.
$_n^f(t)$	The AoI value of the file requested by the vehicle n.
T_r^f	The time consumption for file f to be updated.
E_f	The energy consumed by the traffic sensing and the information uploading for each HDM file f update operation.
V_{max}	The AoI limitation for all the vehicular requests.
$_{n;f}^0(t)$	The AoI difference of the vehicular request on the RSU.
$_{n;f}(t)$	The AoI difference of the vehicular request onboard.
$\bar{}_n(t)$	The average AoI difference cost of all the HDM files vehicle n requested.
$_{AoI}(t)$	The AoI related cost during each time step t.
$P_{AoI}(t)$	The AoI relevant penalty during each time step t.
$P_E(t)$	The energy relevant penalty during each time step t.
$!_{AoI}$	The factor to nondimensionalize $P_{AoI}(t)$.
$!_E$	The factor to nondimensionalize $P_E(t)$.

the RSU responds the vehicular requests with its cached HDM files.

In our network model, we consider that the RSU is assigned with limited transmission resource blocks, which can be optimally allocated to the transmission from the RSU to each vehicle (V2I) based on service requirements of the vehicle's request. The max transmission rate $_n(t)$ from the vehicle n to the RSU in the time step t is given by:

$$_n(t) = BH_n(t)\ _n(t) \tag{2}$$

where B is the available bandwidth of each transmission resource block, $H_n(t)$ is the number of the consecutive resource blocks allocated to the vehicle n, $_n(t)$ is the spectrum efficiency of vehicle n associated with the RSU. Here, the unit of $_n(t)$ is KB per second. Thus, the file transmission latency from the RSU to the vehicle n can be determined as $\frac{l}{_n(t)}$ in the time step t. We consider a more realistic time-varying channel between each vehicle and the RSU. The channel is modeled as a finite-state Markov channel (FSMC) [52] without loss of generality. The spectrum efficiency is divided into Z levels. Let $Z = \{_0, _1, ..., _{Z-1}\}$ denote the state space of the spectrum efficiency, if $_0 < _n(t) < _1$, $_1$, if $_1 < _n(t) < _2$, ..., $_{Z-1}$, if $_n(t) \geq _{Z-1}$. In each time step, the $_n(t)$ can change from one state in the set Z to another with a certain transition probability.

Meanwhile, the total number of resource blocks allocated to downlink transmissions of N vehicles is no more than H_b, i.e.,

$$\sum_{n=1}^{N} H_n(t) \leq H_b, \tag{3}$$

where $H_n(t)$ represents the number of the resource blocks used by the vehicle n in the time step t.

As for the uplink transmission process of file update, we consider the update time of each HDM file keeps unchanged at different time steps due to the identical file size and transmission time. The update time can be depicted as $T_r = \{T_r^1, T_r^2, ..., T_r^F\}$, where T_r represents the file update time consumption set, T_r^f represents the time consumption for file f to be updated, $T_r^f < T(t); f \in \{1, 2, ..., F\}$. $T(t)$ denotes the duration of time step t. To avoid uplink channel congestion, the maximum number of updated files in each time slot is no more than a constant Y, i.e.,

$$\sum_{f=1}^{F} u_f(t) \leq Y \tag{4}$$

where $Y < F$. Particularly, an update operation for each file needs E_f unit energy consumed by the traffic sensing and the information uploading.

B. AoI Analysis

The real-time AoI value of the cached HDM files is of great importance for the RSU to conduct a file update decision. Since the AoI values of the same HDM file may be different in the RSU and vehicles at the same time slot, we analyse the real-time AoI value of cached HDM files on the RSU and the influence of file response latency on AoI when the requested file received by the vehicle.

We define a metric $_{max}$ which represents the maximum system AoI a file cached on the RSU can reach. For any

HDM le f in the RSU, its real-time AoI value $^{f}_{0}(t)$ can be expressed as

$$^{f}_{0}(t) = \begin{cases} T(t_n - 1); & o \quad u_f(t-1) = 1; \\ \min\{^{f}_{0}(t-1) + T(t-1); \ _{max}\}; & \text{otherwise.} \end{cases} \tag{5}$$

For a vehicle n, the transmission latency brought by the le respond process also increases the staleness of the information. Additionally, the instant le updating[1] also brings extra response latency to the request. Thus, the AoI value of the le f requested by the vehicle can be expressed as

$$^{f}_{n}(t) = \begin{cases} ^{f}_{0}(t-1) + \frac{l}{_n(t)}; & u_f(t) = 0 \\ T^{f}_{r} + \frac{l}{_n(t)}; & \text{Otherwise} \end{cases} \tag{6}$$

To ensure that the AoI of the requested HDM le meets the demand of each vehicle, we set an AoI limitation for all the requests in the time step as

$$^{f}_{n}(t) \quad ^{V}_{max} \tag{7}$$

where $^{V}_{max}$ $_{max}$. Fig. 2 illustrates the AoI variation of the HDM le cached on the RSU. To meet the communication needs of other vehicles, the RSU can allocate redundant downlink transmission resources to vehicles once if the requested le's AoI exceeds the threshold $^{V}_{max}$. This will avoid the occurrence of instant le update and reduce the update energy consumption. The le update decision and the downlink transmission resource allocation can be jointly optimized to realize a reduction in the update energy consumption at the cost of downlink transmission resource while ensuring a relatively low AoI experience of the vehicles.

Fig. 2. The AoI variation for an HDM le on the RSU.

C. Problem Formulation

Our objective is to minimize the average AoI experienced by the vehicles and also to reduce the extra sensor energy consumption caused by le update. We design a joint scheduling mechanism for the HDM le update and downlink transmission resource allocation.

To better characterize the satisfaction with the AoI of the requested HDM le, we de ne a new metric, namely AoI difference cost, as the gap between the real-time AoI value of the le and $^{V}_{max}$. The following equations (8) and (9) express the AoI difference of each vehicular request on the RSU and on-board, respectively.

$$^{0}_{n;f}(t) = ^{f}_{0}(t) \quad ^{V}_{max}; \tag{8}$$

$$_{n;f}(t) = ^{f}_{n}(t) \quad ^{V}_{max}: \tag{9}$$

As for a vehicle n, we use the average AoI difference cost $\bar{}_n(t)$ of all the HDM les it requested as the representative of its AoI satisfaction within time step t, which can be expressed as:

$$\bar{}_n(t) = \frac{1}{\sum_{f=1}^{F} d^{f}_{n}(t)} \sum_{f=1}^{F} {}_{n;f}(t) d^{f}_{n}(t) \tag{10}$$

According to the above analysis, the AoI related cost during each time step can be expressed as the weighted sum of each vehicle's average AoI difference cost, i.e.,

$$_{AoI}(t) = \sum_{n=1}^{N} {}_n \bar{}_n(t) \tag{11}$$

where $\sum_{n=1}^{N} {}_n = 1$, $_n \in [0,1]$, and the value of each $_n$ depends on the automatic driving level of the vehicle. Vehicle with higher automatic driving level possesses a higher value $_n$[2]. In this paper, we consider the case that each request only contains one HDM le to simplify the analysis.

Meanwhile, the total energy consumption of the roadside sensors in each time step can be expressed as $E(t) = \sum_{f=1}^{F} u_f(t) E_f$. Here, we denote $P_{AoI}(t)$ as the AoI relevant penalty brought by the average AoI difference cost, and denote $P_E(t)$ as the energy relevant penalty brought by the total energy consumption in each time step, respectively.

$$\begin{cases} P_{AoI}(t) = \frac{_{AoI}(t)}{^{V}_{max}} \\ P_E(t) = \frac{E(t)}{\sum_{f=1}^{F} E_f} \end{cases} \tag{12}$$

Therefore, the overall system cost in each time step t can be expressed as:

$$C_{tot}(t) = \ !_{AoI} P_{AoI}(t) + \ !_E P_E(t) \tag{13}$$

where $!_{AoI}$ and $!_E$ are used to nondimensionalize the function and can realize a tradeoff between the AoI relevant penalty and the energy relevant penalty in each time step. Since the AoI requirement of the requested HDM le is more important than the energy consumption of the roadside sensors. we consider $!_{AoI}$ is larger than $!_E$.

Based on the cost function (13), as the time T_{max} goes to in nity, the average cost of the requesting HDM le can be de ned as

$$C_{ave} = \lim_{T_{max} \to 1} \frac{1}{T_{max}} E\left(\sum_{t=0}^{T_{max}} C_{tot}(t)\right) \tag{14}$$

[1]It will occur when the current AoI of the requested le can not meet the vehicular AoI requirements.

[2]For vehicles with higher automatic driving level, receiving stale information has a greater impact on their judgment on driving behavior.

Our objective can be formulated as

$$\begin{aligned} \text{minimize} \quad & C_{ave} \\ s.t: \quad & {}^{f}_{0}(t) \quad {}^{V}_{max}; f \ 2 f 1; 2; :::; F g \\ & (3); (4); (5); (8) \end{aligned} \tag{15}$$

This is a nonlinear and nonconvex optimization problem. It is generally dif cult to solve such a problem. In the following section, we propose a DRL-based algorithm to solve it.

IV. DEEP REINFORCEMENT LEARNING-BASED ALGORITHM

This section rst formulates the HDM content update and downlink resource allocation process on the RSU as an MDP. Then, the PRD-DRN algorithm is further proposed to minimize the long-term average cost of the requesting HDM le by jointly optimizing the HDM content update and downlink transmission resource allocation.

A. MDP Model

The MDP in this paper is modeled as a 4-tuple $(S; A; P; R)$, i.e.,

State Space S: $s(t) = ({}_{0}(t); {}^{0}_{1}(t); :::; {}^{0}_{N}(t); {}_{1}(t); :::; {}_{N}(t))$ is de ned as the system state at time step t, which is composed of the real-time HDM le AoI value on the RSU ${}_{0}(t) = ({}^{1}_{0}(t); {}^{2}_{0}(t); :::; {}^{F}_{0}(t))$, the AoI difference of each vehicle on the RSU ${}^{0}_{n}(t) = ({}^{0}_{n;1}(t); {}^{0}_{n;2}(t); :::; {}^{0}_{n;F}(t)); n \ 2 f 1; 2; :::; N g$ and the spectrum ef ciency ${}_{n}(t)$ for each vehicle. The whole state space S at time step t is nite due to the constraint of the system maximum AoI value ${}_{max}$ and the FSMC model.

Action set A: $a(t) = (u_1(t); :::; u_F(t); H_1(t); :::; H_N(t))$ is de ned as the system action set in time step t, which represents the HDM le update decision and the downlink transmission resource allocation of the RSU.

State Transition Probability P: $P = S \quad A \quad S ! [0; 1]$ represents the distribution of the transition probability $P(s^0 j s; a)$ from the system state s to a new system state s^0 $(s; s^0 \ 2 \ S)$ when an action $a \ 2 \ A$ is chosen, which is largely effected by the real environment conditions, such as the HDM le request rate, the request popularity of each cached HDM le, etc.

Reward Function R: $S \quad A ! R$ maps a state-action pair to a value $R(s(t); A(t))$. Our objective in this paper is to minimize the long-term average cost $C_{ave}(t)$ given in equation (14) under the constrained conditions, and then we can de ne the reward function as $R(s(t); a(t)) = C_{ave}(t)$.

Here, we de ne the policy as an action $a \ 2 \ A$ that the RSU will execute by given a speci c system state $s \ 2 \ S$. Then, the objective function in (15) can be rewritten as:

$$\arg\max \lim_{T_{max} !1} \frac{1}{T_{max}} E \left[\sum_{t=0}^{T_{max}} R(s(t); a(t)) \ j \ s(0) \right] \tag{16}$$

B. PRD-DRN Algorithm

With the MDP model aforementioned, we can well characterize the effects of the diverse HDM AoI on vehicles under different le update actions based on the vehicular autonomous driving requirements. Here, we need to design an adaptive and ef cient HDM dynamic layer update strategy, which can proactively make le update decision in each state, so as to earn a higher reward by considering the long-term system performance.

In our scenario, the rewards obtained by the agent in different time steps are considered to have the same importance. Thus, this paper is to maximize the long-term average reward rather than the long-term discount reward. We modify the state value function $V(s)$ and the state-action value function $Q(s; a)$ by combining the idea used in the R-learning [46] as follows:

$$V_{\pi}(s) = E \left[\sum_{k=0}^{1} (R(s(t+k); (t+k)) \quad \bar{R}) \ j \ s(t) = s \right] \tag{17}$$

$$Q_{\pi}(s; a) = E \left[\sum_{k=0}^{1} (R(s(t+k); a(t+k)) \quad \bar{R}) \ j \ s(t) = s; a(t) = a \right] \tag{18}$$

where $\bar{R}$ is the long-term average reward of taking policy in state s, which can be written by:

$$\bar{R} = \lim_{k !1} \frac{1}{k+1} \sum_{k=0}^{1} R(s(t+k); (t+k)): \tag{19}$$

The optimal policy can be obtained by utilizing the Bellman Optimality Equation:

$$V (s) = \max_{a2A} Q (s; a): \tag{20}$$

The architecture of our DRL-based HDM dynamic layer update mechanism is presented in Fig. 3 and are the DNN parameters of the main network and the target network respectively. The agent interacts with the environment and observes the real-time system state. Based on the current state $s(t)$, the agent selects an action using the greedy strategy. Under such a strategy, the action $a \ R(s; a;)$ is selected with probability , and the action $a \ 2 \ A$ is selected with probability $(1 \quad)$, where $2 [0; 1])$. Notice that, the agent not only uses the previous experience to maximize current rewards, but also keeps exploration and exploitation to improve the $Q(s; a)$ and the $\bar{R}$. After the agent performs an action $a(t)$, the corresponding reward $R(s(t); a(t))$ can be obtained from environment, and the system state $s(t)$ transfers to $s(t+1)$. Thus, a new experience tuple $E(t) = (s(t); a(t); R(s(t); a(t)); s(t+1))$ is generated and will be cached in the experience replay buffer . Then, the former steps go into a loop to obtain enough experience in the replay buffer for the future training. Notice that, the oldest experience tuple will be discarded when the experience buffer is full.

As for the training procedure, we rst utilize a prioritized experience sampling scheme [47] to acquire a mini-batch

Fig. 3. The architecture of PRD-DRN

of the cached experience tuples $W = \{E_1, E_2, :::, E_{W_m}\}$ based on the pre-defined batch size W_m, where $E_j = (s_j ; a_j ; R(s_j ; a_j); s_j^0)$, $j \in \{1, 2, :::, W_m\}$. Unlike the random sampling, the prioritized experience sampling tends to replay experiences with high priority more frequently, which is measured by the magnitude of their temporal-difference (TD) error $_j$ defined as:

$$_j = R(s_j ; a_j) \quad \overline{R} + \max_{a_j^0} Q^0(s_j^0; a_j^0; \ ) \quad Q(s_j ; a_j ; \ ) \quad (21)$$

where $\overline{R}$ is the average reward of the cached experience in the replay buffer. Meanwhile, the sampling priority of each cached experience tuple E_j can be determined as $p_t = |_j|$. The sampling can be executed by utilizing a SumTree method [47], where the experience tuple with higher sampling priority has a higher probability of being selected.

After obtaining the batch of sampled experiences and the relevant TD errors of these experiences, the average reward $\overline{R}$ will be updated as:

$$\overline{R} = \overline{R} + \sum_{j=1}^{W_m} {}_j : \quad (22)$$

After obtaining the update of the sampled experiences set, the Q value of the target network Q_{target} of the PRD-DRN algorithm can be expressed as:

$$Q_{target}(s_j ; a_j) = R(s_j ; a_j) \quad \overline{R} + Q^0(s_j^0; \arg\max_a (s_j^0; a; \ ); \ ): \quad (23)$$

Meanwhile, the main network and the target network can be trained by minimizing the loss function $L( )$, which can be expressed as

$$L( ) = E[(Q_{target}(s_j ; a_j) \quad Q(s_j ; a_j ; \ ))^2]: \quad (24)$$

In this paper, we use the stochastic gradient descent (SGD) method to update the DNN parameter iteratively as equation (24):

$$^0 = \ + \ r \ L( ) \quad (25)$$

where is the learning rate. The parameter of the main network is updated every step while the parameter of the target network will be updated every steps. Then, $_t = _{t \ i}$.

The pseudo code in Algorithm 1 shows the details of the proposed PRD-DRN algorithm. The replay buffer and the parameters of the main network and the target network will be initialized at the beginning of the algorithm. During each episode, the environment needs to be reset at first. Then, the agent starts to explore the environment. For T_{pre} loops. For a given state $s(t)$, the agent selects an action $a(t)$ by utilizing the -greedy method. With the -greedy method, the agent tends to take a random action from the action set at the beginning of the iteration since it doesn't know much about the environment. After executing multiple iterations, the agent becomes more aware of the environment, and it will select the action with the maximum Q-value with a higher probability. The immediate reward $R(s(t); a(t))$ and the following state $s(t + 1)$ of $s(t)$ can be obtained based on the selected action

Algorithm 1: PRD-DRN Algorithm

Input: exploration rate ε, decay factor ε_0, replay buffer size $|\mathcal{M}|$, state $\mathcal{S}$, pre-training step T_{pre}, training episode T_{tran}, learning rate α and β

Initialize model parameters θ and θ^-, $\theta^- = \theta$;
Initialize average reward $\bar{R} = 0$;
Initialize $m, n = 0$;
for $n \leq T_{tran}$ **do**
  Initialize environment;
  Initialize $m = 0$;
  while True **do**
    Action Selection;
    Given the state $s(t)$;
    Output the corresponding $Q(s(t), a; \theta)$ of actions;
    Select action $a(t)$ with probability ε;
    Select action $a(t) = \max_a Q(s(t), a; \theta)$ with probability $(1 - \varepsilon)$;
    Replay Buffer Refreshing;
    Execute $a(t)$, obtain the reward $R(s(t), a(t))$ and the following state $s(t+1)$, append the new experience tuple $(s(t), a(t), R(s(t), a(t)), s(t+1))$ to $\mathcal{M}$;
    $m = m + 1$;
    if $m \geq T_{pre}$ **then**
      break;
    end
  end
  Training;
  Sample a mini-batch $\mathcal{W}$ from $\mathcal{M}$ by utilizing the prioritized experience replay method in [47];
  Update the average reward $\bar{R}$ with the equation (19);
  Update parameters of the main network by minimizing the loss function $L(\theta)$ value with SGD;
  Update parameters θ^- of the target network with each i steps;
  Record the model parameters and θ;
  $n = n + 1$;
  $\varepsilon = \varepsilon + \varepsilon_0$;
end

$a(t)$. Then, the agent can get a corresponding experience tuple $E(t) = (s(t), a(t), R(s(t), a(t)), s(t+1))$ and cache the tuple in the experience replay buffer $\mathcal{M}$ for the subsequent training. So far, the training of the network model starts and a mini-batch will be sampled from the experience replay buffer $\mathcal{M}$ by utilizing the prioritized experience replay method in [47]. The loss function will be minimized by the SGD procedure to update network parameters until it is converged. The exploration rate is set to 0 initially and increased by ε_0 during each training episode.

C. Algorithm Complexity Analysis

The time complexity of the Arti cial Neural Network based algorithm can be deduced based on its number of network neurons [48]. Assuming that a fully-connected network has x_I input neurons, x_O output neurons and H hidden layers with x_h neurons each layer $h \in \{2, 3, \ldots, H\}$, the time complexity can be expressed as $O(x_I x_1 + x_O x_H + \sum_{h=1}^{H-1} x_h x_{h+1})$ [49]. Meanwhile, the time complexity of the SumTree method is $O(\log|\mathcal{M}|)$ [47]. Thus, for the proposed PRD-DRN algorithm, the time complexity of each episode can be determined as

$$O(|\mathcal{S}| x_1 + |\mathcal{A}| x_H + \sum_{h=1}^{H} x_h x_{h+1} + \log|\mathcal{M}|) \quad (26)$$

where $|\mathcal{S}|$ and $|\mathcal{A}|$ are the dimensions of the state and action space, respectively. Combining with the network model proposed in this paper, the value $|\mathcal{S}|$ and $|\mathcal{A}|$ can be deduced as:

$$\begin{aligned} |\mathcal{S}| &= (1 + N)F + N; \\ |\mathcal{A}| &= F + N. \end{aligned} \quad (27)$$

We know that the training process of a DRL is extremely time-consuming [50]. Similar to [51], we can of ine perform the training procedure in our proposed DRL-based algorithm for many episodes under different channel states. The trained model needs to be updated when there is a signi cant change in the environment characteristics. Speci cally, we can update the trained model when computing resources are idle (e.g., midnight).

V. SIMULATION RESULTS AND DISCUSSIONS

In this section, we evaluate the performance of our proposed PRD-DRN algorithm. We describe our simulation settings rstly, which consists of the parameters of the network model and the hyper-parameters of the PRD-DRN. Meanwhile, the con gurations of the baseline algorithms are also been presented. Then, we show the performance comparison of the PRD-DRN with the benchmarks in different environments and give the relevant analysis. The whole experiment is implemented by the Tensor ow frame and runs on a PC with an Intel Core i7-6700 CPU @2.6GHz, Memory 16G.

A. Simulation Settings

Simulation Scenario. We build a simulation scenario, where there are one RSU with an MEC server, N connected vehicles and 10 traf c information acquisition sensors. The value of N ranges from 10 to 40 with the interval of 10. The wireless channels between the vehicles and the RSU follow the nite-state Markov channel (FSMC) model. We assume that the state of the channel is considered to be bad when spectrum ef ciency is 1 or good when spectrum ef ciency is 2. The transition probability of staying at the same state is set to be 0.7, and the transition probability from one state to another is set to be 0.3 [52]. The value of AoI limitation V_{max} is set to be 20 slots. Meanwhile, the allocated bandwidth B of each resource block is 1 MHz while the number of resource blocks H_b is set to be 50.

In each time step, the arrival vehicular requests for each edge-cached HDM file follows a Zipf distribution, where the value of the distribution parameter is set to be 1.5 [53]. This kind of distribution is representative for the practical vehicular networks which has been widely adopted in many relevant references [54]- [58]. The proposed algorithm needs to be retrained if the distribution changes. For each roadside sensor, the relevant file update latency is randomly selected from the value set $f0.5\ ;\ 0.6\ ;\ 0.7\ ;\ 0.8\ ;\ 0.9\ g$, where is the length of the unit time slot. The value of is set to be 1. Once the file update latency of each roadside sensor has been determined, their values will remain unchanged during the whole simulation process. Based on this, we set the extra request latency of a specific file to be the same as its update latency. In this paper, we consider the edge nodes (e.g., road side units and base stations) are equipped with stable power supply facilities, which are less affected by energy consumption. Thus, we do not consider the energy consumption of the edge nodes. Meanwhile, the transmission power for each roadside sensor is set to be 1W, where the value of belongs to [0.5,1]. As for a specific sensor, the energy consumption for traffic status sensing per file updating is set to be the same as that for data uploading [59].

Training model architecture: As for our proposed PRD-DRN model, the main network and the target network are made up of two identical fully-connected ANNs. Each ANN is consisted of four layers, i.e., an input layer, an output layer and two hidden layers. The input layer is consisted of $(N+1)\ F+N$ cells including the pre-processed system state. Each hidden layer has 256 cells while the output layer gives the HDM file update action. We utilize ReLU as the activation function and Adam [60] as the optimizer. To make the model easier to train, the input state has been normalized by the maximum allowable AoI $_{max}$, i.e.,

$$X_{norm} = \frac{X}{max} \tag{28}$$

where X is the input value. The learning rate of the PRD-DRN parameters, are set to be $4\ \ 10^{-4}$. The learning rate of the average reward is also set to be $4\ 10^{-4}$. The target network update interval i is set to be 2000 steps. The average return is calculated by the agent interacting with the environment for 10^4 steps. The memory buffer size is set to be $2\ 10^5$, and the mini-batch size M_m is set to be 32, 64 and 128. The exploration rate increases linearly from 0 to 1 and keeps fixed.

We compare our proposed PRD-DRN algorithm with the following baseline ones.

Random Policy: During each time step, the RSU randomly selects an update action for the current state. This policy doesn't take into account the transmission resource allocation.

Greedy Policy: During each time step, the RSU manages to maximize the immediate reward by executing the update action. This policy doesn't take the transmission resource allocation into consideration.

DQN-based Policy: The DQN-based policy is based on the traditional double DQN (DDQN) algorithm [61]. Its network architecture is similar to the PRD-DRN, which consists of a main network and a target network. The objective of the DDQN is to maximize the cumulative discount reward instead of the average one. Notice that, the state value function $V^0(s)$ and the state-action value function $Q^0(s;a)$ in the DDQN model are defined as

$$V^0(s) = E\left[\sum_{k=0}^{1} {}^{k} R(s(t+k);\ (t+k))\ j\ s(t) = s\right]\ ; \tag{29}$$

and

$$Q^0(s;a) = E\left[\sum_{k=0}^{1} {}^{k} R(s(t+k);a(t+k))\ j\ s(t) = s;a(t) = a\right] \tag{30}$$

where is the discount factor equal to 0.95 in the subsequent simulation. Meanwhile, the DDQN does not adopt the priority experience sampling method, and its loss function is given by Equation (31).

Except the differences described above, all the network training configurations are set to be the same as those in the PRD-DRN.

B. Simulation Results

1) Convergence Performance: To ensure the reliability of our proposed PRD-DRN algorithm, we first verify its convergence performance.

Fig. 4 shows the reward of PRD-DRN under different mini-batch size (32,64,128). To control variable, we set the $!_{Adj}$ to 0.6 and N to 30. We can see from Fig. 4 that different value of the batch-size has a significant effect on the reward of PRD-DRN, and a higher value of the mini-batch size helps PRD-DRN converge faster to a certain extent. This can be explained as follows. Firstly, the mini-batch size determines the number of the experience samples used for training per round. A smaller mini-batch size increases the randomness of the experience sample, which may impede the convergence speed of the model. Meanwhile, the PRD-DRN utilizes the prioritized sampling method, which can reduce the influence of mini-batch size on the convergence speed to a certain extent when the mini-batch size becomes larger. However, a large batch-size can result in a single-direction gradient descending during the training process, and this may cause a local optimal solution. We set the mini-batch size to 64 in the subsequent simulation.

Fig. 5 shows the convergence comparison of the PRD-DRN and the baseline policies when $N = 30$. Here, we also consider that the discount factor of the DQN-based policy can affect its performance in the convergence and the average reward [61]. Generally, a smaller means that the agent pays more attention to the immediate interests, and the training difficulty may be smaller. On the other hand, a bigger means that the agent pays attention to the long-term interests, which may make the algorithm unstable. It can be seen from Fig. 5 that a smaller (0.89, 0.92, 0.95) of the DQN-based policy indeed ensure its convergence speed, while the average reward becomes higher with a bigger. However, an extremely big (0.98) makes the DQN-based model difficult to converge during the training process. Meanwhile, the greedy policy and the random policy obtain a relatively low reward. By

$$L(\ ) = E\ (R(s_j\ ; a_j\ ) \quad Q^{\ 0}(s_j^0; \text{argmax}_a(s_j^0; a;\ );\quad)\quad Q(s_j\ ; a_j\ ;\ ))^2 \tag{31}$$

Fig. 4. The training rewards comparison under different batch-size.

comparison, our PRD-DRN obtains a higher reward while ensuring a faster convergence speed. It can be explained as follows. The prioritized experience sampling reduces the amount of experience the agent required to learn since it always selects the more valuable experience sample. Although the sampling process consumes some computation resources (as the analysis in Section III-C), the increased computational overhead is acceptable relative to the increased performance gain. Moreover, the PRD-DRN can achieve a better performance without adjusting the discount factor, which also reduces the training cost to a certain extent.

Based on the above analysis, we set the discount factor of the the DQN-based policy to 0.95 in the subsequent simulation.

Fig. 5. The training rewards comparison under different policy.

2) Efficiency Analysis: To verify the efficiency of our proposed method, we make performance comparison with the mentioned baseline policies.

Fig. 6 shows the average AoI cost when vehicles receive their requested HDMaps under different number of vehicles. It can be observed from Fig. 6 that the RSU with the PRD-DRN policy maintains relatively low AoI cost compared with the baseline policies. Meanwhile, it is interesting to find that although the number of state-action pairs increases exponentially with the increase of vehicular number N, the performance of PRD-DRN keeps stable with respect to it. It is due to the fact that the PRD-DRN is to maximize the long-term average reward, and thus it can execute optimal update actions in response to the vehicular requests. Even when there are no vehicular request arrival in a specific time step, the RSU may execute appropriate file update actions based on the historical request record and the real-time AoI of the cached files. We can also find out that the average AoI cost performance of the proposed PRD-DRN algorithm is under the pre-defined value of the AoI limitation (A_{max}^V=20). Thus, the PRD-DRN can ensure the stability of the AoI cost performance and realize a reasonable utilization of the network resources.

Fig. 6. Performance comparison in average AoI cost.

Fig. 7 shows the average file updating energy consumption of the system at each time step under different number of vehicles. It can be seen from Fig. 7 that the PRD-DRN keeps a relatively low energy consumption compared with the baseline policies with the increasing number of vehicles. This is due to the following reason. To achieve a high average reward, the agent will adjust the balance between the AoI cost and the energy consumption appropriately. Meanwhile, available transmission resource is allocated to the vehicle whose requested file is close to the AoI threshold, which avoids the unnecessary updates, and thus reducing the file update energy consumption.

Fig. 7. Performance comparison in energy consumption.

Fig. 8 shows the average file update time of the system at each time step under different number of vehicles. We can see from Fig. 8 that the greedy policy has worse performance than PRD-DRN and DQN-based policy. This is because the greedy policy only considers the instant system performance and is prone to fall into the dilemma of local optimal. Moreover, the greedy policy ignores the benefit brought by the optimal transmission resource allocation in reducing the number of instant file updates. The agents of the PRD-DRN and DQN-based policy can jointly schedule the file update and downlink transmission resource allocation from the long-term interactions with environment. The available downlink transmission resources are reasonably allocated to different vehicles to ensure that the AoI of the requested file meets the requirement of the vehicle with minimum file update times.

Fig. 8. Performance comparison in average update time.

VI. Conclusion

This paper studied the file update and downlink transmission resource allocation in the vehicular HDM edge caching system. For this purpose, we first formulated the file update and resource allocation as a nonlinear and nonconvex optimization problem. To solve this challenging problem, we then proposed a PRD-DRN algorithm combining the perceive capability of R-learning and the scheduling capability of reinforcement learning. Under the proposed PRD-DRN algorithm, the content update and transmission resource allocation procedures on the RSU were modeled as an MDP. Based on the advantages of deep R-learning and prioritized experience sampling, we obtained the optimal decision to minimize the long-term average cost related to the AoI and energy consumption. The extensive simulation results show that our PRD-DRN algorithm can achieve high long-term reward without managing the discounted factor. Remarkably, in comparison with the baseline policies, our algorithm can achieve lower average AoI and energy consumption with relative low file update time during a fixed period. An interesting study is to explore the joint content update and transmission resource allocation in the multiple edge nodes scenario with overlapping service region in our future work.

References

[1] R. Liu, J. Wang, and B. Zhang, "High definition map for automated driving: Overview and analysis," J. Navig., vol. 73, no. 2, pp. 324-341, 2020.

[2] H. G. Seif and X. Hu, "Autonomous driving in the iCity-HD maps as a key challenge of the automotive industry," Engineering, vol. 2, no. 2, pp. 159-162, 2016.

[3] H. Masuda, O. E. Marai, M. Tsukada, T. Taleb and H. Esaki, "Feature-Based Vehicle Identification Framework for Optimization of Collective Perception Messages in Vehicular Networks," in IEEE Transactions on Vehicular Technology, vol. 72, no. 2, pp. 2120-2129, Feb. 2023.

[4] E. Bastug, M. Bennis and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," in IEEE Communications Magazine, vol. 52, no. 8, pp. 82-89, Aug. 2014.

[5] P. Lin, Q. Song, J. Song, A. Jamalipour and F. R. Yu, "Cooperative Caching and Transmission in CoMP-Integrated Cellular Networks Using Reinforcement Learning," in IEEE Transactions on Vehicular Technology, vol. 69, no. 5, pp. 5508-5520.

[6] A. Aissioui, A. Ksentini, A. M. Gueroui and T. Taleb, "On Enabling 5G Automotive Systems Using Follow Me Edge-Cloud Concept," in IEEE Transactions on Vehicular Technology, vol. 67, no. 6, pp. 5302-5316.

[7] W. Qi, Q. Song, L. Guo and A. Jamalipour, "Energy-Efficient Resource Allocation for UAV-Assisted Vehicular Networks With Spectrum Sharing," in IEEE Transactions on Vehicular Technology, vol. 71, no. 7, pp. 7691-7702, July 2022.

[8] K. S. Khan and A. Jamalipour, "Coverage Analysis for Multi-Request Association Model (MRAM) in a Caching Ultra-Dense Network," in IEEE Transactions on Vehicular Technology, vol. 68, no. 4, pp. 3882-3889, April 2019.

[9] X. Xu, S. Gao and M. Tao, "Distributed Online Caching for High-Definition Maps in Autonomous Driving Systems," in IEEE Wireless Communications Letters, vol. 10, no. 7, pp. 1390-1394, July 2021.

[10] S. Kaul, R. Yates and M. Gruteser, "Real-time status: How often should one update?," 2012 Proceedings IEEE INFOCOM, 2012, pp. 2731-2735.

[11] A. Kosta, N. Pappas, and V. Angelakis, "Age of Information: A New Concept, Metric, and Tool," 2017.[Online]. Available: https://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=8187436.

[12] H. Tang, P. Ciblat, J. Wang, M. Wigger and R. Yates, "Age of Information Aware Cache Updating with File- and Age-Dependent Update Durations," 2020 18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT), 2020, pp. 1-6.

[13] L. Yang, Y. Zhong, F.-C. Zheng, and S. Jin, "Edge caching with real-time guarantees," 2019, arXiv:1912.11847. [Online]. Available: http://arxiv.org/abs/1912.11847.

[14] C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier and A. Ephremides, "Information freshness and popularity in mobile caching," 2017 IEEE International Symposium on Information Theory (ISIT), 2017, pp. 136-140.

[15] S. Zhang, L. Wang, H. Luo, X. Ma and S. Zhou, "AoI-Delay Tradeoff in Mobile Edge Caching With Freshness-Aware Content Refreshing," in IEEE Transactions on Wireless Communications, vol. 20, no. 8, pp. 5329-5342, Aug. 2021.

[16] J. Cao, X. Zhu, Y. Jiang and Z. Wei, "Can AoI and Delay be Minimized Simultaneously with Short-Packet Transmission?," IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2021, pp. 1-6.

[17] S. Zhang, J. Li, H. Luo, J. Gao, L. Zhao and X. Sherman Shen, "Low-Latency and Fresh Content Provision in Information-Centric Vehicular Networks," in IEEE Transactions on Mobile Computing, vol. 21, no. 5, pp. 1723-1738, 1 May 2022.

[18] R. D. Yates, P. Ciblat, A. Yener and M. Wigger, "Age-optimal constrained cache updating," 2017 IEEE International Symposium on Information Theory (ISIT), 2017, pp. 141-145.

[19] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," 2015 IEEE International Symposium on Information Theory (ISIT), 2015, pp. 3008-3012.

[20] X. Wu, J. Yang and J. Wu, "Optimal Status Update for Age of Information Minimization With an Energy Harvesting Source," in IEEE Transactions on Green Communications and Networking, vol. 2, no. 1, pp. 193-204, March 2018.

[21] A. Arafa, J. Yang, S. Ulukus and H. V. Poor, "Using Erasure Feedback for Online Timely Updating with an Energy Harvesting Sensor," 2019 IEEE International Symposium on Information Theory (ISIT), 2019, pp. 607-611.

[22] A. Arafa and S. Ulukus, "Timely Updates in Energy Harvesting Two-Hop Networks: Of ine and Online Policies," in IEEE Transactions on Wireless Communications, vol. 18, no. 8, pp. 4017-4030, Aug. 2019.

[23] A. Arafa, J. Yang, S. Ulukus and H. V. Poor, "Age-Minimal Transmission for Energy Harvesting Sensors With Finite Batteries: Online Policies," in IEEE Transactions on Information Theory, vol. 66, no. 1, pp. 534-556, Jan. 2020.

[24] E. T. Ceran, D. Gündüz and A. György, "Average Age of Information With Hybrid ARQ Under a Resource Constraint," in IEEE Transactions on Wireless Communications, vol. 18, no. 3, pp. 1900-1913, March 2019.

[25] Y. Gu, H. Chen, Y. Zhou, Y. Li and B. Vucetic, "Timely Status Update in Internet of Things Monitoring Systems: An Age-Energy Tradeoff," in IEEE Internet of Things Journal, vol. 6, no. 3, pp. 5324-5335, June 2019.

[26] M. Xie, Q. Wang, J. Gong and X. Ma, "Age and Energy Analysis for LDPC Coded Status Update With and Without ARQ," in IEEE Internet of Things Journal, vol. 7, no. 10, pp. 10388-10400, Oct. 2020.

[27] J. Gong, J. Zhu, X. Chen and X. Ma, "Sleep, Sense or Transmit: Energy-Age Tradeoff for Status Update With Two-Threshold Optimal Policy," in IEEE Transactions on Wireless Communications, vol. 21, no. 3, pp. 1751-1765, March 2022.

[28] M. Hatami, M. Leinonen and M. Codreanu, "AoI Minimization in Status Update Control With Energy Harvesting Sensors," in IEEE Transactions on Communications, vol. 69, no. 12, pp. 8335-8351, Dec. 2021.

[29] S. Wang et al., "Distributed Reinforcement Learning for Age of Information Minimization in Real-Time IoT Systems," in IEEE Journal of Selected Topics in Signal Processing(Early Access), 2022.

[30] E. T. Ceran, D. Gündüz and A. György, "A Reinforcement Learning Approach to Age of Information in Multi-User Networks With HARQ," in IEEE Journal on Selected Areas in Communications, vol. 39, no. 5, pp. 1412-1426, May 2021.

[31] C. Kam, S. Kompella and A. Ephremides, "Learning to Sample a Signal through an Unknown System for Minimum AoI," IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2019.

[32] B. Zhou and W. Saad, "Joint Status Sampling and Updating for Minimizing Age of Information in the Internet of Things," in IEEE Transactions on Communications, vol. 67, no. 11, pp. 7468-7482, Nov. 2019.

[33] S. Leng and A. Yener, "Age of Information Minimization for an Energy Harvesting Cognitive Radio," in IEEE Transactions on Cognitive Communications and Networking, vol. 5, no. 2, pp. 427-439, June 2019.

[34] C. Xu, Y. Xie, X. Wang, H. H. Yang, D. Niyato and T. Q. S. Quek, "Optimal Status Update for Caching Enabled IoT Networks: A Dueling Deep R-Network Approach," in IEEE Transactions on Wireless Communications, vol. 20, no. 12, pp. 8438-8454, Dec. 2021.

[35] C. Tunc and S. Panwar, "Optimal transmission policies for energy harvesting age of information systems with battery recovery," in Proc. 53rd Asilomar Conf. Signals, Syst., Comput., Paci c Grove, CA, USA, Nov. 2019, pp. 2012–2016.

[36] E. Sert, C. Sönmez, S. Baghaee and E. Uysal-Biyikoglu, "Optimizing age of information on real-life TCP/IP connections through reinforcement learning," 2018 26th Signal Processing and Communications Applications Conference (SIU), 2018, pp. 1-4.

[37] B. Yang, Y. Dang, T. Taleb, S. Shen and X. Jiang, "Sum Rate and Max-Min Rate for Cellular-Enabled UAV Swarm Networks," in IEEE Transactions on Vehicular Technology, vol. 72, no. 1, pp. 1073-1083, Jan. 2023.

[38] H. Sedjelmaci, S. M. Senouci and T. Taleb, "An Accurate Security Game for Low-Resource IoT Devices," in IEEE Transactions on Vehicular Technology, vol. 66, no. 10, pp. 9381-9393, Oct. 2017.

[39] J. Liu, Y. Xu, Y. Shen, X. Jiang and T. Taleb, "On Performance Modeling for MANETs Under General Limited Buffer Constraint," in IEEE Transactions on Vehicular Technology, vol. 66, no. 10, pp. 9483-9497, Oct. 2017.

[40] H. Huang, D. Qiao and M. C. Gursoy, "Age-Energy Tradeoff in Fading Channels with Packet-Based Transmissions," IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 2020, pp. 323-328.

[41] J. Du et al., "Resource Pricing and Allocation in MEC Enabled Blockchain Systems: An A3C Deep Reinforcement Learning Approach," in IEEE Transactions on Network Science and Engineering, vol. 9, no. 1, pp. 33-44, 1 Jan.-Feb. 2022.

[42] J. Du, F. R. Yu, G. Lu, J. Wang, J. Jiang and X. Chu, "MEC-Assisted Immersive VR Video Streaming Over Terahertz Wireless Networks: A Deep Reinforcement Learning Approach," in IEEE Internet of Things Journal, vol. 7, no. 10, pp. 9517-9529, Oct. 2020.

[43] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. Cambridge, MA, USA: MIT Press, 2018.

[44] N. C. Luong et al., "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey," in IEEE Communications Surveys & Tutorials, vol. 21, no. 4, pp. 3133-3174, Fourthquarter 2019.

[45] Y. -A. Wang and Y. -N. Chen, "Dialogue Environments are Different from Games: Investigating Variants of Deep Q-Networks for Dialogue Policy," 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), 2019, pp. 1070-1076.

[46] A. Schwartz, "A reinforcement learning method for maximizing undiscounted rewards," in Proc. ICML, 1993, pp. 298-305.

[47] Schaul, Tom and Quan, John and Antonoglou, Ioannis and Silver, David, "Prioritized Experience Replay," 2015, arXiv:1511.05952. [Online]. Available: https://arxiv.org/abs/1511.05952.

[48] J. Wu, C. Leng, Y. Wang, Q. Hu and J. Cheng, "Quantized Convolutional Neural Networks for Mobile Devices," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4820-4828.

[49] F. Wu, H. Zhang, J. Wu, Z. Han, H. V. Poor and L. Song, "UAV-to-Device Underlay Communications: Age of Information Minimization by Multi-Agent Deep Reinforcement Learning," in IEEE Transactions on Communications, vol. 69, no. 7, pp. 4461-4475, July 2021.

[50] H. Jun, X. Cong-Cong, G. Shuyang, and B. Erich, "Drop Maslow's Hammer or not: machine learning for resource management in D2D communications," in ACM SIGAPP Applied Computing Review, vol. 22, no. 1, pp. 5–14, March 2022.

[51] 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, Oct. 2019.

[52] Y. He, N. Zhao and H. Yin, "Integrated Networking, Caching, and Computing for Connected Vehicles: A Deep Reinforcement Learning Approach," in IEEE Transactions on Vehicular Technology, vol. 67, no. 1, pp. 44-55, Jan. 2018.

[53] A. Sadeghi, F. Sheikholeslami and G. B. Giannakis, "Optimal and Scalable Caching for 5G Using Reinforcement Learning of Space-Time Popularities," in IEEE Journal of Selected Topics in Signal Processing, vol. 12, no. 1, pp. 180-190, Feb. 2018.

[54] S. Krishnan, M. Afshang and H. S. Dhillon, "Effect of Retransmissions on Optimal Caching in Cache-Enabled Small Cell Networks," in IEEE Transactions on Vehicular Technology, vol. 66, no. 12, pp. 11383-11387, Dec. 2017.

[55] T. Liu, S. Zhou and Z. Niu, "Joint Optimization of Cache Allocation and Content Placement in Urban Vehicular Networks," 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 2018, pp. 1-6.

[56] C. Hou, C. Zhou, Q. Huang and C. -B. Yan, "Cache Control of Edge Computing System for Tradeoff Between Delays and Cache Storage Costs," in IEEE Transactions on Automation Science and Engineering, Early Access, doi: 10.1109/TASE.2022.3228250.