Joint Hybrid Caching and Replacement Scheme for UAV-Assisted Vehicular Edge Computing Networks

Yinan Liu, Chao Yang, Xin Chen, and Fengyan Wu

***Abstract*—Due to the flexible deployment and availability of line-of-sight (LoS) link, unmanned aerial vehicle (UAV) is able to assist the roadside unit (RSU) to provide timely computing resources to the covered vehicle users in the temporary congestion roads. In UAV-assisted vehicular edge computing networks (VECNs), caching necessary data in RSU/UAV reduces task execution delay and bandwidth cost significantly. However, for the constrained storage and computation capacities of UAV and the dynamic requests of users, the efficient caching data selection and replacement schemes are needed. In this article, we propose a novel joint hybrid caching and replacement scheme in a scene that a single UAV assists RSU to cover a set of vehicle users with large number of iterative calculation tasks. In particular, both the content caching and service caching are considered for the RSU and UAV. To minimize the whole task completion delay of users, we joint optimize the hybrid caching data selection of UAV and the task offloading strategy of users, a deep Q-network (DQN)-based solution is proposed to improve the utility of UAV. Then, we design an optimal evaluation function, and propose a caching replacement scheme for both the RSU and UAV. For RSU and UAV can update its caching data separately, a double-DQN (DDQN)-based solution is proposed. Extensive simulation results show that the proposed algorithms have good convergence, and the designed schemes reduce the task completion delay efficiently.**

***Index Terms*—Caching replacement, DDQN, DQN, hybrid caching, UAV, vehicular edge computing networks.**

I. Introduction

ALONG with the development of Internet of Things (IoT) and intelligent transportation technologies, the vehicles become smarter, safer and more efficient, with a plethora of emerging applications for vehicle users (i.e., the vehicles and passengers), such as: intelligent assisted driving, on-line game. The smooth implementation of these applications depends on massive data sharing and fast task processing among users, roadside units (RSUs) and cloud computing center in intelligent transportation systems (ITSs) [1], [2]. To cope with the generated huge delay-sensitive and computation-intensive tasks, vehicular edge computing networks (VECNs) have gained enormous attention, via providing vigorous computing, communication and storage services at the internet of vehicles (IoV) network edge [3]. However, the traffic network has strong spatial-temporal varying characteristics, if a large number of RSUs are deployed directly to meet the short time requests during the peak-traffic periods in some special road segments, it will lead to extremely high infrastructure costs, and low utilization in the normal off-peak periods. It is still a challenge how to utilize the limited computing and storage resources of RSUs in IoV efficiently.

Different from the fixed deployment RSUs, with high mobility, low cost and flexible deployment, unmanned aerial vehicles (UAVs) are regarded as the applicable tools to assist RSUs to provide communication and computation services for the covered vehicle users [4], [5]. In UAV-assisted VECNs, UAV performs as relay or edge node, and it can plan its moving trajectory and hovering position to build the line-of-sight (LoS) links with the ground devices. Caching the necessary data at the edge of VECNs, especially the popular ones, can reduce the IoV network response delay and save bandwidth consumption considerably [6]. According to the type of caching data, the current wireless edge caching strategies are divided into *edge content caching* [7], [8], [9], [10], [11] and *edge service caching* [12], [13], [14], [15]. For the former, the edge nodes (e.g., RSU, BS) can cache the calculation results or the directly output data of the users' requests (e.g., the traffic map download). The optimal content caching copes with the frequent and repeated requests from users. Otherwise, for the service caching, the edge node can cache the related database or the program data response to the computation tasks (e.g., the vehicle language recognition). Specially, the task offloading and service caching should be considered jointly, for the tasks are transmitted to the edge nodes only which had cached the corresponding service caching data [12], [13]. For both the content and service caching, the caching data selection, caching placement, caching data replacement and migration are analyzed and optimized. In the dynamic topology networks, the selected caching data at the edge nodes can be replaced in different time periods to meet the varying requests of users [16], and these data may also be migrated to the destination nodes to cover the moving users on ground [17].

For the UAV-assisted VECNs, UAVs with caching data are the suitable helper for RSUs in the case of sudden road traffic congestion that leads to significant increasing requests of users. From the ITS center's perspective, an optimal caching data

Manuscript received 15 September 2023; accepted 24 September 2023. Date of publication 9 October 2023; date of current version 23 February 2024. This work was supported by the National Natural Science Foundation of China under Grants 62003094 and 62003099. *(Corresponding author: Chao Yang.)*

The authors are with the Guangdong Key Laboratory of IoT Information Technology, School of Automation, Guangdong University of Technology, Guangzhou 510000, China (e-mail: 2112104094@mail2.gdut.edu.cn; yangchaoscut@aliyun.com; xinchen@gdut.edu.cn; 2112204070@mail2.gdut.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2023.3323217.

Digital Object Identifier 10.1109/TIV.2023.3323217

placement, replacement and migration scheme in VECNs can balance the distribution of the whole system communication and computation resources, and the caching cost is minimized. When the services or contents required are not cached in the edge nodes, the offloading tasks cannot be processed and can only be forwarded to the cloud computing center (CCC) for processing, which leads to a considerable transmission latency and a poor travel experience for users. Otherwise, when the corresponding services or contents had been cached, the user experience will be improved directly. Since the task types change with time and vehicles, the cached services and contents are also required to be replaced or migrated accordingly. However, most of the existing works focus on the content or service caching only, cannot satisfy the diverse needs of users. Hybrid caching, including the content and service caching, can improve the utility of caching space efficiently. Therefore, with different storage and computation capacities, and the coverage areas of RSUs and UAVs, it is necessary to design a hybrid caching scheme in UAV-assisted VECNs, to fully utilize the limited caching and computing resources of UAVs and maximize the caching benefits. However, there are three critical problems should be addressed for the hybrid caching scheme design. Firstly, *how to select the caching data to coordinate the percentage of the content and service caching?* Unlike the fixed edge infrastructure caching providers, UAVs can fly to the ground users and the caching capacity is limited, selecting the suitable content/service caching data and optimizing the percentage of them, can improve the effectiveness of UAV. Secondly, *how to perform the task offloading and resource allocation among UAV, RSU and the covered users?* Different caching schemes affect the task offloading directly, especially for the service caching. The users should select the optimal offloading edge node among the UAV or RSU. The communication and computation resources should be allocated corresponding to the caching status to minimize the whole task completion delay. Finally, *how to update the content/service caching data in both the UAV and RSU?* When the requests of users change in real-time, we should give an optimal scheme to make the caching data replacement decisions.

To address the above problems, in this article, we propose a hybrid caching and replacement scheme for the UAV-assisted VECNs, both the content and service caching are considered. We analyze a generalized situation that ITS center dispatches an UAV to assist the RSU to provide timely services for vehicle users in the congestion road intersection. Firstly, we consider the hybrid caching data selection of UAV in an offline scenario. According to the long-term observations of RSUs, the UAV can select the caching data along with flying to the destination. To minimize the whole task completion delay, we propose a joint hybrid caching and task offloading scheme, while the storage capacity of UAV, the requests of users and the cooperation between the RSUs and UAV are mainly considered. For the varying computing resources and transmission conditions, we proposed a deep Q-learning network (DQN)-based algorithm to find the optimal initialize caching strategy. Then, we propose an online optimized caching replacement scheme of both the RSU and UAV, to meet the dynamic requests of users. We design an evaluation function while the remaining caching spaces, the popularity and the hit ratio of the cached data are considered jointly. Then, we give a double-DQN (DDQN)-based algorithm to find the final solutions, for both the RSU and UAV can perform as independent individuals to update its states. The main contributions are summarized as follows:

- We construct a hybrid caching and replacement analytical model for the UAV-assisted VECNs. An UAV performs as a flying BS to assist RSU provides computing resources for the temporal congested vehicle users, both the content and service caching for the UAV and RSU are jointly considered.
- We propose a joint hybrid caching and task offloading scheme to minimize the whole task completion delay of the covered vehicle users. We optimize the caching data selection in UAV and the task offloading of vehicle users via a DQN-based algorithm.
- We design an optimal evaluation function, and propose a caching replacement scheme for both the RSU and UAV. A DDQN-based algorithm is proposed.

The rest of this article is organized as follows. The related works are reviewed in Section II. Section III proposes the details of the system model. In Section IV, a hybrid caching data selection and task offloading scheme is analyzed. Section V elaborates a joint optimization problem for the caching replacement scheme. Numerical examples are given in Section VI to show the performance evaluation of the proposed schemes. We draw the conclusions in Section VII.

II. Related Work

In this article, we design a hybrid caching and replacement scheme for the UAV-assisted VECNs to support the low-latency applications of the covered vehicle users on the congestion roads. In this section, we review the existing relevant works for the UAV-assisted VECNs and the caching schemes in VECNs.

A. UAV-Assisted VECNs

Since UAV-assisted VECNs can provide continuous and timely network service for the covered vehicle users in the dynamic moving environment, it has received an increasing amount of attention in recent years. Due to the easy deployment and high mobility, UAV can perform as edge node or relay, and cooperate with the RSUs to cover the users, especially at some intersections where traffic congestion occurs suddenly. For the limited UAV energy budget, in [18], an energy efficient UAV-enabled computing-communication intelligent offloading scheme was proposed. The ground users were divided into task gathering node and others, based on the amount of data. The UAV flies to the former nodes only. Moreover, the stochastic task offloading and resource allocation scheme had proposed in [19]. In [20], the UAV flying trajectory and ground users' task uploading powers were jointly optimized. Since the dynamic topology of VECNs, the freshness of data packets was considered in [21], an age of information (AoI) aware UAV deployment problem was analyzed. Compared with the single UAV, multi-UAV can provide powerful wireless connections, but the anti-collision and communication interference between

UAVs also need be considered. In [22], a joint multi-UAV trajectory and communication scheduling optimization problem was proposed. Digital twin (DT) is a critical tool to capture the time-varying environment, in [23], the authors analyzed a dynamic DT of UAV-assisted IoV resource allocation scheme.

B. Caching in VECNs

Compared to calculating the requested tasks from remote cloud computing center, caching necessary data in the IoV network edge can empower local calculation and accelerate the request response. The existing works about the design of content/service caching scheme in VECNs, always include the caching data selection, placement, replacement and migration. In [8], the content caching placement and delivery were optimized, for the cooperation among the BS, RSUs on roadside and the vehicles on road was considered. In [24], the future time varying content popularity was predicted firstly, and a joint computation offloading and caching scheme was proposed. The service caching and computation task offloading schemes were jointly optimized in [25]. Moreover, to improve the caching data selection efficiency, the DT based social network [26] and the vehicle moving prediction [27] were also be analyzed. The service caching selection, the computation management were jointly optimized in [9]. For the mobility of vehicles, the service caching migration and task offloading were analyzed in [28]. To capture some complex computation tasks, a service chain caching and task process scheduling scheme were studied in [29]. Moreover, the distributed service placement [15] and the dependency of computation tasks [12] were also considered to improve the QoE of users in VECNs.

Ref. [30] proposed an UAV deployment and content caching placement scheme for the multi-UAV networks, in addition, the service caching, task offloading decisions and the resource allocation were jointly optimized in [31]. Refs. [32], [33], [34] proposed a joint UAV trajectory and caching management scheme for UAV-assisted ITSs. However, the above existing works considered the content caching or the service caching only, cannot utilize the limited storage space effectively. Different from the cited works, in this article, we analyze the hybrid caching and replacement scheme for an UAV-assisted VECNs, to minimize the whole task completion delay. The cooperation between the UAV and RSU, and the time varying requests of vehicle users are considered mainly.

III. System Model

As shown in Fig. 1, an UAV-assisted VECNs system consists a road intersection with a RSU and multiple vehicles, and CCC. When traffic congestion occurs on the road and the total communication and computation requests of vehicle users increase quickly, the ITS center dispatches an UAV to assist the RSU to provide the network services for the covered vehicle users. The UAV, RSU and CCC are connected via 5 G/B5G wireless links, and the wired links exist between RSU and CCC. For the vehicle users, the computation tasks are offloaded to the RSU or UAV firstly, they can further be offloaded to the CCC based on the delay constraint, the computation requests and available

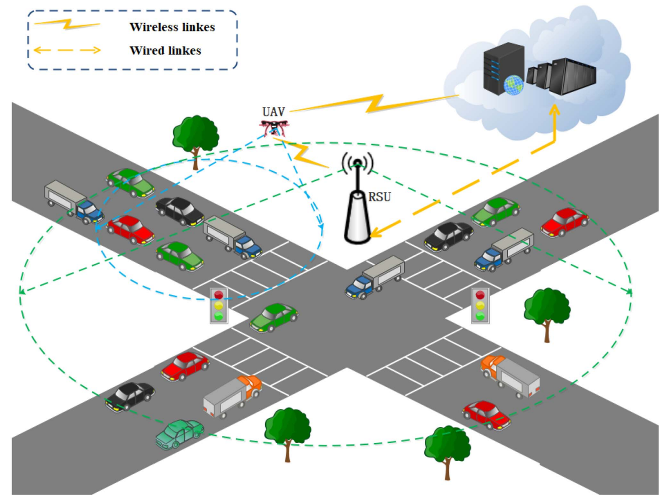

Fig. 1. System model.

caching resources. We set that the CCC has ability to response all requests. Correspondingly, we set the RSU has already cached some suitable services and contents locally via the long-term observation. In this article, we use the historical data recorded in the current RSU, the popularity of the services and contents, and the access demand of the current junction users to consider the caching optimization of the services and contents in UAV, while the UAV is hovering in the current area with high traffic flow, so the wireless channel status of the V2U in the current area basically will not change.

The caching capacities of RSU and UAV are denoted as P_R and P_U, the total computation resources of the RSU and UAV are F_U and F_R, respectively. We firstly consider an offline scenario in which the RSU obtains the service and content caching data in advance, before the UAV flies to the road intersection. The service caching data is represented as $\boldsymbol{S}_1 = \{S_{11}, \ldots, S_{1n}, \ldots, S_{1N}\}$, each service consists of a triples as $S_{1n} = < pop_{s_{1n}}, p_{s_{1n}}, s_{1n} >$, where $pop_{s_{1n}}$ denotes the popularity, $p_{s_{1n}}$ is the data size and s_{1n} denotes the type of service. Similarly, the content caching data is represented as $\boldsymbol{S}_2 = \{S_{21}, \ldots, S_{2m}, \ldots, S_{2M}\}$ and each content is also composed of a triple as $S_{2m} = < pop_{s_{2m}}, p_{s_{2m}}, s_{2m} >$. In the proposed system model, we set that multiple vehicle users can access the same type of service or content simultaneously. The generated task y from vehicle user x is composed of a quaternion as $V_{task}^{x,y} = < s_{x,y}, P_{V,S}^{x,y}, f_{s_{1n}}^{x,y}, t_{S,res}^{x,y}, cope_{x,y} >, x \in \mathbb{X}, y \in \mathbb{Y}$, where $s_{x,y} \in \{s_{1n}, s_{2m}\}$ denotes the task type, $P_{V,S}^{x,y}$ is the task data size, $f_{s_{1n}}^{x,y}$ denotes the required computational resources for the task to be processed by the associated service S_{1n} in the edge server, and $t_{S,res}^{x,y}$ denotes the total latency constraint. $cope_{x,y} \in \{0, l, k\}$ means the task processing selection. $cope_{x,y} = 0$ means the task be processed in CCC, $cope_{x,y} = l$ means it corresponds to the edge node which had cached the special service, $cope_{x,y} = k$ means it can be processed by the node with the corresponding contents. $\mathbb{X}$ denotes the set of vehicle users and $\mathbb{Y}$ denotes the generated tasks. We consider hybrid caching, which includes service caching and content caching. For both the RSU and UAV, we set the vehicle user l accesses the service caching, as S_{1n}^l, $l \in \{1, 2, \ldots, L\}$, L is the total number of users. Similarly, user k accesses the content caching node, as S_{2m}^k, $k \in \{1, 2, \ldots, K\}$.

A. Task Transmission and Calculation Model

We set that the vehicles integrate several communication technologies. The dynamic Time Division Multiple Access (TDMA) is used to ensure the transmission among the vehicles and UAV [35], [36]. Each vehicle user utilizes the entire primary frequency band for a certain time period individually. The bandwidth utilization is handled by the UAV as a central coordinator. It allocates the main frequency band to vehicle users, avoiding the possibility of channel interference caused by others. The time horizon of a UAV-multi-vehicle users is divided into T equal time slices, indexed by $t \in \{1, 2, \ldots, T\}$ [37]. We consider line-of-sight (LoS) propagation models for the transmission links between UAV and users [30], and the path loss for LoS model can be expressed as:

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

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

where $d_u = \{d_{U,x}^t, d_{R,x}^t, d_{UO}\}$ denotes the distances from the UAV to the user x, RSU and CCC, f_c is the carrier frequency, η_{LoS} and η_{NLoS} denote the additional loss of LoS link. For the UAV and a vehicle user, the probabilities of LoS link depend on their elevation angel and the environment, as:

$$Pr_{LoS} = \frac{1}{1 + X \exp(-Y(\Theta_x^t - X))}, \tag{3}$$

$$Pr_{NLoS} = 1 - Pr_{LoS}, \tag{4}$$

where Θ_x^t is the elevation angel between UAV and user x in t, X and Y are constants which depend on the environment. The average path loss is:

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

Then, the transmission rate between UAV and user x is:

$$R_{S,x}^{VU} = B \log_2 \left(1 + \frac{P_{U,x}}{10^{PL/10} \sigma^2} \right). \tag{6}$$

According to the distance between the vehicle users and edge node, the task transmission rate of V2R (vehicle to RSU) R_S^{VR} is given as

$$R_{S,x}^{VR} = B \log_2 \left(1 + \frac{P_{R,x}}{\sigma^2 + N} \right), \tag{7}$$

where the LTE-V2X is used here. B is the allocated bandwidth, N is the interference between multi-channel transmissions, d_{VR} and d_{VU} denote the distances of V2R and V2U. We set that the processed result return transmission rates are fixed as R_S^{back}. The UAV and CCC are connected with 5 G/B5G wireless links, and the transmission rate R_S^{UO} is

$$R_S^{UO} = B_{cloud} \log_2 \left(1 + \frac{P_0}{10^{PL/10} \sigma^2} \right), \tag{8}$$

where B_{cloud} is the wireless channel bandwidth allocated to the UAV by CCC, P_0 is the transmit power, d_{UO} is the distance. RSU and CCC are connected via an optical fiber with a fixed transmission rate of R_S^{RO}.

When a task be offloaded to the edge node (i.e., RSU or UAV), and the corresponding service caching had been cached, the task can be processed directly. For content caching, no computational operations are required. Assuming the UAV and RSU allocate the computing resources $f_{U,s_{1n}}$ and $f_{R,s_{1n}}$ to the service S_{1n}, the computation delay $T_{RC,s_{1n}}^{x,y}$, $T_{UC,s_{1n}}^{x,y}$ are shown as

$$T_{RC,s_{1n}}^{x,y} = \frac{f_{s_{1n}}^{x,y}}{f_{R,s_{1n}}}, \tag{9}$$

$$T_{UC,s_{1n}}^{x,y} = \frac{f_{s_{1n}}^{x,y}}{f_{U,s_{1n}}}. \tag{10}$$

Since CCC has abundant computation resources and service caching data, the task computation latency in CCC is small compared to the transmission one, we consider the latter only. The main notations of this paper are summarized in Table I.

IV. RSU-UAV Hybrid Caching Scheme

In this section, we consider how to design a joint hybrid caching and task offloading scheme for UAV-assisted VECNs in an offline scenario. Part of service/content caching data had been cached in advance before the UAV arriving the intersection. In details, there are two subproblems to be considered, as: 1) how to select the service/content caching data for UAV, under the limited caching capacity? 2) how to perform the task offloading based on the service/content caching?

Denote $\mathcal{C}$ as the caching data selection vector, P_{S_1} and P_{S_2} as the caching space of service caching and content caching in UAV, we have

$$\mathcal{C} = \{C_{s_{1n}}^R \in \{0,1\}, C_{s_{1n}}^U \in \{0,1\}, \\ C_{s_{2m}}^R \in \{0,1\}, C_{s_{2m}}^U \in \{0,1\}\},$$

where $C_{s_{1n}}^R$ denotes whether the service S_{1n} is cached in RSU, $C_{s_{1n}}^U$ denotes whether it be cached in UAV, $C_{s_{2m}}^R$ and $C_{s_{2m}}^U$ indicate the content caching decisions.

The caching data selections for RSU and UAV are different. For the content caching that only consume the storage resources, we set the UAV can cache different contents with the cached data in RSU. Otherwise, for the service caching, both the storage and computation resources are needed, the UAV can cache the same service caching data as the RSU, to cope with the sudden same requests from a set of users.

$$\begin{cases} C_{s_{2m}}^R C_{s_{2m}}^U = 0, \\ C_{s_{1n}}^R C_{s_{1n}}^U \le 1. \end{cases} \tag{11}$$

TABLE I
NOTATIONS USED IN THIS PAPER

Notations	Description
S_1,S_2	Service and content
K,L	The number of vehicles handled by the tasks through content and services
$C^R_{S_{1n}}$,$C^U_{S_{1n}}$,$C^R_{S_{2m}}$,$C^U_{S_{2m}}$	Caching decision parameters for services and content in UAV and RSU
α	Cache rationing parameter for services and content in UAV
$\beta_{x,y}$,τ	Task offloading position decision parameters
T_1^{UR},T_2^{UR},T_3^{UR}	Processing latency of tasks under different offloading scenarios (*s*)
$t^{x,y}_{S,res}$	Time delay constraints for tasks (*s*)
$t^{x,y}_{res}$	Latency of actual processing tasks (*s*)
γ_1,γ_2,γ_3	Evaluation of the weight values of the components in the function
d_{VR},d_{VU}	Distance from vehicle to edge server (*m*)
d_{UO}	Distance between UAV and central cloud (*m*)
P_{S_1},P_{S_2}	Cache space for services and content in UAV (*Mb*)
$t^{x,y}_{cloud}$	Latency of task offload to central cloud processing (*s*)
$t^{x,y}_{OR}$,$t^{x,y}_{OU}$	Latency in returning computation results from the central cloud to the edge servers (*s*)
$P^{x,y}_{V,S_{1n}}$,$P^{x,y}_{V,S_{2m}}$	The size of the result obtained by the task after processing through the service and content (*Mb*)
R^{back}_S	Calculation of the transmission rate returned to the vehicle by the edge server (*bit/s*)
R^{RO}_S	Transmission rate of RSU offload tasks to central cloud (*bit/s*)
$f_{R,S_{1n}}$,$f_{U,S_{1n}}$	RSU and UAV allocated to services of computing resources
$cope_{x,y}$, $\Psi_{x,y}$	The parameter of whether the task is processed by the service or by the content
η_{LoS}, η_{NLoS}	The additional loss of LoS links

A. Task Offloading Scheme

The caching data selection and task offloading are coupled with each other. According to the different caching data of RSU/UAV, the vehicle users select different task offloading schemes. In details, we consider the task offloading scheme under three cases, as:

1) *For a special task, the corresponding type of service had been cached in both RSU and UAV,* $C^R_{s_{1n}} C^U_{s_{1n}} = 1$.
In this case, user x offloads the task y to RSU or UAV based on the transmission delay. The decision variable $\beta_{x,y}$ is shown as

$$\beta_{x,y} = \begin{cases} 0, & if(T^{x,y}_{V2U,s_{1n}} + T^{U,x,y}_{C,s_{1n}}) \geq (T^{x,y}_{V2R,s_{1n}} + T^{R,x,y}_{C,s_{1n}}) \\ 1, & if(T^{x,y}_{V2U,s_{1n}} + T^{U,x,y}_{C,s_{1n}}) < (T^{x,y}_{V2R,s_{1n}} + T^{R,x,y}_{C,s_{1n}}), \end{cases} \quad (12)$$

where $\beta_{x,y} = 0$ means that the task is offloaded to RSU, otherwise, $\beta_{x,y} = 1$. The task completion delay is

$$T_1^{UR} = C^U_{s_{1n}} C^R_{s_{1n}} \Big[(1-\beta_{x,y}) \left(T^{x,y}_{V2R,s_{1n}} + T^{R,x,y}_{C,s_{1n}} + T^{x,y}_{B,s_{1n}} \right) + \beta_{x,y} \left(T^{x,y}_{V2U,s_{1n}} + T^{U,x,y}_{C,s_{1n}} + T^{x,y}_{B,s_{1n}} \right) \Big] \quad (13)$$

where $T^{x,y}_{V2R,s_{1n}} + T^{R,x,y}_{C,s_{1n}} + T^{x,y}_{B,s_{1n}}$ denotes the task offloading delay to RSU with the corresponding type of service, and $T^{x,y}_{V2U,s_{1n}} + T^{U,x,y}_{C,s_{1n}} + T^{x,y}_{B,s_{1n}}$ denotes the delay to UAV.

2) *The corresponding type of service/content had been cached in one of the RSU or UAV.* $C^R_{s_{1n}} + C^U_{s_{1n}} = 1$ *or* $C^R_{s_{2m}} + C^U_{s_{2m}} = 1$. Also let

$$\Psi_{x,y} = \begin{cases} 0, & cope_{x,y} = k, \\ 1, & cope_{x,y} = l \text{ } or \text{ } 0, \end{cases} \quad (14)$$

where $\Psi_{x,y}$ indicates the task processing selection.
In this case, user x can only offload the task to RSU or UAV that had cached the corresponding service/content in advance. The total task completion delay T_2^{UR} is

$$\begin{aligned} T_2^{UR} = & \left(1 - C^U_{s_{1n}}\right) C^R_{s_{1n}} \left(T^{x,y}_{V2R,s_{1n}} + T^{R,x,y}_{C,s_{1n}} + T^{x,y}_{B,s_{1n}} \right) \\ & + \left(1 - C^R_{s_{2m}}\right) \left(T^{x,y}_{V2R,s_{2m}} + T^{x,y}_{B,s_{2m}} \right) \\ & + C^U_{s_{1n}} \left(1 - C^R_{s_{1n}}\right) \left(T^{x,y}_{V2U,s_{1n}} + T^{U,x,y}_{C,s_{1n}} + T^{x,y}_{B,s_{1n}} \right) \\ & + C^U_{s_{2m}} \left(T^{x,y}_{V2U,s_{2m}} + T^{x,y}_{B,s_{2m}} \right), \end{aligned}$$

where $T^{x,y}_{V2R,s_{1n}} + T^{R,x,y}_{C,s_{1n}} + T^{x,y}_{B,s_{1n}}$ is the task completion delay when only RSU had cached the service data, it includes the V2R transmission $T^{x,y}_{V2R,s_{1n}} = \frac{P^{x,y}_{V,S}}{R^{VR}_{S,x}}$, task processing $T^{R,x,y}_{C,s_{1n}}$, and the result backhaul delay $T^{x,y}_{B,s_{1n}}$. For the service/content caching in RSU, the associated results are different. When $C^R_{s_{1n}} = 1$, the backhaul delay is $T^{x,y}_{B,s_{1n}} = \frac{P^{x,y}_{V,s_{1n}}}{R^{back}_S}$, and when $C^R_{s_{2m}} = 1$, the backhaul delay is $T^{x,y}_{B,s_{2m}} = \frac{P^{x,y}_{V,s_{2m}}}{R^{back}_S}$, $P^{x,y}_{V,s_{1n}}$ and $P^{x,y}_{V,s_{2m}}$ are the data size of the associated results. Moreover, $T^{x,y}_{V2U,s_{1n}} + T^{U,x,y}_{C,s_{1n}} + T^{x,y}_{B,s_{1n}}$ and $T^{x,y}_{V2U,s_{2m}} + T^{x,y}_{B,s_{2m}}$ are the total latencies of the corresponding service/content had cached in UAV.

3) *Neither the UAV nor RSU had cached the service or content corresponding to the tasks offloaded from the user,* $C^R_{s_{1n}} = C^U_{s_{1n}} = C^R_{s_{2m}} = C^U_{s_{2m}} = 0$. And $cope_{x,y} = 0$.
In this case, the RSU/UAV performs as relay, the offloaded tasks are processed in CCC finally. The total task completion delay T_3^{UR} is shown as:

$$\begin{aligned} T_3^{UR} = & \left(1 - C^U_{s_{1n}}\right) \left(1 - C^R_{s_{1n}}\right) \left(T^{x,y}_{min,s_{1n}} + t^{x,y}_{cloud} + t^{x,y}_{RU} \right. \\ & \left. + T^{x,y}_{B,s_{1n}} \right), \end{aligned} \quad (15)$$

where $T^{x,y}_{min,s_{1n}}$ is the minimum task offloading delay from the user to RSU/UAV, $T^{x,y}_{B,s_{1n}}$ is the backhaul delay of the calculation result. $t^{x,y}_{RU} = \min\{t^{x,y}_{OR}, t^{x,y}_{OU}\}$. $t^{x,y}_{OR}$, $t^{x,y}_{OU}$ are the backhaul delays from CCC to RSU and UAV. $t^{x,y}_{cloud} = (1-\tau)t^{R,x,y}_{cloud} + \tau t^{U,x,y}_{cloud}$, $t^{R,x,y}_{cloud}$ and $t^{U,x,y}_{cloud}$ denote the transmission delay of the RSU and UAV perform task offloading to CCC, respectively.

Denote $T^{R,x,y}_{s_{1n}}$ and $T^{U,x,y}_{s_{1n}}$ as the total task completion delays, as

$$\begin{aligned}T^{R,x,y}_{s_{1n}} &= T^{x,y}_{V2R,s_{1n}} + C^{R}_{s_{1n}} T^{R,x,y}_{C,s_{1n}} + T^{x,y}_{B,s_{1n}} \\ &\quad + (1 - C^{R}_{s_{1n}})(t^{x,y}_{cloud} + t^{x,y}_{RU}), \\ T^{U,x,y}_{s_{1n}} &= T^{x,y}_{V2U,s_{1n}} + C^{U}_{s_{1n}} T^{U,x,y}_{C,s_{1n}} + T^{x,y}_{B,s_{1n}} \\ &\quad + (1 - C^{U}_{s_{1n}})(t^{x,y}_{cloud} + t^{x,y}_{RU}).\end{aligned}$$

Then, we obtain the service delay constraint $t^{x,y}_{res} \le t^{x,y}_{S,res}$,

$$\begin{aligned}t^{x,y}_{res} = \max\Big\{&T^{R,x,y}_{s_{1n}},\ T^{U,x,y}_{s_{1n}},\ T^{x,y}_{V2R,s_{2m}} + T^{x,y}_{B,s_{2m}}, \\ &T^{x,y}_{V2U,s_{2m}} + T^{x,y}_{B,s_{2m}}\Big\}.\end{aligned}$$

B. Problem Formulation

In summary, the total task completion delay for the UAV-assisted VECNs to provide network service for users, T_{S_1} and T_{S_2} are shown as:

$$\begin{aligned}T_{S_1} = \sum_{n}^{N}\sum_{l}^{L}\Psi_{x,y}\Big\{&\left(1 - C^{U}_{s_{1n}}\right)\left(1 - C^{R}_{s_{1n}}\right) \\ &\left(T^{x,y}_{min,s_{1n}} + t^{x,y}_{cloud} + t^{x,y}_{RU} + T^{x,y}_{B,s_{1n}}\right) \\ &+ C^{U}_{s_{1n}}\left(1 - C^{R}_{s_{1n}}\right)\left(T^{x,y}_{V2U,s_{1n}} + T^{U,x,y}_{C,s_{1n}} + T^{x,y}_{B,s_{1n}}\right) \\ &+ \left(1 - C^{U}_{s_{1n}}\right)C^{R}_{s_{1n}}\left(T^{x,y}_{V2R,s_{1n}} + T^{R,x,y}_{C,s_{1n}} + T^{x,y}_{B,s_{1n}}\right) \\ &+ C^{U}_{s_{1n}}C^{R}_{s_{1n}}\Big[(1-\beta)\left(T_{V2R,s_{1n}} + T^{R,x,y}_{C,s_{1n}} + T^{x,y}_{B,s_{1n}}\right) \\ &+ \beta\left(T^{x,y}_{V2U,s_{1n}} + T^{U,x,y}_{C,s_{1n}} + T^{x,y}_{B,s_{1n}}\right)\Big]\Big\}, \\ T_{S_2} = \sum_{m}^{M}\sum_{k}^{K}(1-\Psi_{x,y})\Big\{&C^{U}_{s_{2m}}\left(T^{x,y}_{V2U,s_{2m}} + T^{x,y}_{B,s_{2m}}\right) \\ &+ \left(1 - C^{R}_{s_{2m}}\right)\left(T^{x,y}_{V2R,s_{2m}} + T^{x,y}_{B,s_{2m}}\right)\Big\}.\end{aligned}$$

Then, the joint optimization problem of the hybrid caching and task offloading scheme is shown as **P1**.

$$\begin{aligned}\mathbf{P1}: \quad &\min_{\{\mathcal{C},\alpha\}}(T_{S_1} + T_{S_2}) \\ \text{s.t. } &C1: \sum_{s_{1n}\in\boldsymbol{S}_1} C^{U}_{s_{1n}} p_{s_{1n}} \le P_{S_1}, \quad \forall n \in \{1,2,3,\ldots,N\} \\ &C2: \sum_{s_{2m}\in\boldsymbol{S}_2} C^{U}_{s_{2m}} p_{s_{2m}} \le P_{S_2}, \quad \forall m \in \{1,2,\ldots,M\} \\ &C3: \sum_{s_{1n}\in\boldsymbol{S}_1} C^{U}_{s_{1n}} f_{U,s_{1n}} \le F_U, \quad \forall n \in \{1,2,\ldots,N\} \\ &C4: \sum_{s_{1n}\in\boldsymbol{S}_1} C^{R}_{s_{1n}} f_{R,s_{1n}} \le F_R, \quad \forall n \in \{1,2,\ldots,N\} \\ &C5: t^{x,y}_{res} \le t^{x,y}_{S,res}, \\ &C6: (1-\alpha)P_{S_1} = \alpha P_{S_2}, \quad P_{S_1} + P_{S_2} \le P_U, \\ &C7: C^{R}_{s_{2m}} C^{U}_{s_{2m}} = 0, \quad C^{R}_{s_{1n}} C^{U}_{s_{1n}} \le 0, \\ &C8: \beta_{x,y}\in\{0,1\}, \tau\in\{0,1\}, \Psi_{x,y}\in\{0,1\}, \alpha\in(0,1).\end{aligned}$$

where C1, C2 are the caching and computing capacity constraints for the service/content cached in UAV, C3, C4 denote the computational resource constraints of UAV and RSU, C5 is the task processing delay constraint, C6 represents the relationship between the caching space of the services and content already cached in UAV, C7 represents the constraints of the types of service/content, C8 denotes the decision to offload tasks to edge servers, the decision to offload tasks to the central cloud, the task processing selection, and the caching rationing parameters for services and content.

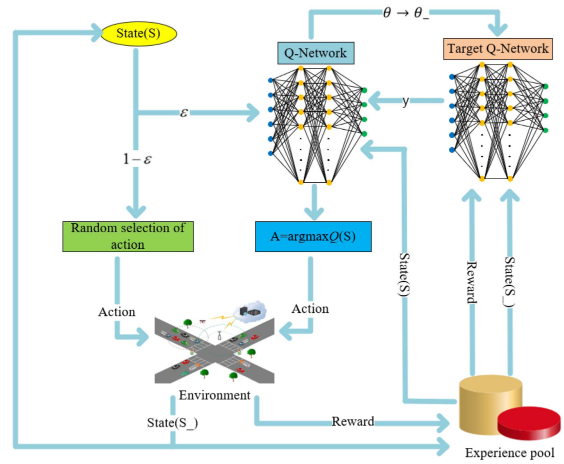

Fig. 2. DQN structural framework diagram.

C. DQN-Based Solution Design

For the analysis in the system model and problem formulation, the task offloading decisions are associated with whether the corresponding service/content caching data had been cached directly. From the perspective of the vehicle user, the hybrid caching and task offloading decisions are coupled with each other, and the computational resources of the edge nodes to handle the tasks are random. Thus, we design a DQN-based algorithm to find the optimal solutions, the block diagram is shown as Fig. 2. **P1** is formulated as a mixed integer nonlinear programming problem, which has non-polynomial computation complexity, it is difficult to be solved by general algorithms.

The overall framework of the DQN is shown in Fig. 2, and the design of DRL in the task offloading processing strategy is as follows. Define a five-tuple $\mathcal{M} =< \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \Pi >$ to represent the Markov decision process(MDP) firstly, as:

State: $\mathcal{S} =< n_{uav}, d_{uav}, n_{rsu}, d_{rsu}, \alpha, t_{uav} >$, where n_{uav} and n_{rsu} denote the number of tasks be offloaded to the UAV and RSU, respectively, and d_{uav}, d_{rsu} denote the straight-line distances of V2U and V2R. α is the proportion of service caching and content caching in UAV, t_{uav} denotes the total task processing latency to UAV. And the tuple parameters are interrelated and affect each other. For example, the distance d_{uav} of V2U and the number of tasks n_{uav} offloaded to UAV affect the total task completion delay t_{uav} directly.

Action: $\mathcal{A} =< a_1, a_2, \ldots, a_\Phi >$, which indicates the different types of service/content be cached at UAV. $a_1, a_2, \ldots, a_\Phi$ represent services or contents of different types.

Reward: The reward R_t is mainly determined by the task processing latencies T_{S_1} and T_{S_2}. Also, a fixed penalty is given to the UAV for any tasks that remain unprocessed after the vehicle user leaves the UAV's coverage area.

Policy: Using Π to represent the policy for selecting service/content to be cached in UAV, which represents the mapping relationship from the state space to the action space in the whole system, i.e., an agent can select an action $a = \Pi(s)$ at any state s, and

$$\sum_{a \in \mathcal{A}} \Pi(a|s) = 1. \tag{16}$$

For example, the UAV selects the suitable type of service/content to cache based on the current state of the environment to handle the offloaded tasks from users. The whole process is considered as a continuous iteration of the loop, the optimal policy Π^*. $V^\pi(s)$ represents the expected value of accumulated rewards obtained by UAV, as

$$\begin{aligned} V^\pi(s_t) &= \mathbb{E}[r_{t+1} + \eta r_{t+2} + \eta^2 r_{r+3} + \cdots] \\ &= \sum_{s_{t+1} \in \mathcal{S}} \mathcal{P}(s_{t+1}|s_t)(R_{t+1}(s_{t+1}) + \eta V(s_{t+1})). \end{aligned} \tag{17}$$

Then, the value function is translated to a Q-function, as

$$Q_\pi(s_t, a_t) = \mathbb{E}_{s_{t+1} \in \mathcal{S}, a_{t+1} \in \mathcal{A}}[R_t + \eta Q_\pi(s_{t+1}, a_{t+1})|s_t, a_t]. \tag{18}$$

DQN designs two structurally identical neural networks to fit the Q-function, the evaluation network to record the current Q-values and the target network to predict the future ones. The loss function of the neural network parameter θ is

$$L(\theta) = (y - Q(s, a|\theta))^2, \tag{19}$$

where $y = R + \eta \max_{a' \in \mathcal{A}} Q(s', a'|\theta\_)$, θ and $\theta\_$ denote the parameters of the evaluation and target network. The gradient of $L(\theta)$ to θ is shown as

$$\nabla_\theta L(\theta) = (y - Q(s, a|\theta)) \nabla Q(s, a|\theta). \tag{20}$$

The Q-value is calculated by two-layer network structure model and the gradient is calculated according to $\nabla_\theta L(\theta)$, the gradient descent method is used to update the network parameters θ, i.e, $\theta \leftarrow \theta - lr \nabla_\theta L(\theta)$, where lr is the learning rate, the optimal parameter structure is obtained by constantly updating $\theta\_$ with θ. The ε-greedy policy is also used for UAV caching action selection, and the processing details are shown in Algorithm 1.

Algorithm 1: DQN-Based Algorithm for Hybrid Caching and Task Offloading.

Initialization: ε; lr; cached services and contents in RSU; Popularity of service/content;
Set $episode = 0$;
Set $P_U, \alpha = 0.5$;
Repeat
Selecting an action randomly with probability ε, or take $\arg\max_{a \in \mathcal{A}} Q(s, a|\theta)$ with probability $1 - \varepsilon$;
Decision $\mathcal{C}$ are obtained according to action a;
Calculating T_{total} and α in UAV;
Obtaining r_{reward}, the next observe $s\_$, and store $(s, a, r, s\_)$ in experience pool;
Randomly sample a batch of $(s, a, r, s\_)$ from experience pool;
$y = r_{reward} + \eta \max_{a \in \mathcal{A}} Q(s\_, a\_|\theta\_)$;
Performing a gradient descend step on $L(\theta)$ with respect to θ;
if Task completed or vehicle leaves the coverage area of RSU **then**
  break;
end if
Update vehicle user's locations, service/content types cached on UAV;
until Convergence;
$episode = episode + 1$;
Output: Optimal caching policies for service/content on UAV $\mathcal{C}$ and α.

In details, the UAV, as an agent, collects the environment information and combines with the task offloading information as the state. The dynamic TDMA model is used to ensure the communications among the vehicles and UAV. The main step of the DQN algorithm is that the deep neural network (DNN) is utilized to find the optimal *Q*-value. In order to speed up the training of the proposed DQN algorithm, we design a two-phase algorithm based on edge-cloud cooperation scheme [38], [39], [40]. In the first offline training phase, the historical experience data is used to train the DNN networks in cloud computing center. And the cloud computing center sends the pre-trained DNN to the UAV. In the second phase, the UAV perform the action selection and obtain the final optimal solutions.

V. RSU-UAV Caching Replacement Scheme

In this section, we consider the hybrid caching replacement scheme for both the RSU and UAV. After the initial hybrid caching in UAV, it provides the communication and computation resources for the covered vehicle users. The caching replacement scheme is needed for the various road conditions and requests. We introduce an evaluation function [41] to determine whether a particular service/content can be cached based on the evaluation values. At fixed intervals, cached service/content in the RSU/UAV can be replaced with service and content with higher cache access hit ratio and improve the quality of the user's experience. The evaluation function consists of three components.

1) The popularity of the service/content caching data, which is obtained based on historical accessed record, as

$$Pop_S^P(\rho, i) = \frac{\frac{1}{i^\rho}}{\sum_i^I \frac{1}{i^\rho}},$$
$$S \in \{S_{1n}, S_{2m}\}, i \in \{n, m\}, I \in \{N, M\} \quad (21)$$

where ρ is the parameter of the Zipf distribution.

2) The size of the service/content caching data in relation to the remaining caching space of the RSU/UAV, as:

$$\lambda_S = \begin{cases} 1, & p_s \leq P_{remainSize}^{RU} \\ 1 - \frac{p_s - P_{remainSize}^{RU}}{P_{RU}}, & p_s > P_{remainSize}^{RU} \\ 0, & p_s \geq P_{remainSize}^{RU} \end{cases} \quad (22)$$

$$P_{remainSize}^{RU} = P_{RU} - P_S^{RU}, \quad (23)$$

where λ_S is the evaluated value if a particular service/content, $P_{remainSize}^{RU}$ denotes the remaining caching space in RSU/UAV, and p_s is the data size, P_S^{RU} is the total data size of the service/content already cached, $S \in \{S_{1n}, S_{2m}\}$.

3) The request frequency of a service/content at the current time interval, as

$$Pop_S^{af} = \frac{R_{q_s}}{t_S^{af}}, \quad S \in \{S_{1n}, S_{2m}\} \quad (24)$$

where R_{q_s} denotes the access number of the service/content during the time period t_S^{af}. In summary, the evaluation function is expressed as

$$F_S^{eva} = \gamma_1 Pop_S^P + \gamma_2 \lambda_S + \gamma_3 Pop_S^{af}, \quad (25)$$

where $\gamma_1, \gamma_2, \gamma_3 \in (0, 1)$, $\gamma_1 + \gamma_2 + \gamma_3 = 1$. Since the edge node in service caching scenario requires computation resources to process tasks, the demand of computation resources for service S_{1n} is set to $Comp_{s_{1n}}^{req}$. For each service S_{1n}, the caching probability increases as $Comp_{s_{1n}}^{req}$ becomes smaller. Setting the service S_{1n} with a constraint latency threshold $T_{s_{1n}}^{res}$. When it becomes smaller, the service data will less be cached. The demand function of service caching is expressed as

$$F_{req} = \frac{T_{s_{1n}}^{req}}{Comp_{s_{1n}}^{req}}. \quad (26)$$

According to the above analysis, the replacement evaluation function for the service and content caching are given as:

$$F_{S_{1n}} = F_S^{eva} F_{req},$$
$$F_{S_{2m}} = F_S^{eva}.$$

A. Problem Formulation

Define *Access hit ratio* to characterize the performance of our proposed approach, which indicates the number of processed tasks compared to the total number of the offloaded tasks in the current time period, as

$$Pop_{ratio}^{hit} = \frac{Num_{hit}}{Num_{offload}}, \quad (27)$$

where Num_{hit} denotes the number of tasks offloaded to the edge node by the vehicle user and processed successfully, and $Num_{offload}$ denotes the number of tasks offloaded to the edge service for the vehicle. The objective of the proposed caching replacement scheme is to maximize the access hit ratio in RSU/UAV, the optimization problem **P2** is shown as:

$$\mathbf{P2}: \max_{\{a_{uav}, a_{rsu}\}} Pop_{ratio}^{hit}$$
$$\text{s.t.} \quad C9: P_S^{RU} \leq P_{RU},$$
$$C10: \gamma_1, \gamma_2, \gamma_3 \in (0, 1), \quad \gamma_1 + \gamma_2 + \gamma_3 = 1, \quad (28)$$

where a_{uav} and a_{rsu} are actions to select service and content respectively, C9 denotes the caching space constraints, C10 denotes the constraints on the weight value parameters.

B. DDQN-Based Algorithm Solution Design

In this subsection, DDQN-based Algorithm is used to find the optimal solutions of **P2**. The optimal replacement strategy is obtained by defining RSU and UAV as two agents that are connected and can update the training model simultaneously. Meanwhile, the basic structure theory of DDQN is essentially the same as DQN described in the previous section, only DDQN is addressed to avoid over-estimating Q-values. Then, the RSU and UAV receive the same reward, the state, action, reward and policy are set as:

State: $s = < s_{uav}, s_{rsu} >$ represents the overall state of the environment, as

$$s_{uav} = < n_{uav}, P_{remainSize}^U, Num_{hit}^{uav}, Eva_{uav}, Pop_{replace}^{uav} >,$$
$$s_{rsu} = < n_{rsu}, P_{remainSize}^R, Num_{hit}^{rsu}, Eva_{rsu}, Pop_{replace}^{rsu} >,$$

where n_{uav} and n_{rsu} denote the number of offloaded tasks from the vehicle users to UAV and RSU. $P_{remainSize}^U$ and $P_{remainSize}^R$ denote the size of the remaining caching spaces of UAV and RSU. Num_{hit}^{uav} and Num_{hit}^{rsu} denote the number of tasks processed on UAV and RSU successfully. Eva_{uav} and Eva_{rsu} denote the evaluation values. $Pop_{replace}^{uav}$ and $Pop_{replace}^{rsu}$ denote the cache replacement ratio on UAV and RSU.

Action: $a = < a_{uav}, a_{rsu} >$, where a_{uav}, a_{rsu} represent the UAV and RSU select some types of service or content, respectively. The selections of UAV and RSU replacement are decided based on the evaluation values of various services or contents evaluated by the evaluation function in the current time period $F_{S_{1n}}, F_{S_{2m}}$. The services and contents with larger evaluation value will be replaced firstly.

Reward: $R = < r_{uav}, r_{rsu} >$ represents the reward of UAV and RSU. We use the cache hit ratio to represent the reward. To express the various system situations effectively, when a vehicle user leaves the coverage range of both the RSU and UAV with

Algorithm 2: Caching Replacement Scheme Based on DDQN-Algorithm.

Initialization: ε; lr; Popularity of services and contents;
Set $episode = 0$;
Repeat
  Obtaining $F_{S_{1n}}$ and $F_{S_{2m}}$ of the already cached service/content data in RSU/UAV;
  Selecting action $a = [a_{uav}, a_{rsu}]$;
  Calculating the evaluation function to obtain a new service S_{1n} or content S_{2m};
  if $p_{s_{1n}}$ or $p_{s_{2m}} \geq P^{RU}_{remainSize}$ **then**
    New service/content be cached directly;
  else
    The service/content with low evaluation values are replaced;
  end if
  Obtaining the replacement ratio and access hit ratio;
  Obtaining $R = [r_{uav}, r_{rsu}]$, update the cache share parameter α;
until The fluctuation range of the access hit ratio is small;
$episode = episode + 1$;
Output: Optimal caching replacement polices $\{a_{uav}, a_{rsu}\}$.

processing tasks, a penalty of a fixed value is given and there is no additional positive reward.

Policy: DDQN compensates for the over-estimation problem of DQN for Q-values. It achieves the optimal actions via the objective function as

$$y_t^{DDQN} = R_{t+1} + \eta Q(s_{t+1}, \eta \max_a Q(s_{t+1}, a|\theta_t)\theta_t^-).$$

The optimal action selections are based on the parameter θ_t of the evaluation network that is currently being updated. The result reduces the overestimation and make the Q-value be closer to the true values. The details are shown in Algorithm 2. Both the UAV and RSU perform as agents, and the proposed two-phase algorithm based on edge-cloud cooperation scheme in hybrid caching scheme is also used here.

VI. Simulation Results

A. Parameter Setting

In this section, number of experimental simulations are proposed to evaluate the designed schemes. We consider a remote area where there is a sudden and severe traffic congestion on a roadway with vehicles moving at an average speed of $2\,\mathrm{m/s}$. ITS center dispatches an UAV to assist RSU for covering the ground vehicle users. The coverage radius of RSU is $500\,\mathrm{m}$ and the storage space is $3000\,\mathrm{Mb}$. The number of vehicle users is 10, the locations are randomly set at the road intersection, each user can generate 10 random tasks and offload them to the RSU or UAV. The number of types of tasks is 20, including the service caching task and content caching one. The UAV hovering height is $50\,\mathrm{m}$, the coverage radius is $300\,\mathrm{m}$ and caching space is $1500\,\mathrm{Mb}$. The hidden layers of the neural network are set as two layers $64 * 64$, the learning rate is 0.005, the discount factor is 0.9. In terms of the simulation software, we used Python 3.9 and Pytorch 1.10.2 to build the DRL and DDQN simulation models. In the hardware part, the computer CPU model is AMD R3600. Other parameters are detailed in Table II.

TABLE II
Default Parameter Setup

Parameter	Value	Parameter	Value
P_0	[1, 6]W	X, Y	11.9, 0.13
P_w^V	[0.1, 1.0]W	η_{LoS}	1.6dBm
σ^2	-96 dBm	η_{NLoS}	25dBm
Φ	10	R_S^{back}	50Mb/GHz
B	[2, 3] MHz	F_U	[2, 2.5]G
B_{cloud}	[20, 25] MHz	F_R	[2.5, 3]G

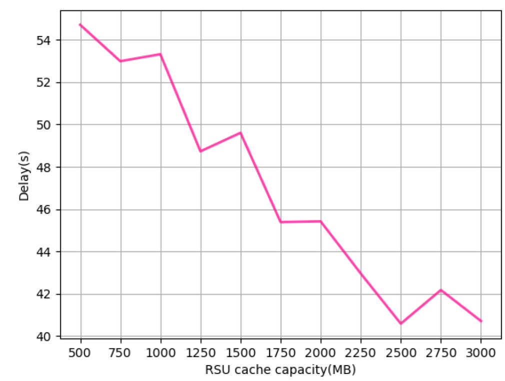

Fig. 3. Comparison of task completion latency of RSU with different caching capacities.

B. Discussion of Simulation Results

a) Hybrid caching: We denote the proposed hybrid caching scheme for UAV as **DQN-based UVA-assisted Scheme** (DUS). To verify the performance of the proposed scheme, we give three benchmarking schemes:

Only-RSU Scheme (ORS): Only the RSU provides service for the vehicle users. Tasks generated by the vehicle can only be offloaded to the RSU and processed on the RSU if the corresponding type of service/content is cached, otherwise they can be further offloaded to CCC via RSU.

Q-learning-based UAV-assisted Scheme (QUS): When the RSU is overloaded, an UAV is dispatched to provide assistance to the RSU, and the UAV caches the service/content via the Q-learning-based algorithm.

Soft Q-learning-based UAV-assisted Scheme (SQUS): Similarly, an UAV is dispatched to provide assistance to the RSU, and the UAV caches the service/content via the Soft Q-learning-based algorithm [42], [43].

Popularity-based UAV-assisted Scheme (PUS): The UAV assists RSU to cover the overloaded areas. The UAV is mixed cached based on the popularity [44], [45] of the service and content.

Fig. 3 shows a comparison of the task completion latency of the RSU in serving vehicles at different caching capacities. The

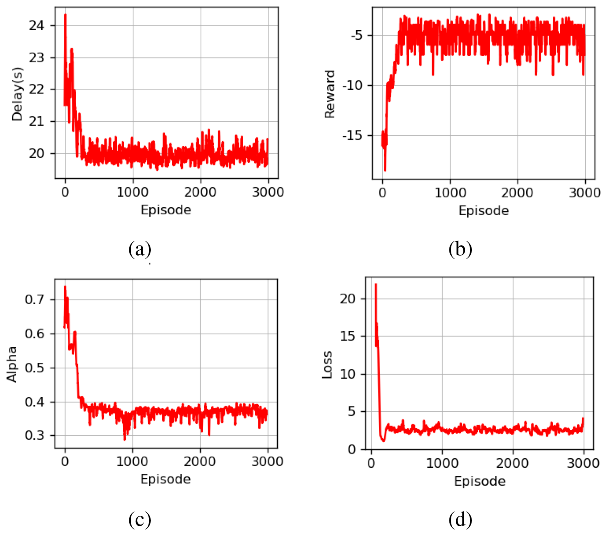

Fig. 4. Convergence of the UAV hybrid caching scheme.

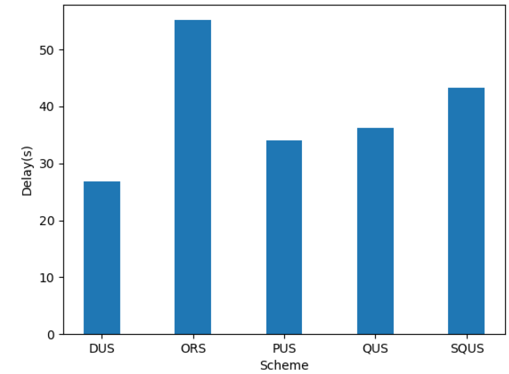

Fig. 5. Comparison of the task completion delay under different schemes.

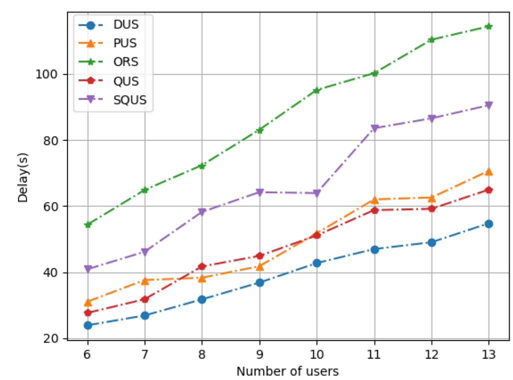

Fig. 6. Comparison of delay with the number of users under different schemes.

caching capacity of RSU increases from $500\,\mathrm{Mb}$ to $3000\,\mathrm{Mb}$ in steps of $250\,\mathrm{Mb}$. From Fig. 3, we can find that the total task completion latency to serve vehicles decreases in a large downward trend as the caching capacity of the RSU increases. When the available caching capacity of RSU is enough, the RSU can cover the vehicle users, and the UAV is not needed. Otherwise, ITS center should dispatch an UAV to assist the RSU for the congested road intersections.

Fig. 4 represents the convergence of the UAV hybrid caching scheme. Fig. 4(a) represents the total task completion delay of the UAV-assisted VECNs to provide network service for the ground users, Fig. 4(b) shows the reward variation during the UAV hybrid caching simulation, Fig. 4(c) shows the convergence of the cache ratio parameters for service and content during the UAV hybrid caching process, and Fig. 4(d) shows the variation of the loss function. As shown in Fig. 4(a), the proposed DUS is able to make the total delay decreases after 300 iterations, and converge to an accepted time range. The results are consistent with our expected requirements for the hybrid caching of UAV. In Fig. 4(c), we can find that the service and content cache ratio parameters also converge, indicating that it is in the best interest of the vehicle to cache the UAV with this cache ratio. Fig. 4(b) and (d) show the other parameters: reward and loss converge over the course of the iterations.

Fig. 5 represents the system task completion delay comparison among the proposed schemes. To evaluate the perform of the proposed DUS, we set the same communication techniques used for all the given schemes. Thus, the transmission time costs are the same. Compared to the other four benchmarking schemes: ORS, PUS, SQUS and QUS, the proposed DUS can better enable UAV to cache more appropriate services and contents, so that the offloaded tasks are processed on the edge node as much as possible, and the total latency is reduced. For QUS and SQUS, the large action and state spaces of the cache policy optimization makes it difficult to converge, and may fluctuate within the upper and lower range of a certain value, and thus the task completion delay is larger than DUS. Moreover, the performance of ORS and PUS are lower, for not all of the associated factors are considered. In the proposed DUS, the UAV can select suitable service and content caching data optimal via the DQN-based algorithm, and the system performance is the best. Similarly, from Fig. 6, we find that the proposed method minimizes the overall service delay as the number of vehicles increases, compared to other methods.

TABLE III
ALGORITHM TIME COMPLEXITY COMPARISON

Algorithm	DUS	ORS	QUS	SQUS	PUS
Time(s)	4.104865	0.596696	1.909851	2.045823	3.499441

We analyze the average time consumption for each iteration in the iterative optimization process. From Table III, it is clear that the time complexity of the proposed DUS is larger compared to the time complexity of other four benchmark algorithms, but from Figs. 5 and 6, we can find that the performance of DUS is the best. The reason is that in DUS, the types of UAV cached services and contents are optimized, making the total system task completed latency be the smallest.

b) Caching replacement: Denote the caching replacement of both RSU and UAV as **DDQN-based Replacement Scheme** (DDRS). The replacement of services and contents cached in RSU and UAV are decided by the vehicle user's access hit ratio

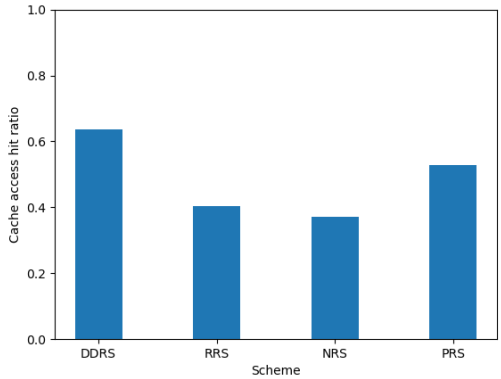

Fig. 7. Comparison of access hit ratio under different schemes.

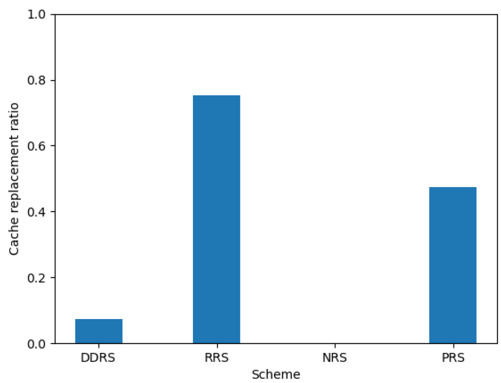

Fig. 8. Comparison of visit replacement ratio under different schemes.

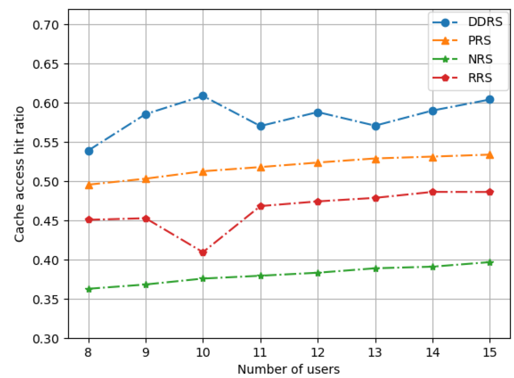

Fig. 9. Comparison of cache hit ratio with the number of users under different schemes.

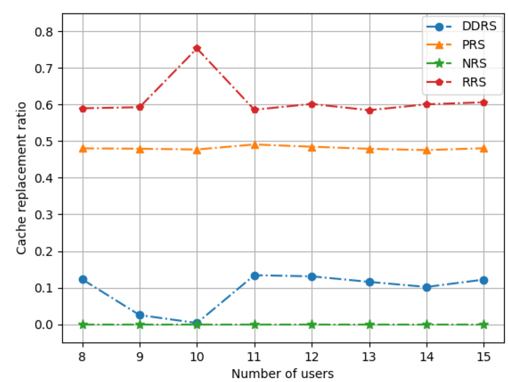

Fig. 10. Comparison of cache replacement ratio with the number of users under different schemes.

and the obtained evaluation values. For comparison, three basic schemes are proposed as:

Random Replacement Scheme (RRS): Random replacement of services and content cached by the RSU and UAV.

No Replacement Scheme (NRS): No replacement of services and content cached in the RSU and UAV, only those types of services and content that were initially cached are maintained.

Popularity Replacement Scheme (PRS): When replacing the caching data in RSU and UAV, the probability of each type of service and content is determined based on its popularity, with higher popularity being more likely to be cached.

In addition to the overall cache replacement evaluation value, we define an additional metric to represent the performance of our proposed method, as:

Cache replacement ratio: It represents the number of successful replacements of service/content over the total number in the current time period, as:

$$Pop_{ratio}^{rep} = \frac{Num_{hit}^{rep}}{Num_{rep}}, \tag{29}$$

where Num_{hit}^{rep} denotes the number of successful replacements of service and content, and Num_{rep} denotes the total number of replacement operations performed.

Fig. 7 shows the comparison of the access hit ratio of the vehicle users under different replacement schemes, Fig. 8 shows the comparison of the replacement ratio in RSU-UAV. In Figs. 7 and 8, it is obvious that the proposed DDRS method has the highest cache hit ratio compared to other. The replacement ratio is zero because the NRS does not perform replacement operations, but the DDRS can better reduce the frequency of replacement operations in the edge server compared to other two schemes. Therefore, the proposed scheme can better reduce the overall task completion delay in the continuous service state of dynamic vehicle users' requests, and thus it can provide better service for vehicle users to meet the comfort experience.

In terms of access hit ratio and replacement ratio, these two metrics change over time as the various tasks, while the proposed scheme allows the hit ratio to fluctuate in a small range above and below a larger value, and the replacement ratio fluctuates in a small range. It allows most of the offloaded tasks are processed directly on the edge node. As vehicle user's task changes over time, the types of task also change accordingly. Under the proposed DDRS, the RSU/UAV provides high quality of service under dynamic users' requests.

Fig. 9 shows the comparison of the cache access hit ratio in RSU/UAV under different number of users, Fig. 10 shows the comparison of the cache replacement ratio. In Fig. 9, we can find that with fixed type of tasks, the cache access hit ratio increases as the number of users increases, due to the increase in the number of tasks of a single type. We can find that the proposed scheme has better results in improving the cache access hit ratio, compared to other three benchmark schemes, and the trend of improvement is greater when excluding the random variability of tasks. In Fig. 10, we can find that the proposed scheme can better reduce the cache replacement ratio compared to the other three schemes. It makes the replacement operation frequency decrease, which better satisfies the vehicle users' various demands and improves the quality of experience of users.

VII. Conclusion

In this article, we propose a hybrid caching and caching replacement scheme for UAV-assisted VECNs to provide network services for the covered vehicle users. UAV performs as a temporary BS to cache services and contents. We design a hybrid caching scheme to satisfy the vehicle users' demands with the limited caching space. An UAV hybrid caching and task offloading scheme based on DQN is proposed, and a joint RSU and UAV caching replacement scheme based on DDQN algorithm is proposed to ensure better service under various requests. Simulation results show that the proposed schemes have good convergence. It can reduce the task completion delay and improve the quality of experience of users.

References

[1] Y. Zhou, F. Tang, Y. Kawamoto, and N. Kato, "Reinforcement learning-based radio resource control in 5G vehicular network," *IEEE Wireless Commun. Lett.*, vol. 9, no. 5, pp. 611–614, May 2020.

[2] X. Jiang, F. R. Yu, T. Song, and V. C. M. Leung, "Resource allocation of video streaming over vehicular networks: A survey, some research issues and challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 5955–5975, Jul. 2022.

[3] Y. Hui et al., "Personalized vehicular edge computing in 6G," *IEEE Netw.*, vol. 35, no. 6, pp. 278–284, Nov./Dec. 2021.

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

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

[6] F. Tang, B. Mao, N. Kato, and G. Gui, "Comprehensive survey on machine learning in vehicular network: Technology, applications and challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 2027–2057, Thirdquarter 2021.

[7] M. A. Javed and S. Zeadally, "AI-empowered content caching in vehicular edge computing: Opportunities and challenges," *IEEE Netw.*, vol. 35, no. 3, pp. 109–115, May/Jun. 2021.

[8] G. Qiao, S. Leng, S. Maharjan, Y. Zhang, and N. Ansari, "Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 247–257, Jan. 2020.

[9] J. Zhao, X. Sun, Q. Li, and X. Ma, "Edge caching and computation management for real-time internet of vehicles: An online and distributed approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2183–2197, Apr. 2021.

[10] K. Zhang, J. Cao, H. Liu, S. Maharjan, and Y. Zhang, "Deep reinforcement learning for social-aware edge computing and caching in urban informatics," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5467–5477, Aug. 2020.

[11] A. A. Khuwaja, Y. Zhu, G. Zheng, Y. Chen, and W. Liu, "Performance analysis of hybrid UAV networks for probabilistic content caching," *IEEE Syst. J.*, vol. 15, no. 3, pp. 4013–4024, Sep. 2021.

[12] Q. Shen, B.-J. Hu, and E. Xia, "Dependency-aware task offloading and service caching in vehicular edge computing," *IEEE Trans. Veh. Technol.*, vol. 71, no. 12, pp. 13182–13197, Dec. 2022.

[13] S. Bi, L. Huang, and Y.-J. A. Zhang, "Joint optimization of service caching placement and computation offloading in mobile edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 4947–4963, Jul. 2020.

[14] X. Ma, A. Zhou, S. Zhang, and S. Wang, "Cooperative service caching and workload scheduling in mobile edge computing," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 2076–2085.

[15] Z. Ning et al., "Distributed and dynamic service placement in pervasive edge computing networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 6, pp. 1277–1292, Jun. 2021.

[16] C. Chen, J. Jiang, R. Fu, L. Chen, C. Li, and S. Wan, "An intelligent caching strategy considering time-space characteristics in vehicular named data networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 19655–19667, Oct. 2022.

[17] Y. Peng et al., "Computing and communication cost-aware service migration enabled by transfer reinforcement learning for dynamic vehicular edge computing networks," *IEEE Trans. Mobile Comput.*, early access, Nov. 28, 2022, doi: 10.1109/TMC.2022.3225239.

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

[19] L. Sun, L. Wan, J. Wang, L. Lin, and M. Gen, "Joint resource scheduling for UAV-enabled mobile edge computing system in Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, early access, Nov. 29, 2022, doi: 10.1109/TITS.2022.3224320.

[20] Y. Wang, M. Chen, C. Pan, K. Wang, and Y. Pan, "Joint optimization of UAV trajectory and sensor uploading powers for UAV-assisted data collection in wireless sensor networks," *IEEE Internet Things J.*, vol. 9, no. 13, pp. 11214–11226, Jul. 2022.

[21] R. Han, Y. Wen, L. Bai, J. Liu, and J. Choi, "Age of information aware UAV deployment for intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2705–2715, Mar. 2022.

[22] X. Liu, B. Lai, B. Lin, and V. C. M. Leung, "Joint communication and trajectory optimization for multi-UAV enabled mobile Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15354–15366, Sep. 2022.

[23] W. Sun, P. Wang, N. Xu, G. Wang, and Y. Zhang, "Dynamic digital twin and distributed incentives for resource allocation in aerial-assisted Internet of Vehicles," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5839–5852, Apr. 2022.

[24] H. Tian et al., "CoPace: Edge computation offloading and caching for self-driving with deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 12, pp. 13281–13293, Dec. 2021.

[25] Z. Ning et al., "Intelligent edge computing in Internet of Vehicles: A joint computation offloading and caching solution," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2212–2225, Apr. 2021.

[26] K. Zhang, J. Cao, S. Maharjan, and Y. Zhang, "Digital twin empowered content caching in social-aware vehicular edge networks," *IEEE Trans. Computat. Social Syst.*, vol. 9, no. 1, pp. 239–251, Feb. 2022.

[27] L. Yao, X. Xu, J. Deng, G. Wu, and Z. Li, "A cooperative caching scheme for VCCN with mobility prediction and consistent hashing," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 20230–20242, Nov. 2022.

[28] Q. Yuan, J. Li, H. Zhou, T. Lin, G. Luo, and X. Shen, "A joint service migration and mobility optimization approach for vehicular edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9041–9052, Aug. 2020.

[29] Y. Liu, S. Wang, M. S. Obaidat, X. Li, and P. Vijayakumar, "Service chain caching and workload scheduling in mobile edge computing," *IEEE Syst. J.*, vol. 16, no. 3, pp. 4389–4400, Sep. 2022.

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

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

[32] A. Al-Hilo, M. Samir, C. Assi, S. Sharafeddine, and D. Ebrahimi, "UAV-assisted content delivery in intelligent transportation systems-joint trajectory planning and cache management," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5155–5167, Aug. 2021.

[33] F. Cheng, G. Gui, N. Zhao, Y. Chen, J. Tang, and H. Sari, "UAV-relaying-assisted secure transmission with caching," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3140–3153, May 2019.

[34] N. Zhao et al., "Caching UAV assisted secure transmission in hyper-dense networks based on interference alignment," *IEEE Trans. Commun.*, vol. 66, no. 5, pp. 2281–2294, May 2018.

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

[36] H. Touati, A. Chriki, H. Snoussi, and F. Kamoun, "Cognitive radio and dynamic TDMA for efficient UAVs swarm communications," *Comput. Netw.*, vol. 196, 2021, Art. no. 108264.
[37] S. Araf, A. S. Saha, S. H. Kazi, N. H. Tran, and M. G. R. Alam, "UAV assisted cooperative caching on network edge using multi-agent actor-critic reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 72, no. 2, pp. 2322–2337, Feb. 2023.
[38] Y. Liu, H. Yu, S. Xie, and Y. Zhang, "Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11158–11168, Nov. 2019.
[39] L. Huang, S. Bi, and Y.-J. A. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Trans. Mobile Comput.*, vol. 19, no. 11, pp. 2581–2593, Nov. 2020.
[40] X. Wei, L. Cai, N. Wei, P. Zou, J. Zhang, and S. Subramaniam, "Joint UAV trajectory planning, DAG task scheduling, and service function deployment based on DRL in UAV-empowered edge computing," *IEEE Internet Things J.*, vol. 10, no. 14, pp. 12826–12838, Jul. 2023.
[41] X. Li, J. Shen, Y. Sun, Z. Wang, and X. Zheng, "A smart content caching and replacement scheme for UAV-assisted fog computing network," in *Proc. Int. Conf. Wireless Commun. Signal Process.*, 2020, pp. 1040–1045.
[42] J. Grau-Moya, F. Leibfried, and H. Bou-Ammar, "Balancing two-player stochastic games with soft Q-learning," 2018, *arXiv:1802.03216*.
[43] C. Yan, Q. Zhang, Z. Liu, X. Wang, and B. Liang, "Control of free-floating space robots to capture targets using soft Q-learning," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2018, pp. 654–660.
[44] T. Fukushima, M. Iio, K. Hirata, and M. Yaoamoto, "Popularity-based content cache management for in-network caching," in *Proc. Int. Conf. Inf. Netw.*, 2019, pp. 411–413.
[45] E. Wang, Q. Dong, Y. Li, and Y. Zhang, "Content placement considering the temporal and spatial attributes of content popularity in cache-enabled UAV networks," *IEEE Wireless Commun. Lett.*, vol. 11, no. 2, pp. 250–253, Feb. 2022.

Yinan Liu received the B.E. degree from Nanchang Hangkong University, Nanchang, China, in 2021. He is currently working toward the M.S. degree with the School of Automation, Guangdong University of Technology, Guangzhou, China. His research interests include wireless communication networks, cooperative communications, and intelligent edge computing.

Chao Yang received the Ph.D. degree in signal and information processing from the South China University of Technology, Guangzhou, China, 2013. He currently with the School of Automation, Guangdong University of Technology, Guangzhou, China. From 2014 to 2016, he was a Research Associate with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His research interests include VANETs, edge computing, and smart grid.

Xin Chen received the Ph.D. degree in bioinformatics from Harbin Medical University, Harbin, China, in 2012. After that, she was a Postdoctoral Fellow with the Faculty of Health Sciences, University of Macau, Macau. Since 2016, she has been with the Institute of Intelligent Information Processing, Guangdong University of Technology, Guangzhou, China. Her research interests focuses on computational biology, wireless Big Data analysis, and convex optimization.

Fengyan Wu received the B.E. degree in 2022 from the Guangdong University of Technology, Guangzhou, China, where she is currently working toward the M.S. degree with the School of Automation. Her research interests include wireless communication networks, cooperative communications, and intelligent edge computing.