


Article

# Deep Reinforcement Learning-Based Resource Allocation for Content Distribution in IoT-Edge-Cloud Computing Environments

Tongke Cui <sup>1</sup>, Ruopeng Yang <sup>1</sup>, Chao Fang <sup>2,\*</sup>  and Shui Yu <sup>3</sup>

<sup>1</sup> College of Information and Communication, National University of Defense Technology, Changsha 410073, China

<sup>2</sup> Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

<sup>3</sup> School of Computer Science, University of Technology Sydney, Sydney, NSW 2007, Australia

\* Correspondence: fangchao@bjut.edu.cn

**Abstract:** With the emergence of intelligent terminals, the Internet of Vehicles (IoV) has been drawing great attention by taking advantage of mobile communication technologies. However, high computation complexity, collaboration communication overhead and limited network bandwidths bring severe challenges to the provision of latency-sensitive IoV services. To overcome these problems, we design a cloud-edge cooperative content-delivery strategy in asymmetrical IoV environments to minimize network latency by providing optimal computing, caching and communication resource allocation. We abstract the joint allocation issue of heterogeneous resources as a queuing theory-based latency minimization objective. Next, a new deep reinforcement learning (DRL) scheme works in each network node to achieve optimal content caching and request routing on the basis of the perceptive request history and network state. Extensive simulations show that our proposed strategy has lower network latency compared with the current solutions in the cloud-edge collaboration system and converges fast under different scenarios.

**Keywords:** cloud-edge cooperation; queuing theory; deep reinforcement learning; in-networking caching; content popularity



**Citation:** Cui, T.; Yang, R.; Fang, C.; Yu, S. Deep Reinforcement Learning-Based Resource Allocation for Content Distribution in IoT-Edge-Cloud Computing Environments. *Symmetry* **2023**, *15*, 217. <https://doi.org/10.3390/sym15010217>

Academic Editor: Jose Carlos R. Alcantud

Received: 29 November 2022

Revised: 18 December 2022

Accepted: 24 December 2022

Published: 12 January 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

High-speed interconnection between devices and networks becomes possible by applying the developed wireless communication technologies in intelligent terminals. To ensure the effectiveness and safety of vehicle driving, it is urgent to develop a more sustainable transportation system [1]. As a new paradigm, the Internet of Vehicles (IoV) has been drawing great attention and is supported by the ubiquitous perception and connection capabilities of the Internet of Things (IoT) [2]. The IoV can provide efficient and low-latency transmission services by fast-exchanging vehicle information over the network [3]. However, the communication in the IoV is affected by road layouts, obstacles and dynamic transportation environments [4]. Moreover, high computational complexity, collaboration communications and limited bandwidths bring severe challenges to the provision of latency-sensitive IoV services. Therefore, how to ensure low-latency network communication in asymmetrical IoV systems is becoming a crucial issue [5].

Although cloud computing can efficiently cope with high-complexity vehicular computing tasks on the basis of its powerful processing and caching capabilities, this leads to the high latency problem as all the requests have to be routed to cloud servers to proceed [6]. In addition, the extra delay caused by the vehicle mobility cannot be ignored and has a direct influence on the transportation performance and safety [7,8]. Multi-access edge computing (MEC) has recently been widely studied to reduce the computation and transmission latency in IoV systems by deploying caching and computing resources in

road side units (RSUs) and base stations (BSs) and satisfying end-user requests in the edge servers [9,10]. To strengthen the connectivity between vehicles, related transport protocols can be designed to shorten the communication time in the MEC-aided IoV [11,12]. Moreover, the adopted in-network storage and offloading schemes can significantly reduce the network latency and handle the challenges caused by mobile vehicles [13,14]. In addition, the introduction of deep reinforcement learning (DRL) to MEC-enabled IoV environments can intelligently allocate network resources to improve transportation performance [15,16].

The constrained service capacities of the MEC-assisted IoV system make it difficult to satisfy the massive requirements from mobile vehicles. Therefore, cloud-edge collaborative computing is considered to leverage their merits, where network services can be partitioned and offloaded to leverage MEC and cloud resources [17,18]. Although current cloud-edge cooperation offloading schemes in the IoV system improve communication delay and reliability by partitioning tasks and optimizing resource allocation, the cross-layer cooperative caching and routing problem is largely ignored. In addition, the challenges caused by the heterogeneous IoV environments have not been discussed in depth. In the article, we design a DRL-based cloud-edge collaborative resource allocation strategy in a heterogeneous IoV system where asymmetrical network environments are considered. The proposed solution can reduce network latency and improve content delivery by realizing optimal computing, caching and communication resource allocation.

The key innovations of this paper can be summarized as:

- We tackle the joint resource allocation issue by minimizing network delay, where cross-layer cooperative content caching and request routing are designed to improve the content distribution and network quality of service (QoS) in the asymmetrical IoV environment, including RSUs, BSs and the cloud.
- We propose a new deep Q network (DQN) policy to handle the proposed delay optimization issue by making content caching and request routing decisions on the basis of the perceptive request history and network state.
- The performance of our solution is evaluated in different system conditions. Extensive real data-based simulations show that our proposed strategy has lower network latency compared with the current solutions in the cloud-edge collaboration system. In addition, the proposed DQN model can adapt to the changes of network states and user requirements and achieve fast convergence.

The remainder of the paper is structured as follows. In Section 2, the related delay-sensitive resource allocation work in the IoV environment is reviewed under the MEC-aided and cloud-edge collaboration scenarios. The delay minimization model is formulated in Section 3. In Section 4, the latency optimization objective is tackled by utilizing the proposed DQN scheme. Simulation is conducted and discussed in Section 5. Finally, our paper is concluded in Section 6.

## 2. Related Work

In this part, the related delay-sensitive resource allocation work is summarized from the perspectives of MEC-aided and cloud-edge collaboration IoV scenarios.

### 2.1. Delay-Sensitive Resource Allocation in Multi-Access Edge Computing

In order to handle a large number of high-complexity computing tasks in the IoV, cloud computing is considered as the initial solution because of its powerful processing and caching capabilities [6]. Zhang et al. [19] designed a novel IoV system under a joint cloud environment, where cloud vendors cooperatively operated to overcome the scalability problem caused by large-scale vehicle data processing. Chaqfeh et al. [20] presented a cloud-assisted IoV paradigm to improve the transportation performance by collecting vehicle knowledge in real time.

However, high latency will be caused in the cloud computing framework due to the long transmission distance between the cloud and vehicles, which makes it difficult to ensure the lower delay required in the IoV system. Moreover, the extra delay incurred

by the high speed vehicle movement cannot be ignored, which directly affects the system performance [7,8]. MEC-assisted mobile vehicles can reduce the computing and transmitting latency by satisfying content requests from the vehicles in the edge servers [9]. Cao et al. [10] proposed an edge-computing-aided IoV framework by choosing the quality of experience (QoE)-based vehicles to reduce service latency.

Given that wireless channels are dynamic and complicated, packet loss and distortion are inevitable during the data transmission in IoV environments [21,22]. To strengthen the connectivity between vehicles, the target transport protocols are designed to shorten the communication latency in the MEC-enhanced IoV. Hadded et al. [11] utilized a TDMA-like MAC protocol to evaluate the media access mechanism in a centralized way and improved the access efficiency and network delay by dynamically constructing vehicle clusters. Das et al. [12] designed a communication embrace collision strategy in the IoV scene, which avoided huge communication overhead among vehicles and promoted content delivery. In addition, the introduced in-network caching policies in the IoV can significantly reduce network latency and cope with the challenges caused by the mobile vehicles. Chen et al. [13] thoroughly summarized content caching solutions in the vehicle naming data network (VNDN) framework, where cache choice and replacement policies were adopted to improve the data transmission time. Kang et al. [14] analyzed how to store data packets in advance at the edge nodes of the IoV system to improve the end-user QoE, which supported seamless switching between edge servers and an increased data transmission rate. Zhang et al. [23] proposed an on-demand adaptive caching policy in the IoV, which updated stored files according to the dynamic changes of content popularity. Moreover, cooperative offloading schemes can be exploited in the MEC-enabled vehicular environments to promote efficiency of task scheduling and response latency while ensuring load balance [7,8].

Considering the merits of artificial intelligence in data process and analysis, the utilization of DRL to MEC-assisted IoV systems can improve the network performance by achieving smart resource allocation [15]. Zhou et al. [16] formulated a finite-state Markov model and built a DRL-based IoV system to realize intelligent offloading. Based on the obtained vehicular information, Chen et al. [24] optimized service time by using a DRL-based mobile edge offloading scheme. Zhou et al. [25] presented a new traffic light control scheme, which aggregated information from the adjacent edge servers and made distributed decisions via reinforcement learning (RL). Qi et al. [26] optimized the multi-armed bandit-based calculation task offloading model by adaptively learning knowledge from the neighboring vehicles. Zou et al. [27] proposed a DRL-based double offloading paradigm, which made computation offloading decisions to balance the energy consumption and system delay by smartly allocating the workload among edge nodes.

## 2.2. Delay-Sensitive Resource Allocation in IoT-Edge-Cloud Computing Environments

In MEC-assisted IoV environments, edge coordination can effectively reduce transmission delay, and the redesigned transmission protocols can improve communication reliability. However, the limited service capacities make it difficult to deal with the growing tasks from mobile vehicles. To achieve reliable and low-delay communication, the cooperation problem between the cloud and edge computing is considered, where the tasks are partitioned and offloaded to the adjacent nodes for parallel or sequential execution [17,18]. Recently, some works have studied the field of task partitioning with different working modes and design goals, which can improve network delay while realizing load balance between edge and cloud computing [28,29]. Time-critical tasks are offloaded to MEC servers while other services are assigned to cloud servers [30]. Ren et al. [31] formulated the collaborative allocation problem of cloud-edge resources as a centralized convex optimization model to minimize the service delay of mobile users. Kadhim et al. [32] designed a software-defined IoV architecture to improve network delay by migrating the tasks to edge servers rather than the cloud. In addition, MEC and cloud servers minimized system delay by meeting partitioned tasks in local vehicles [33]. Shen et al. [34] minimized service delay by sharing the overload among network nodes in a cloud-edge cooperation scenario.

How to leverage energy efficiency and delay is a key research issue in cloud-side collaboration environments. To optimize the execution time caused by power-hungry services, Wang et al. [28] jointly optimized the transmission power, calculation speed and task division ratio in different network nodes. Abbasi et al. [35] leveraged the power consumption and network latency by cooperatively distributing workloads among network nodes. Regarding a tradeoff between the task-processing latency and energy consumed by edge terminals, Bozorgchenani et al. [30] discussed the impact of task classification on task offloading, local computing energy efficiency and service time. Li et al. [36] minimized the system cost by making offloading and resource allocation decisions in an IoV-edge-cloud system.

At present, IoT-edge-cloud collaboration has drawn increasing attention, however, is lacking deep investigation and discussion. In addition, the related DRL-aided solutions need to be improved to adapt to dynamic and complicated IoT environments. Rahman et al. [37] designed a DRL-based computation offloading scheme in resources-constrained fog radio access networks, which optimized the system latency by intelligently assigning computation tasks between edge nodes and the cloud, computing capabilities and transmit power of BSs. To minimize the network delay and adapt to dynamic wireless environments, Van et al. [38] utilized the DRL policy to make optimal computation offloading and resource allocation decisions according to raw network states. In [39], a two-timescale DRL model was built in MEC-enabled 5G ultradense networks to jointly optimize offload decisions, resource allocation and storage deployment. Ren et al. [40] designed an intelligent service offloading and migration strategy in the IoV system to optimize system delay, energy efficiency and network throughput.

### 3. System Model

In this part, we first present the network, content popularity and delay models and then formulate the minimal delay problem in the IoV system.

The notations of main variables are in Table 1.

**Table 1.** Notations of main variables.

Symbols	Notations
$N_R, \mathcal{N}_R$	Amount and set of RSUs
$A_i, \mathcal{A}_i$	Number and set of directly connected edge devices of node $i$ in the same layer
$B_i$	Upper access vertex of node $i$
$A_{B_i}, \mathcal{A}_{B_i}$	Number and set of nodes horizontally connecting to $B_i$
$M_i$	Number of mobile vehicles accessed to RSU $i$
$F, \mathcal{F}$	Amount and set of different files
$b_{m,i}, f_{m,i}^k$	Available wireless bandwidth of the link from the $m$ th vehicle to the $i$ th RSU and its traffic for content $k$
$b_{i,j}, f_{i,j}^k$	Available wired bandwidth of the link $l_{i,j}$ and its traffic for content $k$
$C_i$	Caching capacity for node $i$
$\lambda_i, \lambda_c$	Average arriving rate of node $i$ and the cloud
$\mu_i, \mu_c$	Average serving rate of each server in node $i$ and the cloud
$k_{i,s}, k_{c,s}$	Amount of servers in node $i$ and the cloud

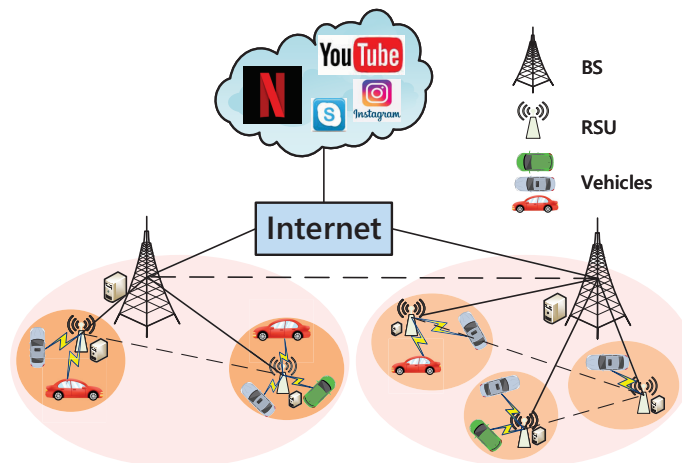
**Table 1.** Cont.

Symbols	Notations
$\rho_i, \rho_c$	Average utilization rate of node $i$ and the cloud
$P_{i,m}, P_{c,n}$	Probability that $n$ requests enter the queuing system of node $i$ and the cloud
$P_{i,Q}, P_{c,Q}$	users' waiting probability in node $i$ and the cloud
$N_{i,Q}, N_{c,Q}$	Amount of requests to process in the queue of node $i$ and the cloud
$T_i^d, T_c^d$	Average response time of node $i$ and the cloud
$\theta_i, \theta_c$	Maximal response latency that node $i$ and the cloud tolerate
$B_{m,i}, B_{i,m}, B_{i,j}$	Maximal bandwidths of the link $l_{m,i}, l_{i,m}$ and $l_{i,j}$

3.1. Network Model

The cloud-edge cooperation IoV system in Figure 1 is asymmetrical due to the multi-layered heterogeneous network paradigm. We abstract the network model to be a directed graph  $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ .  $\mathcal{N}$  is the set of nodes, including all the RSUs and BSs, which are expressed by  $\mathcal{N}_R$  and  $\mathcal{N}_B$ , respectively.  $\mathcal{L}$  is the set of links, where the network link from node  $i$  and  $j$  is denoted as  $l_{i,j}$ . We assume that all the contents are cached in the cloud, while RSUs and BSs have limited caching capabilities. For the sake of simplicity, in this paper, the Greek and non-Greek forms of a variable represent a set and the number of its elements, and we interchangeably replace a RSU or BS with the term “node”.

To fetch an interested file, a car first sends the corresponding request to its accessed RSU. If the content is buffered in the RSU, the file will be sent back to the vehicle. Otherwise, this content request will be forwarded to its adjacent RSUs, the attached BS and its directly connected BSs in sequence. If the request from the vehicle cannot be satisfied in the nodes above, it will fetch the content from the cloud.



**Figure 1.** Network model of the IoV with cloud-edge cooperation.

3.2. File Popularity Model

In our IoV system, the file set is defined as  $\mathcal{F} = \{1, 2, \dots, F\}$ , which indicates that there are  $F$  different kinds of network content. The end-users' request characteristics give the distribution of content popularity certain features. According to the Zipf distribution,

the popularity of network contents declines from 1 to  $F$  [41,42]. Therefore, the probability that a vehicle fetches the content  $k$ ,  $P_k$  is expressed as

$$P_k = \frac{k^{-\alpha}}{\sum_{f=1}^F f^{-\alpha}}, k = 1, 2, \dots, F. \quad (1)$$

A large value of  $\alpha$  means that popular content has a higher request probability in the network.

### 3.3. Delay Model

The latency for the vehicles to obtain their interested contents consists of transmission latency consumed to transmit network data and the sojourn time of service nodes (e.g., RSUs, BSs and the cloud) to proceed with content requests.  $\mathcal{A}_i$  consists of all the RSUs directly connected to the  $i$ th RSU accessed by multiple vehicles denoted as  $\mathcal{M}_i$ .  $B_i$  is the attached BS of RSU  $i$ . Similarly,  $\mathcal{A}_{B_i}$  consists of network nodes horizontally connecting to  $B_i$ .  $X_i^k$  and  $X_{B_i}^k$  are two boolean variables, which are set to be 1 if the two nodes cache file  $k$ . Otherwise, they are set to be 0.

#### 3.3.1. Transmission Delay

There are two kinds of transmission delay models, which depend on the link types in the network. The round-trip delay between the mobile vehicle  $m$  and its accessed RSU  $i$  ( $i \in \mathcal{N}_R$ ) to fetch the content  $k$ , denoted as  $T_{m,i}^{tr,k}$ , can be expressed as

$$T_{m,i}^{tr,k} = \frac{f_{m,i}^k}{b_{m,i}} + \frac{f_{i,m}^k}{b_{i,m}} \quad (2)$$

where  $b_{m,i}$  and  $b_{i,m}$  are the available bandwidths of the wireless links  $l_{m,i}$  and  $l_{j,i}$ , respectively.  $f_{m,i}^k$  and  $f_{i,m}^k$  are their traffic load for content  $k$ .

Similarly, the round-trip delay of the node  $i$  and its neighboring node  $j$  ( $i, j \in \mathcal{N}$ ) in the IoV system to obtain the content  $k$  is written as

$$T_{i,j}^{tr,k} = \frac{f_{i,j}^k}{b_{i,j}} + \frac{f_{j,i}^k}{b_{j,i}} \quad (3)$$

where  $b_{i,j}$  and  $b_{j,i}$  are the available bandwidths of the wired links  $l_{i,j}$  and  $l_{j,i}$ , respectively.  $f_{i,j}^k$  and  $f_{j,i}^k$  are their traffic load for content  $k$ .

#### 3.3.2. Sojourn Delay

The sojourn delay in the IoV system includes the waiting and serving latency caused by a request arriving at a node. Queuing time means the average latency consumed by a request waiting to process in its arriving node, which is determined by the service rate and request arrival rate of the node. The  $M/M/k_s$  queuing system is adopted in the IoV system to present the processing of network requests [43,44].  $\lambda_i$  and  $k_{i,s}$  represent the request arrival rate and the amount of servers for the  $i$ th node, respectively.  $\mu_i$  is the service rate of a server in the  $i$ th node, which depends on the CPU speed and the amount of CPU cycles consumed by requests.

Hence, the utilization rate of node  $i$  is written as

$$\rho_i = \frac{\lambda_i}{k_{i,s}\mu_i} \quad (4)$$

According to the state-transition characteristics in Figure 2, the equilibrium equation of a system in a steady state can be written as

$$\begin{cases} \lambda_i P_{i,n-1} = n\mu_i P_{i,n}, & (1 \leq n < k_{i,s}) \\ \lambda_i P_{i,n-1} = k_{i,s}\mu_i P_{i,n}, & (k_{i,s} \leq n); \end{cases} \tag{5}$$

where  $P_{i,n}$  is the probability that  $n$  requests wait in node  $i$ .

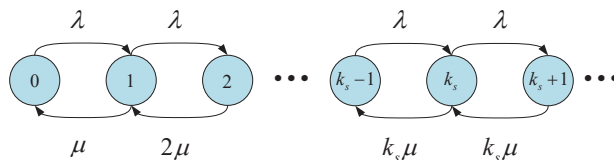


Figure 2. State transition diagram of  $M/M/k_s$  queuing model.

Based on  $\sum_{n=0}^{\infty} P_{i,n} = 1$  and Equation (4), the steady probability that no request waits in node  $i$ , donated as  $P_{i,0}$ , can be expressed as

$$P_{i,0} = \left[ \sum_{n=0}^{k_{i,s}-1} \frac{(k_{i,s}\rho_i)^n}{n!} + \frac{(k_{i,s}\rho_i)^{k_{i,s}}}{(1-\rho_i)k_{i,s}!} \right]^{-1}, \tag{6}$$

and  $P_{i,n}$  can be written as

$$P_{i,n} = \begin{cases} \frac{1}{n!} (k_{i,s}\rho_i)^n P_{i,0}, & (0 < n \leq k_{i,s}) \\ \frac{k_{i,s}^{k_{i,s}}}{k_{i,s}!} \rho_i^n P_{i,0}, & (n > k_{i,s}) \end{cases} \tag{7}$$

When all servers of the  $i$ th node are occupied, its arriving requests have to queue. The corresponding probability can be calculated by

$$P_{i,Q} = \sum_{n=k_{i,s}}^{\infty} P_{i,n} = P_{i,0} \frac{k_{i,s}^{k_{i,s}}}{k_{i,s}!} \frac{\rho_i^{k_{i,s}}}{1-\rho_i} \tag{8}$$

The number of requests in the queue is

$$N_{i,Q} = P_{i,Q} \frac{\rho_i}{1-\rho_i} \tag{9}$$

The average queuing time of requests is

$$T_i^q = \frac{N_{i,Q}}{\lambda_i} = \frac{\rho_i}{\lambda_i(1-\rho_i)} P_{i,Q} \tag{10}$$

The service time for a content request in the queue refers to the latency fetching the interesting file of an end-user in a node or the cloud. Therefore, the average serving latency can be expressed as

$$T_i^s = \frac{1}{\mu_i} \tag{11}$$

Based on the average queuing and service latency equations in Equations (10) and (11), the sojourn delay for node  $i$ , denoted as  $T_i^d$ , can be written as

$$T_i^d = T_i^q + T_i^s = \frac{\rho_i}{\lambda_i(1-\rho_i)} P_{i,Q} + \frac{1}{\mu_i} \tag{12}$$

Similarly, the sojourn delay in the cloud can be expressed as follows

$$T_c^d = \frac{\rho_c}{\lambda_c(1-\rho_c)} P_{c,Q} + \frac{1}{\mu_c} \quad (13)$$

where  $\rho_c$ ,  $\lambda_c$ ,  $\mu_c$  and  $P_{c,Q}$  are the utilization rate of the cloud, its request arriving and serving rates and the probability that a request waits in the cloud, respectively.

Based on the transmission and sojourn delay models, the latency to obtain the content  $k$  for the  $m$ th vehicle accessed in the  $i$ th RSU, denoted as  $T_{m,i}^k$ , is expressed as Equation (14).

$$T_{m,i}^k = T_{m,i}^{tr,k} + X_i^k T_i^d + (1 - X_i^k) \left\{ \left[ 1 - \prod_{i' \in \mathcal{A}_i} (1 - X_{i'}^k) \right] (T_{i,i'}^{tr,k} + T_{i'}^d) + \prod_{i' \in \mathcal{A}_i} (1 - X_{i'}^k) \left\{ T_{i,B_i}^{tr,k} + X_{B_i}^k T_{B_i}^d + (1 - X_{B_i}^k) \left\{ \left[ 1 - \prod_{j \in \mathcal{A}_{B_i}} (1 - X_j^k) \right] (T_{B_i,j}^{tr,k} + T_j^d) + \prod_{j \in \mathcal{A}_{B_i}} (1 - X_j^k) (T_{B_i,c}^{tr,k} + T_c^d) \right\} \right\} \right\} \quad (14)$$

### 3.4. Problem Formulation

In this paper, the joint optimization problem of heterogeneous resources can be formulated as a queuing theory-based latency-minimization model, where cross-layer collaborative caching and routing are considered in an asymmetrical IoV environment. Therefore, the minimal delay problem of the IoV system is written as

$$\begin{aligned} \text{Min} \quad & \sum_{i=1}^{N_R} \sum_{m=1}^{M_i} \sum_{k=1}^F T_{m,i}^k \\ \text{s.t.} \quad & C_1 : \sum_{k=1}^F X_i^k s^k \leq C_i, \forall i \in \mathcal{N} \\ & C_2 : \sum_{m=1}^{M_i} \sum_{k=1}^F f_{m,i}^k \leq B_{m,i}, \sum_{m=1}^{M_i} \sum_{k=1}^F f_{i,m}^k \leq B_{i,m}, \\ & \quad \forall i \in \mathcal{N}_R \\ & C_3 : \sum_{k=1}^F f_{i,j}^k \leq B_{i,j}, \forall i \in \mathcal{N}, j \in \mathcal{A}_i \cup B_i \\ & C_4 : T_i^d \leq \theta_i, T_c^d \leq \theta_c, \forall i \in \mathcal{N} \\ & C_5 : X_i^k \in \{0, 1\}, \forall i \in \mathcal{N}, k \in \mathcal{F} \\ & C_6 : \rho_i \leq 1, \rho_c \leq 1, \forall i \in \mathcal{N} \end{aligned} \quad (15)$$

where  $\theta_c$  and  $\theta_i$  are the upper limits of the response latency caused by edge nodes and the cloud, respectively.  $s^k$  is the size of file  $k$ .  $C_i$  represents the maximal caching capacity node  $i$  provides.  $B_{m,i}$ ,  $B_{i,m}$  and  $B_{i,j}$  are the bandwidth capacities of the link  $l_{m,i}$ ,  $l_{i,m}$  and  $l_{i,j}$ , respectively.

$C_1$  indicates that the caching contents in a node cannot exceed its cache capacity.  $C_2$  and  $C_3$  require that the traffic on a link must be less than its maximal bandwidth.  $C_4$  presents that the response latency consumed by the cloud and other network devices cannot exceed the maximal latency that they can tolerate, which ensures end-user QoE.  $C_5$  means that the caching decision variable  $X_i^k$  only takes the value of 0 or 1.  $C_6$  indicates that the utilization of RSUs, BSs and the cloud cannot exceed their maximal serving capacities.

## 4. Intelligent Caching and Routing Policy

In our IoV system, the crucial problem is to collaboratively allocate computation, storage and transmission resources to quickly exchange information and make efficient caching and routing decisions, which improves the network latency [45]. Although the RL algorithms can achieve optimal resource allocation by dynamically collecting environment



information, it is difficult to overcome the problems caused by dynamic and heterogeneous scenarios [46]. Given the advantages of DL in network routing [47], the integration of RL and DL can be utilized to handle the dimensionality disaster issue generated by our asymmetrical IoV system [48]. Therefore, we designed a DQN-enabled cross-layer collaborative caching and routing scheme to minimize system latency and improve content distribution.

Evaluate and target networks depicted in Figure 3 are two neural ones with the same structure. Based on the known state information, DQN uses the evaluate network with weighting variable  $\omega$  to estimate the Q function and obtain action values. In our proposed solution, the network state for node  $i$  at the time  $t$  is expressed by  $s_{i,t} = \{\mathbf{X}_{i,t}, \mathcal{G}_t, \mathbf{R}_{i,t}, \forall i \in \mathcal{N}\}$ , where  $\mathbf{X}_{i,t} = \{X_{i,t}^1, \dots, X_{i,t}^k, \dots, X_{i,t}^F, \forall i \in \mathcal{N}, k \in \mathcal{F}\}$  is the caching decision vector for the network contents,  $\mathcal{G}_t$  is the network topology, and  $\mathbf{R}_{i,t} = \{R_{i,t}^1, \dots, R_{i,t}^k, \dots, R_{i,t}^F, \forall i \in \mathcal{N}, k \in \mathcal{F}\}$  is the content request vector from the  $i$ th node at the time  $t$ , respectively.

The action for node  $i$  at the time  $t$  is written as  $a_{i,t} = \{\mathbf{X}_{i,t+1}, \mathbf{n}_{i,t+1}, \forall i \in \mathcal{N}\}$ , where  $\mathbf{n}_{i,t+1} = \{n_{i,t+1}^1, \dots, n_{i,t+1}^k, \dots, n_{i,t+1}^F, \forall i \in \mathcal{N}, k \in \mathcal{F}\}$  is a vector indicating the next hop information for content requests of node  $i$  at the following training cycle. The reward obtained by the  $i$ th node at the time  $t$  is written as  $r_{i,t} = \left\{ \sum_{t=1}^{T_{ep}} \frac{\gamma^{T_{ep}-t}}{T_{m,i}}, \forall i \in \mathcal{N}_R, m \in \mathcal{M}_i \right\}$ , where  $T_{ep}$  is the number of training cycles, and  $\gamma$  is a weighting parameter indicating the influence of previous training rewards on the current process. Therefore, based on Equation (14), we can obtain the delay caused by the accessed vehicles of the  $i$ th RSU, denoted as  $T_{m,i}^k$ , which is written as  $\mathbf{T}_{m,i} = \{T_{m,i}^1, \dots, T_{m,i}^k, \dots, T_{m,i}^F, \forall i \in \mathcal{N}_R, m \in \mathcal{M}_i, k \in \mathcal{F}\}$ , can be obtained from Equation (14).

The workflow of the proposed DQN-based cooperative caching and routing policy is summarized in Algorithm 1. The system first initializes the weighting variable  $\gamma$ , exploration speed  $\varepsilon$ , learning rate, memory replay, size for batch gradient descent  $C_B$ , number of quaternions and the amount of training cycles  $T_{ep}$ . According to the state  $s_{i,t}$ , our evaluated model outputs the Q value  $Q(s_{i,t}, a_{i,t}; \omega)$ . Based on the obtained Q value, our IoV network chooses an action by utilizing the  $\varepsilon$ -greedy policy. Specifically, our system randomly selects an action with the probability  $\varepsilon \in (0, 1)$  or chooses the most valuable action according to the expression  $a_{i,t} = \operatorname{argmax}_{a_{i,t}} Q(s_{i,t}, a_{i,t}; \omega)$  with probability  $1 - \varepsilon$ .

Therefore, the proposed DQN model can obtain rewards on the basis of the the known information while avoiding the local optimality problems to adapt to the dynamic network environments. After executing the selected action  $a_{i,t}$ , node  $i$  can obtain knowledge about the reward  $r_{i,t}$  and the next state  $s_{i,t+1}$ . The training process is terminated if the current request can be satisfied by the caching contents in the next hop node. The quaternion  $(s_{i,t}, a_{i,t}, r_{i,t}, s_{i,t+1})$  is cached in the memory replay, where a small number of samples are randomly chosen as a label to speed up the training process in an independent and identically distributed way. During the process, the weighting variables of our neural models are modified through backpropagation and gradient descent policies. In addition, a loss function is exploited to continuously reduce the deviation between the label and the output result, denoted as the mean square error (MSE), which can be defined as

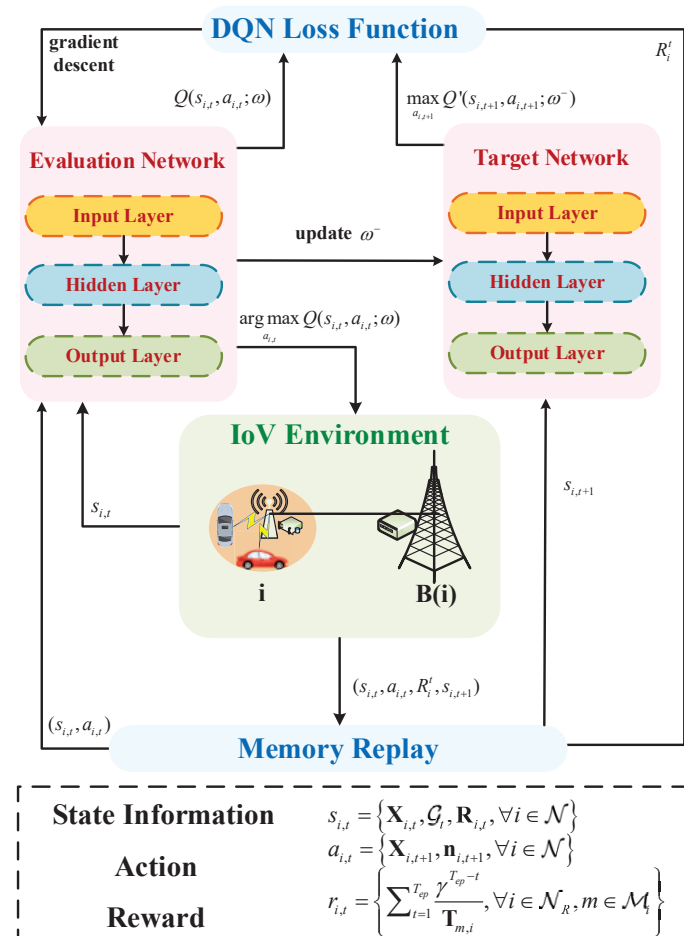
$$L(\omega) = E \left[ \left( r_{i,t} + \gamma \max_{a_{i,t+1}} Q'(s_{i,t+1}, a_{i,t+1}; \omega^-) - Q(s_{i,t}, a_{i,t}; \omega) \right)^2 \right] \quad (16)$$

where  $r_{i,t} + \gamma \max_{a_{i,t+1}} Q'(s_{i,t+1}, a_{i,t+1}; \omega^-)$  is the target Q-value calculated by our target model with the parameter  $\omega^-$ , and  $Q(s_{i,t}, a_{i,t}; \omega)$  is the Q-value achieved by our evaluation model with the parameter  $\omega$ . Our target model provides immovable labels to make the training process converge fast and remain stable. Therefore, the variable update frequency of our target model is lower than that of the evaluation network. To be specific,  $\omega$  is updated at every step while  $\omega^-$  is updated at every fixed step. The proposed DQN algorithm can make optimal caching and routing decisions by making the Q value approximate the target one.

**Algorithm 1** Workflow of the DQN-based cooperative caching and routing algorithm

**Require:** Weighting variable  $\gamma$ , exploration speed  $\varepsilon$ , learning rate, memory replay, size for batch gradient descent  $C_B$ , the number of quaternions, the amount of training cycles  $T_{ep}$

- 1: Initialize replay memory
- 2: Initialize evaluation model with weighted variable  $\omega$
- 3: Initialize target model with weighted variable  $\omega^- = \omega$
- 4: **for** each episode in  $T_{ep}$  **for** node  $i$  **do**
- 5:     Initialize state  $s_{i,1}$
- 6:     **for** each step  $t$  **do**
- 7:         According to the state  $s_{i,t}$ , the  $\varepsilon$ -greedy strategy is adopted in the evaluation network to obtain an action  $a_{i,t}$
- 8:         Execute action  $a_{i,t}$  based on state  $s_{i,t}$  to obtain the next state  $s_{i,t+1}$  and the current reward  $r_{i,t}$  and decide whether the training process is terminated at the step  $t' = t$  according to the caching state of the next hop node
- 9:         Cache quaternion  $(s_{i,t}, a_{i,t}, r_{i,t}, s_{i,t+1})$  in memory replay
- 10:         Randomly select  $C_B$  samples  $(s_{i,t'}, a_{i,t'}, r_{i,t'}, s_{i,t'+1})$  from memory replay
- 11:         **if** training process terminates at step  $t' + 1$  **then**
- 12:             Set  $y_{i,t'} = r_{i,t'}$
- 13:         **else**
- 14:             Set  $y_{i,t'} = r_{i,t'} + \gamma \max_{a_{i,t'+1}} Q'(s_{i,t'+1}, a_{i,t'+1}; \omega^-)$
- 15:         Evaluation network with  $\omega$  operates the gradient descent policy by using MSE function in Equation (16)
- 16:         Replace  $\omega^-$  with  $\omega$  every fixed steps



**Figure 3.** The schematic diagram of our deep reinforcement learning model.

## 5. Simulation Results and Discuss

In this part, we describe the evaluation environment and analyze the simulation results in the asymmetrical IoV network.

### 5.1. Simulation Settings

In this paper, the proposed model is evaluated in a three-layer IoT-edge-cloud network topology. The skewness factor  $\alpha$  related to the content popularity varies from 0.6 to 2 [49,50]. In the simulation, the storage capacity of a node is abstracted as a ratio to  $F$  different kinds of network files. Given that the storage capacity for edge devices in realistic networks is limited, the range of storage capacity is from 0.1% to 1% in our IoV system [51,52].

In the simulation, our proposed “DQN” method is compared with current popular strategies in the IoT-edge-cloud environments, referred to as “Popularity” [53], “LRU” [54] and “Without Cache”, to demonstrate the advantages of our solution. In “DQN”, collaborative caching and routing decisions are made according to the perceptive request history and network state, which can adapt to the changes in network states and user requirements to realize timely and optimal resource allocation. In “Popularity”, the network contents are cooperatively stored in the nodes on the basis of the known file popularity distribution of our IoV system. Specifically, RSUs and their horizontally connecting nodes collaboratively store network files on the basis of the descending rank of file popularity, while BSs store data in a complementary manner with the connected RSUs. Therefore, requests for “hot” files are satisfied by edge nodes while serving those for “cold” files as much as possible. In “LRU”, RSUs and BSs cache the passing data and modify their stored contents by utilizing the least recently used (LRU) policy. “Without Cache” represents the optimal solution where storage capacities are deployed in the RSUs and BSs, and every request is forwarded to the cloud to obtain the corresponding file. Each node can make optimal routing decisions by sharing its network knowledge. In our simulation, the cooperative routing policy mentioned in network model is adopted in all the comparative schemes.

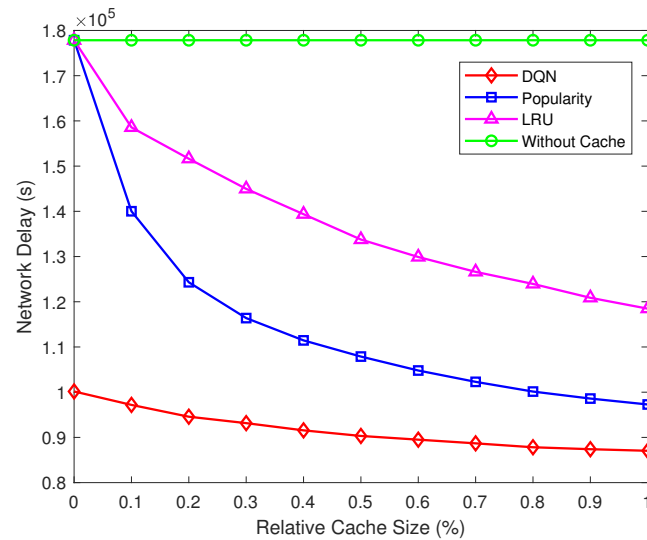
### 5.2. Simulation Results

Figure 4 shows the latency of different solutions when the storage capacities varies. With the growth of storage capacities of RSUs and BSs, more of the files that end-users are interested in are cached to significantly reduce the latency of different strategies with caches. However, our designed “DQN” model performs much better than other solutions due to the timely and intelligent caching and routing decisions. As the cache size grows, the performance gap caused by different caching strategies is narrowed. For “Without Cache”, its performance barely changes because all the requests finally obtain the corresponding contents in the cloud.

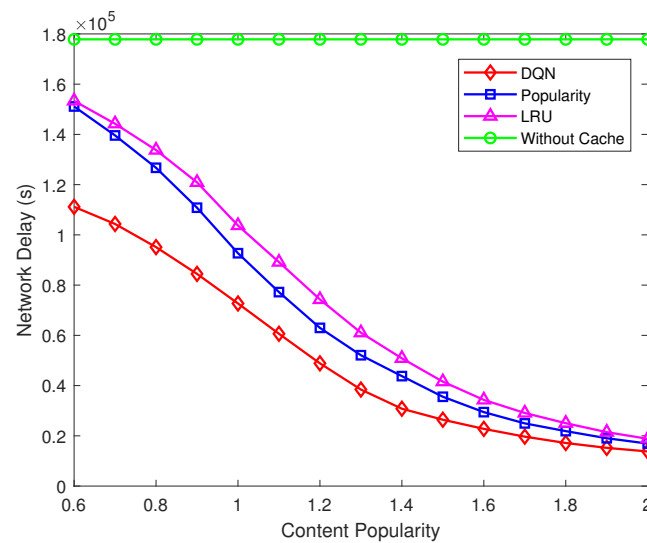
Figure 5 shows the latency of different solutions when the content popularity varies. When the content popularity grows, end-users in the IoV system pay more attention to the popular content, thereby, reducing the network latency of the solutions with caches and bridging their performance gap. Based on the perceptive request history and network state, the “DQN” scheme can make collaborative caching and routing decisions, which gives it better performance compared with “Popularity” and “LRU”. When the value of the parameter  $\alpha$  for the content popularity is small, the static cooperative caching in “Popularity” leads to a high routing overhead, and it is difficult to represent the dynamics and differences of the accessed content requests, therefore, bridging the gap between “Popularity” and “LRU”. Moreover, content requests in “Without Cache” fetch the targeted files in the cloud, which means that its performance does not greatly vary.

Figure 6 shows the latency of different solutions when the content diversity changes. As the amount of different content grows, we can see that the performance of solutions with caches declines and that their performance gap is enlarged in Figure 6. The growth of content diversity in the network means that the amount of requests for popular files are reduced, which deteriorates the cache hit rate and increases the network latency. Based on the information of user requests, available resources and caching contents, the proposed

“DQN” model can always achieve optimal caching and routing to adapt to the changes of network environments and user preference, which makes it perform much better than other schemes. In “Without Cache”, each request is routed to the cloud to obtain the data, which leaves its performance unchanged.

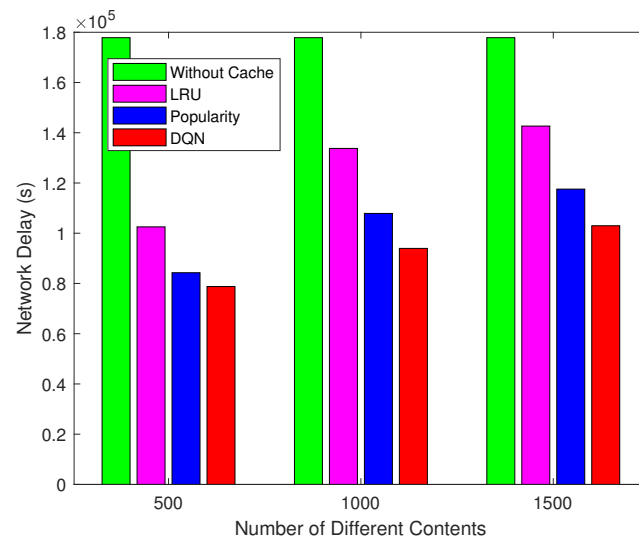


**Figure 4.** Network delay versus cache size.

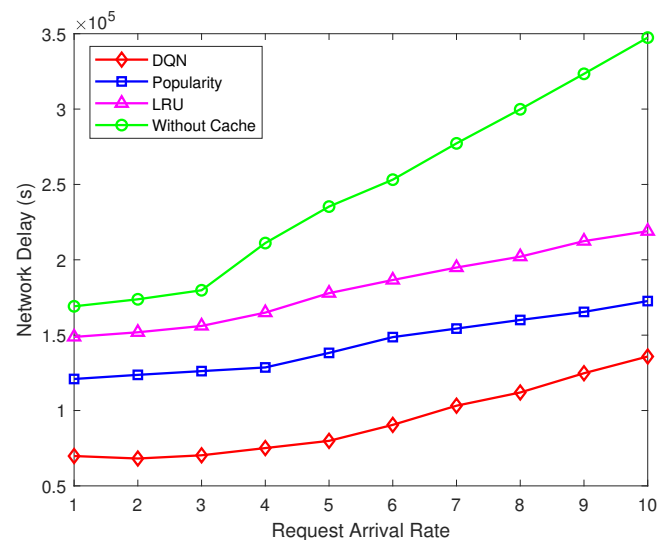


**Figure 5.** Network delay versus content popularity.

Figure 7 shows the latency of different solutions when the request arrival rates vary. Due to the constraint service capacity and network resources, the increase of request arrival rates indicates that more requests are queued and lost in the network, thereby, resulting in a significant growth of network latency for the four strategies. However, the proposed “DQN” policy still performs better than other schemes with larger request arrival rates. The reason is that the increasing content requests arriving at each node can improve the predictive accuracy for caching and routing decisions.



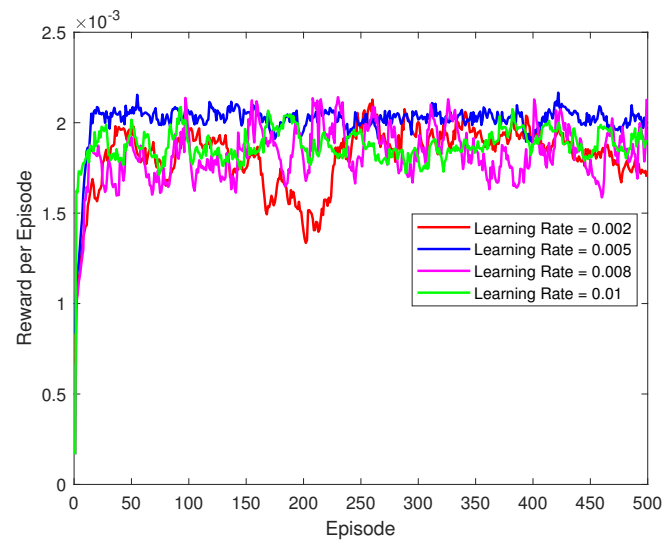
**Figure 6.** Network delay versus the number of different contents.



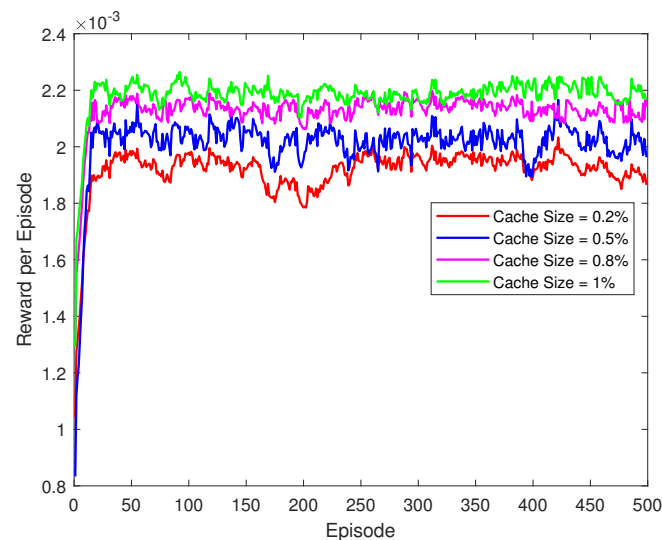
**Figure 7.** Network delay versus request arrival rate.

Figure 8 shows the accumulated reward of “DQN” in each episode when the learning rates vary. Our proposed model always converges fast when the learning rates change. As shown in Figure 8, a large value of the learning rate in the proposed “DQN” model indicates that the occupied proportion of learned information in the whole knowledge increases. However, a large learning rate does not mean better convergence performance and higher accumulated reward. When the learning rate was 0.005, the system achieved the best performance. Therefore, existing and newly explored results must be taken into consideration when choosing the value for the learning rate.

Figure 9 shows the accumulated reward of “DQN” in each episode when storage capacities vary. The reward value obtained by “DQN” increases when the cache capacities grow. This is because a larger storage capacity can improve the caching and routing decisions, which will further reduce the network delay. However, the fluctuation degree is different when the storage capacities change. The reason is that the greedy strategy of “DQN” has random characteristics, and the information obtained by partial cycles is incomplete.



**Figure 8.** Reward versus learning rate.



**Figure 9.** Reward versus cache size.

## 6. Summary and Future Work

In this article, we designed a cloud-edge cooperative content-delivery scheme in an asymmetrical IoV network to improve the latency by jointly optimizing the computing, caching and communication resources. We first formulated the joint optimization issue of heterogeneous resources as a queuing-theory-based delay model. Then, based on the request history information and currently available network resources in the system, a new reinforcement learning policy was proposed to make collaborative caching and routing decisions by predicting the content popularity. Finally, we discussed the performance of the proposed solution in different network scenarios. Extensive simulations demonstrated that our designed strategy performed much better than the current common policies. The proposed model can adapt to the changes in network states and user requirements and quickly converge to a stable state.

In future work, the end-user mobility problem will be investigated and modeled to improve the proposed mechanism in a more complex network environment. In addition, in-depth collaboration among mobile users, edge and cloud networks will be considered to promote the QoE of network terminals. Finally, the tradeoff between network delay and energy consumption will be discussed to compromise among multiple network indicators.

**Author Contributions:** Conceptualization, C.F. and T.C.; methodology, C.F. and R.Y.; software, C.F. and T.C.; validation, C.F. and T.C.; formal analysis, C.F., T.C. and S.Y.; investigation, C.F.; resources, C.F. and S.Y.; data curation, C.F.; writing—original draft preparation, C.F., T.C., R.Y. and S.Y.; writing—review and editing, C.F., T.C., R.Y. and S.Y.; supervision, C.F., R.Y. and S.Y.; project administration, C.F.; funding acquisition, C.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is partially supported by the Beijing Nova Program of Science and Technology Z191100001119094, Urban Carbon Neutral Science and Technology Innovation Fund Project of Beijing University of Technology (040000514122607) and the Beijing Natural Science Foundation L202016.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Qureshi, K.N.; Din, S.; Jeon, G.; Piccialli, F. Internet of Vehicles: Key Technologies, Network Model, Solutions and Challenges With Future Aspects. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 1777–1786. [\[CrossRef\]](#)
2. Li, X.; Zheng, Y.; Alshehri, M.D.; Hai, L.; Balasubramanian, V.; Zeng, M.; Nie, G. Cognitive AmBC-NOMA IoV-MTS Networks With IQI: Reliability and Security Analysis. *IEEE Trans. Intell. Transp. Syst.* **2021**, 1–12. [\[CrossRef\]](#)
3. Singh, P.K.; Nandi, S.K.; Nandi, S. A tutorial survey on vehicular communication state of the art, and future research directions. *Veh. Commun.* **2019**, *18*, 100164. [\[CrossRef\]](#)
4. Cooper, C.; Franklin, D.; Ros, M.; Safaei, F.; Abolhasan, M. A Comparative Survey of VANET Clustering Techniques. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 657–681. [\[CrossRef\]](#)
5. Qureshi, K.N.; Abdullah, A.H.; Lloret, J.; Altameem, A. Road-aware routing strategies for vehicular ad hoc networks: Characteristics and comparisons. *Int. J. Distrib. Sens. Netw.* **2016**, *12*, 1605734. [\[CrossRef\]](#)
6. Luo, X.; Liu, Y.; Chen, H.H.; Meng, W. Artificial Noise Assisted Secure Mobile Crowd Computing in Intelligently Connected Vehicular Networks. *IEEE Trans. Veh. Technol.* **2021**, *70*, 7637–7651. [\[CrossRef\]](#)
7. Hu, X.; Wong, K.K.; Yang, K.; Zheng, Z. UAV-Assisted Relaying and Edge Computing: Scheduling and Trajectory Optimization. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 4738–4752. [\[CrossRef\]](#)
8. Cheng, N.; Lyu, F.; Quan, W.; Zhou, C.; He, H.; Shi, W.; Shen, X. Space/Aerial-Assisted Computing Offloading for IoT Applications: A Learning-Based Approach. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1117–1129. [\[CrossRef\]](#)
9. Liu, Y.; Wang, W.; Chen, H.H.; Lyu, F.; Wang, L.; Meng, W.; Shen, X. Physical Layer Security Assisted Computation Offloading in Intelligently Connected Vehicle Networks. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 3555–3570. [\[CrossRef\]](#)
10. Cao, Y.; Chen, Y. QoE-based node selection strategy for edge computing enabled Internet-of-Vehicles (EC-IoV). In Proceedings of the 2017 IEEE Visual Communications and Image Processing (VCIP), St. Petersburg, FL, USA, 10–13 December 2017; pp. 1–4. [\[CrossRef\]](#)
11. Hadded, M.; Muhlethaler, P.; Laouiti, A.; Zagrouba, R.; Saidane, L.A. TDMA-Based MAC Protocols for Vehicular Ad Hoc Networks: A Survey, Qualitative Analysis, and Open Research Issues. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2461–2492. [\[CrossRef\]](#)
12. Das, T.; Chen, L.; Kundu, R.; Bakshi, A.; Sinha, P.; Srinivasan, K.; Bansal, G.; Shimizu, T. CoReCast: Collision resilient broadcasting in vehicular networks. In Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services, Munich, Germany, 10–15 June 2018; pp. 217–229.
13. Chen, C.; Wang, C.; Qiu, T.; Atiquzzaman, M.; Wu, D.O. Caching in Vehicular Named Data Networking: Architecture, Schemes and Future Directions. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2378–2407. [\[CrossRef\]](#)
14. An, K.; Yan, X.; Liang, T.; Lu, W. Mobility Prediction Based Vehicular Edge Caching: A Deep Reinforcement Learning Based Approach. In Proceedings of the 2019 IEEE 19th International Conference on Communication Technology (ICCT), Xi'an, China, 16–19 October 2019; pp. 1120–1125. [\[CrossRef\]](#)
15. Xu, X.; Li, H.; Xu, W.; Liu, Z.; Yao, L.; Dai, F. Artificial intelligence for edge service optimization in Internet of Vehicles: A survey. *Tsinghua Sci. Technol.* **2022**, *27*, 270–287. [\[CrossRef\]](#)
16. Zhou, J.; Wu, F.; Zhang, K.; Mao, Y.; Leng, S. Joint optimization of Offloading and Resource Allocation in Vehicular Networks with Mobile Edge Computing. In Proceedings of the 2018 tenth International Conference on Wireless Communications and Signal Processing (WCSP), Hangzhou, China, 10–20 October 2018; pp. 1–6. [\[CrossRef\]](#)
17. Sharma, S.; Kaushik, B. A survey on internet of vehicles: Applications, security issues & solutions. *Veh. Commun.* **2019**, *20*, 100182.
18. Yang, L.; Cao, J.; Cheng, H.; Ji, Y. Multi-User Computation Partitioning for Latency Sensitive Mobile Cloud Applications. *IEEE Trans. Comput.* **2015**, *64*, 2253–2266. [\[CrossRef\]](#)
19. Zhang, Y.; Zhang, M.; Wo, T.; Lin, X.; Yang, R.; Xu, J. A Scalable Internet-of-Vehicles Service over Joint Clouds. In Proceedings of the 2018 IEEE Symposium on Service-Oriented System Engineering (SOSE), Bamberg, Germany, 26–29 March 2018; pp. 210–215. [\[CrossRef\]](#)
20. Chaqfeh, M.; Mohamed, N.; Jawhar, I.; Wu, J. Vehicular Cloud data collection for Intelligent Transportation Systems. In Proceedings of the 2016 third Smart Cloud Networks Systems (SCNS), Dubai, United Arab Emirates, 19–21 December 2016; pp. 1–6. [\[CrossRef\]](#)

21. Fang, C.; Yao, H.; Wang, Z.; Wu, W.; Jin, X.; Yu, F.R. A Survey of Mobile Information-Centric Networking: Research Issues and Challenges. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2353–2371. [[CrossRef](#)]
22. Fang, C.; Guo, S.; Wang, Z.; Huang, H.; Yao, H.; Liu, Y. Data-Driven Intelligent Future Network: Architecture, Use Cases, and Challenges. *IEEE Commun. Mag.* **2019**, *57*, 34–40. [[CrossRef](#)]
23. Zhang, Y.; Li, C.; Luan, T.H.; Yuen, C.; Fu, Y.; Wang, H.; Wu, W. Towards Hit-Interruption Tradeoff in Vehicular Edge Caching: Algorithm and Analysis. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 5198–5210. [[CrossRef](#)]
24. Chen, X.; Jiao, L.; Li, W.; Fu, X. Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing. *IEEE/ACM Trans. Netw.* **2016**, *24*, 2795–2808. [[CrossRef](#)]
25. Zhou, P.; Chen, X.; Liu, Z.; Braud, T.; Hui, P.; Kangasharju, J. DRLE: Decentralized Reinforcement Learning at the Edge for Traffic Light Control in the IoV. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 2262–2273. [[CrossRef](#)]
26. Qi, Q.; Wang, J.; Ma, Z.; Sun, H.; Cao, Y.; Zhang, L.; Liao, J. Knowledge-Driven Service Offloading Decision for Vehicular Edge Computing: A Deep Reinforcement Learning Approach. *IEEE Trans. Veh. Technol.* **2019**, *68*, 4192–4203. [[CrossRef](#)]
27. Zou, J.; Hao, T.; Yu, C.; Jin, H. A3C-DO: A Regional Resource Scheduling Framework Based on Deep Reinforcement Learning in Edge Scenario. *IEEE Trans. Comput.* **2021**, *70*, 228–239. [[CrossRef](#)]
28. Wang, Y.; Sheng, M.; Wang, X.; Wang, L.; Li, J. Mobile-Edge Computing: Partial Computation Offloading Using Dynamic Voltage Scaling. *IEEE Trans. Commun.* **2016**, *64*, 4268–4282. [[CrossRef](#)]
29. You, C.; Huang, K.; Chae, H.; Kim, B.H. Energy-Efficient Resource Allocation for Mobile-Edge Computation Offloading. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 1397–1411. [[CrossRef](#)]
30. Bozorgchenani, A.; Tarchi, D.; Corazza, G.E. Centralized and Distributed Architectures for Energy and Delay Efficient Fog Network-Based Edge Computing Services. *IEEE Trans. Green Commun. Netw.* **2019**, *3*, 250–263. [[CrossRef](#)]
31. Ren, J.; Yu, G.; He, Y.; Li, G.Y. Collaborative cloud and edge computing for latency minimization. *IEEE Trans. Veh. Technol.* **2019**, *68*, 5031–5044. [[CrossRef](#)]
32. Kadhim, A.J.; Naser, J.I. Proactive load balancing mechanism for fog computing supported by parked vehicles in IoV-SDN. *China Commun.* **2021**, *18*, 271–289. [[CrossRef](#)]
33. Feng, M.; Krunz, M.; Zhang, W. Joint Task Partitioning and User Association for Latency Minimization in Mobile Edge Computing Networks. *IEEE Trans. Veh. Technol.* **2021**, *70*, 8108–8121. [[CrossRef](#)]
34. Shen, B.; Xu, X.; Dai, F.; Qi, L.; Zhang, X.; Dou, W. Dynamic Task Offloading with Minority Game for Internet of Vehicles in Cloud-Edge Computing. In Proceedings of the 2020 IEEE International Conference on Web Services (ICWS), Beijing, China, 19–23 October 2020; pp. 372–379. [[CrossRef](#)]
35. Abbasi, M.; Yaghoobikia, M.; Rafiee, M.; Khosravi, M.R.; Menon, V.G. Optimal Distribution of Workloads in Cloud-Fog Architecture in Intelligent Vehicular Networks. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 4706–4715. [[CrossRef](#)]
36. Li, Y.; Xu, S. Collaborative optimization of Edge-Cloud Computation Offloading in Internet of Vehicles. In Proceedings of the 2021 International Conference on Computer Communications and Networks (ICCCN), Online, 19–22 July 2021; pp. 1–6. [[CrossRef](#)]
37. Rahman, G.M.S.; Dang, T.; Ahmed, M. Deep reinforcement learning based computation offloading and resource allocation for low-latency fog radio access networks. *Intell. Converge. Netw.* **2020**, *1*, 243–257. [[CrossRef](#)]
38. Dat Tuong, V.; Phung Truong, T.; Tran, A.T.; Masood, A.; Shumeye Lakew, D.; Lee, C.; Lee, Y.; Cho, S. Delay-Sensitive Task Offloading for Internet of Things in Nonorthogonal Multiple Access MEC Networks. In Proceedings of the 2020 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 21–23 October 2020; pp. 597–599. [[CrossRef](#)]
39. Yu, S.; Chen, X.; Zhou, Z.; Gong, X.; Wu, D. When Deep Reinforcement Learning Meets Federated Learning: Intelligent Multitimescale Resource Management for Multiaccess Edge Computing in 5G Ultradense Network. *IEEE Internet Things J.* **2021**, *8*, 2238–2251. [[CrossRef](#)]
40. Ren, Y.; Chen, X.; Guo, S.; Guo, S.; Xiong, A. Blockchain-Based VEC Network Trust Management: A DRL Algorithm for Vehicular Service Offloading and Migration. *IEEE Trans. Veh. Technol.* **2021**, *70*, 8148–8160. [[CrossRef](#)]
41. Katsaros, K.; Xylomenos, G.; Polyzos, G.C. MultiCache: An incrementally deployable overlay architecture for information-centric networking. In Proceedings of the Proceedings IEEE INFOCOM'10, San Diego, CA, USA, 14–19 March 2010.
42. Fang, C.; Yu, F.R.; Huang, T.; Liu, J.; Liu, Y. Distributed Energy Consumption Management in Green Content-Centric Networks via Dual Decomposition. *IEEE Syst. J.* **2017**, *11*, 625–636. [[CrossRef](#)]
43. Wu, J.; Zhou, S.; Niu, Z. Traffic-aware base station sleeping control and power matching for energy-delay tradeoffs in green cellular networks. *IEEE Trans. Wirel. Commun.* **2013**, *12*, 4196–4209. [[CrossRef](#)]
44. Wu, J.; Bao, Y.; Miao, G.; Zhou, S.; Niu, Z. Base-station sleeping control and power matching for energy-delay tradeoffs with bursty traffic. *IEEE Trans. Veh. Technol.* **2015**, *65*, 3657–3675. [[CrossRef](#)]
45. Fadlullah, Z.M.; Tang, F.; Mao, B.; Kato, N.; Akashi, O.; Inoue, T.; Mizutani, K. State-of-the-Art Deep Learning: Evolving Machine Intelligence Toward Tomorrow's Intelligent Network Traffic Control Systems. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2432–2455. [[CrossRef](#)]
46. Zhu, H.; Cao, Y.; Wang, W.; Jiang, T.; Jin, S. Deep reinforcement learning for mobile edge caching: Review, new features, and open issues. *IEEE Netw.* **2018**, *32*, 50–57. [[CrossRef](#)]



47. Mao, B.; Fadlullah, Z.M.; Tang, F.; Kato, N.; Akashi, O.; Inoue, T.; Mizutani, K. Routing or Computing? The Paradigm Shift Towards Intelligent Computer Network Packet Transmission Based on Deep Learning. *IEEE Trans. Comput.* **2017**, *66*, 1946–1960. [[CrossRef](#)]
48. Xu, C.; Liu, S.; Zhang, C.; Huang, Y.; Lu, Z.; Yang, L. Multi-Agent Reinforcement Learning Based Distributed Transmission in Collaborative Cloud-Edge Systems. *IEEE Trans. Veh. Technol.* **2021**, *70*, 1658–1672. [[CrossRef](#)]
49. Breslau, L.; Cao, P.; Fan, L.; Phillips, G.; Shenker, S. *Web Caching and Zipf-Like Distributions: Evidence and Implications*; IEEE: Piscataway, NJ, USA, 1999.
50. Choi, N.; Guan, K.; Kilper, D.C.; Atkinson, G. In-network caching effect on optimal energy consumption in content-centric networking. In Proceedings of the IEEE International Conference on Communications (ICC), Ottawa, ON, Canada, 10–15 June 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 2889–2894.
51. Xie, H.; Shi, G.; Wang, P. TECC: Towards collaborative in-network caching guided by traffic engineering. In Proceedings of the 2012 Proceedings IEEE INFOCOM, Orlando, FL, USA, 25–30 March 2012; pp. 2546–2550. [[CrossRef](#)]
52. Li, J.; Liu, B.; Wu, H. Energy-efficient in-network caching for content-centric networking. *IEEE Commun. Lett.* **2013**, *17*, 797–800. [[CrossRef](#)]
53. Wang, X.; Chen, M.; Taleb, T.; Ksentini, A.; Leung, V. Cache in the air: Exploiting content caching and delivery techniques for 5g systems. *IEEE Commun. Mag.* **2014**, *52*, 131–139. [[CrossRef](#)]
54. Kim, Y.; Yeom, I. Performance analysis of in-network caching for content-centric networking. *Comput. Netw.* **2013**, *57*, 2465–2482. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.