




Article

# COPP-DDPG: Computation Offloading with Privacy Preservation in a Vehicular Edge Network

Yancong Wang <sup>1</sup>, Jian Wang <sup>1</sup>, Hongchang Ke <sup>2,\*</sup> and Zemin Sun <sup>1</sup><sup>1</sup> College of Computer Science and Technology, Jilin University, Changchun 130012, China<sup>2</sup> School of Computer Technology and Engineering, Changchun Institute of Technology, Changchun 130012, China

\* Correspondence: kehongchang1981@163.com; Tel.: +86-13500882674

**Abstract:** Vehicular edge computing (VEC) is emerging as a prospective technology in the era of 5G and beyond to support delay-sensitive and computation-intensive vehicular applications. However, designing an efficient approach for joint computation offloading and resource allocation is challenging due to the limited resources of VEC servers, the highly dynamic vehicular networks (VNs), different priorities of vehicular applications, and the threat of privacy disclosure. In this work, we propose a cooperative optimization for privacy-preserving and priority-aware offloading and resource allocation in VEC network (VECN) based on deep reinforcement learning (DRL). Firstly, we employed a privacy-preserving framework where the certificate authority (CA) is integrated into the VEC architecture. Furthermore, we formulated the dynamic optimization problem as a Markov decision process (MDP) by constructing a weighted cost function that integrates the priority of stochastic arrival tasks, privacy-preserving of offloading, and dynamic interaction between the edge servers and intelligent connected vehicles (ICVs). To solve this problem, a cooperative optimization for privacy and priority based on deep deterministic policy gradient (COPP-DDPG) is proposed by learning the optimal actions to minimize the weighted cost function. The simulation results show that COPP-DDPG has good convergence and outperforms the other four comparison algorithms in many aspects.

**Keywords:** computation offloading; DRL; privacy preservation; vehicular edge network



**Citation:** Wang, Y.; Wang, J.; Ke, H.; Sun, Z. COPP-DDPG: Computation Offloading with Privacy Preservation in a Vehicular Edge Network. *Appl. Sci.* **2022**, *12*, 12522. <https://doi.org/10.3390/app122412522>

Academic Editor: Jie Lin

Received: 24 October 2022

Accepted: 4 December 2022

Published: 7 December 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 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

The development of beyond 5G (B5G) networks [1] and vehicular networks (VNs) and the ever-increasing vehicles on the road hasten the flourishing of vehicular applications, such as online gaming [2], autonomous driving [3], and augmented reality [4], which is generally computation intensive and delay-sensitive [5–7]. Although cloud computing provides powerful computational capabilities, remote resources could lead to high latency due to unpredictable transmission latency. By deploying the cloud capabilities close to the end-users, mobile edge computing (MEC) is emerging as a promising solution to reduce the long latency between the users and the cloud [8]. Vehicular edge computing (VEC) is further emerging as a prospective way to support the stringent requirements of vehicular applications by installing the MEC servers on roadside units (RSUs) to support the stringent requirements of vehicular applications [9]. Thanks to the closely deployed resources, the communication and computing delay could be significantly reduced by offloading the tasks to the VEC servers [10].

However, the VEC network is characterized by the salient features of both MEC and VNs, which brings several challenges for efficient task offloading and resource allocation in the VEC network (VECN) [11]. First, compared to conventional cloud servers, VEC servers have limited resources, which may be insufficient to meet the strict requirements of vehicular applications, especially for the explosive arrival of applications in dense scenarios.

Second, the high mobility and dynamics of VNs lead to spatial-temporally dynamic and uneven distribution of tasks, which further brings difficulties for designing a dynamic and elastic task offloading and resource allocation mechanism for a VECN [12]. Third, some vehicular applications are safety-related and life-critical and should be processed preferentially, and some are non-safety related (such as the entertainment applications) and could be deferred, which makes the traditional offloading strategy infeasible for the heterogeneous tasks with different priorities [13]. Fourth, besides the above-mentioned concerns on the quality of service (QoS), the privacy issue is critical but generally isolated from task offloading and resource allocation in VECNs [14,15]. The highly dynamic and open nature of VN channels, together with the frequent task offloading behaviors of vehicles, make it possible for an attacker to infer the location or personal information of the vehicle users, leading to the threat of privacy disclosure [16]. The leakage of privacy information poses significant risks to the social privacy of drivers. Therefore, designing an efficient and dynamic task-offloading and computational-resource-allocation approach that is able to ensure the stringent and heterogeneous requirements of vehicles, satisfy the resource constraints of VEC servers, and protect the privacy of vehicles in real-time is a fundamental problem.

In this work, we propose a dynamic task-offloading and computational-resource-allocation approach based on decentralized multi-agent deep reinforcement learning (DRL), where privacy preservation for computation offloading, the priority of different tasks, the high mobility of ICVs, and the random arrival of the vehicular tasks are jointly considered. The main contributions of this work follow:

1. We employ a privacy-preserving computation-offloading framework that integrates the security component of the certificate authority (CA) into the edge-servers-assisted vehicular network, which involves multiple RSUs and intelligent connected vehicles (ICVs). In this proposed system model, due to the high mobility of multi-ICVs with random arrival tasks, the offloading strategies, computational resources assigned to the ICVs, and the pseudonym-changing decisions vary with time slots.
2. Based on the architecture, we formulated the cooperative optimization problem of computation offloading and resource allocation with the consideration of privacy preservation and task priority as a Markov decision process (MDP) based on three aspects: the end-to-end delay, the computational resource cost, and the ICV's privacy level to minimize the weighted cost in the VEN system. Furthermore, the state, action, and reward states are designed subsequently.
3. In order to effectively solve the above-mentioned problem with continuous variables and meet the requirement of convergence, cooperative optimization for privacy and priority based on DDPG (COPP-DDPG) is proposed.
4. The convergence of the proposed approach is verified by the simulation results. Furthermore, the sets of simulation results of the performance comparison demonstrate the proposed approach exhibits superior performance to the other four baseline algorithms.

## 2. Related Work

A growing number of works have been devoted to computation offloading and resource allocation for delay minimization [17,18], the trade-off between delay and fairness [19], handover management [20], and under-utilized resource exploration [21–23]. Zhang et al. [17] proposed a load-balancing computation offloading scheme for VECNs to reduce the task-processing delay by efficiently utilizing the edge resources. To cope with the dynamic and uncertain environment of VECNs, Sun et al. [18] aimed to enable vehicles to decide on the optimal offloading strategy by learning the offloading delay performance of their neighbors. An adaptive-learning-based task offloading mechanism was designed based on the multi-armed bandit theory to minimize the average offloading delay. Zhou et al. [24] designed a two-stage VFC framework for joint resource management and task offloading, which consists of a contract theory-based vehicular computational

resource management scheme and a matching-learning-based task offloading mechanism. Tong et al. [19] proposed a collaborative method for optimal task offloading and resource allocation in VECN to achieve the trade-off between delay and fairness of collaboration. Considering the highly dynamic of VECN, the authors of [20] designed an intelligent task offloading scheme based on deep Q-learning to deal with the frequent handover with rapid changes in communication and computing. The studies in [21–23] aimed to exploit the under-utilized resources of the nearby vehicles. In [21], the authors focused on exploiting multi-hop vehicle computational resources for a task-offloading approach in VECNs based on the mobility analysis of vehicles. Wei et al. [22] focused on resource allocation and task assignment jointly by designing a cooperative vehicular fog computing architecture from an overlapping perspective, to fully utilize the under-explored resources of nearby vehicles. In [23], the authors proposed a joint offloading and resource allocation scheme for the parked-and-moving-vehicles-assisted VECN. This work aims to minimize the total offloading delay by employing a two-stage heuristic algorithm to determine the optimal strategies for offloading, channel allocation, and resource allocation. However, the above-mentioned works mainly focus on computation offloading without considering the threat of privacy disclosure during computation offloading.

Some studies are aware that user privacy leaks during the computation offloading [25–28]. Wei et al. [25] aimed at minimizing the offloading action and transmitting power with the objective of minimizing the system's cost under the privacy requirement of task offloading. Wang et al. [26] proposed a privacy-preserving VEC framework to minimize the delay of task execution by jointly optimizing the task offloading and resource-allocation algorithm. Xu et al. [27] adopted the non-dominated sorting genetic algorithm II (NSGA-II) for multi-objective optimization of execution time, energy consumption, and the privacy of the computing tasks in the VECN. A privacy-oriented task offloading method was designed in [28] based on deep reinforcement learning to resist attacks from privacy attackers with prior knowledge. However, these mechanisms formulate the privacy level as an indicator or the constraint of the optimization problem but lack the design of a privacy protection strategy for vehicles to defend the offloading actively against the privacy threats.

From the perspective of theoretical mechanisms that are applied for problem solutions, most of the above studies adopted the methods of game theory [17,23,25], contract theory and a matching mechanism [19], heuristic approaches [23,24,27], or an optimization approach [22,26]. However, because of the high mobility of the ICVs and the dynamics of communication connection in VECN, these mechanisms are insufficient to enable the strategies of computation offloading, resource allocation, and privacy preserving to adapt to the dynamic environment in real time. Unlike the previous studies, we studied the cooperative computation offloading, resource allocation, and privacy preservation in a VEC network based on decentralized multi-agent deep reinforcement learning, where the priority of different tasks, the mobility of vehicles, and the stochastic arrival of tasks are jointly considered.

### 3. Model of VEN with Privacy Preservation

#### 3.1. Scenario Description

In this work, we implemented a vehicular edge network (VEN) including an MEC server, a set of RSUs denoted by  $\mathcal{R} = \{1, 2, \dots, R\}$ , and a set of ICVs, which are denoted as  $\mathcal{V} = \{1, 2, \dots, V\}$ , on a length  $L$  multi-lane road.

As shown in Figure 1, a regional MEC server receives and transmits workloads between several RSUs via the optical fiber link. Each RSU's coverage range is assumed as  $r$ , which equally divides the road into  $R$  segments. The ICVs are randomly and independently distributed in lanes on the road with an arrival rate  $\lambda$ . Within the coverage of the RSU, ICV transmits task data between RSU by V2I communication. RSU is assumed as a traditional communication node in the VEN without computation capacity, which is considered as an equipment for message forwarding between the MEC server and the ICVs. Equipped with an onboard unit (OBU) and massive sensors such as cameras, mmWave radar, and

lidar, ICV has a certain amount of computation capacity, collecting information from the surrounding environment and sending different types of messages periodically to the RSU or other ICVs within its communication range.

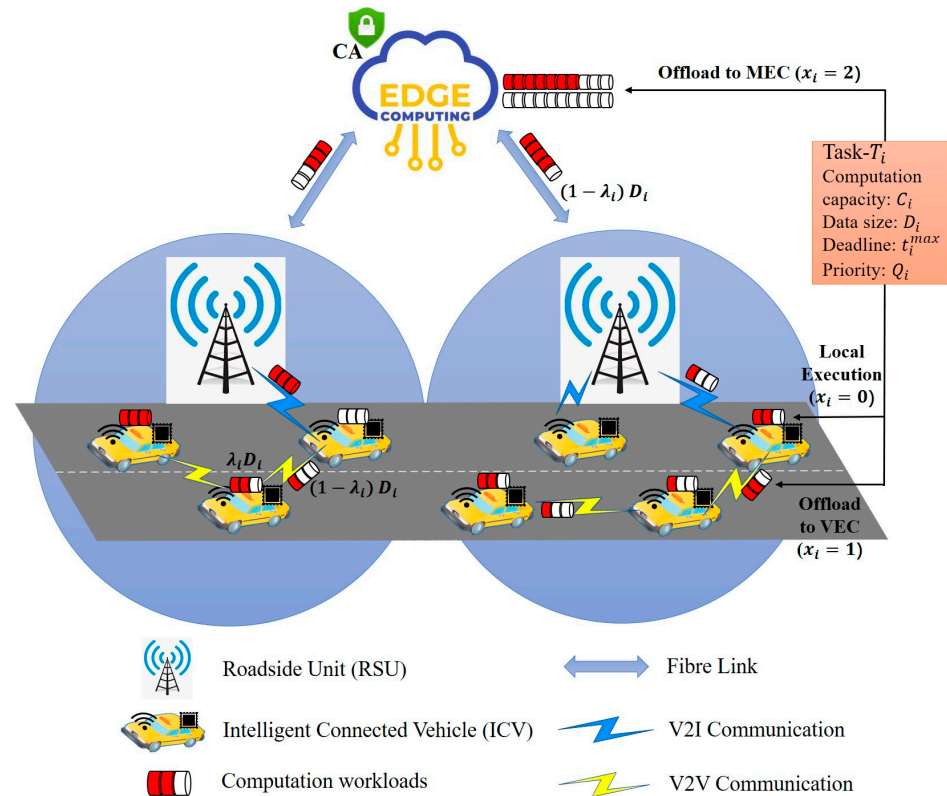


Figure 1. The system model of the VEN with privacy preservation.

Suppose that the speed of the ICVs follows a truncated Gaussian distribution, which is more appropriate with the practical environment to avoid dealing with the negative speeds [29]. In addition, the ICVs in the single lane follow the Poisson spatial distribution with density  $\frac{V}{L}$  [17], which makes the ICVs' equivalent speed  $\bar{s} = \frac{L\lambda}{V}$  [30]. The driving characteristics  $\{p_i, s_i\}$  for ICV- $i$  are periodically broadcasted to RSU and other ICVs with wireless links, where  $p_i$  represents the position coordinates of ICV- $i$  on the road space and  $s_i$  is the immediate speed of ICV- $i$  at its current position;  $\min\{s_i\} \leq \bar{s} \leq \max\{s_i\}$ ,  $i \in V$ . In our proposed road segment, the distance between different lanes is relatively small compared with the road length; therefore, the distance difference between different lanes is not considered in this work. Here, we assume that  $p_i$  is a one-dimensional position coordinate and that the starting point of the road is zero. In this case,  $p_i$  represents the distance between the position of ICV- $i$  and the starting point of the road.

Assume that each ICV produces a computation task in a certain time period. The vehicular task developed by ICV- $i$  is denoted as  $T_i$ , and the task arrival probability is denoted as  $P_{task}^{arr}$ . The key parameters of  $T_i$  can be characterized by a quintuple  $\{C_i, D_i^{in}, D_i^{out}, t_i^{max}, Q_i\}$  to characterize the profile of a mobile application, where  $C_i$  is the computing resource required to finish the vehicular task. The sizes of input and output data generated by the task execution  $D_i^{in}$  and  $D_i^{out}$  are related to the data bits for the task's input and output. The maximum completion deadline  $t_i^{max}$  denotes the maximum number of successive time slots before the vehicular task must be completed. The priority variable of the vehicular task  $Q_i$  is associated with the security level and urgency of the vehicular task. According to the existing standards and use cases of the vehicular network, we divided the vehicular tasks into three categories: traffic-safety-related tasks, traffic-efficiency-related tasks, and other vehicular entertainment tasks. Tasks related to ICV safety concern the lives of drivers and occupants; therefore, these vehicular tasks have the highest priority. Although

traffic-efficiency-related tasks have strict delay requirements, which may not involve security threats, we define these vehicular tasks as having normal priority. Other vehicular entertainment tasks provide value-added services, which can tolerate the completion delay exceeding the maximum deadline but with some degradation of data availability.

In this work, the ICV tasks are assigned to be executed locally, offloaded to another ICV with spare computational resources, or offloaded to the MEC server. ICVs with available computing capacity for task offloading are regarded as the vehicular edge computing (VEC). The offloading decision of ICV- $i$  is defined as  $\mathcal{X} = \{x_i | x_i \in \{0, 1, 2\}, i \in N\}$ . The ICV- $i$  chooses the local execution strategy of task  $T_i$  when  $x_i = 0$ . The offloading decision of calculating the task  $T_i$  on VEC is selected when  $x_i = 1$ . Task  $T_i$  is offloaded and processed on the MEC server when  $x_i = 2$ . In addition, the delay-sensitive vehicular tasks are not suitable for the situation of the binary offloading [31]. Partial offloading is introduced in this work to make better usage of the computational resources, which means that the vehicular task  $T_i$  can be executed separately onboard and edge computing as two portions. The offloading ratio  $\lambda_i (0 \leq \lambda_i \leq 1)$  is depicted as the offloading part ratio to the entire vehicular task  $T_i$ . ICV- $i$  computes  $\lambda_i D_i^{in}$  locally and offloads the rest  $(1 - \lambda_i) D_i^{in}$  to the VEC or MEC server [32]. For the limited computing and communication capacity of the ICV, each VEC is considered to only be able to execute one offloaded task from another ICV. The main notation is shown in Table 1.

**Table 1.** List of main notations.

Notation	Definition
$\mathcal{R}/R$	Set/number of RSUs
$\mathcal{V}/V$	Set/number of ICVs
$i/j$	The ICV index $i \in \mathcal{V}$ /the RSU index $j \in \mathcal{R}$
$\lambda$	ICVs' arrival rate
$p_i/s_i$	ICV's position/speed
$C_i/D_i^{in}/D_i^{out}/t_i^{max}/Q_i$	Vehicular task $T_i$ 's required computational resource/input data size/output data size/maximum completion deadline/priority
$t_{ij}^{link}/t_{ik}^{link}$	Link duration of V2I communication/V2V communication
$r_{ij}^{UL/DL}, /r_{ik/ki}$	Transmission rate of V2I communication/V2V communication
$f_{ICV}/f_{VEC}/f_j^{MEC}/F$	Computation resource of the ICV/VEC/allocated to MEC- $j$ /MEC server
$N_{i(t)}/N(t)$	ICV- $i$ pseudonym changing decision/number of ICVs changing the pseudonym at time $t$
$P_{max}/P_{loss}/P_i(t)$	ICV's maximum privacy/privacy loss/actual privacy at time slot $t$
$U_i^{ICV}, U_i^{VEC}, U_i^{MEC}$	Weighted cost function of ICV- $i$ under different offloading strategies
$\mathcal{X}/A/\mathcal{N}$	Offloading strategy/computational resource allocation/pseudonym changing decision sets of ICVs

### 3.2. Mobility Model

In a VEN, the high mobility of the ICVs may lead to changes in communication connection and affect the task offloading progress. When the ICV is out of the coverage of the RSU, the connection between the ICV and the MEC server for task offloading through V2I communication is disconnected.  $t_{ij}^{link}$  is described as the duration of the V2I transmission link between ICV- $i, i \in V$  and RSU- $j, j \in R$ , given by

$$t_{ij}^{link} = \frac{(r \lceil \frac{p_i}{r} \rceil - p_i) \cdot V}{L\lambda}. \tag{1}$$

When the distance between two ICVs exceeds their maximum communication range, the V2V communication will be disconnected. The duration between ICV- $i$  and VEC- $k$  staying connected is represented as  $t_{ik}^{link}$ , which can be calculated as [33]



$$t_{ik}^{link} = \chi(|p_i - p_k| \leq T_r) \frac{T_r - (p_i - p_k)\text{sign}(v_i - v_k)}{|v_i - v_k|}, \tag{2}$$

where  $T_r$  is the V2V transmission range under the fixed transmission power  $P$ .  $\chi(\cdot)$  is the indicator function. If  $z$  is false, then  $\chi(\cdot) = 0$ , indicating ICV- $i$  and VEC- $k$  are disconnected.  $\text{sign}(\cdot)$  is the sign function, when  $(p_i - p_k)\text{sign}(v_i - v_k) < 0$  indicates that ICV- $i$  and VEC- $k$  are approaching;  $(p_i - p_k)\text{sign}(v_i - v_k) > 0$  indicates that ICV- $i$  and VEC- $k$  are moving apart.

### 3.3. Communication Model

Two wireless communication modes, vehicle-to-vehicle and vehicle-to-infrastructure (V2V and V2I), are involved in the proposed VEN. The V2V communication mode is adopted when the ICV- $i$  decides to offload the vehicular task  $T_i$  to the surrounding ICV within one hop. The available ICV, which has spare computational resource VEC- $k$ , processes the computational task and returns the calculation results to ICV- $i$ . The transmission rate  $r_{ik/ki}$  between ICV- $i$  and VEC- $k$  is

$$r_{ik/ki} = B_{ik/ki} \log_2 \left[ 1 + \frac{\rho_0 P_i / P_k}{\sigma^2 [d_{ik/ki}]^\epsilon} \right]. \tag{3}$$

The two-way communication delay between ICV- $i$  and VEC- $k$  is obtained as

$$t_{comm}^{VEC} = \frac{\lambda_i D_i^{in}}{r_{ik}} + \frac{\lambda_i D_i^{out}}{r_{ki}}. \tag{4}$$

After offloading the vehicular task to the MEC server, the communication process includes the communication between the RSUs and the MEC server, and between the ICVs and the RSUs. In general, the optical fiber transmission time between the RSUs and the MEC server is negligible [17]. In this work, the ICVs and the RSUs communicate in the mode which is assumed to use the protocol of IEEE 802.11p. According to [34], the uplink and downlink transmission rate between ICV- $i$  and RSU- $j$  is

$$r_{ij}^{UL/DL} = \frac{\lambda_i D_i^{in} / D_i^{out} N_j \tau_{ij} \rho}{(1 - \tau_{ij})^{N_j} \sigma + T_{ij}^{success} N_j \tau_{ij} \rho + 1 - (1 - \tau_{ij})^{N_j} - N_j \tau_{ij} \rho (\text{RTS} + \text{AIFS} + \delta)}, \tag{5}$$

where  $N_j$  is the number of ICVs offloading the vehicular task to the MEC server through the RSU- $j$ .  $\tau_{ij}$  is the connection probability between ICV- $i$  and RSU- $j$  in the random time slot.  $\sigma$  is a time slot duration,  $\rho = (1 - \tau_{ij})^{N_j - 1}$ , and  $\delta$  is the propagation delay.  $T_{ij}^{success}$  is the successful transmission period between ICV- $i$  and RSU- $j$ , which is expressed as

$$T_{ij}^{success} = \Phi + \frac{\lambda_i D_i^{in} / D_i^{out}}{\omega_j \log(1 + P_i h_{ij})}, \tag{6}$$

where  $\Phi$  is related to the media access control protocol.  $\omega_j$  is the RSU- $j$ 's bandwidth,  $P_i$  stands for the ICV- $i$ 's transmission power, and  $h_{ij}$  denotes the channel gain between ICV- $i$  and RSU- $j$ .

The uplink and downlink transmission time between ICV- $i$  and RSU- $j$  is calculated as

$$t_{UL/DL}^{MEC} = \frac{\lambda_i D_i^{in} / D_i^{out}}{r_{ij}^{UL/DL}}. \tag{7}$$

The two-way communication delay between ICV- $i$  and RSU- $j$  is given by

$$t_{comm}^{MEC} = t_{UL}^{MEC} + t_{DL}^{MEC}. \tag{8}$$

### 3.4. Computation Model

There are three different modes with which to calculate the computation time according to the above-mentioned different offloading strategies. We assume that each ICV is equipped with the same OBU which provides the vehicular computation capacity as  $f_{ICV}$ . In terms of computational resources, the VEC is essentially an ICV, and  $f_{VEC} = f_{ICV}$ . The MEC server's whole computation is depicted as  $F$ .

#### 3.4.1. Local Computation

When the ICV- $i$  executes the vehicular task  $T_i$  locally,  $x_i = 0$ . The computation delay  $t_i^{ICV}$  of the vehicular task  $T_i$  is only dependent on the ICV- $i$ 's computational resource. The local computation time  $t_{comp}^{ICV}$  is given as

$$t_i^{ICV} = t_{comp}^{ICV} = \frac{\lambda_i C_i}{f_{ICV}}. \quad (9)$$

#### 3.4.2. VEC Offloading

In the situation of the vehicular task  $T_i$  offloaded to the VEC- $k$ ,  $x_i = 1$ , considering the VEC- $k$  has spare computational resources, the computation time of the ICV- $i$  for the offloaded vehicular task  $T_i$  to VEC- $k$  is formulated as

$$t_{comp}^{VEC} = \frac{(1 - \lambda_i) C_i}{(1 - \lambda_k) f_{VEC}}. \quad (10)$$

The total delay between the ICV- $i$  and the VEC- $k$  involves the task computation time and the communication time, which can be written as

$$t_i^{VEC} = t_{comp}^{VEC} + t_{comm}^{VEC}. \quad (11)$$

#### 3.4.3. MEC Offloading

In the situation of the vehicular task  $T_i$  offloaded to the MEC server,  $x_i = 2$ , the computational time of ICV- $i$  offloading the vehicular task  $T_i$  to the MEC server is obtained by

$$t_{comp}^{MEC} = \frac{(1 - \lambda_i) C_i}{f_j^{MEC}}, \quad (12)$$

where  $f_j^{MEC}$  is the computational resource allocated to the RSU- $j$ , which is considered as the transmission link between the ICV- $i$  and the MEC server, which is related to the MEC server's CPU cycle frequency. The total delay between the ICV- $i$  and the MEC server is expressed as

$$t_i^{MEC} = t_{comp}^{MEC} + t_{comm}^{MEC}. \quad (13)$$

### 3.5. Privacy Model

In the secure communication of VEN, the privacy protection of ICV's identity and location are very important. There are two privacy preservation schemes in the Internet of Vehicles, pseudonymous-based scheme, and signature-based scheme. According to the European Telecommunications Standards Institute (ETSI) standard, pseudonyms are considered a main method for the security and privacy preservation in secure communication of intelligent transportation system [35]. The pseudonym is a digital certificate issued to an ICV by the Certificate Authority (CA) based on its real identity after the ICV is registered with the CA. Using pseudonyms rather than real identities in the process of vehicular communication can effectively protect the identity and the location privacy of the ICV to some extent.

Pseudonymous privacy preservation scheme is widely used in the IoV. Pseudonym exchanging and pseudonym changing are two methods in this scheme. Pseudonym

exchanging is when an ICV's pseudonym expires, the ICV negotiates with another ICV to exchange their pseudonyms. Although pseudonym exchanging saves on pseudonym resources, the situation where the ICV cannot reach an agreement with its surrounding ICVs may occur, which may not be able to meet the ICV's timely pseudonym update requirement. By contrast, it is much simpler to implement ICV's pseudonym updating by pseudonym changing. The CA has to suffer more computational and communicational stress and requires additional storage in the method of pseudonym changing. However, in our considered model, the CA is integrated into the MEC server, which provides the CA's required computation, communication, and storage resources. In addition, the signature privacy preservation scheme in the IoV includes the group signature and ring signature. The group signature scheme can effectively implement anonymous communication, but it is difficult to update system parameters effectively when new members join the group and remove malicious members. Ring signatures can provide spontaneous anonymity, but the difficulty is how to form a ring between ICVs.

Therefore, pseudonym changing is the most appropriate method of the above-mentioned methods for the proposed VECN scenario. Here we assume that ICVs should change their pseudonyms under certain conditions to reduce the risk of a single pseudonym being associated with the real identity of the ICV [36].

Let  $N_{i(t)}$  indicate whether the ICV- $i$  changes its pseudonym at time slot  $t$ .

$$N_{i(t)} = \begin{cases} 0, & \text{ICV} - i \text{ does not change the pseudonym} \\ 1, & \text{ICV} - i \text{ changes the pseudonym} \end{cases} \quad (14)$$

$N_{(t)}$  depicts the number of ICV units changing the pseudonym at time slot  $t$ .

$$N_{(t)} = \sum_{i=1}^V N_{i(t)}. \quad (15)$$

According to the information entropy principle, the maximum privacy  $P_{max}$  that an ICV can obtain at time slot  $t$  is [37]

$$P_{max} = \sum_{a=1}^N P_{a|b} \log_2 P_{a|b}, \quad (16)$$

where  $P_{a|b}$  is the case that the pseudonym is changed from  $b$  to  $a$ , and the pseudonym is available when  $P_{a|b} = \frac{1}{N_{(t)}}$ .

There is a certain cost when changing the ICV's pseudonym. Here, we comprehensively consider the cost in two parts. One part is the cost of changing the radio's routing and addressing table when acquiring a new pseudonym, and another is the time cost of keeping silent during the process of changing the pseudonym [38]. It is assumed that there is no limit to the number of pseudonym changing in this work, and the costs are collectively denoted as  $P_{loss}$ . The real-time loss of ICV- $i$ 's privacy at time slot  $t$  is

$$P_{loss} = \vartheta * (t - t_{c,i}), \quad (17)$$

where  $\vartheta$  represents the location privacy loss constant, and  $t_{c,i}$  indicates the last time slot that the pseudonym has been changed. The privacy of ICV- $i$  at time slot  $t$  is

$$P_{i(t)} = P_{max} - P_{loss} = \sum_{d=1}^V \frac{1}{N_{(t)}} \log_2 \frac{1}{N_{(t)}} - \vartheta * (t - t_{c,i}). \quad (18)$$

#### 4. Problem Formulation

We formulated the cooperative optimization problem of computation offloading and resource allocation with privacy preservation to minimize the weighted system cost. In this



work, the cost function is negatively correlated with the ICV’s satisfaction. The satisfaction function is a term in economics and has been widely used by recent studies [39].

Firstly, the total delay of the ICV’s task is negatively correlated with satisfaction. With the increment of the delay, the vehicular tasks’ information become less valuable. Even for some security-related tasks, delay beyond their required maximum execution deadline will endanger the safety of vehicles. When the task end-to-end delay exceeds the vehicular task’s completion deadline, the satisfaction of the ICV is considered to be extremely poor, and the corresponding penalty will be imposed. Therefore, the longer the total delay of the vehicular task, the lower the satisfaction of the ICV.

Secondly, there is no cost for ICVs to use their own configured computing resources. As a supplement to the ICV’s own computing resources, the computing resources obtained by the ICV from the VEC or MEC server are considered to require the payment. Therefore, the more computational resources consumed that are not ICVs, the lower the satisfaction of the ICV.

Thirdly, privacy is also a reference quantity that affects the ICV’s satisfaction. The ICV’s privacy concerns the vehicle safety and the driver’s personal privacy. A low privacy level will lead to the disclosure of the ICV’s related information. Therefore, the lower the amount of privacy the ICV has, the lower the satisfaction of the ICV.

In addition, the sensitivity of the vehicular task priority to delay is also different. We also take the priority into the consideration. Therefore, the cost function for ICV-*i* to process the task locally is given by

$$U_i^{ICV} = \begin{cases} \beta Q_i \log \left( 1 + \left( t_i^{actual} - t_i^{ICV} \right)^+ \right) \\ + \gamma \rho f_{ICV} - (1 - \beta - \gamma) P_i(t), t_i^{ICV} \leq t_i^{max} \\ P, t_i^{ICV} > t_i^{max} \end{cases} \quad (19)$$

where  $\beta \in (0, 1)$  is the weight of the delay part and  $\gamma$  represents the weigh of computational resource cost part.  $(\cdot)^+ = \max(\cdot, 0)$  guarantees  $U_i^l$  to be non-negative. The unit cost of the computational resource unit cost is depicted as  $\rho$ . Additionally,  $P$  denotes the penalty for the vehicular task’s total delay exceeding its maximum execution deadline.

The cost function of ICV-*i* offloaded the task to the VEC is

$$U_i^{VEC} = \begin{cases} \beta Q_i \log \left( 1 + \left( t_{i-VEC}^{actual} - t_i^{VEC} \right)^+ \right) \\ + \gamma \rho f_{VEC} - (1 - \beta - \gamma) P_i(t), t_i^{VEC} \leq t_i^{max} \\ P, t_i^{VEC} > t_i^{max} \end{cases} \quad (20)$$

where  $t_{i-VEC}^{actual} = \min \{ t_{ik}^{link}, t_i^{max} \}$  is denoted as the delay actually generated by ICV-*i*. As soon as the V2V transmission connection is disconnected, the task offloading process of the ICV-*i* to the VEC-*k* will be terminated.

The cost function of ICV-*i* offloaded the task to the MEC server is

$$U_i^{MEC} = \begin{cases} \beta Q_i \log \left( 1 + \left( t_{i-MEC}^{actual} - t_i^{MEC} \right)^+ \right) \\ + \gamma \rho f_i^{MEC} - (1 - \beta - \gamma) P_i(t), t_i^{MEC} \leq t_i^{max} \\ P, t_i^{MEC} > t_i^{max} \end{cases} \quad (21)$$

Similarly,  $t_{i-MEC}^{actual} = \min \{ t_{ij}^{link}, t_i^{max} \}$  is described as the actual delay of the vehicular task. The task offloading to the MEC server interrupt occurs when the V2I communication is disconnected.

Combining (19), (20), and (21), the ICV-*i*'s cost function is generated as

$$U_i = \begin{cases} U_i^{ICV}, & x_i = 0, \\ U_i^{VEC}, & x_i = 1, \\ U_i^{MEC}, & x_i = 2. \end{cases} \quad (22)$$

In this work, to minimize the total cost of all the ICVs in the VEN by determining three objects—the offloading strategies of ICVs, the computational resource allocation from the MEC server, and the pseudonym-changing decisions by ICVs—the cooperative optimization problem with corresponding constraints can be described as follows:

$$\begin{aligned} & \min_{\mathcal{X}, \mathcal{A}, \mathcal{N}} \sum_{i=1}^V U_i, \\ & s.t. C1 : 0 \leq f_{ICV} < F, \\ & C2 : 0 \leq f_{VEC} < F, \\ & C3 : 0 \leq f_i^{MEC} \leq x_i F, \forall i \in \mathcal{V}, \forall j \in \mathcal{R}, \\ & C4 : \sum_{j=1}^R f_j^{MEC} \leq F, \forall j \in \mathcal{R}, \\ & C5 : x_i = \{0, 1, 2\}, \forall i \in \mathcal{V}. \end{aligned} \quad (23)$$

Constraints C1 and C2 exact the constraints on the available computational resources of ICV and VEC, which should be non-negative and less than the MEC server's computational resources. Constraint C3 limits the computational resources assigned by the MEC server to each ICV. Constraint C4 guarantees that the sum of the computational resources assigned to ICV's offloading tasks should not exceed the total computational resources of the MEC server. Constraint C5 ensures that there are only three offloading strategies provided to ICV-*i* for the vehicular task.

### 5. Cooperative Optimization for Privacy and Priority Based on DDPG

For the above problems, a large number of time-varying variables makes the problem become nonlinear and have high computational complexity. DDPG is a model-free method that relies on the actor-critic structure. DDPG can be used to solve the DRL problems on continuous-action spaces steadily to obtain the optimal solution instead of the deep Q-network (DQN) which is used to deal with the discrete action problems. Therefore, we propose a cooperative optimization for privacy and priority based on DDPG, i.e., COPP-DDPG, to solve the above optimization problem. As shown in Figure 2, the  $s_{i(t)}$ ,  $a_{i(t)}$  and  $r_i(t)$  in the structure of COPP-DDPG represent the state, action, and reward in the MDP, respectively.

#### 5.1. State Space

The state of the ICV-*i* at time slot *t* is denoted as  $s_{i(t)}$ , which consists of the parameters of the vehicular task  $T_i$ , such as partial offloading ratio  $\lambda_{i(t)}$ , the computational resource required by the vehicular task  $C_{i(t)}$ , the sizes of input and output data generated by the task execution  $D_{i(t)}^{in}$  and  $D_{i(t)}^{out}$ , the maximum completion deadline  $t_{i(t)}^{max}$ , and the vehicular task priority  $Q_{i(t)}$ . The ICV-*i*'s time-variable driving characteristics at time slot *t* as the current position and speed  $p_{i(t)}$  and  $s_{i(t)}$  are also the components of  $s_{i(t)}$ . In addition,  $s_{i(t)}$  involves the allocated RSU-*j*'s computational resource  $f_{j(t)}^{mec}$  and two privacy-related variables,  $N_{(t)}$  and  $t_{c,i(t)}$ , which represent the ICV number of the pseudonym changing and the last time slot ICV-*i* changed its pseudonym, respectively. The state  $s_{i(t)} \in S$  can be written collectively as

$$s_{i(t)} = \left\{ \begin{array}{l} \lambda_{i(t)}, C_{i(t)}, D_{i(t)}^{in}, D_{i(t)}^{out}, t_{i(t)}^{max}, Q_{i(t)}, \\ p_{i(t)}, s_{i(t)}, f_{j(t)}^{MEC}, N_{(t)}, t_{c,i(t)} \end{array} \right\}_{\forall i \in \mathcal{V}, \forall j \in \mathcal{R}} \quad (24)$$

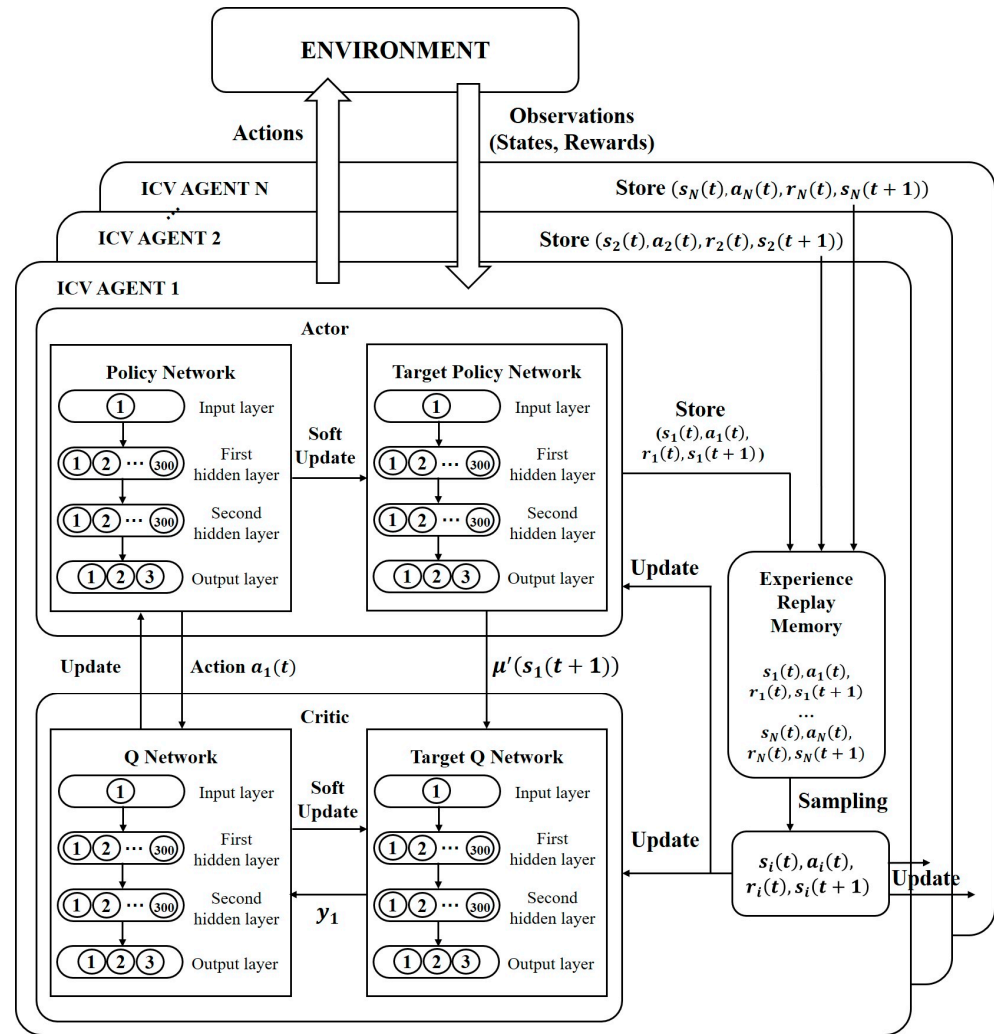


Figure 2. The structure of COPP-DDPG.

### 5.2. Action Space

The agent ICV- $i$  selects an action  $a_{i(t)}$  from the action space at time slot  $t$  by state  $s_{i(t)}$ . The action considers the cooperative strategies of computation offloading, resource allocation, and privacy preservation. The offloading strategy for vehicular task  $x_{i(t)}$  needs to be selected from three ways, local calculation, offloading computation at VEC, or execution of the offloaded task on the MEC server. The computational resource is allocated to the ICV- $i$  by the MEC server via its connected RSU once the task is offloaded to the MEC server. Considering the privacy of ICV, pseudonym changing decision  $N_{i(t)}$  should also be determined in this process. The action  $a_{i(t)} \in A$  is depicted as

$$a_{i(t)} = \left\{ x_{i(t)}, f_{i(t)}^{MEC}, N_{i(t)} \right\}_{\forall i \in \mathcal{V}} \tag{25}$$

### 5.3. Reward Space

The reward is used to evaluate the performance of the selected action. All the ICVs are assumed to use the same reward function. To achieve the goal of minimizing the cost function in the optimal problem, we designed the reward function  $r_i(t)$  of the agent ICV- $i$  during the period  $T$  as

$$r_{i(t)} = - \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T U_i(t) \Big|_{s_{i(t)}} \tag{26}$$

The average reward of multi-agent ICVs in time slot  $t$  is given by

$$r(t) = \frac{1}{V} \sum_{i=1}^V r_{i(t)}, \forall i \in \mathcal{V}. \tag{27}$$

The overall flow of COPP-DDPG algorithm is based on DDPG, as shown in Algorithm 1. DDPG algorithm is widely used and introduced in a large number of studies [40]. Here we will not describe the details of the algorithm. The goal of DDPG is to make the agents obtain the maximum reward, which is guided by the rewards observed by the interaction with the environment. Due to the interaction between multiple ICVs in the VEN environment, the behavior of one ICV has the potential to influence and change the decisions of other ICVs. Therefore, multi-agent DDPG is considered in this work to meet the increasing complexity of environment and task requirements, which takes full account of the cooperation and competition between agents to maximize joint returns. However, with the preliminary experiments, we found that the centralized multi-agent DDPG leads to the problem of poor convergence. Therefore, the COPP-DDPG we finally chose is a decentralized multi-agent DDPG method.

---

**Algorithm 1:** Cooperative Optimization for Privacy and Priority based on Deep Deterministic Policy Gradient COPP-DDPG

---

1 Randomly initialize the critic network  $Q(s, a | \theta_i^Q)$  and the actor network  $\mu(s | \theta_i^\mu)$  with weights  $\theta_i^Q$  and  $\theta_i^\mu$ ;

2 Initialize the target critic network  $Q'$  and the target actor network  $\mu'$  with weights  $\theta_i^{Q'}$  and  $\theta_i^{\mu'}$ ;

3 Initialize the memory replay buffer  $B$ ;

4 **for** episode  $k = 1, 2, \dots, K$  **do**

5     Initialize a random process  $n_t$ ;

6     Receive initial observation state  $S = \{s_{i(1)}, s_{i(2)}, \dots, s_{i(N)}\}$ ;

7     **for**  $i = 1, 2, \dots, N$  **do**

8         **for**  $t = 1, 2, \dots, T$  **do**

9             Select action  $a_{i(t)} = \mu(s_{i(t)} | \theta_i^\mu) + n_t$  according to the current policy and exploration noise  $n_t$ ;

10            Execute action  $a_{i(t)}$  and observe reward  $r_{i(t)}$ , the next state  $s_{i(t+1)}$ ;

11            Store all transitions  $(s_{i(t)}, a_{i(t)}, r_{i(t)}, s_{i(t+1)})$  in  $B$ ;

12            Sample a random mini-batch of  $Z$  transitions from  $B$ ;

13            Set

14            

$$y_i = r_{i(t)} + \gamma Q \pi(s_{i(t)}, a_{i(t)} | \theta^Q) \Big|_{a_i^{k+1} = \mu'(s_i^{k \neq 1})} \tag{28}$$

15            Update the critic network  $Q(s, a | \theta_i^Q)$  by minimizing the loss

16            

$$L(\theta^Q) = \frac{1}{Z} \sum_i y_i - Q \pi(s_i, a_{i(t)} | \theta_i^Q)^2 \tag{29}$$

17            Update the actor policy by using the sampling policy gradient

18            

$$\nabla_{\theta_i^\mu} J \approx \frac{1}{Z} \sum_i \nabla_a Q(s_{i(t)}, a_{i(t)} | \theta_i^Q) \Big|_{a_i = \mu(s_i)} \nabla_{\theta_i^\mu} \mu(s_{i(t)} | \theta_i^\mu) \tag{30}$$

19            Update the target networks for each agent  $i$ :

20            

$$\theta_i^{Q'/\mu'} \leftarrow \eta \theta_i^{Q/\mu} + (1 - \eta) \theta_i^{Q'/\mu'} \tag{31}$$

21            

22            **end**

23            **end**

24 **end**

---

## 6. Numerical Results

### 6.1. Simulation Setup

The system hardware configuration required for constructing the experimental environment was as follows: the CPU was i9 9900K, the GPU was a NVIDIA RTX GEFORCE 2080TI, and the memory capacity was 32GB. The system software configuration for the experimental environment was as follows: the experimental environment was simulated through TensorFlow1.15.0 and ran under the PyCharm-integrated development environment. The parameters of COPP-DDPG are given in Table 2.

**Table 2.** Main hyperparameters of the De-DDPG.

Parameters	Value
Size of the first hidden layer for actor and critic	300
Size of the second hidden layer for actor and critic	300
Learning rate of actor and critic $\alpha'/\alpha$	0.0001/0.001
Size of experience memory $B$	20,000
Parameters for OU noise $\theta, \mu, \sigma$	0.15, 0.15, 0.10
Discount factor $\gamma$	0.95
Penalty for failed tasks execution $P$	8
Total number of all episodes $K$	1000
Total time periods of one episode $T$	110

There was an MEC server with sufficient computational resources and communication resources. Two groups of RSUs were connected to the MEC server and two RSUs were designed in each group, which included a primary RSU and a secondary RSU. As the multi-agent model, 20 ICVs were randomly driving on the proposed road segment. We set the time slot as  $t = 1$ . The required computational resource for completing the vehicular task  $T_i$  was set as  $C_i = 0.48$ . The size of all the arrival vehicular tasks' input data in each time slot  $t$  follows Uniform distribution with  $D_i^{in} \sim U(2.0, 4.8)$ . The size of output data for vehicular tasks computation were set as  $D_i^{out} = 0.05 \times D_i^{in}$ . Meanwhile, the vehicular tasks' arrival probability was set to 0.45. The maximum completion deadline of the vehicular task was set as  $t_i^{max} = 4t$ . The available uplink and downlink transmission rate of V2I communication were derived as  $r_{ij}^{UL} \sim U(1.1, 1.25)$  and  $r_{ij}^{DL} \sim U(1.0, 1.15)$ , respectively. The probability of successful connection between ICV- $i$  and RSU- $j$  in the random time slot was set as  $\tau_{ij} = 0.95$ . The propagation delay was set as  $\delta = 0.1t$  and the RSU- $j$ 's bandwidth was set as  $\omega_j = 10$ . The ICV's, VEC's and MEC server's computational resources were set as 0.5, 0.9  $\sim$  1.75 and 4.0 Gigacycles, respectively. The unit cost of the MEC server's computational resource was set as  $\rho = 0.58$  Gigacycles/Mb. In terms of the priority of the arrival vehicular tasks, we set three priorities in this work as  $Q_i = 0.6 + 0.3i$ ,  $i = 1, 2, 3$ . The partial offloading ratio of the vehicular task  $\lambda_i$  and the initial probability of vehicle updating pseudonym  $P(N_{i(t)})$  about privacy preservation will be discussed in Section 6.3.

### 6.2. Performance Comparison

We set four algorithms (benchmarks) to compare the performance and advantage of COPP-DDPG, including binary offloading policy with DDPG (BOP-DDPG), offloading all tasks to MEC (OT-MEC), offloading all tasks to VEC (OT-VEC), and local execution all tasks by ICV's processor (LE-VP).

1. BOP-DDPG: Although we adopt the DDPG algorithm, BOP-DDPG does not consider the situation of partial offloading, but only offloading all tasks to the MEC server, VEC, or local execution. However, the neural network has the same structure as each network as COPP-DDPG and the allocated computational resources for the vehicles from all RSUs are the same as COPP-DDPG.



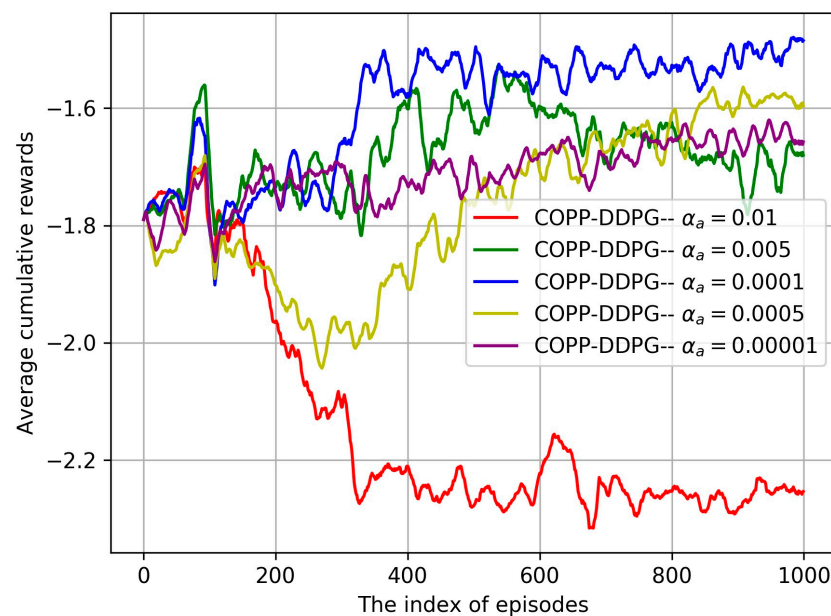
2. OT-MEC: The vehicular tasks generated by the ICVs are all offloaded and executed to the MEC server through their linked RSUs. The VEC is not considered in the simulation environment.
3. OT-VEC: The vehicular tasks generated by the ICVs are all offloaded and calculated to VEC with spare computational resources. The MEC offloading strategy is not considered in the simulation.
4. LE-VP: All computation tasks are executed by the local processor of ICV. VEC and the MEC server are not applicable for offloading tasks.

### 6.3. Simulation Results

Firstly, the analysis of the COPP-DDPG algorithm's convergence is given in Section 6.3.1, and then we compare the performance and advantage of COPP-DDPG with those of the other four baseline algorithms in Section 6.3.2. Here, we consider using the average cumulative reward of each episode to evaluate the convergence of COPP-DDPG and the average cumulative reward of all 1000 episodes to compare the performance of COPP-DDPG to that of other four baseline algorithms.

#### 6.3.1. Convergence Performance

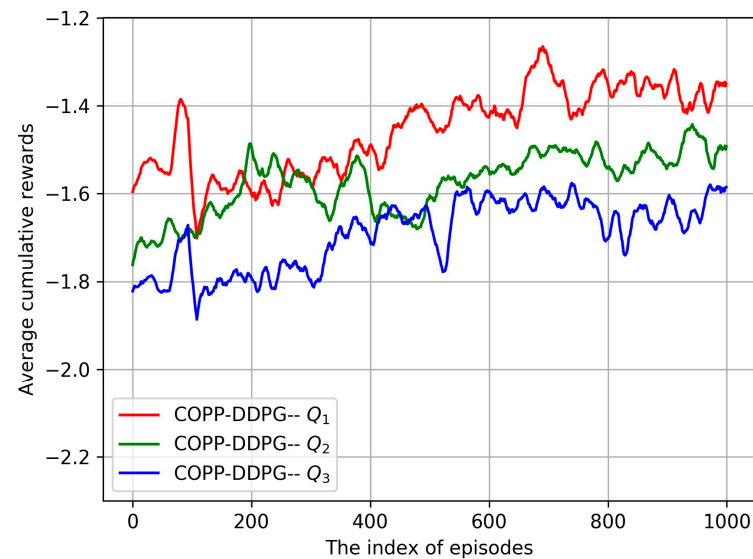
Figure 3 reveals the convergence of COPP-DDPG with different actor learning rates,  $\alpha_a$ . The choice of learning rate  $\alpha_a$  can obviously affect the convergence effect and the speed of COPP-DDPG. From Figure 3, COPP-DDPG cannot be convergent due to the greater learning rate when  $\alpha_a = 0.01$ . The convergence of COPP-DDPG is obviously improved when  $\alpha_a = 0.005$ , but the result is not satisfactory. When  $\alpha_a = 0.0005$  and  $\alpha_a = 0.00001$ , COPP-DDPG can converge on certain results, and the proposed algorithm does not perform well. Considering the stability and performance of COPP-DDPG, we set the actor's learning rate as  $\alpha_a = 0.0001$ , which is shown as the blue curve in Figure 3.



**Figure 3.** Convergence of COPP-DDPG with different actor learning rates  $\alpha_a$ .

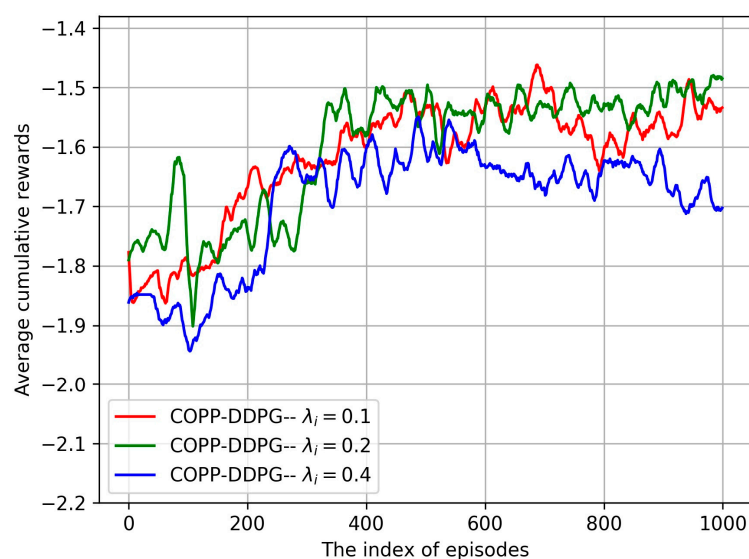
In this paper, we consider the priority of arriving tasks generated by ICVs in each time slot. Figure 4 shows the convergence of the proposed COPP-DDPG with three different priorities,  $Q_i$ . As described in subsection V.A, the values of priorities were set as  $Q_i = 0.6 + 0.3i$ ; that is to say,  $Q_1 = 0.9$ ,  $Q_2 = 1.2$ , and  $Q_3 = 1.5$ . As illustrated in Figure 4, due to the fewer values of low priority, the performance of the vehicular task with priority  $Q_1$  is better than the cases where the vehicular task priority is  $Q_2$  or  $Q_3$ . The fewer the values of the priorities are, the greater the immediate reward and the less the cost.

It can be observed in Figure 4 that COPP-DDPG converges under the situation of three different priorities.



**Figure 4.** Convergence of COPP-DDPG with three different priorities  $Q_i$ .

In terms of the partial offloading for vehicular edge computing, Figure 5 illustrates the convergence of proposed COPP-DDPG with different partial offloading rates  $\lambda_i$ . When  $\lambda_i = 0.1$ , the convergence of COPP-DDPG can be satisfied. However, from the perspective of the average cumulative rewards, the performance of COPP-DDPG whose offloading rate was set as 0.1 was not good due to the local processor of ICVs having remaining resources. When  $\lambda_i = 0.4$ , the partial offloading ratio of vehicular tasks reserved locally was too large, which led to the shortage of the local processor resources. Furthermore, due to more incomplete tasks, which caused the punishment, the performance of COPP-DDPG was greatly reduced. When  $\lambda_i = 0.2$ , it made better use of the computational resources of the ICV's processor and the resources of the VEC and MEC server. Its performance was the best. Therefore, we set the ratio for tasks offloading as  $\lambda_i = 0.2$  in this work.



**Figure 5.** Convergence of COPP-DDPG with different partial offloading rates  $\lambda_i$ .

### 6.3.2. Performance Comparison

Figures 6–9 verify the performance and advantages of the COPP-DDPG compared to the other four algorithms described in Section 6.2: BOP-DDPG, OT-RSU, OT-VEC, and LE-VP.

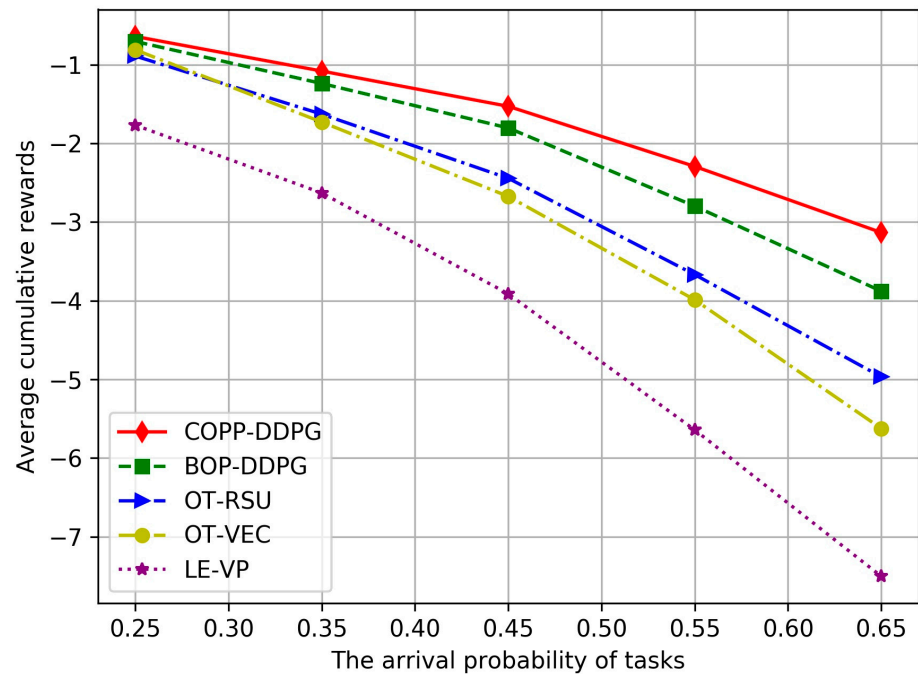


Figure 6. Comparison of five algorithms with different arrival probabilities of tasks  $D_{task}^{arr}$ .

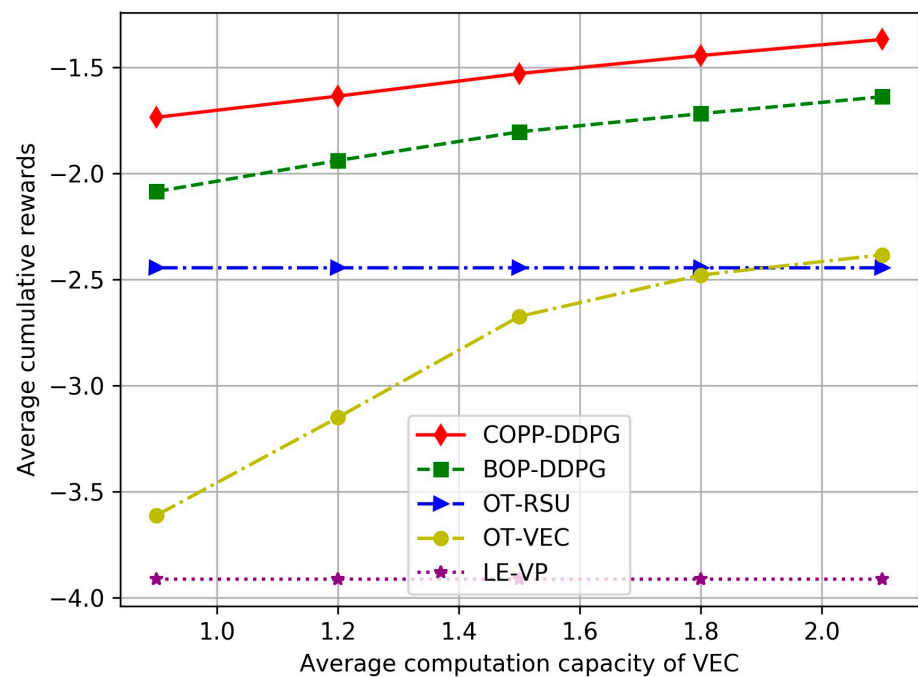


Figure 7. Comparison of all algorithms with different computational capacities of VEC.

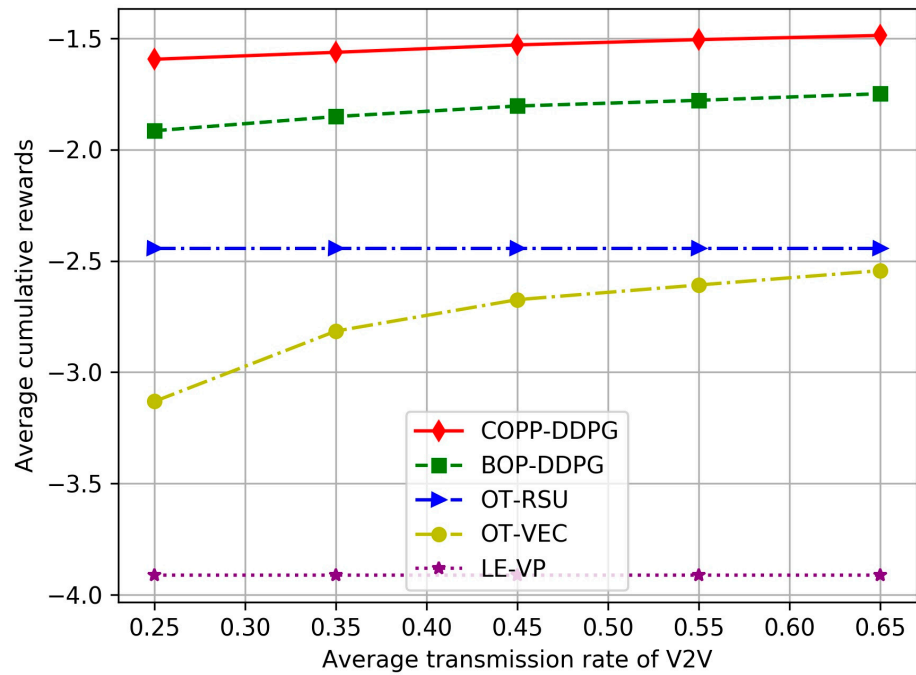


Figure 8. Comparison of all algorithms with different transmission rates of V2V.

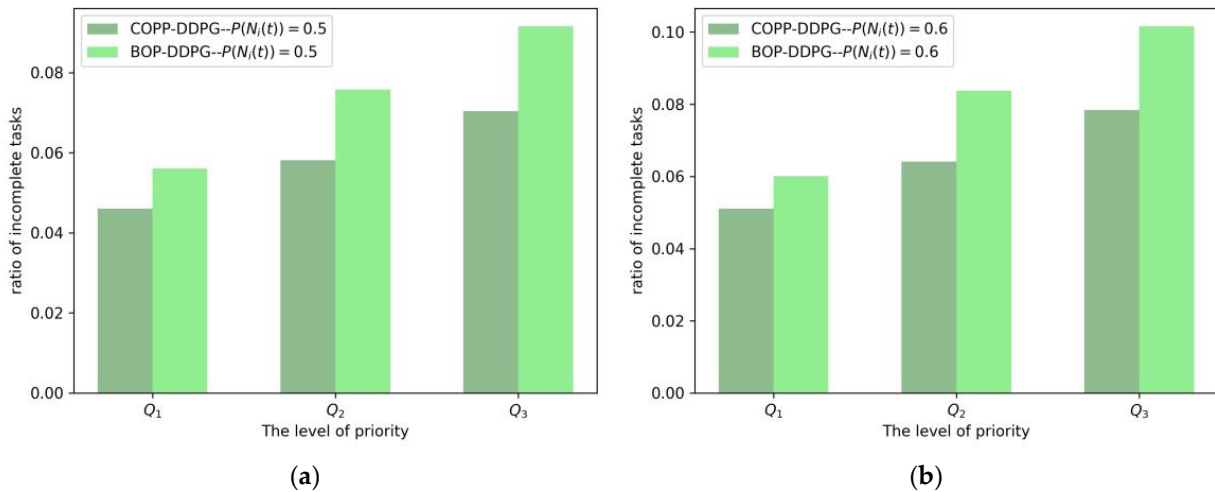


Figure 9. Comparison of ratios of incomplete tasks for two learning algorithms with different priorities: (a) performance comparison; (b) performance improvement.

Figure 6 indicates the comparison of five algorithms with different arrival probabilities of tasks  $P_{task}^{arr}$ . The vehicular task arrival rate represents the probability that ICVs can generate computation tasks in each time slot. When the vehicular task arrival rate is low, the vehicular task is processed in a timely pattern due to the small number of vehicular tasks, so that the computational tasks' delay is low and the proportion of tasks not processed is greatly reduced, which means that there is less punishment, so the average cumulative reward is greater. As shown in Figure 6, the COPP-DDPG algorithm keeps the best performance no matter what the vehicular task arrival rate is. As ICV's local processing capacity is the lowest, the performance of LE-VP is the worst. Compared with all vehicular tasks offloaded to the MEC server strategy, although the transmission rate of VEC is better, its computing capacity is lower. Therefore, when the task arrival rate is low, such as  $P_{task}^{arr} = 0.25$ , the performance of OT-VEC is better than that of OT-RSU. However, when the task arrival rate is high, the performance of OT-VEC is worse than that of OT-RSU.

Figure 7 shows the comparison of all five algorithms with different computation capacities of VEC  $f_{VEC}$ . We set the range of average computation capacity of all available VECs as [0.9, 1.2, 1.5, 1.8, 2.1] GHz. The curves of OT-RSU and LE-VP show that although the computational capacity of VECs is time-varying, the performances of these two algorithms cannot be affected because the average cumulative rewards of them are independent of the computational capacity of V2V. The performances of COPP-DDPG and BOP-DDPG are better than those of other algorithms due to their outstanding learning ability.

Besides the uplink and downlink transmission rate of the V2I communication between ICV- $i$  and RSU- $j$ ,  $r_{ij}^{UL}$  influences the performances of the five algorithms; see Figure 8. The transmission rates between ICV and VEC (V2V communication) are also significant for all algorithms. Figure 8 shows the performances of the five algorithms with the different transmission rates of V2V. Given the curves of OT-RSU and LE-VP, the communication between V2V does not affect the performances of the two algorithms. However, with the increase in the transmission rate of V2V, the average cumulative rewards of COPP-DDPG, BOP-DDPG, and OT-VEC increased gradually, because the higher the transmission rate is, the lower the delay is, and the average cumulative reward increases.

Furthermore, we will verify the performances of the algorithms considering the privacy preservation of ICVs. As mentioned in Section 3.5, pseudonym changing is an effective means for ICV privacy preservation. Figure 9 displays the comparison of the ratios of incomplete tasks for two learning algorithms with different priorities  $Q_i$ . As shown in Figure 9a, we set the change in the ICV's pseudonym to follow the Bernoulli distribution with parameter 0.5—that is, the probability of the ICV changing pseudonyms is 0.5;  $P(N_{i(t)}) = 0.5$ . In terms of average cumulative rewards, COPP-DDPG's performance is better than that of BOP-DDPG. In addition, the completion rate of tasks with high priority is significantly higher than that of tasks with low priority. This is because the setting of the reward function  $r(t)$ , which is related to the cost function  $U_i$ , can guide the training of the two learning algorithms (COPP-DDPG, BOP-DDPG). As shown in Figure 9b, the probability of ICV changing pseudonyms was set as  $P(N_{i(t)}) = 0.6$ . With the incrementing of pseudonym change probability, the higher the privacy-preserving level of ICV, which can be used as VEC, the higher its corresponding privacy cost, resulting in higher cost and smaller average cumulative return.

## 7. Conclusions

In a VECN, there are massive dynamic cooperative and competitive interactions between ICVs, and the privacy exposure will threaten the security of ICVs, especially for the high-priority security-related tasks. Therefore, we propose a privacy-preserving and priority-aware task offloading and computational-resource-allocation approach based on DRL to satisfy the stringent requirements of QoS and privacy for vehicles in dynamic VECN while guaranteeing the constraints of the limited resources for VEC servers. First, we employ a privacy-preserving framework that integrates the CA into the VECN. Furthermore, the cooperative optimization problem is formulated as an MDP to minimize the total weighted cost of the system. To solve the problem, we propose an approach named COPP-DDPG to learn the optimal actions of task offloading, resource allocation, and pseudonym changing decision in the dynamic VEC networks. The simulation results demonstrated the converge and superior performance of the proposed approach in comparison with the benchmark methods. In the future, we will continue to research using ring signature to further improve the VECN system's privacy protection under greater ICV density in the environment.

**Author Contributions:** Conceptualization, methodology, Y.W. and J.W.; software, validation, formal analysis, Y.W. and H.K.; investigation, writing—original draft preparation, Z.S.; writing—review and editing, J.W. and H.K. All authors have read and agreed to the published version of the manuscript.



**Funding:** This research was funded by National Natural Science Foundation of China, grant number 61572229 and 6171101066, Jilin Provincial Science and Technology Department of China (20210101415JC).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data used to support the findings of this study are available from the corresponding author upon request.

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

## References

1. Chien, W.C.; Cho, H.H. Intelligent architecture for mobile HetNet in B5G. *IEEE Netw.* **2019**, *33*, 34–41. [[CrossRef](#)]
2. Hu, L.; Tian, Y. Ready player one: UAV-clustering-based multi-task offloading for vehicular VR/AR gaming. *IEEE Netw.* **2019**, *33*, 42–48. [[CrossRef](#)]
3. Liu, S.; Liu, L. Edge computing for autonomous driving: Opportunities and challenges. *Proc. IEEE* **2019**, *107*, 1697–1716. [[CrossRef](#)]
4. Qiao, X.; Ren, P. A new era for web AR with mobile edge computing. *IEEE Internet Comput.* **2018**, *22*, 46–55. [[CrossRef](#)]
5. Zhao, J.; Dong, P. Mobile-aware and relay-assisted partial offloading scheme based on parked vehicles in B5G vehicular networks. *Phys. Commun.* **2020**, *42*, 101163. [[CrossRef](#)]
6. Vo, N.; Duong, T.Q. Editorial: The Key Trends in B5G Technologies, Services and Applications. *Mob. Netw. Appl.* **2022**, *27*, 1716–1718. [[CrossRef](#)]
7. Alshamrani, S.S.; Jha, N. B5G Ultrareliable Low Latency Networks for Efficient Secure Autonomous and Smart Internet of Vehicles. *Math. Probl. Eng.* **2021**, *2021*, 3697733. [[CrossRef](#)]
8. Xiong, R.; Zhang, C. Reducing Power Consumption for Autonomous Ground Vehicles via Resource Allocation Based on Road Segmentation in V2X-MEC With Resource Constraints. *IEEE Trans. Veh. Technol.* **2022**, *71*, 6397–6409. [[CrossRef](#)]
9. Liu, L.; Chen, C. Vehicular Edge Computing and Networking: A Survey. *Mob. Netw. Appl.* **2021**, *26*, 1145–1168. [[CrossRef](#)]
10. Karimi, E.; Chen, Y. Task offloading in vehicular edge computing networks via deep reinforcement learning. *Comput. Commun.* **2022**, *189*, 193–204. [[CrossRef](#)]
11. Duan, W.; Gu, X. Resource Management for Intelligent Vehicular Edge Computing Networks. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 9797–9808. [[CrossRef](#)]
12. Zhang, Y.; Nalam, V.A. RAVEN: Resource Allocation using Reinforcement Learning for Vehicular Edge Computing Networks. *IEEE Commun. Lett.* **2022**, *26*, 2636–2640. [[CrossRef](#)]
13. Li, Y.; Tao, X. Privacy-Preserved Federated Learning for Autonomous Driving. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 8423–8434. [[CrossRef](#)]
14. Onieva, J.A.; Rios, R. Edge-Assisted Vehicular Networks Security. *IEEE Internet Things J.* **2019**, *6*, 8038–8045. [[CrossRef](#)]
15. El-Sayed, H.; Zeadally, S. Edge-centric trust management in vehicular networks. *Microprocess. Microsyst.* **2021**, *84*, 104271. [[CrossRef](#)]
16. Liu, J.; Zhang, S.; Liu, H. Distributed Collaborative Anomaly Detection for Trusted Digital Twin Vehicular Edge Networks. In Proceedings of the WASA, Nanjing, China, 25–27 June 2021. [[CrossRef](#)]
17. Zhang, J.; Guo, H. Task Offloading in Vehicular Edge Computing Networks: A Load-Balancing Solution. *IEEE Trans. Veh. Technol.* **2020**, *69*, 2092–2104. [[CrossRef](#)]
18. Sun, Y.; Guo, X. Adaptive Learning-Based Task Offloading for Vehicular Edge Computing Systems. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3061–3074. [[CrossRef](#)]
19. Zhou, Z.; Liao, H. When Vehicular Fog Computing Meets Autonomous Driving: Computational Resource Management and Task Offloading. *IEEE Netw.* **2020**, *34*, 70–76. [[CrossRef](#)]
20. Tong, S.; Liu, Y. Joint Task Offloading and Resource Allocation: A Historical Cumulative Contribution Based Collaborative Fog Computing Model. *IEEE Trans. Veh. Technol.* **2022**, early access. [[CrossRef](#)]
21. Guo, H.; Liu, J. Intelligent Task Offloading in Vehicular Edge Computing Networks. *IEEE Wirel. Commun.* **2020**, *27*, 126–132. [[CrossRef](#)]
22. Liu, L.; Zhao, M. Mobility-aware multi-hop task offloading for autonomous driving in vehicular edge computing and networks. *IEEE Trans. Intell. Transp. Syst.* **2022**; early access. [[CrossRef](#)]
23. Wei, Z.; Li, B. OCVC: An Overlapping-Enabled Cooperative Vehicular Fog Computing Protocol. *IEEE Trans. Mob. Comput.* **2022**, early access. [[CrossRef](#)]
24. Fan, W.; Liu, J. Joint Task Offloading and Resource Allocation for Multi-Access Edge Computing Assisted by Parked and Moving Vehicles. *IEEE Trans. Veh. Technol.* **2022**, *71*, 5314–5330. [[CrossRef](#)]
25. Wei, D.; Zhang, J. Privacy-Aware Multiagent Deep Reinforcement Learning for Task Offloading in VANET. *IEEE Trans. Intell. Transp. Syst.* **2022**; early access. [[CrossRef](#)]

26. Wang, S.; Li, J. Joint Optimization of Task Offloading and Resource Allocation Based on Differential Privacy in Vehicular Edge Computing. *IEEE Trans. Comput. Soc. Syst.* **2022**, *9*, 109–119. [[CrossRef](#)]
27. Xu, X.; Xue, Y. An edge computing-enabled computation offloading method with privacy preservation for internet of connected vehicles. *Future Gener. Comput. Syst.* **2019**, *96*, 89–100. [[CrossRef](#)]
28. Gao, H.; Huang, W. PPO2: Location Privacy-Oriented Task Offloading to Edge Computing Using Reinforcement Learning for Intelligent Autonomous Transport Systems. *IEEE Trans. Intell. Transp. Syst.* **2022**; *early access*. [[CrossRef](#)]
29. Durrani, S.; Zhou, X. Effect of vehicle mobility on connectivity of vehicular ad hoc networks. In Proceedings of the IEEE 2010 IEEE 72nd Vehicular Technology Conference-Fall, Ottawa, ON, Canada, 6–9 September 2010. [[CrossRef](#)]
30. Wisitpongphan, N.; Bai, F. Routing in sparse vehicular ad hoc wireless networks. *IEEE J. Sel. Areas Commun.* **2007**, *25*, 1538–1556. [[CrossRef](#)]
31. Wang, Y.; Sheng, M. Mobile-edge computing: Partial computation offloading using dynamic voltage scaling. *IEEE Trans. Commun.* **2016**, *64*, 4268–4282. [[CrossRef](#)]
32. You, C.; Huang, K. Energy-Efficient resource allocation for mobile-edge computation offloading. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 1397–1411. [[CrossRef](#)]
33. Zhang, Y.; Qin, X.; Song, X. Mobility-Aware Cooperative Task Offloading and Resource Allocation in Vehicular Edge Computing. In Proceedings of the WCNCW Workshops, Seoul, Republic of Korea, 6–9 April 2020. [[CrossRef](#)]
34. Dai, Y.; Xu, D. Joint Load Balancing and Offloading in Vehicular Edge Computing and Networks. *IEEE Internet Things J.* **2019**, *6*, 4377–4387. [[CrossRef](#)]
35. *ETSI TS 102 941*; Intelligent Transport Systems (ITS): Security; Trust and Privacy Management. European Telecommunications Standards Institute (ETSI): Valbonne, France, 2021.
36. Förster, D.; Kargl, F.; Löhr, H. A framework for evaluating pseudonym strategies in vehicular ad-hoc networks. In Proceedings of the WISEC 15, New York, NY, USA, 22–26 June 2015. [[CrossRef](#)]
37. Wang, J.; He, N. Optimization and non-cooperative game of anonymity updating in vehicular networks. *Ad Hoc Netw.* **2019**, *88*, 81–97. [[CrossRef](#)]
38. Freudiger, J.; Manshaei, M.H.; Hubaux, J.P. On non-cooperative location privacy: A game-theoretic analysis. In Proceedings of the CCS 2009, Chicago, IL, USA, 9–13 November 2009. [[CrossRef](#)]
39. Su, Z.; Hui, Y. Distributed Task Allocation to Enable Collaborative Autonomous Driving with Network Softwarization. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 2175–2189. [[CrossRef](#)]
40. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous Control with Deep Reinforcement Learning. *arXiv* **2015**, arXiv:1509.02971.