

Joint Optimization of Vehicular Sensing and Vehicle Digital Twins Deployment for DT-Assisted IoVs

Lun Tang , Zhangchao Cheng , Jun Dai , Hongpeng Zhang , and Qianbin Chen , Senior Member, IEEE

Abstract—Emerging Intelligent Transportation Systems (ITS) applications in the B5G/6G era have higher demands on real-time information in real traffic scenarios, and the current Internet of Vehicles (IoVs) can hardly support the operation of such applications. To realize the efficient operation of ITS applications, we integrate Digital Twin (DT) technology into IoVs and propose a framework of DT-assisted cloud-edge collaboration IoVs for intelligent transportation. DT synchronization requires vehicular sensing and data uploading, where the sensing capability and sensing policy of different vehicles affect the accuracy of DT, and the simultaneous sensing of neighboring vehicles creates data redundancy. The deployment strategy of Vehicle DTs (VDTs) at the network edge affects the real-time performance of DT. The mobility of vehicles, the low-latency requirements of DT, and the limited heterogeneous resources of edge servers pose great challenges to the deployment of VDTs. To address the above problems, we established the vehicular sensing model considering sensing quality, cost and redundancy. Then, A DT synchronization mechanism is designed and an improved Age of Information (AoI) metric is used to measure the freshness of the real vehicle state data received by the cloud during the DT synchronization process. We proposed a joint optimization problem of vehicular sensing and VDTs deployment to maximize the system Quality of Services (QoS), which is reflected through the vehicular sensing quality, AoI and system cost in DT synchronization process. We develop an algorithm based on Multi-Agent Deep Reinforcement Learning (MADRL) to solve this optimization problem, called DTSD-MAPPO. Numerical results show that the scheme reduces the system cost and improves the accuracy and real-time in DT synchronization.

Index Terms—Internet of vehicles (IoVs), digital twin (DT), age of information (AoI), deep reinforcement learning.

I. INTRODUCTION

A S AN important paradigm of 5G communication system, Internet of Vehicles (IoVs) interconnects vehicles, roads, pedestrians, and other traffic-related factors by integrating emerging technologies such as Mobile Edge Computing

Manuscript received 9 August 2023; revised 27 November 2023 and 8 January 2024; accepted 29 February 2024. Date of publication 5 March 2024; date of current version 15 August 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62071078, and in part by the Sichuan Science and Technology Program under Grant2021YFQ0053. The review of this article was coordinated by Dr. Yao Ma. (*Corresponding author: Zhangchao Cheng*)

Lun Tang, Zhangchao Cheng, Jun Dai, Hongpeng Zhang, and Qianbin Chen are with the School of Communications and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China, and also with the Chongqing Key Laboratory of Mobile Communications Technology, Chongqing 400065, China (e-mail: tangl@cqupt.edu.cn; chengzhangchao1454@163.com; 878607471@qq.com; 759974869@qq.com; chenqb@cqupt.edu.cn).

Digital Object Identifier 10.1109/TVT.2024.3373175

(MEC), network slicing, and Artificial Intelligence (AI) to realize real-time data transmission and information sharing in road traffic scenarios, which promotes the modernization and intelligent development of Intelligent Transportation System (ITS) [1]. However, the current capabilities of ITS in real-time traffic sensing and big data analysis are relatively limited [2], insufficient to meet the demanding requirements of emerging traffic applications in the B5G/6G era, such as path planning, traffic flow prediction, and autonomous driving [3].

Digital Twin (DT) is expected to provide new technological support and development direction for IoVs. The concept of DT was proposed by Dr. Michael Grieves and refers to the complete and consistent replication of physical objects in a virtual space, realizing the research and control of real objects by observing, analyzing, deducing, and manipulating the DT [4]. DT can be utilized to comprehensively and continuously collect and store vehicle state data and vehicular sensing data from real traffic scenarios, thus building a digital model of the entire road traffic scene, which can be presented in a visualized form. Training and deploying AI algorithm models in the DT enable efficient verification and low-cost experimentation of innovative technologies [5], and provide ITS with intelligent traffic services such as traffic flow prediction, vehicle path pre-planning, accident deduction and warning to improve the efficiency and safety of road traffic. Additionally, DT can also provide full lifecycle management services for ITS, enabling visual perception, quick fault localization, real-time diagnostics, and closed-loop control of the entire transportation system [6]. This comprehensive enhancement elevates the functionality and performance of ITS, offering a more sophisticated level of support for the intelligent and digital transformation of transportation.

DT require real data of the whole elements of the physical space and the whole life-cycle to support their evolution, which requires high-precision sensing of various transportation-related factors in the whole traffic scenario [7]. Currently, the sensing data in IoVs mainly originates from Intelligent Connected Vehicles (ICVs) equipped with on-board sensors (e.g., cameras, LIDAR, and millimeter-wave radar, etc.). However, it is a major challenge to effectively enable the collaborative sensing of massive data by thousands of mobile vehicles to realize the accuracy of data collection in the whole domain, given the different sensing capabilities of different vehicles. Firstly, due to the varying sensing capabilities of vehicles and the influence of adverse environmental conditions, vehicles may generate inaccurate sensing data, and low-quality data can reduce the precision of DT. Secondly, there is a high correlation

between the continuous sensing data, and a large number of sensing data packets all uploaded may greatly consume the available wireless communication bandwidth and the computational resources of the edge servers[8], which leads to a high synchronization latency between the DT and the real entities. Meanwhile, neighboring vehicles may sense the same entity causing data redundancy during sensing, and a large amount of redundant data uploading will lead to a waste of communication and computational resources. In order to cope with the above challenges, it is necessary to reasonably set the sensing frequency of vehicles and control the proportion of vehicles participating in sensing, so as to improve the accuracy of DT construction while reducing the system sensing cost and the data transmission volume. Additionally, it is important to evaluate the quality of vehicular sensing data and restrain the upload of low-quality data.

How to deploy Vehicle DTs (VDTs) in the IoVs is a fundamental question that requires further research. Since MEC technology reduces system latency and network transmission stress by executing computations at the network edge, **deploying VDTs at the edge ensures low-latency interactions with their corresponding object entities [9]**. The deployment location of VDTs causes changes of communication latency between vehicles and twins, so how to realize efficient and low-latency interactions between the vehicles and VDTs by optimizing the deployment of VDTs is also one of the main challenges currently faced [10]. Due to the high-speed mobility of vehicles, their VDT may dynamically migrate as the vehicle moves, incurring communication costs. Therefore, how to choose the optimal VDT migration strategy based on different VDT migration requirements and available network resources is also a significant challenge [11].

To address the above problems, this paper develops an effective joint optimization scheme for vehicular sensing and DT dynamic deployment, called DTSD. The scheme aims to maximize the Quality of Services (QoS) of DT-assisted cloud-side-end collaborative intelligent transportation system for the Internet of Vehicles by optimizing the vehicular sensing policy and the VDT deployment policy. The main contributions of this paper can be summarized as follows:

- 1) We propose a new framework of DT-assisted cloud-edge collaboration IoVs for intelligent transportation. Considering ICVs in the physical entity layer uploading vehicle state data and sensing data to the edge twinning layer, VDTs are dynamically deployed at the network edge. VDTs interact with the cloud on behalf of real vehicles to provide real data support for large-scale AI model training and traffic application development in the cloud. At the same time, the edge VDTs also deploys the cloud-trained AI models to realize traffic assistance for real vehicles on the road.
- 2) In this paper, we focus on the vehicular sensing and the VDTs dynamic deployment problem at the network edge. For vehicular sensing, we model it from three aspects: sensing quality, sensing cost, and sensing redundancy. A DT synchronization mechanism is designed and an improved AoI metric is used to measure the freshness of the real vehicle data received by the cloud during the DT

synchronization process. At the same time, considering that the high mobility of vehicle nodes may trigger the DT migration process thereby bringing some migration cost.

- 3) In this paper, a system QoS model including sensing quality, AoI and system cost is developed and a joint optimization model to maximize system QoS by optimizing vehicular sensing and vehicle DT dynamic deployment is formulated. We develop an algorithm to solve the above optimization problem based on Multi-Agent Deep Reinforcement Learning (MADRL), called DTSD-MAPPO. Finally, simulation results demonstrate the convergence and effectiveness of the proposed algorithm.

The remaining sections of this paper are organized as follows. Section II introduces an overview of related work in three aspects. We define the system model in Section III. The problem formulation and solutions are proposed in the Section IV. Simulation results are provided and discussed in Section V, and the paper is summarized in Section VI.

II. RELATED WORKS

This paper is applied to the IoVs scenario and involves technologies such as DT, mobile crowdsensing, and MEC. In this section, we introduce a number of studies on the above areas.

A. DT in IoVs with MEC

DT has shown significant effectiveness in processing real-time data streams, and when combined with MEC technology, it has already found extensive applications in the field of the Internet of Things, especially in the field of IoVs [12], [13], [14], [15] and Space-Air-Ground Integrated Network (SAGIN) [16], [17].

Dynamic DTs of vehicles and Road Side Units (RSUs) is designed to capture time-varying resource demand information, and a social-aware vehicular edge caching mechanism is proposed to realize optimal cache resource allocation [12]. In [13], DT is used to realize the test of real autonomous driving vehicles in virtual complex road scenes. In [14], a cooperative driving system is designed for unsignalized intersections using DT technology, which enables vehicles to cooperate with each other to pass through intersections without any traffic lights. In [15], a safer and smarter vehicle lane-changing method using DT and MEC is proposed. MEC is leveraged to enhance the vehicle's capabilities of perception and computation, ensuring real-time driving safety. The virtualization and offline learning functions of the DT enable the vehicle to achieve foresight lane-changing. In the field of SAGIN, DT is considered for comprehensive resource orchestration and heterogeneous access management, assisting in ensuring secure transmissions of heterogeneous downlink communications [16]. In [17], DT is introduced into the satellite-ground integrated vehicle network, where it collects channel feedback information from vehicles to assist in securing network transmissions.

However, the above research is mainly descriptive from the perspective of DT application, mainly focusing on the benefits and advantages that DT can bring to the system. However, few

studies have addressed how constructing DT affects the system in terms of the DT itself, such as the system cost incurred by DT, the latency between DT and physical entities, and the accuracy of DT mapping.

B. Vehicular sensing

Vehicular sensing technology can provide important real-time data and information for IoVs. In recent years, many researches have been conducted for vehicular crowdsensing and data collection in IoVs. Vehicular crowdsensing takes advantage of the mobility of vehicles and utilizes the sensing capabilities provided by on-board sensors equipped on vehicles recruit vehicles to collect sensory data in a large coverage area to accomplish challenging sensing tasks [18].

In [19], a vehicular social network is proposed, and a dynamic sensing mechanism is designed to improve the long-term benefits of vehicles. All vehicles get the optimal long term sensing policy by using deep reinforcement learning. However, this work did not take into account the timeliness of vehicular crowdsensing. The communication and computation latency is considered during vehicular crowdsensing in [20], and a cost-and-quality aware data collection scheme is proposed to realize the reliable and timely data uploads while ensuring the trade-off between sensing precision and communication overhead. But this work does not consider the issue of data redundancy. Sensing data redundancy and communication overhead during crowdsensing is cosidered in [21]. A preprocessing mechanism is designed in this work for sensing data to reduce the transmitted data volume under the premise of ensuring the quality of sensing data, and proposes a grid selection algorithm to realize the sensing of abundant information regions, and an on-line parameter adjustment algorithm to adjust the sensing mode according to the current data redundancy, thus reducing the communication overhead.

However, the vehicular sensing discussed in the above work are almost all in mobile crowdsensing scenario, and some of the ideas can be borrowed to the DT construction and DT synchronization process, which is the main topic of this paper, but they are truly different. First, vehicular crowdsensing is generally a small number of vehicles to sense a target in a certain area, while DT sensing requires a large number of vehicles in the road to sense the whole scene. Second, vehicular crowdsensing is generally performed when there is a service request, while DT sensing is continuously performed throughout the whole life cycle. Additionally, DT sensing has higher requirements for low latency and high precision.

C. DT deployment

The DT deployment differs from the classical service placement problem in that it considers both the computational and storage resources required to maintain the DTs, as well as the continuous synchronization of data updates between the physical entities and the DTs. Several studies have explored how to deploy DTs in edge networks, and current research generally considers placing DTs at the network edge to ensure low-latency interactions with their corresponding physical entities. In [22],

a problem about how to place DTs with social capabilities at the edge is addressed, and the target is to minimize the sum of two latency, including the latency between physical devices and the corresponding DTs and the latency among DTs of friend devices. But the processing time of DT data by the edge server is not taken into account. In [23], DRL and transfer learning are applied to solve the optimal DT placement and DT migration problems respectively with the objective of minimizing the average system delay. However, using two algorithms to solve the DT deployment and the migration problem separately increases the overhead and reduces the system efficiency. Data age target is introduced to the DT deployment problem with the aim of minimizing the application response latency to the maximum data request among all physical entities [24], but it does not consider the mobility of the end users and the DT migration process.

Different from the above work, in this paper, we consider how to deploy vehicle DTs dynamically in an IoT scenario. We consider the low latency requirement for DT synchronization and introduce AoI to represent the freshness of real vehicle information received by DTs. At the same time, we consider how to deploy vehicle DTs while taking into account the high-speed mobility of vehicles that causes the DT migration problem.

III. SYSTEM MODEL

This section outlines a general overview of the proposed framework of DT-assisted cloud-edge collaboration IoVs for intelligent transportation and summarizes our modeling assumptions. We consider that the system operates according to a discrete timing, discretized as $\mathcal{T} = \{0, 1, \dots, t, t + 1, \dots\}$. The duration of each time slot $t \in \mathcal{T}$ is τ , which is assumed to be sufficiently small, ensuring that the policy of vehicular sensing and VDT deployment remains unchanged within a single time slot.

A. Network scenario

The proposed framework of DT-assisted cloud-edge collaboration IoVs for intelligent transportation is shown in Fig. 1. In the proposed framework, it consists of three layers: 1) physical entity layer; 2) edge twinning layer; 3)cloud twin layer.

- 1) *Physical entity layer:* The physical entity layer mainly consists of mobile vehicles with limited computing and storage resources. The set of vehicles is denoted as $V = \{1, 2, \dots, I\}$. In time slot $t \in \mathcal{T}$, vehicle i accesses the edge server (ES) by connecting to a single base station (BS), uploading its sensing data and status data to ES for DT synchronization.
- 2) *Edge twinning layer:* The edge twinning layer consists of BSs that provide access services and ESs that provide computation and storage services, and each ES is associated with all ESs. The set of BSs is denoted as $BS = \{1, 2, \dots, J\}$. The set of ESs is denoted as $ES = \{1, 2, \dots, K\}$, and the heterogeneous resources of ES k is denoted as $R_k = \{ES_k^{CPU}, ES_k^{Disk}\}$, where ES_k^{CPU} and ES_k^{Disk} denote the total CPU computing resources and disk storage resources of ES k , respectively.

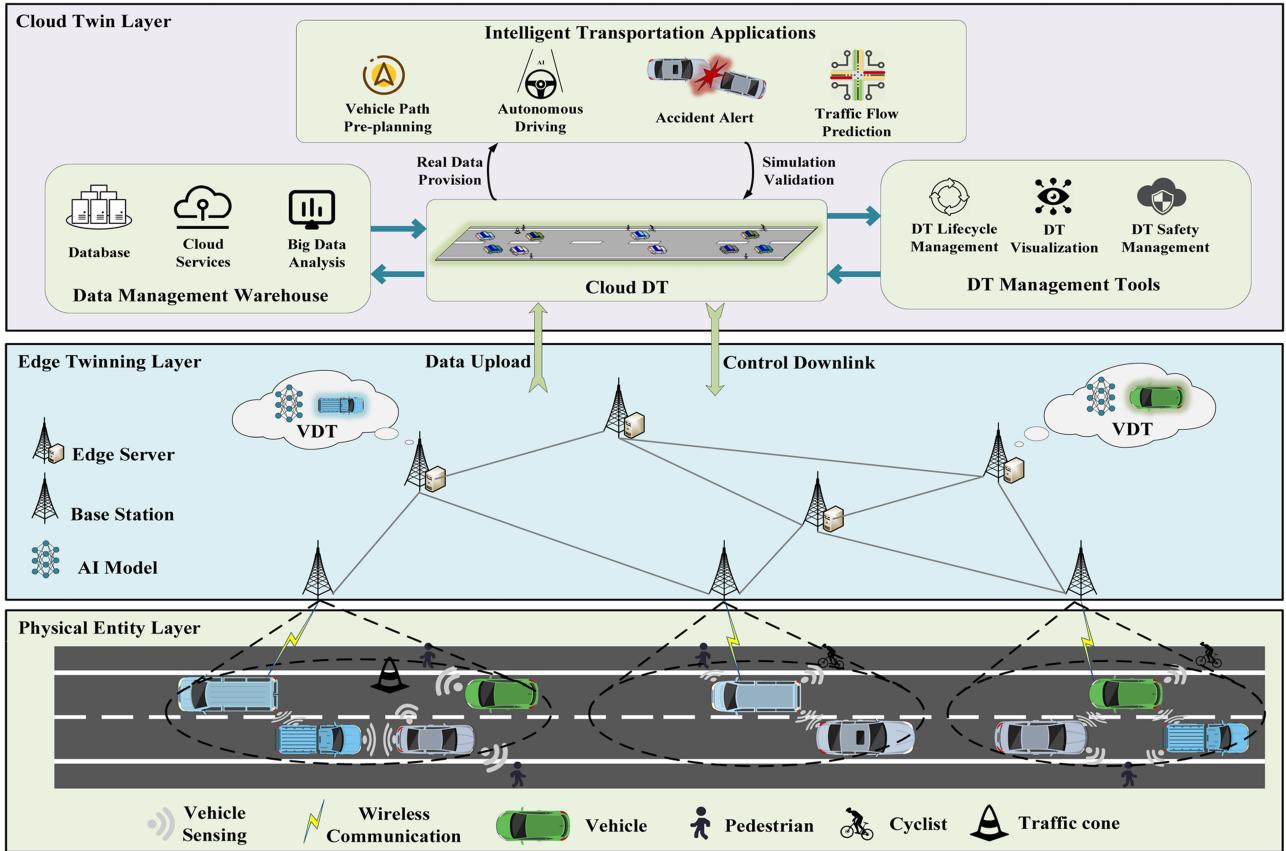


Fig. 1. Framework of DT-assisted cloud-edge collaboration IoVs for intelligent transportation

The edge twinning layer maintains the VDTs, and the VDT of vehicle i is denoted as:

$$DT_i = \{Conf_i, H_i, Env_i(t), S_i(t), Act_i, \Delta S_i(t+1)\}. \quad (1)$$

As shown in the above expression, the DT_i retains the entire information of vehicle i , including hardware configuration, historical operation data, surrounding environment, and real-time operational status. Maintaining a VDT requires certain computational and storage resources, denoted as $R_{DT_i} = \{DT_i^{CPU}, DT_i^{Disk}\}$, where DT_i^{CPU} and DT_i^{Disk} denote the CPU computational and disk storage resources required to maintain DT_i , respectively.

- 3) *Cloud twin layer*: The cloud twin layer maintains a large-scale DT management platform and multiple application servers. The cloud DT operates by receiving data of all VDTs from the edge twinning layer, denoted as:

$$DT = \{H, Env(t), V(t)\}. \quad (2)$$

The above expression indicates that the cloud DT stores all the historical data and environment information of the whole traffic scene, as well as the status information of all vehicles. The cloud twin layer is equipped with sufficient heterogeneous resources, enabling management and control of the physical layer through DT simulation, prediction and optimization.

B. Vehicular sensing model

We define $f_i^{se}(t)$ as the sensing frequency of the vehicle i in the time slot t , and the data size generated by the vehicle's single sensing process is D_0^{se} , then the sensing data size generated by the vehicle i in the time slot t can be written as:

$$D_i^{se}(t) = D_0^{se} \cdot f_i^{se}(t) \cdot \tau. \quad (3)$$

We assume that different vehicles are equipped with sensing devices with different sensing capabilities, and define dis_i^r as the sensing capability of the vehicle i , i.e., the maximum sensing range. The quality of single sensing of the sensing vehicle is evaluated based on the distance between the vehicle and the sensing target, the speed of the vehicle, and the external environment at that time. We define the sensing quality of the vehicle i to sense the target o in a single sensing process as:

$$q_i^o = \alpha_1 \cdot \frac{dis_i^r - dis(i, o)}{dis_i^r} \cdot \frac{1}{1 + ve_i}, \quad (4)$$

where $\alpha_1 \in (0, 1)$ is defined as the influence factor of the external environment on the sensing quality, and its value becomes higher with the excellence of the external environment. dis_i^r is the maximum sensing distance of the vehicle i , which is determined by the performance of the on-board sensor, and the target that exceeds the maximum sensing distance is regarded as invalid sensation. $dis(i, o)$ indicates the sensing distance between the vehicle i and the target o . ve_i is the vehicle speed

at the moment of sensing. The sensing quality of the vehicle decreases with the increase of vehicle speed. We assume that the sum of targets sensed by vehicle i in a single sensing is O , the sensing quality of vehicle i is defined as the average sensing quality of all targets, which is expressed as:

$$q_i = \frac{1}{O} \cdot \sum_{o=1}^O q_i^o. \quad (5)$$

The sensing quality of a sensing vehicle in a time slot is related to the quality of a single sensation and the sensing frequency. Assuming that the sensing frequency of vehicle i in time slot t is $f_i^{se}(t)$, the sensing quality of vehicle i in time slot t is defined as:

$$q_i(t) = \frac{1}{f_i^{se}(t)} \cdot \sum_{t=1}^{f_i^{se}(t)} q_i. \quad (6)$$

To ensure the quality of the uploaded data, an upload trigger threshold q_{th} is introduced and when the sensing quality is greater than q_{th} , it indicates that the sensing data is qualified to upload the data. The average sensing quality of the system in time slot t is defined as:

$$Q_{se} = \frac{\sum_{i=1}^I q_i(t)}{I}. \quad (7)$$

We assume that the cost of a vehicle sensing once is c^{se} , then the sensing cost of vehicle i in one time slot is $c^{se} \cdot f_i^{se}$. In time slot t , the total sensing cost of the system is:

$$Cost^{se} = \sum_{i=1}^I c^{se} \cdot f_i^{se}(t). \quad (8)$$

The BS is responsible for evaluating the sensing redundancy within its coverage area after vehicles upload sensing data. Evaluating the sensing redundancy of vehicles in an area can be calculated based on the locations of the vehicles and the maximum sensing range of the vehicles. If $dis(i, i') < dis_i^r + dis_{i'}^r$ is satisfied for vehicle i and vehicle i' , it means that there exists a region of sensing redundancy between the two vehicles. The sensing redundancy in one time slot is also related to the sensing frequency of vehicles. We define the sensing redundancy of the whole system as:

$$W_R = \frac{2 \cdot \sum_{i=1}^I \sum_{i'=i+1}^I \omega_{i,i'} \cdot \frac{f_i^{se} + f_{i'}^{se}}{2f_{\max}^{se}} \cdot \frac{dis_i^r + dis_{i'}^r}{dis(i, i')}}{I^2 - I}, \quad (9)$$

where $\omega_{i,i'}$ indicates whether vehicle i and vehicle i' are neighboring or not, denoted as:

$$\omega_{i,i'} = \begin{cases} 1, & \text{if } dis(i, i') < dis_i^r + dis_{i'}^r \\ 0, & \text{else} \end{cases}, \quad (10)$$

where $dis(i, i') > 0$, it indicates that it is impossible for the two cars to coincide in position.

C. DT synchronization model

In this paper, we design a DT synchronization mechanism: the data upload interval of vehicles is dynamically determined based on the previous packet upload delay and the processing delay

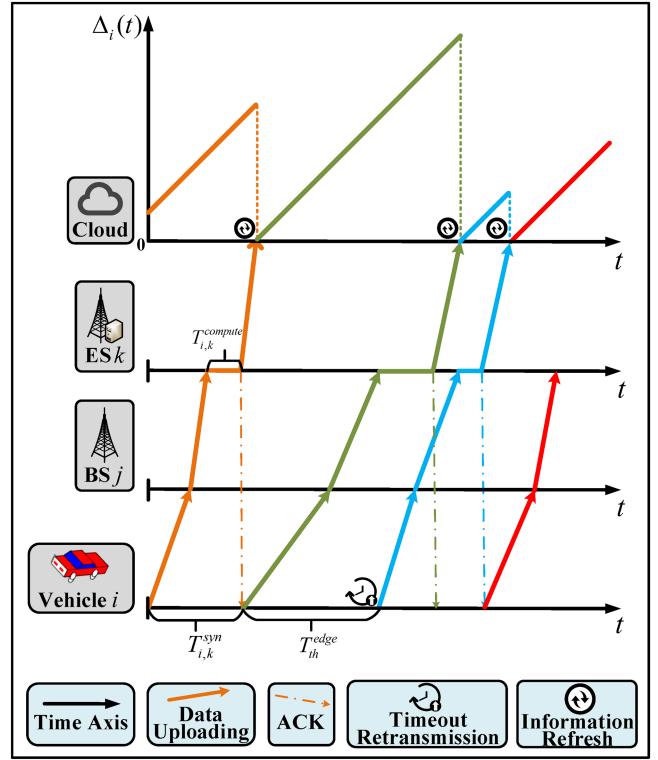


Fig. 2. DT synchronization process and cloud AoI.

of the edge server. When the edge server finishes processing the data uploaded by one vehicle, it returns an acknowledgement packet called ACK to the vehicle, and when the vehicle receives the acknowledgement message, it then uploads the next data packet.

In time slot t , vehicles upload data includes vehicle status data and vehicle sensing data. It can be written as:

$$D_i^{up}(t) = D_i^{se}(t) + D_i^{st}, \quad (11)$$

where D_i^{st} represents the vehicle status data, which can be considered as a small fixed value for each upload. $D_i^{se}(t)$ is the sensing data size of vehicle i in time slot t which varies with the change of sensing frequency f_i^{se} . In time slot t , when vehicle i does not sense, i.e. $f_i^{se}(t) = 0$, the upload data in time slot t contains only the status data of vehicle i .

As shown in Fig. 2, each color represents a single DT synchronization process, and a single DT synchronization process consists of the following stages.

- 1) *Vehicles upload data to the BS*: We consider that the communication between the vehicles and the BSs is in wireless transmission mode. The maximum uplink information rate between the vehicle i and the BS j is:

$$r_{i,j} = B \log_2 \left(1 + \frac{p_i \cdot h_{ij} \cdot dis(i, j)^{-\gamma_0}}{N_0 \cdot B} \right), \quad (12)$$

where B is the transmission channel bandwidth, p_i is the transmission power of the vehicle i , h_{ij} is the channel gain, N_0 is the power spectral density of noise, and $dis(i, j)^{-\gamma_0}$ is the transmission path loss. $dis(i, j)$ is the distance

between the vehicle i and the BS j , and γ_0 is a attenuation factor.

We assume that the vehicle i uploads the data D_i to the BS j in time slot t , the upload delay is noted as:

$$T_{i,j} = \frac{D_i}{r_{i,j}}, \quad (13)$$

where D_i is the size of data uploaded by the vehicle i and $r_{i,j}$ is the uplink information rate between the vehicle i and the BS j .

- 2) *BS uploads data to ES*: We consider that the wired transmission delay is related to the size of the transmitted data volume and the transmission distance. The wired transmission delay between the BS j and the ES k is written as:

$$T_{j,k}^{tran} = \psi \cdot D_i \cdot dis(j, k), \quad (14)$$

where ψ is the delay required to transmit a unit of data per unit of distance, and $dis(j, k)$ is the distance between the BS j and the ES k .

- 3) ES processes data to maintain the VDTs. The computational resources of the ES can be allocated to multiple vehicles for maintaining their VDTs. The processing time for the ES k to process the uploaded data from the vehicle i is denoted as:

$$T_{i,k}^{compute} = \frac{f(D_i)}{DT_i^{CPU}}, \quad (15)$$

where $f(D_i)$ represents the amount of computation when the uploaded data size is D_i .

The transmission delay of ACK can be neglected because data size is small enough. Therefore, when DT_i is deployed at the ES k , the DT synchronization delay is written as:

$$T_{i,k}^{syn} = T_{i,j} + T_{j,k}^{tran} + T_{i,k}^{compute}. \quad (16)$$

In the DT synchronization process, we assess the system performance by AoI, a metric that evaluate freshness of arriving information [25]. In this paper, we consider two types of AoI: AoI for cloud DT and AoI for edge DT, which capture the real-time requirements of cloud and VDT.

- 1) *AoI for edge DT*: The DT synchronization period (i.e., the time interval between two DT updates) should be less than or equal to a threshold value, it can be written as:

$$T^{syn} \leq T_{th}^{edge}, \quad (17)$$

and this constraint keeps the AoI of the edge DT from becoming too large. If the vehicle i has not received the ACK when the time reaches the threshold T_{th}^{edge} , then the vehicle i will retransmit the data to ensure the freshness of the edge DT.

- 2) *AoI for cloud DT*: During the DT synchronization, we use the truncated-AoI to quantify the freshness of the information in the cloud DT, denoted by Δ . Δ denotes the length of time that the current moment has elapsed since the last time data was refreshed, and it is set to 0 whenever the cloud information is refreshed.

We assume that at the moment of t_0 , the cloud happens to receive the DT information of DT_i uploaded by the ES, from the moment of t_0 until the next data update, the AoI of the cloud about DT_i is written as:

$$\Delta_i(t) = \sum_{k=1}^K z_{i,k} \cdot (t - t_0) \quad t \in [t_0, t_0 + T_{i,k}^{syn} + T_{k,Cloud}^{tran}], \quad (18)$$

where $z_{i,k} = \{0,1\}$ is a binary variable indicating whether the DT_i is deployed on the ES k or not. The AoI of the cloud about DT_i is related to the deployment location of DT_i , because the transmission delay is different when DT_i deployed in different locations.

We define the average AoI of the vehicle i in DT synchronization as:

$$\bar{\Delta}_i(t) = \frac{1}{t} \int_0^t \Delta_i(t) dt. \quad (19)$$

We define the peak AoI of vehicle i in DT synchronization is $\Delta_i^m = \max\{\Delta_i(t)\}$, i.e., the maximum value of AoI at all moments during DT synchronization. The cloud DT also has AoI objective, i.e., the maximum tolerable delay of the data received from the vehicle i , which is denoted by T_{th}^{Cloud} . The peak AoI of each vehicle should satisfy the cloud AoI objective constraint in DT synchronization, which can be written as:

$$\forall i \in V, \quad \Delta_i^m \leq T_{th}^{Cloud}. \quad (20)$$

The average AoI of cloud DT about all vehicles can be written as:

$$\bar{\Delta} = \frac{1}{I} \sum_{i=1}^I \bar{\Delta}_i(t). \quad (21)$$

D. DT deployment and migration model

We define the matrix $Z = [z_{i,k}]$ to denote the deployment strategy of VDTs, where $z_{i,k}$ is a binary variable indicating whether the DT_i is deployed on the ES k or not. The deployment strategy matrix is denoted as:

$$Z = \begin{pmatrix} z_{1,1} & \dots & z_{1,K} \\ \vdots & \ddots & \vdots \\ z_{I,1} & \dots & z_{I,K} \end{pmatrix}. \quad (22)$$

Due to the high mobility of vehicles, they may gradually move away from the ES currently maintaining their VDT and move near another ES. At this point, it is necessary to adjust the deployment location of their VDT and trigger the DT migration process if the deployment location is changed.

In this paper, we consider that the migration process mainly consists of two stages. Firstly, the newly determined ES needs to open a storage space for maintaining VDTs and perform VDT initialization operations. Then, the old ES will transmit the DT history data to the new ES. Both stages lead to system cost. The opening cost increases every time the DT migration process is triggered. The transmission cost is mainly related to the data size to be migrated and the distance between old and new ES.

When the deployment location of DT_i is migrated from ES k_1 to ES k_2 , the migration cost is denoted as:

$$Cost_i^{mig} = c^{ou} + c^{mig} \cdot DT_i^{Disk} \cdot dis(k_1, k_2), \quad (23)$$

where c^{ou} denotes the opening cost, and c^{mig} denotes the unit migration cost, i.e., the cost of transmitting a unit data per a unit distance.

For each vehicle i , if the migration process is not triggered when the deployment strategy for the current time slot is the same as the previous time slot, then $z'_i = 0$, otherwise $z'_i = 1$. The migration strategy for the whole system can be denoted by a matrix:

$$Z' = (z'_1, z'_2, \dots, z'_I). \quad (24)$$

The migration cost of the whole system can be described as:

$$Cost^{mig} = \sum_{i=1}^V z'_i \cdot Cost_i^{mig}. \quad (25)$$

E. QoS model

In order to quantify and measure the QoS of system, several aspects are considered in terms of DT construction accuracy, average AoI of cloud DT, system cost, and sensing redundancy. The system QoS of DT construction accuracy can be measured in the average sensing quality of vehicles in the system, which is denoted in (7).

The system QoS of AoI can be written as:

$$Q_{\bar{\Delta}} = 1 - \frac{\bar{\Delta}}{T_{th}^{Cloud}}. \quad (26)$$

The system cost consists of two parts, the system sensing cost recorded in (8) and the DT migration cost recorded in (23). The system QoS of vehicle sensing cost can be denoted as:

$$Q_{Cost^{se}} = 1 - \frac{Cost^{se}}{\sum_{i=1}^I c^{se} \cdot f_{max}^{se}}, \quad (27)$$

where f_{max}^{se} denotes the maximum vehicle sensing frequency, and $\sum_{i=1}^I c^{se} \cdot f_{max}^{se}$ denotes the maximum system sensing cost.

We assume that all VDTs migrate in one time slot, and they migrate to the furthest ES. Then, the maximum system DT migration cost can be denoted as:

$$Cost_{max}^{mig} = \sum_{i=1}^I c^{ou} + c^{mig} \cdot DT_i^{Disk} \cdot dis_{max}^k, \quad (28)$$

where dis_{max}^k denotes the maximum distance between ESs. So, the system QoS of DT migration cost can be denoted as:

$$Q_{Cost^{mig}} = 1 - \frac{Cost^{mig}}{Cost_{max}^{mig}}. \quad (29)$$

Equation (9) defines the sensing redundancy of the system. We define the system QoS of sensing redundancy as:

$$Q_R = 1 - W_R. \quad (30)$$

Therefore, the QoS of system can be written as:

$$\begin{aligned} Q = & \beta_1 \cdot Q_{se} + \beta_2 \cdot Q_{\bar{\Delta}} + \beta_3 \cdot Q_{Cost^{se}} \\ & + \beta_4 \cdot Q_{Cost^{mig}} + \beta_5 \cdot Q_R, \end{aligned} \quad (31)$$

where β denotes the weights of each components, and satisfies $\beta_1, \beta_2, \beta_3, \beta_4, \beta_5 \in (0, 1)$, and $\beta_1 + \beta_2 + \beta_3 + \beta_4 + \beta_5 = 1$.

IV. PROBLEM FORMULATION AND SOLUTION

In this section, we propose a joint optimization problem of vehicular sensing and VDT deployment to maximize the system QoS. Subsequently, we develop a MADRL-based algorithm, called DTSD-MAPPO, to solve the optimization problem.

A. Problem formulation

The main objective of this paper is to find the optimal vehicular sensing and VDT deployment strategy to maximize the QoS of the DT-assisted cloud-edge collaboration IoVs system for intelligent transportation. Simultaneously, it ensures the accuracy of DT construction and the AoI target in DT synchronization, while also preventing any exhaustion of heterogeneous resources in each ES. Specifically, the optimization objectives are as follows:

$$P : \max_{F, Z} Q$$

$$\begin{aligned} \text{s.t. } C1 : & \forall t \in \mathcal{T}, \forall i \in V, \sum_{k=1}^K z_{i,k} = 1 \\ C2 : & \forall t \in \mathcal{T}, \forall k \in ES, \sum_{i=1}^I z_{i,k} \cdot DT_i^{CPU} \leq ES_k^{CPU} \\ C3 : & \forall t \in \mathcal{T}, \forall k \in ES, \sum_{i=1}^I z_{i,k} \cdot DT_i^{Disk} \leq ES_k^{Disk} \\ C4 : & t \in \mathcal{T}, \forall i \in V, q_i(t) \geq q_{th} \\ C5 : & \forall t \in \mathcal{T}, \forall i \in V, T_{syn}^i(t) \leq T_{th}^{edge} \\ C6 : & \forall t \in \mathcal{T}, \forall i \in V, \Delta_m \leq T_{th}^{Cloud} \\ C7 : & \beta_1, \beta_2, \beta_3, \beta_4, \beta_5 \in (0, 1) \\ C8 : & \beta_1 + \beta_2 + \beta_3 + \beta_4 + \beta_5 = 1 \\ C9 : & \forall t \in \mathcal{T}, \forall i \in V, f_i^{se}(t) \in [0, f_{max}^{se}] \\ C10 : & \forall t \in \mathcal{T}, \forall i \in V, z_{i,k} \in \{0, 1\} \end{aligned} \quad (32)$$

In the above constraints, C1 ensures that each VDT can only be deployed on a unique ES within a single time slot. C2-C3 ensures that the deployment of VDTs will not exhaust the computational and storage resources of any ES. C4 ensures that the sensing data uploaded by all vehicles in each time slot meets the sensing quality thresholds. C5-C6 ensures that the synchronization process of the DT satisfies both the edge end and cloud AoI requirements. C7-C8 illustrates the weights of each component of the system QoS model. C9 indicates the

range of sensing frequency of each vehicle. C10 illustrates that the VDT deployment variable is a binary variable.

B. MA-POMDP

Since the joint optimization problem proposed in this paper is NP-hard and contains discrete variables, the problem cannot be solved using traditional optimization algorithms. So, we proposed a MAPPO-based multi-agent deep reinforcement learning algorithm to solve it. First, we model the optimization problem as a MA-POMDP (Multi-Agent Partially Observable Markov Decision Process). The MA-POMDP can be represented as a quadruple, denoted as:

$$\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}\}, \quad (33)$$

where $\mathcal{S} = \prod_{j=1}^J \mathcal{S}^j$ and $\mathcal{A} = \prod_{j=1}^J \mathcal{A}^j$ are the action spaces of the joint state space of all agents, \mathcal{S}^j and \mathcal{A}^j are the state space and action space of the agent j , respectively. $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the set of state transition probabilities. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the set of agent rewards. In time slot t , we assume that the global state is $s_t = (s_t^1, s_t^2, \dots, s_t^J) \in \mathcal{S}$ and the agent perform the joint action $a_t = (a_t^1, a_t^2, \dots, a_t^J) \in \mathcal{A}$, and then each agent receives the reward $r_t = (r_t^1, r_t^2, \dots, r_t^J) \in \mathcal{R}$. The environment transfers to a new state $s_{t+1} \in \mathcal{S}$ with probability $P(s_{t+1}|s_t, a_t) \in \mathcal{P}$.

In this paper, each BS in the environment is treated as an agent that observes local state information and makes decisions based on a policy function to maximize its own long-term cumulative discount rewards. The detailed MA-POMDP elements are defined as follows:

- 1) *State*: The global state space \mathcal{S} contains real-time sensing quality information of all vehicles, all VDTs deployment locations, vehicle-BS association information, resource utilization of all ESs, and the AoI information about all vehicles.
- 2) *Observable state*: The local observable state space s^j of each agent j contains the current real-time sensing quality information, VDT deployment locations, and AoI information of the vehicles associated with agent j , and the remaining resources of the corresponding ES.
- 3) *Action*: The action of each agent j can be denoted as $a^j = \{f, z\}$, including the sensing frequency and VDT deployment of vehicles in the coverage area of agent j .
- 4) *Reward*: We define the reward of the agent j in time slot t as:

$$r_t^j = \frac{\sum_{i=1}^I x_{i,j} \cdot Q_i}{\sum_{i=1}^I x_{i,j}}, \quad (34)$$

where the reward is denoted by the contribution to the system QoS of vehicles associated with agent j . $x_{i,j}$ is a binary variable which denotes the association of vehicle i with agent j . Q_i denotes the contribution of vehicle i to the system QoS. The global reward in time slot t is

$$R_t = \sum_{j=1}^J r_t^j. \quad (35)$$

C. DTSD-MAPPO

In this paper, an online optimization algorithm based on MAPPO is proposed for solving the joint optimization problem of vehicular sensing and VDT deployment, called DTSD-MAPPO (Digital Twin Sensing and Deployment-MAPPO). MAPPO is a deep reinforcement learning algorithm that introduces the Proximal Policy Optimization (PPO) algorithm into a multi-agent environment, using a centralized training-distributed execution framework based on Actor-Critic [26]. In the training phase, centralized value functions are used to stabilize the training process by considering the states and actions of all agents. In the execution phase, the distributed agents do not need any additional information and make decisions based solely on the local information they observe. The network architecture of DTSD-MAPPO is shown in Fig. 3.

The policy of the agent j is approximated by a Deep Neural Network (DNN) with parameter θ^j , called Actor network. To enable collaborative training of the agents, a virtual center node is established to deploy a DNN with parameter ϕ , called Critic network. This virtual node communicates with each node and can be located in any of the ESs. A twin Actor network with the same parameters θ^j will be created at the center node for each agent j to reduce the additional communication overhead.

We define π^j as the policy of the agent j , which is the probability of choosing the action $a^j \in \mathcal{A}^j$ when the observable state is $s^j \in \mathcal{S}^j$. We use a DNN to approximate π^j , denoted as $\pi_{\theta^j}(a^j|s^j)$. The joint policy of all agents is denoted as:

$$\pi(a|s) = \prod_{j=1}^J \pi_{\theta^j}(a^j|s^j). \quad (36)$$

The target of MAPPO training is to find the optimal joint policy $\pi(a|s)$ to maximize the discount reward U_t , which is denoted by:

$$U_t = R_t + \gamma \cdot R_{t+1} + \dots + \gamma^k \cdot R_{t+k}, k \rightarrow \infty, \quad (37)$$

where $\gamma \in (0, 1]$ is the discount factor, which measures the degree of attenuation of future rewards in the calculation of cumulative rewards.

We define the global action value function as:

$$Q^{\pi_\theta}(s_t, a_t) = \mathbb{E}[U_t|s = s_t, a = a_t]. \quad (38)$$

We define the global state value function as:

$$V^{\pi_\theta}(s_t) = \sum_{a \in \mathcal{A}} \pi_\theta(a|s_t) \cdot Q^{\pi_\theta}(s_t, a). \quad (39)$$

The advantage function $A^{\pi_\theta}(s_t, a_t)$ is used to evaluate how good or bad it is to take an action in a given state relative to the average, defined as:

$$A^{\pi_\theta}(s_t, a_t) = Q^{\pi_\theta}(s_t, a_t) - V^{\pi_\theta}(s_t). \quad (40)$$

When training MAPPO, the process of sampling data and optimizing policy can be separated. Initially, agents use the current policy to interactively sample the environment, perform a series of actions, observe the corresponding rewards and state

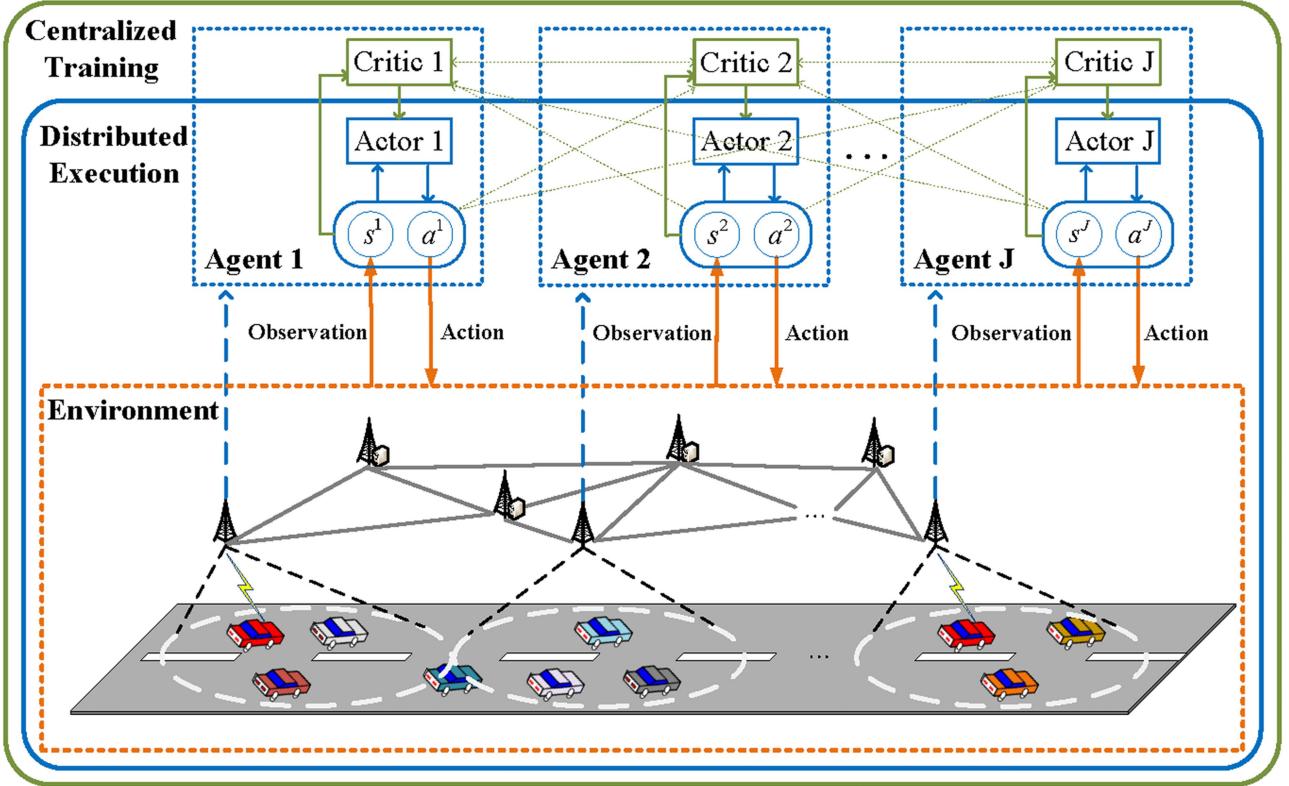


Fig. 3. DTSD-MAPPO framework.

transitions. The samples are denoted as:

$$\tau = \{s_0, a_0, r_0, s_1, a_1, \dots, s_{T-1}, a_T\}. \quad (41)$$

Then, during the training phase, the samples are used to update the Actor network parameters. MAPPO uses an approximate optimization method to maximize the cumulative reward while keeping the updated policy within a certain range from the old one. During each update, the Actor network updates the network parameters by means of a policy loss function, and the loss function of the Actor network corresponding to the agent j is computed as:

$$L_{Actor}(\theta^j) = \mathbb{E}_{\pi_{\theta^j}} \left\{ \min \left[\frac{\pi_{\theta^j}^{new}(a_t^j | s_t^j)}{\pi_{\theta^j}^{old}(a_t^j | s_t^j)} \cdot \hat{A}_j^{\pi_{\theta^j}}(s_t, a_t), \right. \right. \\ \left. \left. clip \left(\frac{\pi_{\theta^j}^{new}(a_t^j | s_t^j)}{\pi_{\theta^j}^{old}(a_t^j | s_t^j)}, 1 - \varepsilon, 1 + \varepsilon \right) \cdot \hat{A}_j^{\pi_{\theta^j}}(s_t, a_t) \right] \right\}, \quad (42)$$

where $\pi_{\theta^j}^{old}$ denotes the old policy of the agent j and $\pi_{\theta^j}^{new}$ denotes the current new policy of the agent j . $clip()$ is a truncation function that ensures that the gap between the updated parameters and the old parameters is not too large. ε is a hyperparameter that sets the limiting range of the $clip()$ function. $\hat{A}_j^{\pi_{\theta^j}}(s_t, a_t)$ is an estimate of the advantage function $A_j^{\pi_{\theta^j}}(s_t, a_t)$, which is obtained by Generalized Advantage Estimation (GAE) [27],

defined as:

$$\hat{A}_j^{\pi_{\theta^j}}(s_t, a_t) = \delta_t + \gamma \lambda \delta_{t+1} + \dots + (\gamma \lambda)^{T-1-t} \delta_{T-1}, \quad (43)$$

where $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$, and $\lambda \in [0, 1]$ is a hyperparameter for balancing bias and variance.

The parameters of the Actor network of the agent j are iteratively updated by stochastic gradient ascent, denoted as:

$$\theta_{t+1}^j \leftarrow \theta_t^j + \alpha_{Actor} \cdot \nabla_{\theta^j} L_{Actor}(\theta^j), \quad (44)$$

where α_{Actor} is the learning rate of Actor network.

The Critic network updates the network parameters by means of a global state value loss function. The loss of the Critic network is:

$$L_{Critic}(\phi) = \frac{1}{2} [V_{\phi}(s_t) - V_j(s(t))]^2, \quad (45)$$

and the Critic network parameter ϕ is iteratively updated by stochastic gradient descent:

$$\phi_{t+1} \leftarrow \phi_t - \alpha_{Critic} \cdot \nabla_{\phi} L_{Critic}(\phi), \quad (46)$$

where α_{Critic} is the learning rate of the Critic network.

Based on above-mentioned discussions, we proposed the DTSD-MAPPO algorithm, which is summarized in Algorithm 1.

Algorithm 1: DTSD-MAPPO Algorithm.

```

1: Initialize each Actor network parameters  $\theta_0^j$ , each Critic
   network parameters  $\phi_0^j$ , and the experience buffer.
2: Input: the maximum length of episodes  $epi^{\max}$ , the
   maximum length of steps  $step^{\max}$ , the total number of
   epochs  $epoch^{\max}$ .
3: Output: all trained Actor networks.
4: for  $episode = 1, 2, \dots, epi^{\max}$  do
5:   Get the initial state of the environment  $s_0$ .
6:   for  $t = 1, 2, \dots, step^{\max}$  do
7:     for  $j = 1, 2, \dots, J$  do
8:       The agent  $j$  get the observable state  $s_t^j$ .
9:       The agent  $j$  executes the action  $a_t^j$  based on the old
          policy, and gets the reward  $r_t^j$  and the next state
           $s_{t+1}^j$ .
10:      The agent  $j$  put the sample  $\tau = \{s_t^j, a_t^j, r_t^j, s_{t+1}^j\}$ 
          into the experience buffer.
11:    end for
12:    The environment state  $s_t$  updates to  $s_{t+1}$ .
13:  end for
14:  Calculate the estimated advantage function by the (43).
15:  Calculate the target value of the Critic network  $V_j(s_t)$ .
16:  for  $epoch = 1, 2, \dots, epoch^{\max}$  do
17:    for  $j = 1, 2, \dots, J$  do
18:      Get the sample  $\tau = \{s_t^j, a_t^j, r_t^j, s_{t+1}^j\}$  from the
          experience buffer.
19:      Calculate the loss function of Actor network by the
          (42).
20:      Calculate the loss function of Critic network by
          the (45).
21:      Update the parameter of Actor network by the (44).
22:      Update the parameter of Critic network by the (46).
23:    end for
24:  end for
25:  Update all Actor network and Critic network
     parameters.
26: end for

```

V. NUMERICAL SIMULATIONS

In this section, we aim to assess the performance of the proposed DTSD-MAPPO algorithm. First, we introduce the simulation settings, including the traffic dataset, parameter settings, and simulation procedure. Then, we present the convergence performance of the DRL-based algorithm. Finally, we show the evaluation results of the performance of our proposed approach from multiple perspectives.

A. Simulation settings

We construct a simulation environment based on Python3.7 and implement our proposed DTSD-MAPPO scheme using Pytorch framework. In order to enhance the reliability and authenticity of the simulation, we adopt the trajectory dataset of more than 700,000 vehicles in a 24-hour period in a 400 square kilometer area of Cologne, Germany [28], and the sampling data

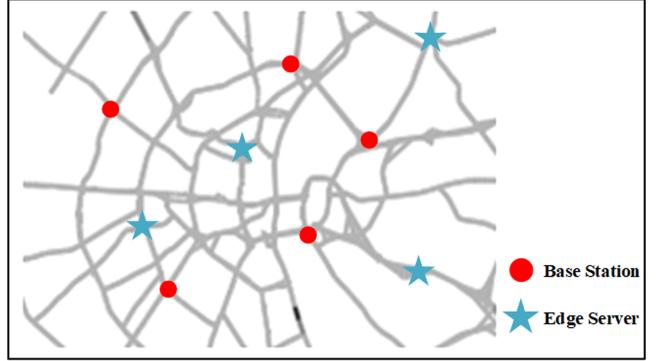


Fig. 4. Distribution of BSs and ESs.

contains vehicle coordinates, vehicle speed and time stamps, with a sampling interval of 1 second. In the actual simulation process, all the vehicle trajectory data in the coordinate range of (7900,12000)-(9840,15000) between 11:00-12:00 are used, containing 86,199 data items for 949 vehicles.

We treat 75% of these vehicles as ICVs with sensing capability, and the remaining 25% are considered as ordinary vehicles not equipped with any sensing devices, and the status information of these ordinary vehicles needs to be sensed by other ICVs to be obtained. In this way, we can simulate the vehicle sensing process reasonably. In the simulation scenario, we assume that there are 5 BSs and 4 ESs, and they are geographically distributed as shown in Fig. 4.

In the DRL training process, lightweight network architectures are designed for both Actor and Critic networks, comprising 1 input layer, 3 hidden layers, and 1 fully connected layer, and each hidden layer contains 128, 64 and 32 neurons, respectively. We use the ReLU nonlinear function as the activation function for all hidden layers to achieve high-efficiency computation. In addition, we experiment by setting multiple sets of hyperparameter combinations to find the optimal configuration for algorithm performance. The detailed hyperparameters and communication-related simulation parameter settings are summarized in Table I.

In order to prove the superiority of the proposed scheme of DTSD-MAPPO, we designed some baseline schemes for comparison.

For vehicular sensing, we designed two baseline schemes for comparison.

- 1) *Full Endeavor Sensing (FES)*: Each vehicle senses with all its sensing ability, i.e., the sensing frequency is maximized.
- 2) *Random Sensing (RS)*: The sensing frequency of each vehicle is set randomly.

For VDT deployment, we also designed two baseline schemes for comparison.

- 1) *Random Deployment (RD)*: the VDT is randomly deployed on any ES at the network edge.
- 2) *Nearest Deployment (ND)*: the VDT is deployed on the ES closest to the current access BS.

Since our proposed DTSD-MAPPO scheme is based on DRL, we also introduce two other DRL methods as comparative

TABLE I
SIMULATION PARAMETER SETTINGS

Parameters	Value
Wireless channel bandwidth B	50MHz
Noise power spectral density N_0	-127dBm/Hz
Coverage of BS	500m
Channel gain per unit distance h	-30dB
Vehicle transmitting power p	[500, 600]mW
Maximum sensing frequency of vehicles f_{max}^{se}	200Hz
VDT required computing/storage resources DT^{CPU}/DT^{Disk}	[1, 2]GHz / [2, 3]GB
VDT open up cost c^{ou}	1×10^{-5}
VDT unit migration cost c^{mig}	1×10^{-10}
ES computing/storage resources ES^{CPU}/ES^{Disk}	[80, 90]GHz / [2, 3]TB
Wired transmission unit delay ψ	1×10^{-12}
QoS weighting coefficients $\beta_1, \beta_2, \beta_3, \beta_4, \beta_5$	0.35, 0.25, 0.15, 0.15, 0.1
Learning rate of Actor network α_{Actor}	0.001
Learning rate of Critic network α_{Critic}	0.001
Discount factor γ	0.9
Balancing hyperparameter λ	0.95
Limiting hyperparameter ε	0.95
Maximum length of episodes epi^{max}	1200
Maximum length of steps $step^{max}$	300
Total number of epochs epo^{max}	5

benchmarks in order to highlight the superiority of the scheme proposed in this paper.

1) *PPO-based DT Sensing and Deployment (DTSD-PPO)*:

PPO is a single-agent DRL algorithm, which optimizes policy iteratively using multiple epochs of data to achieve gradual policy updates.

2) *MADDPG-based DT Sensing and Deployment (DTSD-MADDPG)*:

MADDPG is an off-policy algorithm with deterministic policy. It extends the Deep Deterministic Policy Gradient (DDPG) algorithm by incorporating a centralized critic that observes the joint actions and states of all agents. Unlike MAPPO, MADDPG relies on centralized value estimation for cooperation.

B. Convergence of DRL training

Similar to [10], [26] and [27], we evaluate the algorithm convergence in terms of the average reward function. Fig. 5 shows the average reward curves of three DRL-based schemes after 1200 episodes of training. At the beginning of training, agents are in the exploration phase and their actions are highly randomized, resulting in sparse and fluctuating rewards. As the number of training episodes increases, the agent gradually learns a better policy and hence the reward gradually increases and eventually converges, indicating that the agent has learned an optimal policy. Our proposed scheme DTSD-MAPPO reaches convergence at around 900 training iterations, and the final convergence value is about 0.79, higher than the other two DRL-based schemes.

Among them, the single-agent based PPO algorithm converges faster, but the final converged reward value is lower than

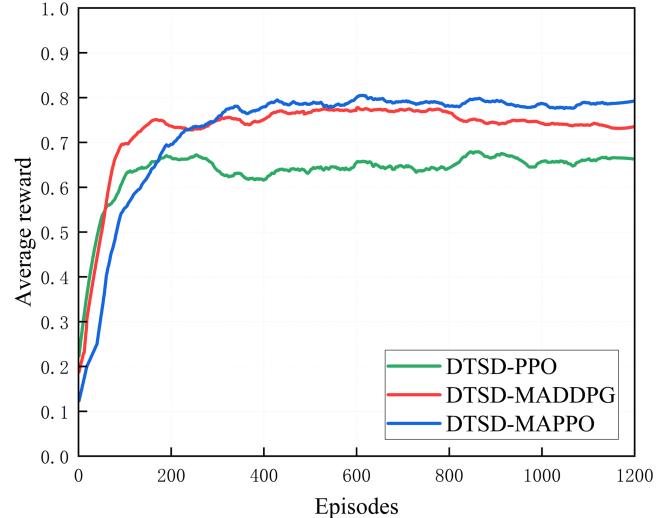


Fig. 5. Average reward in training.

that of the MAPPO-based algorithm. This is due to the fact that in a multi-agent environment, the policy update of each agent receives the influence of the policies of other agents, and this non-stationarity causes the global policy optimization to become difficult and the convergence speed to be slow. Since the agents in a multi-agent reinforcement learning environment collaborate and share information and experience with each other, the reward value of the multi-agent based reinforcement learning algorithm can eventually converge to a higher and more stable value, i.e., a better policy is found.

The MADDPG algorithm, which is also based on the multi-agent environment, adopts the centralized training-distributed

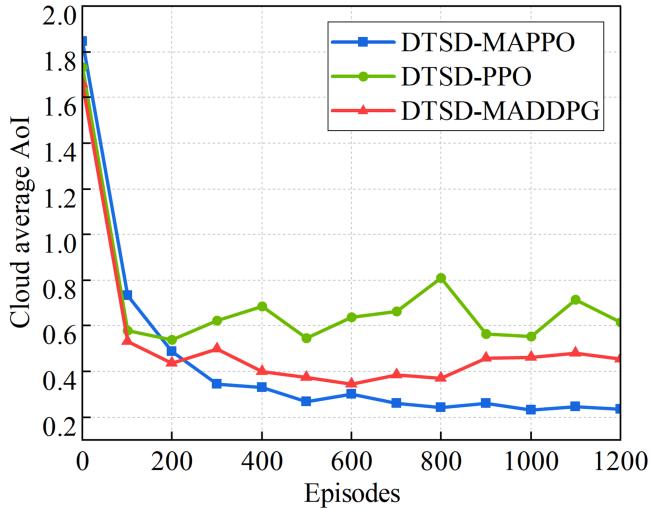


Fig. 6. Cloud average AoI in training.

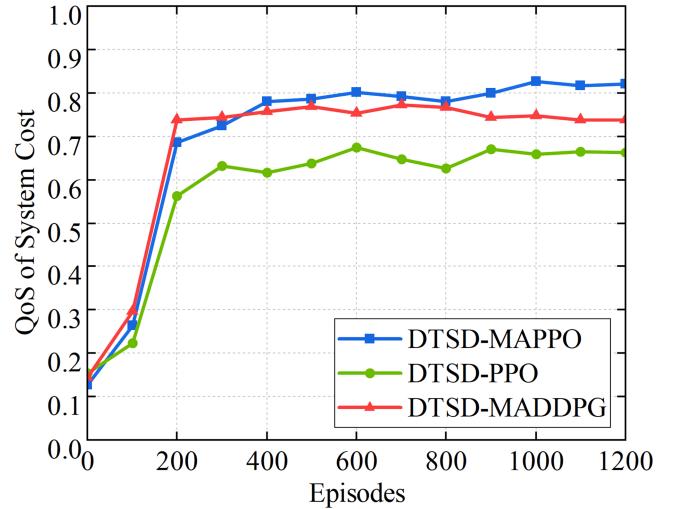


Fig. 7. QoS of system cost in training.

execution framework, but the final convergence result is not as good as that of the MAPPO algorithm. This is due to the fact that MADDPG has relatively poor exploration ability based on deterministic policies and is prone to fall into local optimality. MADDPG can easily lead to overestimation of the Q function, which leads to fluctuations in training. Therefore, compared with MADDPG, MAPPO can receive higher rewards, which fully proves the superiority of the DTSD-MAPPO scheme proposed in this paper.

C. Performance evaluation

Fig. 6 demonstrates the variation of the cloud average AoI during the training process for each DRL-based scheme. As can be seen from the figure, with the increase of training rounds, the cloud average AoI all shows a decreasing trend. However, eventually, the DTSD-MAPPO scheme proposed in this paper is able to reduce the cloud average AoI to a lower value. When training stabilizes, the cloud average AoI in DTSD-MAPPO scheme is about 38.4% less than that in DTSD-MADDPG, and is about 59.8% less than that in DTSD-PPO. Fig. 7 shows the changes of the system cost during the training process, and it is easy to see from the graph that the system cost also decreases with the increase of training iterations, and our scheme can reduce the system cost to the greatest extent, compared to the other two schemes.

Different vehicle sensing policies affect the size of the data volume uploaded to the ES during DT synchronization, and the deployment of VDT consumes computational and storage resources of the ES. Fig. 8 illustrates the variation of system QoS metrics for each scheme with different restrictions on ES resources. As can be seen from the graph, the first two options have some randomness, making it difficult to identify the impact of ES resource changes on QoS. The adequacy of heterogeneous resources for ESs affects system QoS, especially the three DRL-based schemes. With the continuous expansion of ES resources, the system QoS also increases. And no matter how the ES resources change, the DTSD-MAPPO scheme proposed

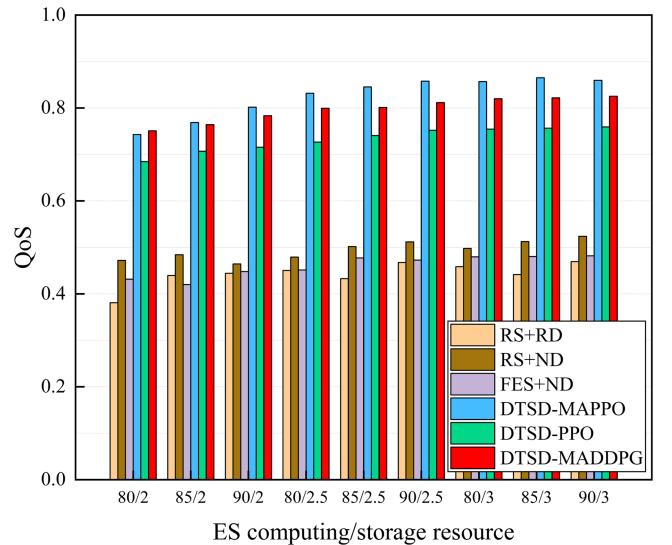


Fig. 8. Impact of ES resource.

in this paper can always make the system QoS metrics better than other schemes.

In order to prove the effectiveness of the proposed scheme in this paper under different traffic flows, we design multiple sets of experiments to randomly retain a portion of the vehicles by region, and record the QoS of different schemes under different traffic flows as shown in Fig. 9. In the case of low traffic flow, each scheme is able to make the system QoS higher, which is due to the fact that the resources of ES are relatively free in the case of fewer vehicles and the vehicle sensing task is not too burdensome. Meanwhile, regardless of traffic congestion, DTSD-MAPPO always maintains a higher system QoS than other schemes.

To further demonstrate the superiority of the DTSD-MAPPO scheme proposed in this paper, vehicle trajectory data during the 11:30-12:00 time period is used for testing, and the current cloud average AoI is recorded every three minutes interval. Fig. 10 demonstrates the comparison between the DTSD-MAPPO scheme proposed in this paper and other schemes in

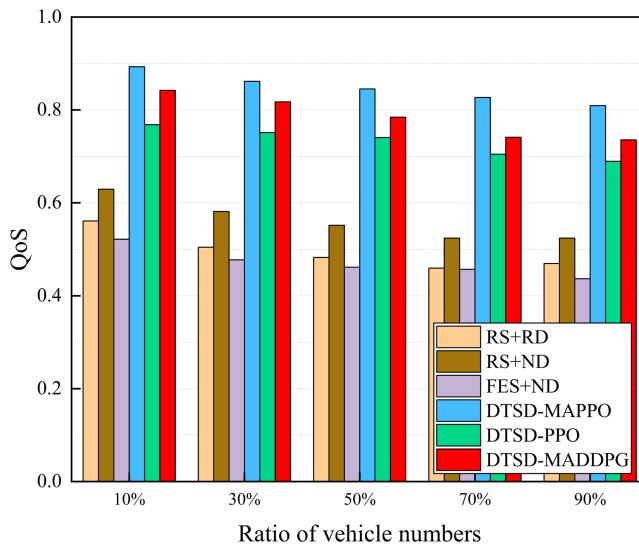


Fig. 9. Impact of traffic flow.

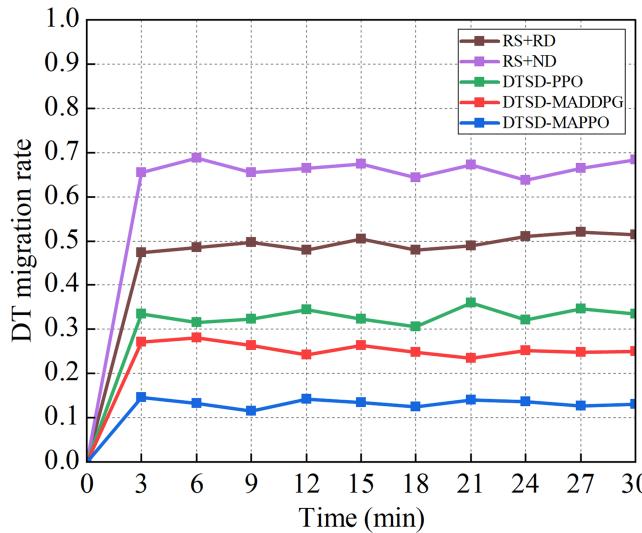


Fig. 10. DT migration rate comparison.

terms of the DT migration rate metrics. We define the DT migration rate as the proportion of ICVs whose VDT deployed in a different location between the current time slot and the previous time slot. As shown in the figure, the RS+ND scheme leads to highest DT migration rate, which is due to the fact that ND scheme requires the VDT to be deployed on the ES closest to the corresponding vehicle, so the DT migration process is constantly triggered as the vehicle moves. The DT migration rate is minimal under the DTSD-MAPPO scheme, which can lower the system cost of DT migration.

The above simulation results show that DTSD-MAPPO can well balance the DT synchronization performance requirements and system cost and maximize the system QoS, indicating the effectiveness and feasibility of the proposed scheme in this paper.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a framework of DT-assisted cloud-edge collaboration IoVs for intelligent transportation. Firstly, we considered the impact of vehicular sensing on the accuracy of DT synchronization and data redundancy, and noted that the real-time performance of DT is affected by the VDTs deployment. Secondly, we constructed a system QoS model in five dimensions, and proposed an optimization problem to maximize system QoS by jointly optimizing vehicular sensing and VDT deployment. Since the problem is NP-hard, we proposed a MADRL-based algorithm called DTSD-MAPPO to find the optimal solution. Through numerous simulations, we demonstrate the effectiveness and excellence of the DTSD-MAPPO scheme compared to others, showing that the proposed scheme is highly practical.

For future work, a promising direction would be to design a DT backup deployment scheme at the network edge, which can avoid the risk of a single point of failure of DTs, realize mutual validation of DTs, and enhance communication efficiency between distant vehicles with social relationships. Additionally, how to optimize DT deployment through efficient data management and data interaction process simplification will also be the focus of our future research.

REFERENCES

- [1] B. Ji et al., "Survey on the internet of vehicles: Network architectures and applications," *IEEE Commun. Standards Mag.*, vol. 4, no. 1, pp. 34–41, Mar. 2020.
- [2] S.-h. An, B.-H. Lee, and D.-R. Shin, "A survey of intelligent transportation systems," in *Proc. 3rd Int. Conf. Computat. Intell., Commun. Syst. Netw.*, Bali, Indonesia, 2011, pp. 332–337.
- [3] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 383–398, Jan. 2019.
- [4] B. R. Barricelli, E. Casiraghi, and D. Fogli, "A survey on digital twin: Definitions, characteristics, applications, and design implications," *IEEE Access*, vol. 7, pp. 167653–167671, 2019.
- [5] S. Mihai et al., "Digital twins: A survey on enabling technologies, challenges, trends and future prospects," *IEEE Commun. Surv. Tut.*, vol. 24, no. 4, pp. 2255–2291, Fourthquarter 2022.
- [6] M. Vaezi et al., "Digital twins from a networking perspective," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 23525–23544, Dec. 2022.
- [7] C.-X. Wang et al., "On the road to 6G: Visions, requirements, key technologies, and testbeds," *IEEE Commun. Surv. Tut.*, vol. 25, no. 2, pp. 905–974, Secondquarter 2023.
- [8] B. Gao, Z. Zhou, F. Liu, and F. Xu, "Winning at the starting line: Joint network selection and service placement for mobile edge computing," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Paris, France, 2019, pp. 1459–1467.
- [9] Q. Fan and N. Ansari, "On cost aware cloudlet placement for mobile edge computing," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 4, pp. 926–937, Jul. 2019.
- [10] Z. Chen, H. Li, K. Ota, and M. Dong, "Deep reinforcement learning for AoI aware VNF placement in multiple source systems," in *Proc. IEEE Glob. Commun. Conf.*, Rio de Janeiro, Brazil, 2022, pp. 2873–2878.
- [11] Y. Hui, Y. Qiu, Z. Su, Z. Yin, T. H. Luan, and K. Aldubaikhy, "Digital twins for intelligent space-air-ground integrated vehicular network: Challenges and solutions," *IEEE Internet Things Mag.*, vol. 6, no. 3, pp. 70–76, Sep. 2023.
- [12] K. Zhang, J. Cao, S. Maharjan, and Y. Zhang, "Digital twin empowered content caching in social-aware vehicular edge networks," *IEEE Trans. Comput. Soc. Syst.*, vol. 9, no. 1, pp. 239–251, Feb. 2022.
- [13] Y. Ge, Y. Wang, R. Yu, Q. Han, and Y. Chen, "Demo: Research on test method of autonomous driving based on digital twin," in *Proc. IEEE Veh. Netw. Conf.*, Los Angeles, CA, USA, 2019, pp. 1–2.

- [14] Z. Wang, K. Han, and P. Tiwari, "Digital twin-assisted cooperative driving at non-signalized intersections," *IEEE Trans. Intell. Veh.*, vol. 7, no. 2, pp. 198–209, Jun. 2022.
- [15] B. Fan, Y. Wu, Z. He, Y. Chen, T. Q. S. Quek, and C. -Z. Xu, "Digital twin empowered mobile edge computing for intelligent vehicular lane-changing," *IEEE Netw.*, vol. 35, no. 6, pp. 194–201, Nov./Dec. 2021.
- [16] Z. Yin, N. Cheng, T. H. Luan, Y. Song, and W. Wang, "DT-assisted multi-point symbiotic security in space-air-ground integrated networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 5721–5734, 2023.
- [17] Z. Yin, N. Cheng, T. H. Luan, and P. Wang, "Physical layer security in cybertwin-enabled integrated satellite-terrestrial vehicle networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4561–4572, May 2022.
- [18] L. Xiao, T. Chen, C. Xie, H. Dai, and H. V. Poor, "Mobile crowdsensing games in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1535–1545, Feb. 2018.
- [19] Y. Zhao and C. H. Liu, "Social-aware incentive mechanism for vehicular crowdsensing by deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2314–2325, Apr. 2021.
- [20] L. Liu, L. Wang, Z. Lu, Y. Liu, W. Jing, and X. Wen, "Cost-and-quality aware data collection for edge-assisted vehicular crowdsensing," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 5371–5386, May 2022.
- [21] C. Liu, L. Wang, X. Wen, L. Liu, W. Zheng, and Z. Lu, "Efficient data collection scheme based on information entropy for vehicular crowdsensing," in *Proc. IEEE Int. Conf. Commun. Workshops*, Seoul, Republic of Korea, 2022, pp. 1–6.
- [22] O. Chukhno, N. Chukhno, G. Araniti, C. Campolo, A. Iera, and A. Molinaro, "Placement of Social digital twins at the edge for beyond 5G IoT networks," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 23927–23940, Dec. 2022.
- [23] Y. Lu, S. Maharjan, and Y. Zhang, "Adaptive edge association for wireless digital twin networks in 6G," *IEEE Internet Things J.*, vol. 8, no. 22, pp. 16219–16230, Nov. 2021.
- [24] M. Vaezi, K. Noroozi, T. D. Todd, D. Zhao, and G. Karakostas, "Digital twin placement for minimum application request delay with data age targets," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11547–11557, Jul. 2023.
- [25] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?," in *Proc. Int. Conf. Comput. Commun.*, Orlando, FL, USA, 2012, pp. 2731–2735.
- [26] Y. Cheng, L. Huang, and X. Wang, "Authentic boundary proximal policy optimization," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 9428–9438, Sep. 2022.
- [27] H. Kang, X. Chang, J. Mišć, V. B. Mišć, J. Fan, and Y. Liu, "Cooperative UAV resource allocation and task offloading in hierarchical aerial computing systems: A MAPPO-based approach," *IEEE Internet Things J.*, vol. 10, no. 12, pp. 10497–10509, Jun. 2023.
- [28] S. Uppoor, O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas, "Generation and analysis of a large-scale urban vehicular mobility dataset," *IEEE Trans. Mobile Comput.*, vol. 13, no. 5, pp. 1061–1075, May 2014.



Lun Tang received the Ph.D. degree in communication and information system from Chongqing University, Chongqing, China, in 2010. He is currently a Professor with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing. His current research interests include digital twin networks, 5G/6G, industrial Internet of Things, and Internet of Vehicles.



Zhangchao Cheng received the B.S. degree from Hubei Normal University, Huangshi, China, in 2020. He is currently working toward the M.S. degree with the Key Laboratory of Mobile Communication Technology, Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include Internet of Vehicles, digital twin networks and deep reinforcement learning.



Jun Dai received the B.S. degree from Anhui Agricultural University, Hefei, China, in 2021. He is currently working toward the M.S. degree with the Key Laboratory of Mobile Communication Technology Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include digital twin networks, autonomous driving route plan, reinforcement learning and federated learning.



Hongpeng Zhang received the B.S. degree from Hefei University, Hefei, China, in 2019. He is currently working toward the M.S. degree with the Key Laboratory of Mobile Communication Technology, Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include digital twin networks, 5G radio access network slicing, resource allocation and age of information.



Qianbin Chen (Senior Member, IEEE) received the Ph.D. degree in communication and information system from the University of Electronic Science and Technology of China, Chengdu, China, in 2002. He is currently a Professor with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China, and the Director of the Chongqing Key Laboratory of Mobile Communication Technology. He has authored or coauthored more than 100 papers in journals and peer-reviewed conference proceedings, and has coauthored seven books. He holds 47 granted national patents.