Model-Assisted Learning for Adaptive Cooperative Perception of Connected Autonomous Vehicles

Kaige Qu, *Member, IEEE*, Weihua Zhuang, *Fellow, IEEE*, Qiang Ye, *Senior Member, IEEE*, Wen Wu, *Senior Member, IEEE*, and Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—Cooperative perception (CP) is a key technology to facilitate consistent and accurate situational awareness for connected and autonomous vehicles (CAVs). To tackle the network resource inefficiency issue in traditional broadcast-based CP, unicast-based CP has been proposed to associate CAV pairs for cooperative perception via vehicle-to-vehicle transmission. In this paper, we investigate unicast-based CP among CAV pairs. With the consideration of dynamic perception workloads and channel conditions due to vehicle mobility and dynamic radio resource availability, we propose an adaptive cooperative perception scheme for CAV pairs in a mixed-traffic autonomous driving scenario with both CAVs and human-driven vehicles. We aim to determine when to switch between cooperative perception and stand-alone perception for each CAV pair, and allocate communication and computing resources to cooperative CAV pairs for maximizing the computing efficiency gain under perception task delay requirements. A model-assisted multi-agent reinforcement learning (MARL) solution is developed, which integrates MARL for an adaptive CAV cooperation decision and an optimization model for communication and computing resource allocation. Simulation results demonstrate the effectiveness of the proposed scheme in achieving high computing efficiency gain, as compared with benchmark schemes.**

***Index Terms*—Connected and autonomous vehicles (CAVs), cooperative perception, data fusion, autonomous driving, multi-agent reinforcement learning, model-assisted learning.**

I. Introduction

The advances in sensing, artificial intelligence (AI), and vehicles-to-everything (V2X) communication technologies have paved the way for autonomous driving, leading to a potential paradigm shift in future transportation systems towards improved road safety and traffic efficiency [1]–[3]. Reliable and real-time environment perception is a key component in autonomous driving that facilitates the connected and autonomous vehicles (CAVs) to accurately and continuously perceive the surrounding objects, such as traffic participants, by using on-board cameras, light detection and ranging (LiDAR) sensors, and radar sensors [4], [5]. To enhance the perception reliability in terms of both coverage and accuracy, *cooperative perception* (CP) has been proposed to enable the sensory information sharing among CAVs by leveraging V2X communication, as a supplement to the *stand-alone perception* (SP) by individual CAVs based on their own viewpoints [6]–[12]. In case of unreliable network connectivity or network congestion due to limited network resources, CAVs can switch back to the default SP mode [13].

According to the type of shared sensory information, there are three CP levels, including raw level, feature level, and decision level. In the raw-level CP, complete [7], [14] or partial raw data [13], [15] are shared among CAVs, which preserves the most fine-grained environmental information and leads to the highest perception performance gain at the cost of huge communication overhead due to the large data volume. The decision-level CP integrates lightweight perception results of individual CAVs, which is communication-efficient but with limited perception performance gain [10]. The feature-level CP, which has gained significant attention in the computer vision field, can balance between communication overhead and perception performance gain [16], [17]. Research studies on feature-level CP have focused on the design of AI-based fusion schemes, but the underlying communication scheme is usually simple, e.g., via broadcast-based vehicle-to-vehicle (V2V) communication. Specifically, each CAV fuses all the received feature data broadcast from neighboring CAVs with its own data and processes the fused data for inference [16], [17]. Although the feature data are compressed from the raw data, the data size is still large, e.g., in the scale of Mbits. Hence, the broadcast-based CP schemes are communication inefficient especially in dense-traffic scenarios and even not applicable when the available transmission resources are limited. Moreover, due to individual computation at each CAV, the overall computation demand is intensive, which is roughly proportional to the number of CAVs and the data volume processed at each CAV [13]. The communication and computation in such broadcast-based CP schemes lead to large network resource consumption for satisfying the stringent delay requirement of real-time perception tasks.

Recently, there are some studies on the deployment of CP schemes in a practical network environment by considering the limited V2X communication bandwidth and on-board computing resources [7]–[10]. As nearby CAVs collect sensing data of common objects in a shared environment from diverse viewpoints, adding sensing data from more CAVs for data fusion potentially improves the perception performance with a diminishing marginal gain, at the cost of almost linearly increasing network resources. For resource efficiency, unicast-based CP schemes have been studied in [7]–[9], which deal

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

Kaige Qu, Weihua Zhuang, and Xuemin (Sherman) Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (emails: {k2qu, wzhuang, sshen}@uwaterloo.ca).

Qiang Ye is with University of Calgary, Calgary, AB T2N 1N4, Canada (email: qiang.ye@ucalgary.ca).

Wen Wu is with Peng Cheng Laboratory, Shenzhen, Guangdong, China, 518055 (email: wuw02@pcl.ac.cn). He contributed to this study while working as a postdoctoral fellow at the University of Waterloo, Canada.

with the association of CAV pairs that perform the cooperative perception via unicast-based V2V communication. Two CAVs with complementary or enhancing sensory information usually provide higher perception performance gain through cooperation and tend to be associated, due to more even spatial distribution or higher intensity of fused sensing data. In comparison with the broadcast-based counterparts, unicast-based CP schemes significantly improve the network resource efficiency without a remarkable compromise on the perception performance through proper association of CAV pairs.

In this work, we investigate unicast-based feature-level CP among CAVs. Different from the existing works on CAV pair association to improve the perception performance gain, we investigate how to support CP for predetermined CAV pairs in a complex and dynamic network environment with high resource efficiency. Specifically, we consider a practical mixed-traffic autonomous driving scenario where a cluster of CAVs and human-driven vehicles (HDVs) traverse a road segment with intermittent road-side-unit (RSU) coverage due to the high RSU deployment cost. Each CAV pair works in either the SP mode by default or the unicast-based feature-level CP mode by selection. Considering the radio resource sharing among vehicles, the radio resource availability for CAVs dynamically changes over time. Due to vehicle mobility, the perception workloads and channel conditions for CAVs are dynamic in different perception task periods. In such a network scenario, it is challenging to constantly support all CAV pairs to work in the CP mode with delay satisfaction.

To accommodate the network dynamics, we propose an adaptive cooperative perception scheme, which facilitates a dynamic selection of CAV pairs for cooperative perception. The selected CAV pairs are referred to as *cooperative CAV pairs*, and the non-selected CAV pairs work in the SP mode by default. For each cooperative CAV pair, we dynamically allocate a fraction of available radio resources to support the feature data transmission, and adjust the CPU frequency at the CAVs on demand, to balance between the transmission and computation delays under the network dynamics, while satisfying a perception delay requirement. For the joint adaptive CAV cooperation and resource allocation, there is a trade-off between a total computing efficiency gain and a total cost for dynamically switching between the SP and CP modes for the CAV pairs. Specifically, for a CAV pair, the total *computing demand* is significantly reduced in the CP mode, by performing the data fusion and inference at one CAV and allowing the computation result sharing within the CAV pair. However, due to the on-demand CPU frequency allocation, the *CPU frequency* in the CP mode can be occasionally higher than that in the SP mode. As the CAVs work in the SP mode by default, we characterize the computing efficiency gain of a CAV pair as the reduced amount of computing energy consumption in comparison with that in the SP mode, which depends on both computing demand and CPU frequency. We focus on increasing the total computing efficiency gain while reducing the total switching cost, via proper selection of cooperative CAV pairs and optimal resource allocation. The main contributions of this paper are summarized as follows.

- We propose an adaptive cooperative perception scheme for CAV pairs in a moving mixed-traffic vehicle cluster, which allows each CAV pair to dynamically switch between the SP and CP modes over different perception task periods, to adapt to the network dynamics;
- We formulate a joint adaptive CAV cooperation and resource allocation problem, which can be decoupled to an adaptive CAV cooperation subproblem in the long run and a series of instantaneous resource allocation subproblems in each perception task period, to maximize the total computing efficiency gain with minimum switching cost, while satisfying the perception delay requirement;
- We propose a model-assisted multi-agent reinforcement learning (MARL) solution, where MARL is used to learn the adaptive cooperation decisions among CAV pairs, and a model-based solution is used for resource allocation given each cooperation decision.

The remainder of this paper is organized as follows. The system model is presented in Section II, with a performance analysis included in Section III. The joint adaptive CAV cooperation and resource allocation problem is formulated in Section IV, with a model-assisted MARL solution presented in Section V. Simulation results are provided in Section VI, and conclusions are drawn in Section VII.

II. System Model

A. Mixed-Traffic Autonomous Driving Scenario

We consider a vehicle cluster including M HDVs and K CAV pairs in set $\mathcal{K}$, moving on a multi-lane unidirectional road under a consistent base station (BS) coverage and an intermittent RSU coverage. The CAV pairs are predetermined by using existing CAV pair association algorithms [8], [9]. a cluster head is selected among the vehicles based on existing vehicle clustering algorithms, to coordinate the communication, computing, and sensing in the vehicle cluster [4]. Both BS and RSU provide the edge computing capability, facilitated by co-located edge servers. Fig. 1 illustrates three snapshots for a vehicle cluster moving through an RSU's coverage area. Initially, all the vehicles in the cluster have no access to the RSU but the leading vehicle is about to move into the RSU coverage. Then, the vehicles gradually move into and later leave the RSU coverage. Consider a time-slotted system, in which the time slots are indexed by integer n.

Each CAV initiates a perception task in each time slot to identify the surrounding objects, whose results are essential to supporting autonomous driving applications, such as path planning and maneuver control. Each CAV pair $k \in \mathcal{K}$ consists of one transmitter CAV and one receiver CAV, and both CAVs share a similar view but from different angles. Each CAV pair works in the SP mode by default and in the CP mode by selection. Let $\boldsymbol{x}(n) = \{x_k(n), \forall k \in \mathcal{K}\}$ be a binary perception mode selection decision vector for all CAV pairs (also referred to as cooperation decisions for brevity) at time slot n, with $x_k(n) = 1$ indicating the CP mode and $x_k(n) = 0$ indicating the SP mode for CAV pair k. A CAV pair in the CP mode is referred to as a cooperative CAV pair. Let $\mathcal{K}_C(n)$ denote a set of cooperative CAV pairs at time slot n, with $\mathcal{K}_C(n) \subset \mathcal{K}$.

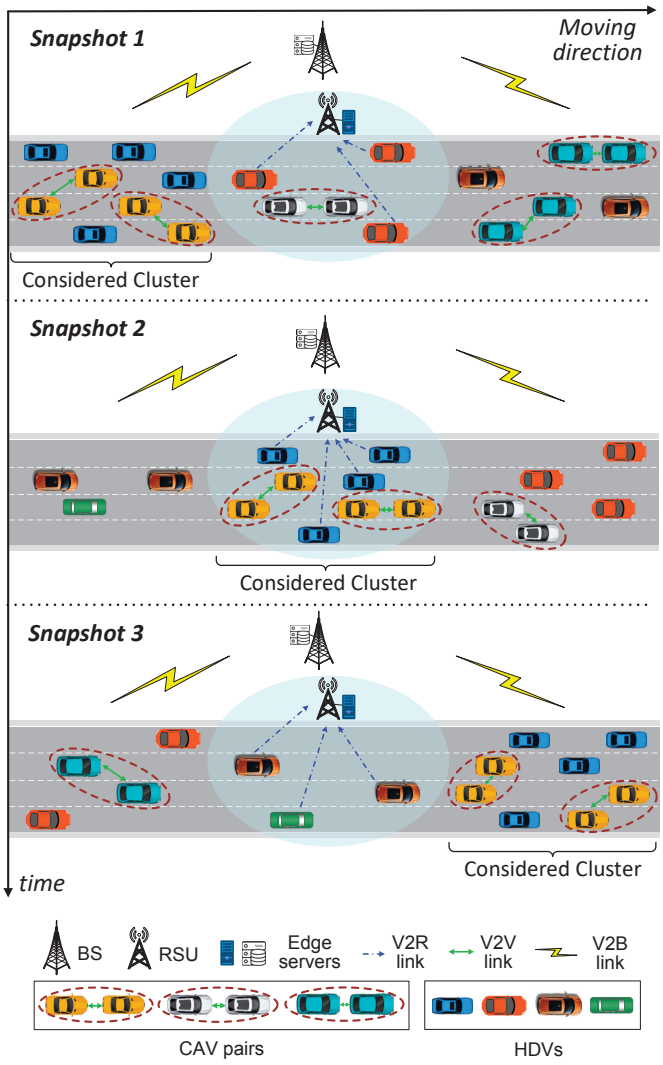


Fig. 1: A mixed-traffic autonomous driving scenario.

Each HDV occasionally requests an infotainment service, e.g., mobile virtual reality, which is throughput sensitive and computation intensive. For energy and delay efficiency, the computation tasks at an HDV can be offloaded to a more powerful edge server either at a BS or at an RSU, and the data transmission is supported by either vehicle-to-BS (V2B) or vehicle-to-RSU (V2R) communications [18].

B. Perception Task Model

For environment perception, lightweight data pre-processing algorithms are used to slice the raw sensing data into *object partitions* each containing one object of interest and *background partitions* that contain only background information [19]. An object tracking algorithm associates the object partitions with existing objects in a maintained object tracking list, by comparing the identified and predicted object locations [20]. Only the new objects and the objects with reduced tracking accuracy are further processed by a deep neural network (DNN) for classification [21]. For CAV pair k, let $W_k(n)$, $W_k^{\mathsf{T}}(n)$, and $W_k^{\mathsf{R}}(n)$ denote the number of objects that require DNN model processing in the overlapping sensing

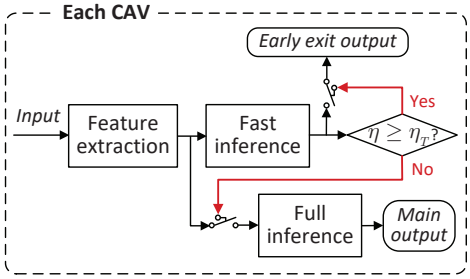


Fig. 2: Object classification by using a default DNN model.

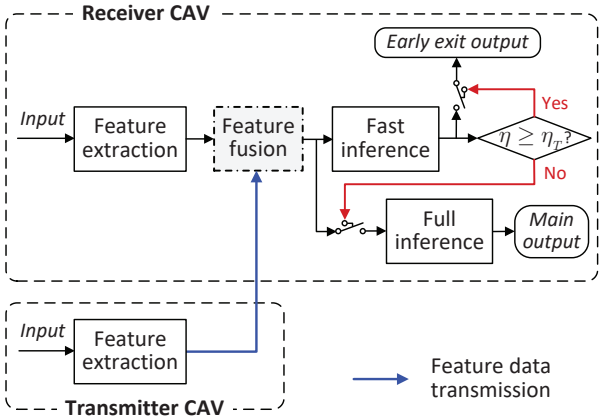


Fig. 3: Object classification by using a feature-fusion DNN model.

range of both CAVs, in the non-overlapping sensing range of the transmitter CAV, and in the non-overlapping sensing range of the receiver CAV, respectively, at time slot n. We refer to the objects in the overlapping and non-overlapping sensing ranges as shared and individual objects respectively. Then, $W_k(n)$ is also referred to as shared workload, and $W_k^{\mathsf{T}}(n)$, $W_k^{\mathsf{R}}(n)$ are referred to as individual workloads, for CAV pair k.

At time slot n, the object classification tasks for all the objects of CAV pair k should be completed within a time duration of Δ, which is typically smaller than the time slot length, with the consideration of 1) the raw data pre-processing and object tracking procedures before the generation of object classification tasks, and 2) the response time for autonomous driving applications based on the object classification results.

1) DNN models: To support the object classification at each CAV pair, we consider a *default DNN model* deployed at both the transmitter and receiver CAVs and a *feature-fusion DNN model* partitioned between them, both employing an early-exit DNN architecture, as illustrated in Fig. 2 and Fig. 3 respectively [22]–[25]. As the objects can be processed independently, we consider that both DNN models operate on a per-object basis [19], [26]. Specifically, in the default DNN model, a feature extraction module, mainly composed of convolution (CONV) layers, first generates compressed feature data based on an input of object sensing data. Then, the feature data are further processed by a fast inference module, composed of both CONV and fully-connected (FC) layers, to generate a DNN inference result, referred to as a fast inference result. Letting Z denote the total number of object classes, a DNN inference result is a Z-dimension estimated class

probability vector, where the classification performance is measured by confidence level defined as one minus normalized entropy [22], [23]. A higher confidence level indicates a less uncertain estimation and implies a higher accuracy [23]. Let η denote the confidence level of a fast inference result. If η reaches a predetermined threshold, η_T, an object classification result is obtained at an early exit output. Otherwise, a full inference module, composed of deeper `CONV` and `FC` layers, is triggered to re-process the feature data and generate another DNN inference result, referred to as a full inference result, at a main output. Let $\rho \in (0,1)$ be the early exit probability for the default DNN model, representing the probability that an object classification result is obtained at the early exit output.

For a shared object of a CAV pair, the transmitter and receiver CAVs have object sensing data from different viewpoints, implying a potential confidence level gain from data fusion. Such an object can be processed by using the feature-fusion DNN model. Specifically, both CAVs process their own object sensing data and extract features based on a feature extraction module. Then, the feature data of the transmitter CAV are transmitted via V2V communication to the receiver CAV, where the feature data from both CAVs are fused and processed by the fast inference and selective full inference modules. The object classification result is obtained at either an early exit output with probability $\tilde{\rho} \in (0,1)$, or a main output with probability $1-\tilde{\rho}$, at the receiver CAV, which is then sent back to the transmitter CAV. Typically, we have $\tilde{\rho} > \rho$, as more fast inference results can satisfy the confidence level requirement due to the confidence level gain from data fusion.

For the DNN models, let δ_1, δ_2, δ_3 and δ_4 be the computing demand (in CPU cycles) for feature extraction, feature fusion, fast inference, and full inference. Typically, we have $\delta_2 \ll \min\{\delta_3, \delta_4\}$, as the feature fusion module can be implemented by simple operations such as concatenation, maxout, and average operations [16], [22] or lightweight attention schemes [17]. The average computing demand for processing one object by the default and feature-fusion DNN models, denoted by δ and $\tilde{\delta}$ respectively, are given by

$$\delta = \delta_1 + \delta_3 + (1-\rho)\delta_4 \tag{1}$$

$$\tilde{\delta} = 2\delta_1 + \delta_2 + \delta_3 + (1-\tilde{\rho})\delta_4. \tag{2}$$

2) Cooperative Perception Mode: For a CAV pair in the CP mode, the shared objects are collaboratively processed by using the feature-fusion DNN model, while the individual objects are independently processed at each CAV by using the default DNN model. Consider two CPU cores at each CAV, for processing the shared and individual objects separately using different DNN models. For each CPU core, the dynamic voltage and frequency scaling (DVFS) technique is used to allow on-demand CPU frequency scaling, to support the dynamic perception workloads [7].

3) Stand-Alone Perception Mode: In the default SP mode, both CAVs in a CAV pair independently perform the object classification tasks. Both the shared and individual objects are processed by using the default DNN model. For ease of analysis, we assume that the shared and individual objects are processed at separate CPU cores, as in the CP mode.

TABLE I: CPU FREQUENCY CONFIGURATION FOR A CAV PAIR

CPU Core	Core 1		Core 2	
Object Type	Shared Objects		Individual Objects	
Perception Mode	CP	SP	CP	SP
CPU Frequency at Transmitter CAV	$f_k(n)$	$f_k^{\mathsf{D}}(n)$	$f_k^{\mathsf{T}}(n)$	$f_k^{\mathsf{T}}(n)$
CPU Frequency at Receiver CAV	$f_k(n)$	$f_k^{\mathsf{D}}(n)$	$f_k^{\mathsf{R}}(n)$	$f_k^{\mathsf{R}}(n)$

C. Computing Model

For CAV pair k, let $\delta_k^C(n)$, $\delta_k^S(n)$, $\delta_k^T(n)$, and $\delta_k^R(n)$ be the average total computing demand for processing the shared objects in the CP mode, the shared objects in the SP mode, the transmitter CAV's individual objects, and the receiver CAV's individual objects, at time slot n, given by

$$\delta_k^C(n) = \tilde{\delta} W_k(n), \quad \delta_k^S(n) = 2\delta W_k(n) \tag{3}$$

$$\delta_k^T(n) = \delta W_k^{\mathsf{T}}(n), \quad \delta_k^R(n) = \delta W_k^{\mathsf{R}}(n). \tag{4}$$

There is a positive total computing demand reduction for CAV pair k through cooperation which increases proportionally to shared workload $W_k(n)$, given by $(2\delta - \tilde{\delta})W_k(n) = [\delta_3 + (1+\tilde{\rho}-2\rho)\,\delta_4 - \delta_2]\,W_k(n) > 0$.

For CAV pair k, let $f_k(n)$, $f_k^{\mathsf{T}}(n)$, and $f_k^{\mathsf{R}}(n)$ denote the CPU frequencies (in Hz or cycle/s) for processing the shared objects at both CAVs, the individual objects at transmitter CAV, and the individual objects at receiver CAV, respectively, at time slot n. We have $f_k(n) = f_k^{\mathsf{D}}(n)$ if $x_n(k) = 0$, where $f_k^{\mathsf{D}}(n)$ is the CPU frequency for processing the shared objects at both CAVs in the default SP mode. Table I summarizes the relationships among CPU cores, object types, perception modes, and CPU frequencies for a CAV pair. As the shared and individual objects can be processed in parallel at the separate CPU cores, we have $f_k^{\mathsf{D}}(n) = \frac{\delta W_k(n)}{\Delta}$, $f_k^{\mathsf{T}}(n) = \frac{\delta W_k^{\mathsf{T}}(n)}{\Delta}$, and $f_k^{\mathsf{R}}(n) = \frac{\delta W_k^{\mathsf{R}}(n)}{\Delta}$ to ensure that the default DNN model processing can be finished within delay bound Δ for the shared objects in the SP mode, and for the individual objects in either SP or CP mode, without CPU frequency over-provisioning. All CPU frequencies are upper limited by a maximum CPU frequency, f_{M}, supported by DVFS, leading to an upper limit, $W_{\mathsf{M}} = \frac{f_{\mathsf{M}}\Delta}{\delta}$, for $W_k(n)$, $W_k^{\mathsf{T}}(n)$, and $W_k^{\mathsf{R}}(n)$.

The total computing energy consumption by all CPU cores of CAV pair k at time slot n, denoted by $e_k(n)$, is given by

$$\begin{aligned} e_k(n) = \kappa \big[& f_k(n)^2 \delta_k^C(n) x_k(n) + f_k^{\mathsf{D}}(n)^2 \delta_k^S(n)\,(1-x_k(n)) \\ & + f_k^{\mathsf{T}}(n)^2 \delta_k^T(n) + f_k^{\mathsf{R}}(n)^2 \delta_k^R(n) \big] \end{aligned} \tag{5}$$

where κ is the energy efficiency coefficient of a CPU core [23]. From (5), we see that only the portion of computing energy for processing the shared objects depends on cooperation decision $x_k(n)$. The computing energy is a comprehensive metric that integrates both CPU frequency and computing demand. We characterize the computing efficiency gain of CAV pair k at time slot n, denoted by $G_k(n)$, as the reduced amount of computing energy in comparison with that in the default SP mode. We have $G_k(n) \equiv 0$ for $k \notin \mathcal{K}_C(n)$. For cooperative

CAV pair $k \in \mathcal{K}_C(n)$, we have

$$G_k(n) = \kappa f_k^{\mathsf{D}}(n)^2 \delta_k^S(n) - \kappa f_k(n)^2 \delta_k^C(n) \\ = 2\kappa\delta f_k^{\mathsf{D}}(n)^2 W_k(n) - \kappa\tilde{\delta} f_k(n)^2 W_k(n),\ \forall k \in \mathcal{K}_C(n) \quad (6)$$

as a decreasing function of $f_k(n) > 0$. Here, $G_k(n)$ is independent of the individual workloads, as only shared objects are processed differently between the SP and CP modes.

D. Communication Model

Consider a radio resource pool with total bandwidth B for V2X sidelink communication, which is shared between the V2R transmission from HDVs and the V2V transmission for cooperative CAV pairs, and is non-overlapping with that for V2B communication. The radio resource sharing between CAVs and HDVs occur only in the RSU coverage. Consider a transmission priority for HDVs, as the CAVs can work in the SP mode without radio resource usage by default. Orthogonal frequency division multiplexing (OFDM) based transmission schemes are employed for the V2X sidelink communication.

Let $B(n)$ be the time-varying available radio spectrum bandwidth for CAVs, with the consideration of dynamic background radio resource usage by HDVs. Let $\boldsymbol{\beta}(n) = \{\beta_k(n), \forall k \in \mathcal{K}\}$ be a radio resource allocation decision vector for all CAV pairs at time slot n, with $\beta_k(n)$ denoting the fraction of available radio resources allocated to CAV pair k for supporting the feature data transmission at time slot n. The average transmission rate for CAV pair k at time slot n is

$$R_k(n) = \beta_k(n) B(n) \log_2\left(1 + \frac{p_k g_k(n)}{\sigma^2}\right), \quad \forall k \in \mathcal{K} \quad (7)$$

where p_k is the transmit power of CAV pair k, $g_k(n)$ is the channel power gain between both CAVs in CAV pair k at time slot n, and σ^2 represents the received noise power. Due to high vehicle mobility, we consider only the large-scale channel conditions, specifically the path loss, for the CAV pairs.

E. Delay Model

Under the assumption of $\max\{W_k(n), W_k^{\mathsf{T}}(n), W_k^{\mathsf{R}}(n)\} \leq W_{\mathsf{M}}$, the CPU frequencies for processing the shared objects in the SP mode and for processing the individual objects in both SP and CP modes can be feasibly scaled up/down to ensure delay satisfaction. Here, we focus on the delay performance of the shared objects in the CP mode. Let w denote the feature data size in the unit of bit. If CAV pair k works in the CP mode, the average object classification delay for each shared object, denoted by $d_k(n)$, depends on the feature-fusion DNN model. The delay is composed of feature extraction delay $\frac{\delta_1}{f_k(n)}$, feature data transmission delay $\frac{w}{R_k(n)}$, feature fusion delay $\frac{\delta_2}{f_k(n)}$, and the average inference delay, $\frac{\delta_3 + (1-\tilde{\rho})\delta_4}{f_k(n)}$. The delay for sending the classification results is negligible due to the small data size. For delay satisfaction, $d_k(n)$ should not exceed a per-object delay budget, $\frac{\Delta}{W_k(n)}$, given by

$$d_k(n) = \frac{w}{R_k(n)} + \frac{\hat{\delta}}{f_k(n)} \leq \frac{\Delta}{W_k(n)}, \quad \forall k \in \mathcal{K}_C(n) \quad (8)$$

where $\hat{\delta} = \delta_1 + \delta_2 + \delta_3 + (1-\tilde{\rho})\,\delta_4$ is a constant.

F. Generalization

For computing efficiency gain and perception accuracy enhancement, nearby CAVs can be grouped for cooperative perception by using a Y-input feature-fusion DNN model, where Y is a general group size. At a given vehicle density, the content similarity within a group tends to reduce as Y increases, due to a lower average percentage of shared objects. As only the shared objects can be collaboratively processed by using the feature-fusion DNN model, the reduced content similarity may gradually compromise the total computing efficiency gain and the average perception accuracy enhancement as Y increases. Additionally, a larger group size increases the accuracy of shared objects with a diminishing marginal gain. However, there is a higher overall communication cost for supporting the feature data transmission of more transmitter CAVs in a larger group. For simplicity, we consider $Y = 2$ and pair the CAVs based on existing works [7]–[9]. How to select the best group size, Y, and how to optimally group the CAVs given Y remain to be investigated in our future work.

III. 2D Performance Region Analysis

For problem formulation, we analyze the performance of an arbitrary cooperative CAV pair, $k \in \mathcal{K}_C(n)$, at time slot n, under different transmission rates and CPU frequencies for supporting the classification of shared objects. The condition for non-negative computing efficiency gain via cooperation, i.e., $G_k(n) \geq 0$, is a CPU frequency requirement, given by

$$f_k(n) \leq f_k^{\mathsf{P}}(n) = \sqrt{\frac{2\delta}{\tilde{\delta}}} f_k^{\mathsf{D}}(n) = \sqrt{\frac{2\delta^3}{\tilde{\delta}}} \frac{W_k(n)}{\Delta} \quad (9)$$

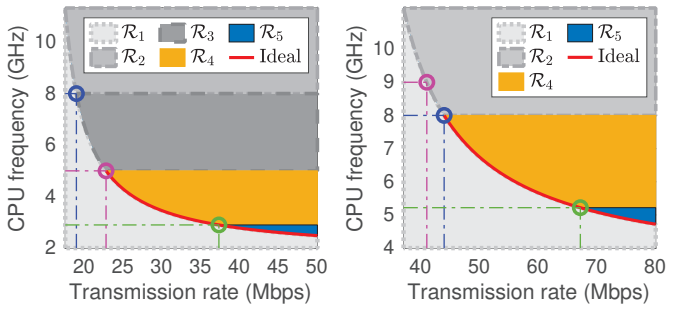
where $f_k^{\mathsf{P}}(n)$ corresponds to zero computing efficiency gain and increases proportionally to shared workload $W_k(n)$. We have $f_k^{\mathsf{P}}(n) > f_k^{\mathsf{D}}(n)$, as $\sqrt{\frac{2\delta}{\tilde{\delta}}} > 1$. Under assumption $W_k(n) \leq W_{\mathsf{M}}$, there are two cases for the relationship among $f_k^{\mathsf{D}}(n)$, $f_k^{\mathsf{P}}(n)$ and f_{M}, depending on shared workload $W_k(n)$:

- *Low shared workload:* For $W_k(n) \leq \sqrt{\frac{\tilde{\delta}}{2\delta}} \frac{f_{\mathsf{M}}\Delta}{\delta} = \sqrt{\frac{\tilde{\delta}}{2\delta}} W_{\mathsf{M}}$, we have $f_k^{\mathsf{D}}(n) < f_k^{\mathsf{P}}(n) \leq f_{\mathsf{M}}$. For a feasible CPU frequency scale-up from $f_k^{\mathsf{D}}(n)$ to f_{M}, computing efficiency gain $G_k(n)$ transits from positive to negative, with a zero value at $f_k(n) = f_k^{\mathsf{P}}(n)$;
- *High shared workload:* For $\sqrt{\frac{\tilde{\delta}}{2\delta}} W_{\mathsf{M}} < W_k(n) \leq W_{\mathsf{M}}$, we have $f_k^{\mathsf{D}}(n) \leq f_{\mathsf{M}} < f_k^{\mathsf{P}}(n)$. As $f_k(n)$ scales up from $f_k^{\mathsf{D}}(n)$ to f_{M}, $G_k(n)$ decreases but remains positive.

Let R_{M}, $R_k^{\mathsf{P}}(n)$, and $R_k^{\mathsf{D}}(n)$ be the minimum transmission rates for delay satisfaction if cooperative CAV pair k operates at CPU frequencies f_{M}, $f_k^{\mathsf{P}}(n)$, and $f_k^{\mathsf{D}}(n)$ respectively for processing the shared objects. The three rate-frequency pairs, $[R_{\mathsf{M}}, f_{\mathsf{M}}]$, $\left[R_k^{\mathsf{P}}(n), f_k^{\mathsf{P}}(n)\right]$, and $\left[R_k^{\mathsf{D}}(n), f_k^{\mathsf{D}}(n)\right]$, all lie on a curve indicating $d_k(n) = \frac{\Delta}{W_k(n)}$. We obtain $R_{\mathsf{M}} = w \Big/ \left(\frac{\Delta}{W_k(n)} - \frac{\hat{\delta}}{f_{\mathsf{M}}}\right)$, $R_k^{\mathsf{P}}(n) = \frac{W_k(n) w \phi}{(\phi-1)\Delta}$ where $\phi = \sqrt{\frac{2\delta^3}{\tilde{\delta}^2\tilde{\delta}}} > 1$ is a constant, and $R_k^{\mathsf{D}}(n) = \frac{W_k(n) w \delta}{[(\tilde{\rho}-\rho)\delta_4 - \delta_2]\Delta}$, all increasing with shared workload $W_k(n)$. For cooperative CAV pair k, there are multiple 2D performance regions with different delay performance, CPU frequency scaling range, and computing

TABLE II: SUMMARY OF DIFFERENT PERFORMANCE REGIONS OF A COOPERATIVE CAV PAIR

Region	Shared Workload	Delay Performance	CPU Frequency Scaling Range	Computing Efficiency Gain
$\mathcal{R}_1$	Low/High	Violation	N/A	N/A
$\mathcal{R}_2$	Low/High	Satisfaction	Infeasible scale-up: $f_k(n) > f_{\mathsf{M}}$	N/A
$\mathcal{R}_3$	Low	Satisfaction	Feasible scale-up: $f_k^{\mathsf{P}}(n) < f_k(n) \le f_{\mathsf{M}}$	Negative
$\mathcal{R}_4$	Low/High	Satisfaction	Feasible scale-up: $f_k^{\mathsf{D}}(n) < f_k(n) \le f_k^{\mathsf{P}}(n)$ at a low workload, $f_k^{\mathsf{D}}(n) < f_k(n) \le f_{\mathsf{M}}$ at a high workload	Non-negative
$\mathcal{R}_5$	Low/High	Satisfaction	Default or scale-down: $f_k(n) \le f_k^{\mathsf{D}}(n)$	Positive



(a) Low workload (b) High workload

Fig. 4: Examples of 2D performance regions for cooperative CAV pair k at a different shared workload, where $[R_{\mathsf{M}}, f_{\mathsf{M}}]$, $[R_k^{\mathsf{P}}(n), f_k^{\mathsf{P}}(n)]$, and $[R_k^{\mathsf{D}}(n), f_k^{\mathsf{D}}(n)]$ are indicated by blue, pink, and green circles.

efficiency gain, for different combinations of $R_k(n)$ and $f_k(n)$, as summarized in Table II. The number of performance regions depends on the shared workload, as illustrated in Fig. 4.

From Fig. 4 and Table II, we obtain some useful principles in the resource allocation for each cooperative CAV pair. First, only the rate-frequency pairs in **Regions** $\mathcal{R}_4$ and $\mathcal{R}_5$ should be selected for both delay satisfaction and non-negative computing efficiency gain at a feasible CPU frequency. Second, a subset of **Region** $\mathcal{R}_4$ and **Region** $\mathcal{R}_5$ rate-frequency pairs that lie on curve $d_k(n) = \frac{\Delta}{W_k(n)}$ are the ideal candidate rate-frequency pairs without transmission rate over-provisioning, as indicated by red curves in Fig. 4. Accordingly, the ideal candidate rate-frequency pairs for each cooperative CAV pair $k \in \mathcal{K}_C(n)$ should satisfy the following constraints,

$$f_k(n) \le \min\left\{f_k^{\mathsf{P}}(n), f_{\mathsf{M}}\right\}, \quad \forall k \in \mathcal{K}_C(n) \tag{10}$$

$$\frac{w}{R_k(n)} + \frac{\hat{\delta}}{f_k(n)} = \frac{\Delta}{W_k(n)}, \quad \forall k \in \mathcal{K}_C(n). \tag{11}$$

IV. PROBLEM FORMULATION

Due to the dynamic radio resource availability, the current available radio resources may be insufficient for supporting all CAV pairs to work in the CP mode with delay satisfaction and non-negative computing efficiency gain. Thus, an adaptive set of cooperative CAV pairs, $\mathcal{K}_C(n) \subset \mathcal{K}$ should be determined for each time slot. Due to environmental changes, the shared workload and the transmitter-receiver distance vary over time for each cooperative CAV pair. As the total computing demand reduction through cooperation increases in proportion to the shared workload, a moderate shared workload increase potentially brings more computing efficiency gain. However, due to a reduced per-object delay budget, a heavier shared workload requires a higher CPU frequency for delay satisfaction under limited radio resources, compromising the computing efficiency gain. For a longer transmitter-receiver distance, the average transmission rate decreases due to a higher path loss, leading to a higher CPU frequency requirement for delay satisfaction, which reduces the computing efficiency gain. Hence, the dynamics in both shared workloads and transmitter-receiver distances should be considered in the adaptive selection of cooperative CAV pairs, to maximize the total computing efficiency gain.

Moreover, the cooperation decisions for each CAV pair should not change too frequently, as the switching between the SP and CP modes incur a CPU process switching overhead between scheduling the default and feature-fusion DNN models [27]. Let $C(n)$ be the total number of CAV pairs that change the cooperation status at time slot n, given by

$$C(n) = \sum_{k \in \mathcal{K}} \left| x_k(n) - x_k(n-1) \right|. \tag{12}$$

The total switching cost increases proportionally to $C(n)$. To maximize the total computing efficiency gain while minimizing the total switching cost in the long run, we study a joint adaptive CAV cooperation and resource allocation problem, to adaptively switch between the SP and CP modes and allocate resources among all CAV pairs for each time slot. Let $\boldsymbol{f}(n) = \{f_k(n), \forall k \in \mathcal{K}\}$ be a CPU frequency allocation decision vector for processing shared objects at all CAV pairs at time slot n. Let $\dot{\boldsymbol{x}} = \{\boldsymbol{x}(n), \forall n\}$, $\dot{\boldsymbol{\beta}} = \{\boldsymbol{\beta}(n), \forall n\}$, $\dot{\boldsymbol{f}} = \{\boldsymbol{f}(n), \forall n\}$ denote the CAV cooperation, radio resource allocation, and CPU frequency allocation decisions for all time slots. Then, the joint problem is formulated as

$$\mathbf{P}_0: \max_{\dot{\boldsymbol{x}}, \dot{\boldsymbol{\beta}}, \dot{\boldsymbol{f}}} \sum_n \left[\left(\sum_{k \in \mathcal{K}} G_k(n) \right) - \tilde{\omega} C(n) \right] \tag{13}$$

$$\text{s.t.} \quad (7), (10), (11)$$

$$\sum_{k \in \mathcal{K}} \beta_k(n) \le 1 \tag{14}$$

$$0 \le \beta_k(n) \le x_k(n), \ k \in \mathcal{K} \tag{15}$$

$$\left(1 - x_k(n)\right) f_k^{\mathsf{D}}(n) \le f_k(n) \le \left(1 - x_k(n)\right) f_k^{\mathsf{D}}(n) + x_k(n)\mathbb{M}, \ k \in \mathcal{K} \tag{16}$$

where $\tilde{\omega}$ is a positive weight that controls the trade-off between gain and cost, and $\mathbb{M}$ is a very large constant. Among the constraints, (7) is the expression of transmission rate $R_k(n)$, (10) and (11) are conditions for the ideal candidate rate-frequency pairs in the 2D performance regions. Constraints (14) and (15) ensure that the total fraction of allocated bandwidth for all CAV pairs does not exceed one, while guaranteeing that no radio resources are allocated to CAV pairs in the SP mode. With constraint (16), we have $f_k(n) = f_k^{\mathsf{D}}(n)$ in the SP mode and $0 \leq f_k(n) \leq \mathbb{M}$ in the CP mode for CAV pair k.

Let $*$ associate the optimal solution to problem $\mathbf{P}_0$. Given $\dot{\boldsymbol{x}}^*$, the resource allocation decisions can be decoupled among time slots in the objective function and all the constraints. Hence, given $\dot{\boldsymbol{x}}^*$, $(\boldsymbol{\beta}^*(n), \boldsymbol{f}^*(n))$ must be the resource allocation solution that maximizes the instantaneous total computing efficiency gain, $\sum_{k\in\mathcal{K}} G_k(n)$, for time slot n, since the total switching cost depends only on $\dot{\boldsymbol{x}}$. Therefore, problem $\mathbf{P}_0$ can be decoupled to a long-term optimization subproblem for the adaptive CAV cooperation and a series of instantaneous optimization subproblems for resource allocation, as follows.

A. Resource Allocation Subproblem

For time slot n, given any CAV cooperation decision $\boldsymbol{x}(n)$, a cooperative CAV pair set, $\mathcal{K}_C(n)$, is determined. For CAV pair k in the SP mode, we have $\beta_k(n) = 0$, $f_k(n) = f_k^{\mathsf{D}}(n)$ based on (15) and (16), and $G_k(n) = 0$. Accordingly, a resource allocation subproblem for time slot n is formulated as

$$\mathbf{P}_1: \max_{\boldsymbol{\beta}_C(n), \boldsymbol{f}_C(n)} \quad \sum_{k\in\mathcal{K}_C(n)} G_k(n) \tag{17}$$

$$\text{s.t.} \quad (7), (10), (11)$$

$$\sum_{k\in\mathcal{K}_C(n)} \beta_k(n) \leq 1 \tag{18}$$

where $\boldsymbol{\beta}_C(n) = \{\beta_k(n), \forall k \in \mathcal{K}_C(n)\}$ and $\boldsymbol{f}_C(n) = \{f_k(n), \forall k \in \mathcal{K}_C(n)\}$ are the resource allocation decisions for cooperative CAV pairs in $\mathcal{K}_C(n)$. If problem $\mathbf{P}_1$ is feasible, we have $G^*(n) = \sum_{k\in\mathcal{K}_C(n)} G_k^*(n)$ as the maximal total computing efficiency gain achieved with optimal resource allocation; otherwise, $G^*(n)$ is undefined.

B. Adaptive CAV Cooperation Subproblem

We formulate an adaptive CAV cooperation subproblem as a multi-agent Markov decision process (MMDP), where each agent corresponds to a CAV pair that makes binary cooperation decisions based on local observations over time. We consider cooperative agents in the MMDP, where all CAV pairs collaboratively maximize an expected total discounted reward. An MMDP is represented as $(\mathcal{K}, \mathcal{S}, \{\mathcal{A}_k\}, P, R, \{\Omega_k\}, \gamma)$, where $\mathcal{K}$ is a set of agents, $\mathcal{S}$ is the state space, $\mathcal{A}_k$ is the action space for agent k, with $\mathcal{A} = \times_k \mathcal{A}_k$ being the set of joint actions, P is an unknown state transition probability matrix, $R: \mathcal{S}\times\mathcal{A} \mapsto \mathbb{R}$ is a reward function, Ω_k is a set of observations for agent k, and γ is a discount factor in $[0, 1)$. The observation, action, and reward of agent k are given as follows.

- *Observation*: The local observation for agent k at time slot n, denoted by $o_k^{(n)}$, includes the available radio spectrum bandwidth for CAVs, $B(n)$, shared workload $W_k(n)$, transmitter-receiver distance $D_k(n)$, and previous cooperation status $x_k(n-1)$, given by
$$o_k^{(n)} = \{B(n), W_k(n), D_k(n), x_k(n-1)\}; \tag{19}$$
- *Action*: At time slot n, agent k's action is the cooperation decision, $x_k(n)$. Joint action $\boldsymbol{x}(n) = \{x_k(n), \forall k\}$ determines a cooperative CAV pair set, $\mathcal{K}_C(n)$;
- *Reward*: The reward for any agent at time slot n, denoted by $r^{(n)}$, is given by
$$r^{(n)} = \begin{cases} G^*(n) - \tilde{\omega}C(n), & \text{if } \mathbf{P}_1 \text{ is feasible} \\ P, & \text{otherwise} \end{cases} \tag{20}$$
where $G^*(n)$ is the maximal total computing efficiency gain associated with optimal resource allocation for $\mathcal{K}_C(n)$. If problem $\mathbf{P}_1$ is infeasible under the selected joint action, a negative penalty, P, is used as the reward. As both the objective function and the delay constraint in problem $\mathbf{P}_1$ are derived based on the early exit probabilities of DNN models, we aim to learn the *statistically* optimal adaptive CAV cooperation actions in the long run by using the reward function in (20).

V. Model-Assisted Multi-Agent Reinforcement Learning Solution

We propose a model-assisted learning solution to the joint problem. First, a model-based optimal solution is derived for the resource allocation subproblem. Then, a model-assisted MARL approach that relies on the model-based optimal resource allocation solution for reward calculation is proposed, to solve the MMDP for adaptive CAV cooperation.

A. Optimal Resource Allocation Solution

With (6), maximizing $\sum_{k\in\mathcal{K}_C(n)} G_k(n)$ in $\mathbf{P}_1$ is equivalent to minimizing $\sum_{k\in\mathcal{K}_C(n)} W_k(n) f_k(n)^2$. By writing $\beta_k(n)$ as a function of $R_k(n)$, we combine constraints (7) and (18) as

$$\sum_{k\in\mathcal{K}_C(n)} \frac{R_k(n)}{B(n)\log_2\left(1 + p_k g_k(n)/\sigma^2\right)} \leq 1. \tag{21}$$

From constraint (11), $R_k(n) = w/\left(\frac{\Delta}{W_k(n)} - \frac{\hat{\delta}}{f_k(n)}\right)$ is a function of $f_k(n)$. Substituting $R_k(n)$ in (21), we transform (21) into a constraint on decision variables $\boldsymbol{f}_C(n)$, given by

$$h(\boldsymbol{f}) = \sum_{k\in\mathcal{K}_C(n)} \frac{c_k}{b_k - \hat{\delta}/f_k(n)} - 1 \leq 0 \tag{22}$$

where $b_k = \frac{\Delta}{W_k(n)}$ and $c_k = \frac{w}{B(n)\log_2(1+p_k g_k(n)/\sigma^2)}$ are known parameters given network status for time slot n. Here, $h(\boldsymbol{f})$ is a monotonically decreasing constraint function of $\boldsymbol{f}_C(n)$, defined in domain $\{f_k(n) > \frac{\hat{\delta}}{b_k}, \forall k \in \mathcal{K}_C(n)\}$ to ensure $R_k(n) > 0$ for $k \in \mathcal{K}_C(n)$. Let $f_k^0(n) = \min\left\{f_k^{\mathsf{P}}(n), f_{\mathsf{M}}\right\}$, which is known to CAV pair k, and let $\boldsymbol{f}_C^0(n) = \left\{f_k^0(n), \forall k \in \mathcal{K}_C(n)\right\}$. Then, problem $\mathbf{P}_1$ is transformed to a CPU frequency allocation problem, given by

$$\mathbf{P}_2: \min_{\boldsymbol{f}_C(n)} \quad \sum_{k\in\mathcal{K}_C(n)} W_k(n) f_k(n)^2 \tag{23}$$

$$\text{s.t.} \quad \boldsymbol{f}_C(n) \preceq \boldsymbol{f}_C^0(n) \tag{24}$$

$$h(\boldsymbol{f}) \leq 0. \tag{25}$$

Theorem 1. Problem $\mathbf{P}_2$ is convex, and strong duality holds if the problem is feasible under condition $h(\boldsymbol{f}^0) < 0$.

The proof of Theorem 1 is given in Appendix. Based on Theorem 1, Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient conditions for optimal solution to problem $\mathbf{P}_2$ [28]. The Lagrangian of $\mathbf{P}_2$ is

$$\mathcal{L}(\boldsymbol{f}, \boldsymbol{\lambda}, \nu) = \sum_{k \in \mathcal{K}_C(n)} W_k(n) f_k(n)^2 + \sum_{k \in \mathcal{K}_C(n)} \lambda_k \left(f_k(n) - f_k^0(n)\right) + \nu\, h(\boldsymbol{f}) \tag{26}$$

where $\boldsymbol{\lambda} = \{\lambda_k, \forall k \in \mathcal{K}_C(n)\}$ and ν are dual variables. Its gradient with respect to primal variable $f_k(n)$ is given by

$$\frac{\partial \mathcal{L}(\boldsymbol{f}, \boldsymbol{\lambda}, \nu)}{\partial f_k(n)} = 2W_k(n) f_k(n) + \lambda_k - \nu \frac{c_k \hat{\delta}}{\left(b_k f_k(n) - \hat{\delta}\right)^2}. \tag{27}$$

Let $\boldsymbol{f}^*$ be a primal optimal point and $(\boldsymbol{\lambda}^*, \nu^*)$ be a dual optimal point. Then, the KKT conditions for the optimal solution to problem $\mathbf{P}_2$ are given by

$$f_k^* \leq f_k^0, \quad \forall k \in \mathcal{K}_C \tag{28a}$$

$$h(\boldsymbol{f}^*) \leq 0 \tag{28b}$$

$$\lambda_k^* \geq 0, \quad \forall k \in \mathcal{K}_C \tag{28c}$$

$$\nu^* \geq 0 \tag{28d}$$

$$\lambda_k^* \left(f_k^* - f_k^0\right) = 0, \quad \forall k \in \mathcal{K}_C \tag{28e}$$

$$2W_k f_k^* + \lambda_k^* - \nu^* \frac{c_k \hat{\delta}}{\left(b_k f_k^* - \hat{\delta}\right)^2} = 0, \ \forall k \in \mathcal{K}_C \tag{28f}$$

where time slot index n is omitted for brevity. Among all conditions, (28a) and (28b) are primal feasible conditions, (28c) and (28d) are dual feasible conditions, (28e) indicates the complementary slackness, and (28f) ensures that the Lagrangian gradient in (27) vanishes at $\boldsymbol{f}^*$ as $\boldsymbol{f}^*$ minimizes $\mathcal{L}(\boldsymbol{f}, \boldsymbol{\lambda}^*, \nu^*)$ [28]. From (28e) and (28f), we obtain

$$\left(\frac{\nu^* c_k \hat{\delta}}{\left(b_k f_k^* - \hat{\delta}\right)^2} - 2W_k f_k^*\right)\left(f_k^* - f_k^0\right) = 0, \forall k \in \mathcal{K}_C. \tag{29}$$

For dual variable ν, let $S(f_k, \nu) = \nu \frac{c_k \hat{\delta}}{\left(b_k f_k - \hat{\delta}\right)^2} - 2W_k f_k$, which is a monotonically decreasing function in domain $f_k > \frac{\hat{\delta}}{b_k}$. Let $f_k^1(\nu)$ be the root of $S(f_k, \nu)$, which corresponds to the intersection point of functions $\nu \frac{c_k \hat{\delta}}{\left(b_k f_k - \hat{\delta}\right)^2}$ and $2W_k f_k$ in domain $f_k > \frac{\hat{\delta}}{b_k}$. Fig. 5 illustrates the root of function $S(f_k, \nu)$ for $\nu_1 < \nu_2$. We see that, for a smaller value of dual variable ν, the root of $S(f_k, \nu)$, i.e., $f_k^1(\nu)$, has a smaller value. According to (28f), we have $\lambda_k^* = S(f_k^*, \nu^*)$, and condition (29) is rewritten as

$$S(f_k^*, \nu^*)\left(f_k^* - f_k^0\right) = 0, \quad \forall k \in \mathcal{K}_C. \tag{30}$$

Then, the primal optimal point, $\boldsymbol{f}^*$, that minimizes the objective function in (23) is given by

$$f_k^* = \begin{cases} f_k^1(\nu^*), & \text{if } f_k^1(\nu^*) \leq f_k^0 \\ f_k^0, & \text{if } f_k^0 < f_k^1(\nu^*) \end{cases} \tag{31}$$

for any $k \in \mathcal{K}_C$, to guarantee condition (30) and ensure that $\lambda_k^* = S(f_k^*, \nu^*) \geq 0$ in (28c). Specifically, if $f_k^1(\nu^*) \leq f_k^0$, we have $\lambda_k^* = 0$; otherwise, we have $\lambda_k^* > 0$.

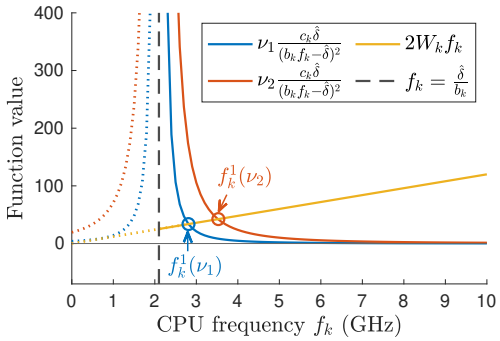


Fig. 5: An illustration of the root of function $S(f_k, \nu)$ for $\nu_1 < \nu_2$.

As ν^* is an unknown optimal dual variable in (31), we will find the primal optimal point, $\boldsymbol{f}^*$, by exploring the possible values of ν^*. Let $\hat{\nu}^*$ be a candidate value of ν^*, and let $\hat{\boldsymbol{f}}^*$ be the corresponding candidate value of $\boldsymbol{f}^*$ for $\nu^* = \hat{\nu}^*$ based on (31). For a smaller $\hat{\nu}^*$ value, $\hat{f}_k^1(\hat{\nu}^*)$ is smaller, then $\hat{f}_k^*$ is potentially smaller according to (31), leading to a potentially smaller objective value in (23). However, as $h(\hat{\boldsymbol{f}}^*)$ is a decreasing function of $\hat{f}_k^*$, the primal feasible condition in (28b) might be violated if the value of $\hat{\nu}^*$ is too small. Therefore, we should find a minimum non-negative value for $\hat{\nu}^*$ that satisfies (28b) and (28d), to obtain ν^* and $\boldsymbol{f}^*$.

To solve problem $\mathbf{P}_2$, the feasibility is first checked by calculating $h(\boldsymbol{f}^0)$. If $h(\boldsymbol{f}^0) > 0$, the problem is infeasible; if $h(\boldsymbol{f}^0) = 0$, we can directly obtain the optimal solution as $\boldsymbol{f}^* = \boldsymbol{f}^0$; if $h(\boldsymbol{f}^0) < 0$, we continue to use a binary-search method to iteratively find ν^* in a gradually reduced interval $[\nu_L, \nu_R]$. The detailed algorithm is described as follows.

For initialization, as $f_k^* \in \left(\frac{\hat{\delta}}{b_k}, f_k^0\right]$, ν_L and ν_R are set to satisfy $S\left(\frac{\hat{\delta}}{\max_{k \in \mathcal{K}_C} b_k} + \epsilon, \nu_L\right) = 0$ and $S\left(\min_{k \in \mathcal{K}_C} f_k^0, \nu_R\right) = 0$. Here, ϵ is a very small number satisfying $0 < \epsilon \ll 1$. The shared workloads among all cooperative CAV pairs determine the initial binary-search interval, $[\nu_L, \nu_R]$. In each iteration, $\hat{\nu}^*$ is set as $\frac{\nu_L + \nu_R}{2}$, and $\hat{\boldsymbol{f}}^*$ is obtained based on (31). For $\boldsymbol{f} = \hat{\boldsymbol{f}}^*$, let $\hat{G}^* = \sum_{k \in \mathcal{K}_C} \hat{G}_k^*$ denote the corresponding total computing efficiency gain in (17). Constraint (28b) is checked by calculating $h(\hat{\boldsymbol{f}}^*)$. Whether or not the binary search ends at the current iteration and how to update interval $[\nu_L, \nu_R]$ for the next iteration depend on $h(\hat{\boldsymbol{f}}^*)$:

- If $h(\hat{\boldsymbol{f}}^*) = 0$, the optimal points, ν^* and $\boldsymbol{f}^*$, are obtained as $\hat{\nu}^*$ and $\hat{\boldsymbol{f}}^*$, and the algorithm is ideally finished. In practice, we set a stopping criteria, $-10^{-4} < h(\hat{\boldsymbol{f}}^*) < 0$, and obtain asymptotically optimal primal and dual variables when the binary search ends;
- If $h(\hat{\boldsymbol{f}}^*) > 0$, constraint (28b) is infeasible, and the candidate value for ν^* should be increased in the next iteration, thus we set $\nu_L = \hat{\nu}^*$;

- If $h(\hat{\boldsymbol{f}}^*) < 0$, the candidate value for ν^* can be further reduced in the next iteration to increase $h(\hat{\boldsymbol{f}}^*)$ and reduce the objective value in (23), thus we set $\nu_R = \hat{\nu}^*$.

Such a centralized iterative algorithm can be executed at the cluster head which is responsible for collecting the overall network dynamics in the vehicle cluster.

B. Model-Assisted Multi-Agent Reinforcement Learning

We use a model-assisted MARL algorithm to solve the MMDP for adaptive CAV cooperation. To address the inherent nonstationary issue due to the partial observability at each agent, we use a multi-agent deep deterministic policy gradient (MADDPG) algorithm which adopts a centralized training distributed execution (CTDE) framework [29], [30]. The model-assisted MADDPG algorithm is presented in Algorithm 1. Each agent trains a critic network and an actor network based on the global state and joint action in a centralized training stage, and uses the trained actor network for decision based on local observation in a distributed execution stage. To enhance the observability at each agent, we augment the local observation by two additional elements, i.e., the average workload and the average transmitter-receiver distance among all agents, both of which can be provided by the cluster head [31]. In this manner, a CAV pair has both local information and some statistical global information for better-informed decisions during the distributed execution stage, without acquiring the full global state. Let $s_k^{(n)}$ be the augmented local observation for agent k at time slot n, given by $s_k^{(n)} = \left\{o_k^{(n)}, \frac{\sum_{k\in\mathcal{K}} W_k(n)}{K}, \frac{\sum_{k\in\mathcal{K}} D_k(n)}{K}\right\}$. Let $\boldsymbol{s}^{(n)} = \{s_k^{(n)}, \forall k \in \mathcal{K}\}$ be the global state at time slot n.

As the MMDP has a discrete action space for each agent, specifically a binary action space for whether or not to cooperate, a Gumbel-Softmax estimator is used by each agent to allow the backpropagation of gradients through the actor network during training [29]. For agent k, an actor network, $\mu_k(s_k)$, parameterized by weights $\boldsymbol{\varphi}_k$, is trained to learn a continuous action, $a_k = \{a_{k,j}, j = 0, 1\}$, based on augmented local observation s_k. Here, a_k is an estimated two-dimension Gumbel-Softmax distribution over the binary action space [32]. Let $a_k^{(n)} = \{a_{k,j}^{(n)}, j = 0, 1\}$ be the continuous action of agent k at time slot n, and let $\boldsymbol{a}^{(n)} = \{a_k^{(n)}, \forall k\}$ be the joint continuous action at time slot n. Agent k obtains the cooperation decision as the binary action with the maximum Gumbel-Softmax probability, i.e., $x_k(n) = \arg\max_{j=0,1} a_{k,j}^{(n)}$.

In addition to actor network $\mu_k(s_k)$, agent k trains a critic network parameterized by weights $\boldsymbol{\theta}_k$ to approximate a centralized Q-function, $Q_k(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}\left[\sum_{n=0}^{N-1} \gamma^n r_k^{(n)} \middle| \boldsymbol{s}, \boldsymbol{a}\right]$, that takes global state $\boldsymbol{s}$ and joint action $\boldsymbol{a}$ as input to estimate a Q-value, where N is the maximum number of learning steps in an episode. In the training stage of such an actor-critic framework, although the agents independently take actions based on the augmented local observations, they evaluate the actions and refine the policies by taking the actions of other agents into consideration in $Q_k(\boldsymbol{s}, \boldsymbol{a})$, thus facilitating a collaborative exploration of the vehicular network environment to maximize a collective reward. To overcome the divergence

Algorithm 1: A Model-Assisted MADDPG Algorithm

```
/* Centralized Training Stage                                   */
All agents initialize networks with random weights.
for each episode do
    Initialize local observation o_k^(0) for k ∈ K.
    for learning step n do
        for agent k do
            Send local observation o_k^(n) to cluster head.
            Collect augmented local observation s_k^(n).
            Decide continuous action a_k^(n) = μ_k(s_k^(n)),
              derive binary cooperation decision x_k(n), and
              send a_k^(n) to cluster head.
        Cluster head solves the resource allocation
          subproblem, obtains G*(n) and r^(n), and
          broadcasts s^(n), a^(n), and r^(n) to all agents.
        for agent k do
            Add (s^(n-1), a^(n-1), r^(n-1), s^(n)) to buffer B
              if n ≥ 1.
            Sample mini-batch of experiences from B.
            Update primary and target critic and actor
              networks based on (33), (34), and (32).
/* Distributed Execution Stage                                  */
for each episode do
    for learning step n do
        for agent k do
            Send local observation o_k^(n) to cluster head.
            Collect augmented local observation s_k^(n).
            Decide continuous action a_k^(n) = μ_k(s_k^(n)),
              derive binary cooperation decision x_k(n), and
              send x_k(n) to cluster head.
        Cluster head allocates resources to cooperative CAV
          pairs.
```

update issue, agent k also has a target critic network, $\hat{Q}_k(\boldsymbol{s}, \boldsymbol{a})$, parameterized by $\hat{\boldsymbol{\theta}}_k$, and a target actor network, $\hat{\mu}_k(s_k)$, parameterized by $\hat{\boldsymbol{\varphi}}_k$, with delayed updates. Agent k initializes the primary and target critic and actor networks with random weights before training (*line* 1) and then continually updates the weights until convergence. MADDPG employs a soft updating strategy, where agent k updates weights $\hat{\boldsymbol{\theta}}_k$ and $\hat{\boldsymbol{\varphi}}_k$ of the target networks in each learning step (*line* 13) as

$$\hat{\boldsymbol{\theta}}_k = \xi\boldsymbol{\theta}_k + (1-\xi)\,\hat{\boldsymbol{\theta}}_k \quad \text{and} \quad \hat{\boldsymbol{\varphi}}_k = \xi\boldsymbol{\varphi}_k + (1-\xi)\,\hat{\boldsymbol{\varphi}}_k \tag{32}$$

with ξ being the soft updating rate of the target networks.

The agents interact with the network environment in a sequence of episodes, each containing a finite number of learning steps, one learning step for one time slot. An episode starts when a vehicle cluster is about to move into an RSU's coverage area and ends once it leaves the coverage. At the beginning of each episode, each agent initializes the local observation (*line* 3). At the beginning of time slot n, each agent k sends local observation $o_k^{(n)}$ via a dedicated control channel to the cluster head (*line* 6), which calculates and then returns the augmented information. Agent k collects augmented local observation $s_k^{(n)}$ (*line* 7), based on which the agent decides continuous action $a_k^{(n)}$ as $\mu_k\left(s_k^{(n)}\right)$ by the primary actor

network, and then discretizes it into a binary cooperation decision, $x_k(n)$ (*line* 8). All agents send the continuous actions to the cluster head (*line* 8). The cluster head allocates resources to cooperative CAV pairs, determines the maximal total computing efficiency gain, $G^*(n)$, using the model-based resource allocation solution, and calculates reward $r^{(n)}$ in (20) which is then broadcast to all agents together with global state $\boldsymbol{s}^{(n)}$ and joint action $\boldsymbol{a}^{(n)}$ (*line* 9). Then, each agent adds a new transition tuple $\left(\boldsymbol{s}^{(n-1)}, \boldsymbol{a}^{(n-1)}, r^{(n-1)}, \boldsymbol{s}^{(n)}\right)$ to an experience replay buffer, $\mathcal{B}$, if $n \geq 1$ (*line* 11).

To train the critic and actor networks at each learning step, each agent samples a mini-batch of I experiences from $\mathcal{B}$, among which $\left(\boldsymbol{s}^{(i)}, \boldsymbol{a}^{(i)}, r^{(i)}, \boldsymbol{s}^{(i+1)}\right)$ represents the i-th experience (*line* 12). Agent k updates the critic network by minimizing a loss function, $\mathbb{L}_k(\boldsymbol{\theta}_k) = \frac{1}{I}\sum_{i=1}^{I}\left[y_k^{(i)} - Q_k\left(\boldsymbol{s}^{(i)}, \boldsymbol{a}^{(i)}\right)\right]^2$, where $y_k^{(i)} = r^{(i)} + \gamma \hat{Q}_k\left(\boldsymbol{s}^{(i+1)}, \hat{\mu}_1\left(s_1^{(i+1)}\right), \ldots, \hat{\mu}_K\left(s_K^{(i+1)}\right)\right)$ is a target value estimated by the target critic and actor networks. Weights $\boldsymbol{\theta}_k$ are updated via a gradient descent (*line* 13), given by

$$\boldsymbol{\theta}_k \leftarrow \boldsymbol{\theta}_k - \alpha_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}_k} \mathbb{L}_k(\boldsymbol{\theta}_k) \tag{33}$$

where $\alpha_{\boldsymbol{\theta}}$ is the learning rate for critic networks. The actor network of agent k aims to maximize a long-term total expected reward, $J_k(\boldsymbol{\varphi}_k) = \mathbb{E}\left[\sum_{n=0}^{N} \gamma^n r_k^{(n)}\right]$, which is the expected Q-value among all state-action pairs, i.e., $J_k(\boldsymbol{\varphi}_k) = \mathbb{E}_{\boldsymbol{s},\boldsymbol{a}} Q_k(\boldsymbol{s}, \boldsymbol{a})$. Thus, weights $\boldsymbol{\varphi}_k$ of the actor network are updated via a gradient ascent (*line* 13), given by

$$\boldsymbol{\varphi}_k \leftarrow \boldsymbol{\varphi}_k + \alpha_{\boldsymbol{\varphi}} \nabla_{\boldsymbol{\varphi}_k} J_k(\boldsymbol{\varphi}_k) \tag{34}$$

where $\alpha_{\boldsymbol{\varphi}}$ is the learning rate for actor networks, and the gradient of $J_k(\boldsymbol{\varphi}_k)$ is given by

$$\nabla_{\boldsymbol{\varphi}_k} J_k(\boldsymbol{\varphi}_k) = \frac{1}{I}\sum_{i=1}^{I}\Big[\nabla_{\boldsymbol{\varphi}_k}\mu_k\left(s_k^{(i)}\right) \nabla_{a_k} Q_k\left(\boldsymbol{s}^{(i)}, a_1^{(i)}, \ldots, a_k, \ldots, a_K^{(i)}\right)\Big|_{a_k=\mu_k\left(s_k^{(i)}\right)}\Big]. \tag{35}$$

After the centralized training stage, each agent uses the trained actor network for decision in a distributed execution stage, with the assistance of the cluster head (*lines* 14-20).

VI. Simulation Results

A. Simulation Setup

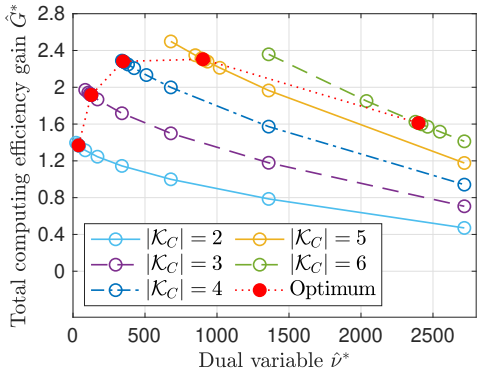
We consider a four-lane unidirectional highway, where $K \in \{2, 3, 4, 5, 6\}$ CAV pairs are moving together with 10 HDVs in a vehicle cluster, with an intermittent RSU coverage. The RSU is 10 m away from the highway and provides a communication radius of 250 m. In the RL task, we consider a 1500 m highway segment for each episode, during which a vehicle cluster moves for a distance of 1000 m through the RSU coverage. The vehicle speed is uniformly set in a range of $[23, 27]$ m/s. The time slot length is 500 ms. Each episode spans over an average time duration of 40 s and contains 80 time slots on average. The delay requirement for object classification in each time slot is $\Delta = 100$ ms.

TABLE III: System parameters in simulation

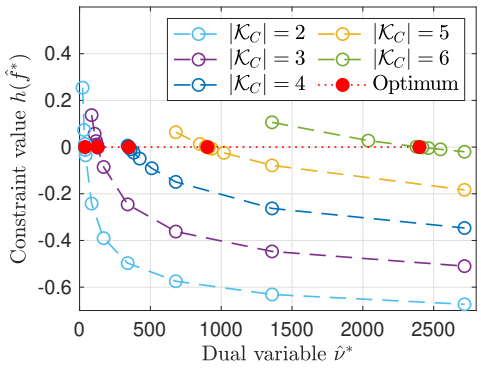
Parameters	Value
Center frequency (f_c)	6 GHz
Noise power (σ^2)	-104 dBm
Transmit power (p_k)	23 dBm
Maximum local CPU frequency (f_{M})	8 GHz
Energy efficiency coefficient (κ)	10^{-28} J/s/Hz^3
Feature extraction computing demand (δ_1)	4×10^6 cycles
Feature fusion computing demand (δ_2)	1000 cycles
Fast inference computing demand (δ_3)	3.1×10^5 cycles
Full inference computing demand (δ_4)	7.7×10^7 cycles
Feature data size (w)	0.29 Mbits
Default early exit probability (ρ)	0.3
Feature-fusion early exit probability ($\tilde{\rho}$)	0.6

We use the Simulation of Urban Mobility (SUMO) traffic simulator to simulate the vehicle trajectories in each episode, based on which the transmitter-receiver distances of each CAV pair can be obtained [9], [31], [33]. To obtain the time-varying channel power gain for each CAV pair, we use the 3GPP NR-V2X 37.885 highway case for the V2V link path loss calculation [34]. The path loss in dB for CAV pair k during time slot n is calculated as $L_{dB}(n) = 32.4 + 20\log_{10} D_k(n) + 20\log_{10} f_c$, where $D_k(n)$ is the transmitter-receiver distance in meter, and f_c is the center frequency in GHz. For CAV pair k, the transitions of shared workload, $W_k(n)$, across different time slots follow a Markov chain with states in $\{4, 5, 6, 7, 8\}$. When residing in the RSU coverage, each HDV generates V2R transmission requests in each time slot according to a $Bernoulli(0.5)$ distribution, with each V2R transmission request occupying a bandwidth of 0.5 MHz. With a total bandwidth of $B = 10.5$ MHz for V2X sidelink communication, the available radio spectrum bandwidth for V2V transmission at time slot n is $B(n) = B - 0.5M(n)$, where $M(n)$ is the number of HDVs that request V2R transmission at time slot n. On average, the $B(n)$ value in an episode follows a decreasing-then-increasing trend when the vehicle cluster drives through the RSU coverage. Other system parameters are given in Table III.

We implement both the iterative algorithm for optimal resource allocation and the MADDPG algorithm for adaptive CAV cooperation using Python 3.9.2. The learning modules are implemented using TensorFlow 2.11.0. Each learning agent has two hidden layers with $(64, 64)$ neurons and `Relu` activation functions in critic and actor networks. The critic network has a one-dimension output with no activation function, and the actor network has a two-dimension output with `Gumbel-SoftMax` activation. We set weight $\tilde{\omega} \in [0, 1]$ and penalty $P = -10$ in reward function (20), and use $\alpha_{\boldsymbol{\theta}} = 10^{-2}$ and $\alpha_{\boldsymbol{\varphi}} = 10^{-3}$ as the critic and actor learning rates, a soft updating rate of $\xi = 0.01$ for target network update, and discount factor $\gamma = 0.95$. For training at each learning step, a mini-batch of $I = 1024$ experiences are sampled from buffer $\mathcal{B}$ with size 100000.

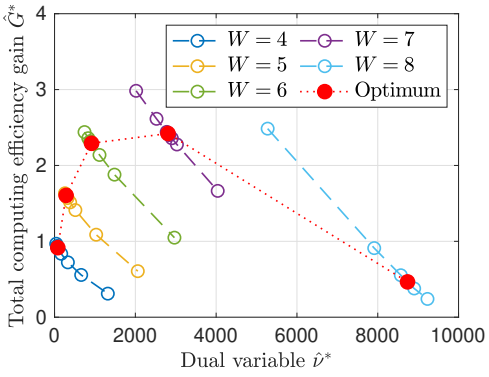


(a)

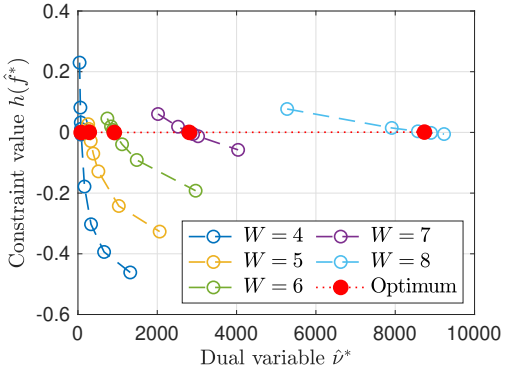


(b)

Fig. 6: Performance of the optimal resource allocation solution for a different number of cooperative CAV pairs ($|\mathcal{K}_C|$) at $W = 6$. (a) Total computing efficiency gain. (b) Constraint value.



(a)



(b)

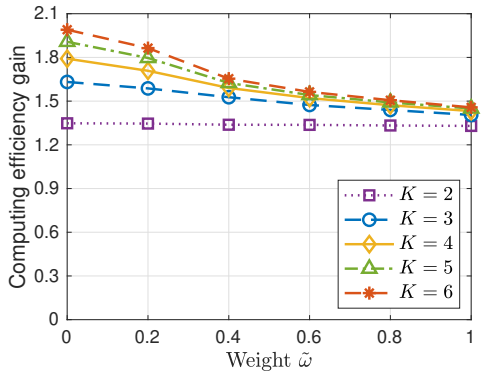
Fig. 7: Performance of the optimal resource allocation solution for a different workload (W) at $|\mathcal{K}_C| = 5$. (a) Total computing efficiency gain. (b) Constraint value.

B. Performance Evaluation

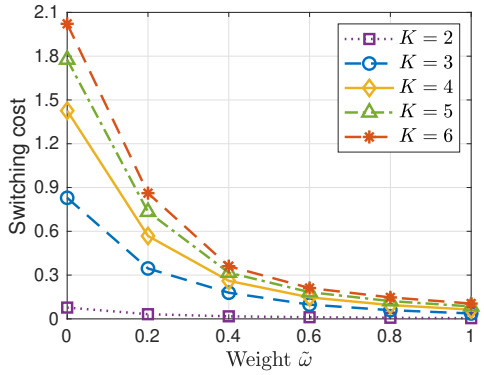
We first evaluate the performance of the optimal resource allocation solution, for a set, $\mathcal{K}_C$, of cooperative CAV pairs with an available bandwidth of 10.5 MHz for V2V transmission. Without loss of generality, all cooperative CAV pairs have an identical shared workload, W. We examine the impact of $|\mathcal{K}_C|$ and W on the total computing efficiency gain.

In the first set of simulations, the performance of the optimal resource allocation solution is evaluated for $|\mathcal{K}_C| \in \{2, 3, 4, 5, 6\}$, with $W = 6$. The transmitter-receiver distances of all the cooperative CAV pairs are set as 20 m. For the constant workload, the initial binary-search interval, $[\nu_L, \nu_R]$, is the same for all the $|\mathcal{K}_C|$ values. Fig. 6 shows the variations of total computing efficiency gain $\hat{G}^*$ and constraint value $h(\hat{\boldsymbol{f}}^*)$ during the binary-search of candidate optimal dual variable $\hat{\nu}^*$, for each $|\mathcal{K}_C|$. The asymptotically optimal total computing efficiency gain and constraint value, G^* and $h(\boldsymbol{f}^*)$, obtained at an asymptotically optimal dual variable, ν^*, are represented by a red dot for each $|\mathcal{K}_C|$ value. As $|\mathcal{K}_C|$ increases, the radio resources are shared among more cooperative CAV pairs for feature data transmission, and each cooperative CAV pair should increase the CPU frequency to compensate for the lower average transmission rate, to achieve delay satisfaction. Accordingly, as $|\mathcal{K}_C|$ increases, the asymptotically optimal CPU frequency allocation variables, $\boldsymbol{f}^*$, are larger, corresponding to a larger ν^* value. Although the total reduced amount of computing demand increases in proportion to $|\mathcal{K}_C|$, the total computing efficiency gain gradually decreases as the CPU frequency further increases, leading to a first-increasing-then-decreasing trend of G^*, as shown in Fig. 6(a). At the asymptotically optimal points, constraint value $h(\boldsymbol{f}^*)$ approaches zero, as shown in Fig. 6(b).

In the second set of simulations, the performance of the optimal resource allocation solution is evaluated for $|\mathcal{K}_C| = 5$, with shared workload $W \in \{4, 5, 6, 7, 8\}$, as shown in Fig. 7. The transmitter-receiver distances of the 5 cooperative CAV pairs are set as $[20.4, 16.5, 11.4, 29.7, 28.3]$ m, respectively. For a heavier workload, there is a right shift for the initial binary-search interval, $[\nu_L, \nu_R]$, as both ν_L and ν_R have a larger value. When W increases, the total computing demand reduction increases proportionally, while the per-object delay budget, $\frac{\Delta}{W}$, decreases inverse proportionally. With a constant radio spectrum bandwidth, the CPU frequency should be increased at each CAV pair to satisfy the more stringent delay requirement as W increases. Hence, in Fig. 7(a), we observe a first-increasing-then-decreasing trend for G^*, due to a trade-off between computing demand and CPU frequency. Fig. 7(b) shows that the constraint value, $h(\boldsymbol{f}^*)$, approaches zero at the asymptotically optimal points. Fig. 6 and Fig. 7 demonstrate that, with a limited amount of radio resources, it is necessary to select the best subset of CAV pairs for cooperation while taking the shared workload into account, to improve the total

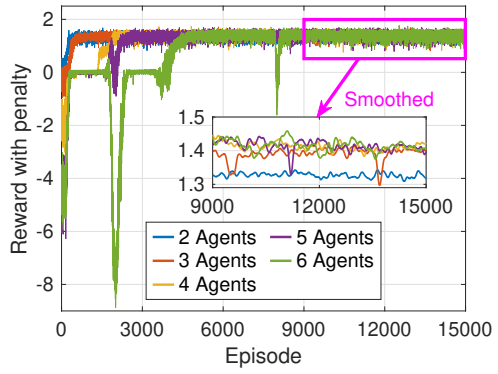


(a)

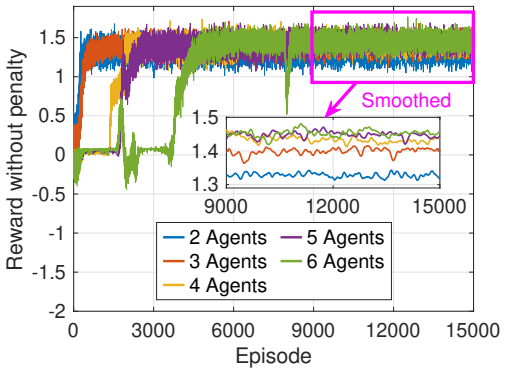


(b)

Fig. 8: An illustration of gain-cost trade-off for different $\tilde{\omega}$ values. (a) Computing efficiency gain. (b) Switching cost.



(a)



(b)

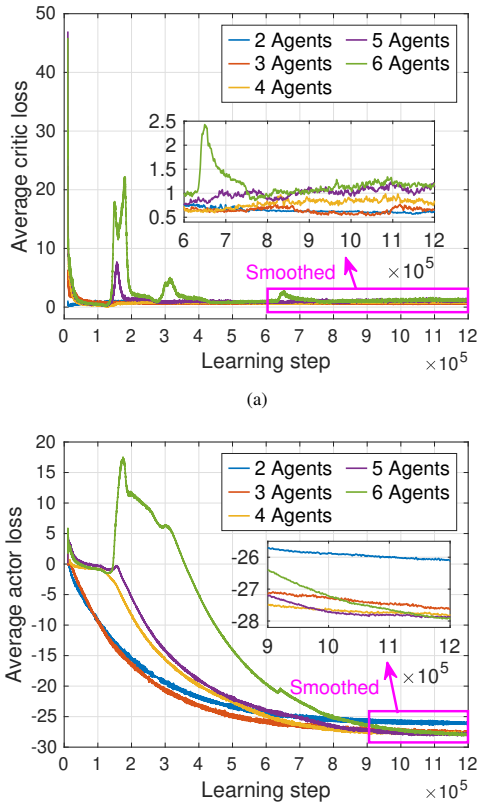
Fig. 9: Convergence of the reward during the training process. (a) With penalty. (b) Without penalty.

computing efficiency gain.

Before the performance evaluation of the MADDPG algorithm for adaptive CAV cooperation, we examine the impact of weight $\tilde{\omega}$ in reward function (20) on the trade-off between computing efficiency gain and switching cost. We set $\tilde{\omega} \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$. For each $\tilde{\omega}$ value, a group of experiments are performed for a different number of CAV pairs (K). In each experiment, a brute-force search is conducted among all possible CAV cooperation decisions for a maximum instantaneous reward in each time slot of 2000 episodes. Fig. 8 shows the slot-average performance in terms of computing efficiency gain and switching cost for different K values as $\tilde{\omega}$ increases. As a larger $\tilde{\omega}$ value puts more emphasis on minimizing the switching cost, we observe a decreasing trend for both gain and cost with the increase of $\tilde{\omega}$. We also observe higher gain and cost for more CAV pairs at a given $\tilde{\omega}$ value, which is to be discussed later. The weight, $\tilde{\omega}$, can be selected according to the desired gain-cost trade-off. In the following, $\tilde{\omega}$ is set to 0.4, as we observe that the cost is reduced by more than 80% at a gain loss of less than 20% in comparison with that achieved at $\tilde{\omega} = 0$, for $K = 6$.

Next, we evaluate the convergence of the MADDPG algorithm, in terms of both reward and training loss, for a different number of CAV pairs. Fig. 9 shows the convergence of the average reward per learning step over 15000 episodes, for a different agent number (K) from 2 to 6. In a practical implementation, when there is a negative penalty in the reward due to infeasible resource allocation, we can refine the joint action and let all the CAV pairs work in the default SP mode, which gives a zero computing efficiency gain and a refined reward without penalty. The original reward with penalty and the original action are used during training to guide the learning agents towards a least penalty after convergence, while the reward without penalty is the true reward for the CAV pairs when the action refinement is enabled. Fig. 9 shows the average rewards with and without penalty during the training process. As K increases, both rewards converge in a slower speed, indicated by a later increase to a converged value interval. By comparing the smoothed rewards with and without penalty after convergence, we see that the penalty has been effectively suppressed, indicated by the limited large negative glitches in the reward with penalty. It implies that the learning agents have collaboratively learn the adaptive CAV cooperation decisions that are feasible for resource allocation under the network dynamics. Moreover, we observe that more learning agents tend to improve the reward, as to be discussed.

Fig. 10 shows the per-agent average critic and actor loss during training for K from 2 to 6. For the critic network, the input dimension grows linearly with K. Thus, it is more difficult to minimize the critic loss if there are more agents, leading to a higher average critic loss after convergence as K increases, as shown in Fig. 10(a). As the reward tends to increase for more agents, as shown in Fig. 9, the average Q-value for the sampled mini-batch of experiences at each



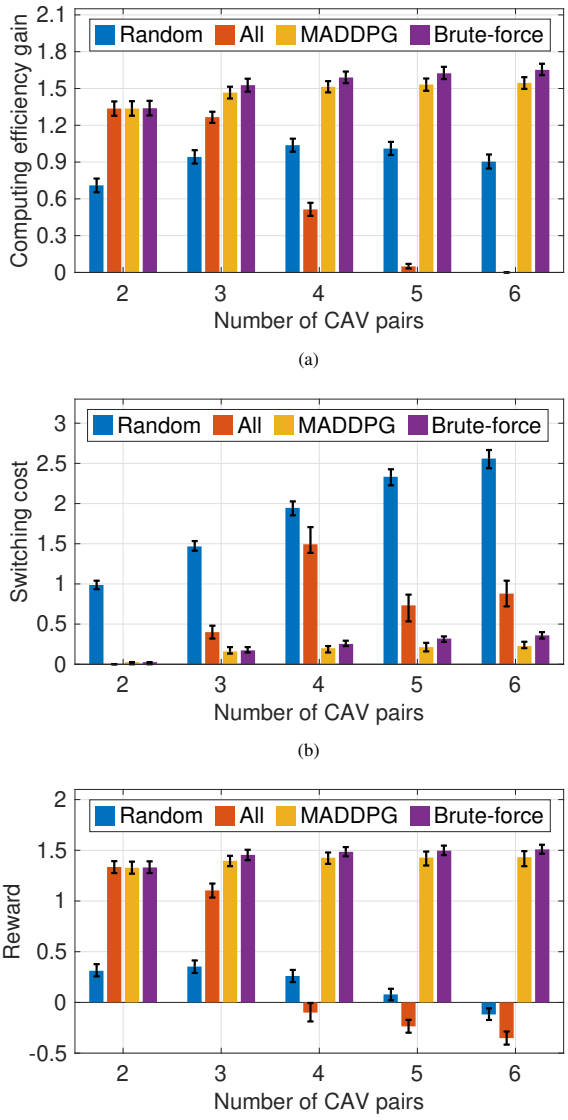
(a)

(b)

Fig. 10: Convergence of the critic loss and actor loss during the training process. (a) Average critic loss. (b) Average actor loss.

learning step increases for more agents. As the actor loss is the negative average Q-value, there is a lower average actor loss after convergence as K increases, as shown in Fig. 10(b).

To evaluate the effectiveness of the MADDPG algorithm in improving the computing efficiency gain and reducing the switching cost, we compare the performance between the trained MADDPG algorithm and three benchmark algorithms, for a different number of CAV pairs (K). The first benchmark is a random CAV cooperation scheme, in which each CAV pair switches between SP and CP modes at random in each time slot. Action refinement is enabled. In the second benchmark, all CAV pairs always cooperate if there exists a feasible resource allocation solution among them. Otherwise, action refinement is triggered to let all CAV pairs work in the default SP mode. Hence, the solution switches between all CAV pairs working in the CP mode and all CAV pairs working in the SP mode, referred to as "all CP mode" and "all SP mode" respectively. In the third benchmark, we conduct a step-wise brute-force search among all candidate joint CAV cooperation decisions in each time slot, for a maximum instantaneous reward. The time complexity for the brute-force benchmark is 2^K times of that for a trained MADDPG algorithm, for solving 2^K resource allocation subproblems given each candidate joint CAV cooperation decision. By using a trained MADDPG algorithm, only one resource allocation subproblem is solved given one learned joint action. For performance comparison, the 25%, 50%, and 75% percentiles of the slot-average total



(a)

(b)

(c)

Fig. 11: Performance comparison between the proposed and benchmark algorithms. (a) Computing efficiency gain. (b) Switching cost. (c) Reward.

computing efficiency gain, slot-average total switching cost, and slot-average reward are evaluated for each K value using the four algorithms, as shown in Fig. 11.

Fig. 11(a) demonstrates an increasing trend of the computing efficiency gain as the number of CAV pairs increases for both the MADDPG algorithm and the brute-force benchmark, both of which significantly outperform the other two benchmarks. Fig. 6(a) demonstrates the first-increasing-then-decreasing computing efficiency gain when more CAV pairs cooperate. Accordingly, for the random benchmark in which the average number of selected cooperative CAV pairs increases linearly with K, we observe such a trend in the computing efficiency gain as K increases, with a turning point at $K = 4$. The second benchmark achieves a gradually degraded computing efficiency gain when K increases, until obtaining a zero gain at $K = 6$. For a larger K value, the chance for infeasible resource allocation among all CAV

pairs increases, leading to more frequent action refinement to the "all SP mode" with zero gain. Nevertheless, in both the MADDPG algorithm and the brute-force benchmark, as there are more candidate CAV pairs in the system, there is a higher flexibility in selecting the best subset of CAV pairs for cooperation, by considering their shared workloads and transmitter-receiver distances. This flexibility contributes to a further increase in the computing efficiency gain as K further increases from 4 to 6. We observe a decreasing speed in the increase of computing efficiency gain as K further increases, as the gain from the higher flexibility gradually saturates. Especially, it is more difficult for the MADDPG algorithm to find the optimal solution in a distributed fashion as the agent number, K, increases, leading to a slightly larger sub-optimality gap from the brute-force benchmark.

From Fig. 11(b), we observe an almost linear increasing switching cost for the random benchmark, while both the MADDPG algorithm and the brute-force benchmark can significantly reduce the switching cost. Especially, as the switching cost has time correlation, the MADDPG algorithm which can minimize the total switching cost in the long run achieves a lower average switching cost in comparison with the step-wise brute-force benchmark which does not take the future states into account. For the second benchmark, as K increases, the dominant solution gradually changes from the "all CP mode" to the "all SP mode", due to a higher action refinement probability. Accordingly, the highest switching cost is obtained at a medium K value. For both the MADDPG algorithm and the brute-force benchmark, due to a higher flexibility in cooperative CAV pair selection as K increases, there are more changes in the selection decision, leading to an increasing switching cost as K increases. The reward, which is a linear combination of the computing efficiency gain and the switching cost, is shown in Fig. 11 (c) for reference.

VII. Conclusion

In this paper, we develop an adaptive cooperative perception framework for CAVs in a moving mixed-traffic vehicle cluster, while considering the dynamic shared workloads and channel conditions due to vehicle mobility, dynamic radio resource availability, and intermittent RSU coverage. A model-assisted multi-agent reinforcement learning solution is developed, which integrates learning-based adaptive CAV cooperation decision over time with model-based resource allocation decision in each time slot. Simulation results demonstrate the necessity for dynamically activating the cooperative perception among CAV pairs to improve the total computing efficiency gain. The effectiveness of the model-assisted MADDPG algorithm is verified, in improving the total computing efficiency gain under a limited switching cost. In future works, we will explore the generalization of the learning model for adaptive CAV cooperation, while considering a varying vehicle cluster size and a non-stationary network environment.

Appendix: Proof of Theorem 1

Lemma 1. Constraint (25) must be active for an optimal solution to problem $\mathbf{P}_2$.

Proof. Assume there is an optimal solution to $\mathbf{P}_2$ which achieves "$<$" in (25). Then, there must be another feasible solution achieving "$=$" in (25) by decreasing the CPU frequency for some CAV pairs, thus further decreasing the objective value in (23) without violating constraint (24), as the objective function is an increasing function of $\boldsymbol{f}_C(n)$ while constraint function $h(\boldsymbol{f})$ is a decreasing function of $\boldsymbol{f}_C(n)$. Therefore, the assumption must be false, and constraint (25) must be active for an optimal solution. Lemma 1 is proved. □

For problem $\mathbf{P}_2$, the objective function is a second-order function of decision variable $f_k(n) \in \boldsymbol{f}_C(n)$ with coefficient $W_k(n) > 0$, which is convex. Constraint (24) is linear. For constraint (25), the second-order derivative of constraint function $h(\boldsymbol{f})$ with respect to $f_k(n)$, is given by

$$\frac{\partial h^2(\boldsymbol{f})}{\partial \left(f_k(n)\right)^2} = \frac{2c_k b_k \hat{\delta}}{\left(b_k f_k(n) - \hat{\delta}\right)^3}. \tag{A1}$$

As $f_k(n) > \frac{\hat{\delta}}{b_k}$, we have $\partial h^2(\boldsymbol{f})/\partial \left(f_k(n)\right)^2 > 0$, thus $h(\boldsymbol{f})$ is convex. Therefore, $\mathbf{P}_2$ is convex.

For a convex problem, strong duality holds if Slater's condition (or a weaker form) is satisfied, which requires the nonlinear inequality constraints to be strictly feasible [28]. For problem $\mathbf{P}_2$, there is only one nonlinear inequality constraint in (25). Suppose the optimal solution is denoted as $\boldsymbol{f}_C^*(n)$ if the problem is feasible. Based on Lemma 1, we have $h(\boldsymbol{f}^*) = 0$. Then, as long as $h(\boldsymbol{f}^0) < 0$, there must exist at least one strictly feasible non-optimal solution $\boldsymbol{f}_C^\diamond(n) \succeq \boldsymbol{f}_C^*(n)$ which gives a larger objective value while satisfying $\boldsymbol{f}_C^\diamond(n) \preceq \boldsymbol{f}_C^0(n)$ and $h(\boldsymbol{f}^\diamond) < 0$. Therefore, strong duality holds for $\mathbf{P}_2$ if $h(\boldsymbol{f}^0) < 0$. Theorem 1 is proved. Note that, if $h(\boldsymbol{f}^0) = 0$, we can directly obtain the optimal solution as $\boldsymbol{f}_C^*(n) = \boldsymbol{f}_C^0(n)$; if $h(\boldsymbol{f}^0) > 0$, the problem is infeasible.

References

[1] W. Zhuang, Q. Ye, F. Lyu, N. Cheng, and J. Ren, "SDN/NFV-empowered future IoV with enhanced communication, computing, and caching," *Proc. IEEE*, vol. 108, no. 2, pp. 274–291, 2019.

[2] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou, and W. Zhuang, "Holistic network virtualization and pervasive network intelligence for 6G," *IEEE Commun. Surv. Tutor.*, vol. 24, no. 1, pp. 1–30, 2021.

[3] Y. Hui, X. Ma, Z. Su, N. Cheng, Z. Yin, T. H. Luan, and Y. Chen, "Collaboration as a service: Digital-twin-enabled collaborative and distributed autonomous driving," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 18 607–18 619, 2022.

[4] J. Wang, J. Liu, and N. Kato, "Networking and communications in autonomous driving: A survey," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 2, pp. 1243–1274, 2018.

[5] J. Zhang and K. B. Letaief, "Mobile edge intelligence and computing for the Internet of vehicles," *Proc. IEEE*, vol. 108, no. 2, pp. 246–261, 2019.

[6] X. Zheng, S. Li, Y. Li, D. Duan, L. Yang, and X. Cheng, "Confidence evaluation for machine learning schemes in vehicular sensor networks," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 4, pp. 2833–2846, 2023.

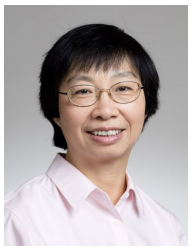
[7] Y. Jia, R. Mao, Y. Sun, S. Zhou, and Z. Niu, "Online V2X scheduling for raw-level cooperative perception," in *Proc. IEEE ICC*, 2022, pp. 309–314.

[8] ——, "Mass: Mobility-aware sensor scheduling of cooperative perception for connected automated driving," *IEEE Trans. Veh. Technol.*, vol. 72, no. 11, pp. 14 962–14 977, 2023.

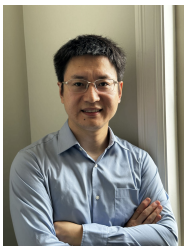
[9] M. K. Abdel-Aziz, C. Perfecto, S. Samarakoon, M. Bennis, and W. Saad, "Vehicular cooperative perception through action branching and federated reinforcement learning," *IEEE Trans. Commun.*, vol. 70, no. 2, pp. 891–903, 2022.

[10] Z. Xiao, J. Shu, H. Jiang, G. Min, H. Chen, and Z. Han, "Perception task offloading with collaborative computation for autonomous driving," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 2, pp. 457–473, 2023.
[11] Y. Sun, J. Xu, and S. Cui, "User association and resource allocation for MEC-enabled IoT networks," *IEEE Trans. Wirel. Commun.*, vol. 21, no. 10, pp. 8051–8062, 2022.
[12] J. Lin, P. Yang, N. Zhang, F. Lyu, X. Chen, and L. Yu, "Low-latency edge video analytics for on-road perception of autonomous ground vehicles," *IEEE Trans. Industr. Inform.*, vol. 19, no. 2, pp. 1512–1523, 2022.
[13] X. Zhang, A. Zhang, J. Sun, X. Zhu, Y. E. Guo, F. Qian, and Z. M. Mao, "EMP: Edge-assisted multi-vehicle perception," in *Proc. 27th Annual International Conf. Mobile Computing and Networking*, 2021, pp. 545–558.
[14] Q. Chen, S. Tang, Q. Yang, and S. Fu, "Cooper: Cooperative perception for connected autonomous vehicles based on 3D point clouds," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2019, pp. 514–524.
[15] H. Qiu, P. Huang, N. Asavisanu, X. Liu, K. Psounis, and R. Govindan, "Autocast: Scalable infrastructure-less cooperative perception for distributed collaborative driving," in *Proc. ACM Int. Conf. Mobile Syst., Appl. and Services (MobiSys)*, 2021, pp. 128–141.
[16] Q. Chen, X. Ma, S. Tang, J. Guo, Q. Yang, and S. Fu, "F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3D point clouds," in *Proc. ACM/IEEE Symp. Edge Comput. (SEC)*, 2019, pp. 88–100.
[17] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng, and R. Urtasun, "V2VNeT: Vehicle-to-vehicle communication for joint perception and prediction," in *Proc. Eur. Conf. Comput. Vision (ECCV)*, 2020, pp. 605–621.
[18] W. Wu, N. Chen, C. Zhou, M. Li, X. Shen, W. Zhuang, and X. Li, "Dynamic RAN slicing for service-oriented vehicular networks via constrained learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2076–2089, 2021.
[19] W. Zhang, Z. He, L. Liu, Z. Jia, Y. Liu, M. Gruteser, D. Raychaudhuri, and Y. Zhang, "Elf: accelerate high-resolution mobile deep vision with content-aware parallel offloading," in *Proc. ACM MobiCom*, 2021, pp. 201–214.
[20] H. Wang, Q. Li, H. Sun, Z. Chen, Y. Hao, J. Peng, Z. Yuan, J. Fu, and Y. Jiang, "Vabus: Edge-cloud real-time video analytics via background understanding and subtraction," *IEEE J. Select. Areas Commun.*, vol. 41, no. 1, pp. 90–106, 2023.
[21] K. Yang, J. Yi, K. Lee, and Y. Lee, "FlexPatch: Fast and accurate object detection for on-device high-resolution live video analytics," in *IEEE Proc. INFOCOM*, 2022, pp. 1898–1907.
[22] S. Teerapittayanon, B. McDanel, and H.-T. Kung, "Distributed deep neural networks over the cloud, the edge and end devices," in *Proc. IEEE ICDCS*, 2017, pp. 328–339.
[23] K. Qu, W. Zhuang, W. Wu, M. Li, X. Shen, X. Li, and W. Shi, "Stochastic cumulative DNN inference with RL-aided adaptive IoT device-edge collaboration," *IEEE Internet Things J.*, vol. 10, no. 20, pp. 18 000–18 015, 2023.
[24] E. Li, L. Zeng, Z. Zhou, and X. Chen, "Edge AI: On-demand accelerating deep neural network inference via edge computing," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 1, pp. 447–457, 2020.
[25] Z. Liu, Q. Lan, and K. Huang, "Resource allocation for multiuser edge inference with batching and early exiting," *IEEE J. Select. Areas Commun.*, vol. 41, no. 4, pp. 1186–1200, 2023.
[26] H. Wang, H. Bao, L. Zeng, K. Luo, and X. Chen, "Real-time high-resolution pedestrian detection in crowded scenes via parallel edge offloading," in *Proc. IEEE ICC*, 2023.
[27] K. Qu, W. Zhuang, Q. Ye, X. Shen, X. Li, and J. Rao, "Dynamic flow migration for embedded services in SDN/NFV-enabled 5G core networks," *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2394–2408, 2020.
[28] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
[29] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. 30th Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 6379–6390.
[30] J. Tian, Q. Liu, H. Zhang, and D. Wu, "Multiagent deep-reinforcement-learning-based resource allocation for heterogeneous QoS guarantees for vehicular networks," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1683–1695, 2022.
[31] Q. Ye, W. Shi, K. Qu, H. He, W. Zhuang, and X. Shen, "Joint RAN slicing and computation offloading for autonomous vehicular networks: A learning-assisted hierarchical approach," *IEEE Open J. Veh. Technol.*, vol. 2, pp. 272–288, 2021.
[32] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *Proc. ICLR*, 2017.
[33] "Simulation of Urban MObility (SUMO) 1.16.0," https://www.eclipse.org/sumo/, 2023, [Online; accessed 8-January-2024].
[34] 3GPP, "Study on evaluation methodology of new Vehicle-to-Everything (V2X) use cases for LTE and NR," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 37.885, 2019, version 15.3.0.

Kaige Qu (S'19−M'21) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2021. She received the B.Sc. degree in communication engineering from Shandong University, Jinan, China, in 2013, and M.Sc. degrees in integrated circuits engineering and electrical engineering from Tsinghua University, Beijing, China, and KU Leuven, Leuven, Belgium, respectively, in 2016. From February 2021 to December 2023, she was a Post-doctoral Fellow and then a Research Associate with the Department of Electrical and Computer Engineering, University of Waterloo. Her research interests include connected and autonomous vehicles, network intelligence, network virtualization, and digital twin assisted network automation.



Weihua Zhuang (M'93−SM'01−F'08) received the B.Sc. and M.Sc. degrees from Dalian Maritime University, China, and the Ph.D. degree from the University of New Brunswick, Canada, all in electrical engineering. Since 1993, she has been a faculty member in the Department of Electrical and Computer Engineering, University of Waterloo, Canada, where she is a University Professor and a Tier I Canada Research Chair in wireless communication networks. Her current research focuses on network architecture, algorithms and protocols, and service provisioning in future communication systems. She is the recipient of Women's Distinguished Career Award in 2021 from IEEE Vehicular Technology Society, R.A. Fessenden Award in 2021 from IEEE Canada, Award of Merit in 2021 from the Federation of Chinese Canadian Professionals (Ontario), and Technical Recognition Award in Ad Hoc and Sensor Networks in 2017 from IEEE Communications Society. Dr. Zhuang is a Fellow of the IEEE, Royal Society of Canada (RSC), Canadian Academy of Engineering (CAE), and Engineering Institute of Canada (EIC). She is the President and an elected member of the Board of Governors (BoG) of the IEEE Vehicular Technology Society. She was the Editor-in-Chief of IEEE Transactions on Vehicular Technology (2007-2013), an editor of IEEE Transactions on Wireless Communications (2005-2009), General Co-Chair of IEEE/CIC International Conference on Communications in China (ICCC) 2021, Technical Program Committee (TPC) Chair/Co-Chair of IEEE Vehicular Technology Conference 2017 Fall and 2016 Fall, TPC Symposia Chair of the IEEE Globecom 2011, and an IEEE Communications Society Distinguished Lecturer (2008-2011).



Qiang Ye (S'15−M'17−SM'22) received the PhD degree in Electrical and Computer Engineering from the University of Waterloo, ON, Canada, in 2016. Since Sept. 2023, he has been an Assistant Professor with the Department of Electrical and Software Engineering, Schulich School of Engineering, University of Calgary, AB, Canada. Before joining UCalgary, he worked as an Assistant Professor with the Department of Computer Science, Memorial University of Newfoundland, NL, Canada from Sept. 2021 to Aug. 2023 and with the Department of Electrical and Computer Engineering and Technology, Minnesota State University, Mankato, USA, from Sept. 2019 to Aug. 2021, respectively. He was with the Department of Electrical and Computer Engineering, University of Waterloo as a Postdoctoral Fellow and then a Research Associate from Dec. 2016 to Sept. 2019. He has published over 70 research articles on top-ranked journals and conference proceedings. He is/was the General, Publication, Program Co-chairs for different reputable international conferences and workshops, e.g., IEEE ICCC'23, CANAI'23, IEEE VTC'22, IEEE INFOCOM'22, and IEEE IPCCC'21. He serves/served as Associate Editors of IEEE Transactions on Vehicular Technology, IEEE Transactions on Cognitive Communications and Networking, IEEE Open Journal of the Communications Society, Peer-to-Peer Networking and Applications, ACM/Wireless Networks, and International Journal of Distributed Sensor Networks. He also serves as the IEEE Vehicular Technology Society (VTS) Regions 1-7 Chapters Coordinator (2022-2023). He is a Senior Member of IEEE.

Wen Wu (S'13-M'20-SM'22) earned the Ph.D. degree in Electrical and Computer Engineering from University of Waterloo, Waterloo, ON, Canada, in 2019. He received the B.E. degree in Information Engineering from South China University of Technology, Guangzhou, China, and the M.E. degree in Electrical Engineering from University of Science and Technology of China, Hefei, China, in 2012 and 2015, respectively. He worked as a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Waterloo. He is currently an Associate Researcher at the Frontier Research Center, Peng Cheng Laboratory, Shenzhen, China. His research interests include 6G networks, network intelligence, and network virtualization.

Xuemin (Sherman) Shen received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on network resource management, wireless network security, Internet of Things, AI for networks, and vehicular networks. Dr. Shen is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Member, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.

Dr. Shen received "West Lake Friendship Award" from Zhejiang Province in 2023, President's Excellence in Research from the University of Waterloo in 2022, the Canadian Award for Telecommunications Research from the Canadian Society of Information Theory (CSIT) in 2021, the R.A. Fessenden Award in 2019 from IEEE, Canada, Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society (ComSoc), and Technical Recognition Award from Wireless Communications Technical Committee (2019) and AHSN Technical Committee (2013). He has also received the Excellent Graduate Supervision Award in 2006 from the University of Waterloo and the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada. Dr. Shen is the President of the IEEE Communications Society. He was the Vice President for Technical & Educational Activities, Vice President for Publications, Member-at-Large on the Board of Governors, Chair of the Distinguished Lecturer Selection Committee, and Member of IEEE Fellow Selection Committee of the ComSoc. Dr. Shen served as the Editor-in-Chief of the IEEE IoT JOURNAL, IEEE Network, and IET Communications.