

From Perception to Computation: Revisiting Delay Optimization for Connected Autonomous Vehicles

TIANEN LIU, Southeast University, Nanjing, China

SHUAI WANG, Southeast University, Nanjing, China

ZHENG DONG, Wayne State University, Detroit, United States

BORUI LI, Southeast University, Nanjing, China

TIAN HE, Southeast University, Nanjing, China

With the development of sensing, wireless communication, and real-time computing technologies, vehicles are gradually becoming more and more intelligent. To provide safe autonomous mobility services, connected autonomous vehicles (CAVs) need to obtain complete information about their environment and process it in real-time to make driving decisions. However, the rapid increase in data volume puts pressure on CAVs to process tasks in real time. This survey analyzes CAVs delay optimization from the perception layer, communication layer, computation layer, and cross-layer. According to different coordination modes, each layer of CAVs is divided, and the problem of delay optimization is classified in fine granularity. This survey will help researchers gain insight into the mechanism of delay optimization on CAVs and highlight the key role of optimized delay in autonomous driving.

CCS Concepts: • **Computer systems organization** → Architectures; Real-time systems; • **Networks** → Network architectures; • **Human-centered computing** → Human computer interaction;

Additional Key Words and Phrases: Connected autonomous vehicle, vehicle edge computing, autonomous driving, delay optimization

ACM Reference Format:

Tianen Liu, Shuai Wang, Zheng Dong, Borui Li, and Tian He. 2025. From Perception to Computation: Revisiting Delay Optimization for Connected Autonomous Vehicles. *ACM Comput. Surv.* 57, 8, Article 200 (March 2025), 45 pages. <https://doi.org/10.1145/3718361>

1 Introduction

With the rapid development of artificial intelligence and sensor technology, traditional cars are gradually moving toward more intelligent **autonomous vehicles (AVs)** [5, 52, 178, 208]. Optimists predict that AVs will be reliable enough and commercially affordable to replace human driving by 2030 [137]. However, due to the limited resources of AVs itself, it is difficult to handle various

This work is partially supported by the National Natural Science Foundation of China (Grant Nos. 62272098, U24B20152, and 62302096).

Authors' Contact Information: Tianen Liu, Southeast University, Nanjing, Jiangsu, China; e-mail: toliutianen@seu.edu.cn; Shuai Wang (Corresponding author), Southeast University, Nanjing, Jiangsu, China; e-mail: shuaiwang@seu.edu.cn; Zheng Dong, Wayne State University, Detroit, MI, USA; e-mail: dong@wayne.edu; Borui Li, Southeast University, Nanjing, Jiangsu, China; e-mail: libr@seu.edu.cn; Tian He, Southeast University, Nanjing, Jiangsu, China; e-mail: tianhe@seu.edu.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 0360-0300/2025/03-ART200

<https://doi.org/10.1145/3718361>

tasks promptly [100, 137]. At the highest level of vehicle automation, **connected autonomous vehicles (CAVs)** will usher in AVs that use wireless communication to connect with transportation infrastructure and vehicle networks to get more resources [2, 70, 101, 172, 214]. Compared to traditional AVs, CAVs cooperatively perceive, communicate and compute with infrastructure or **road-side units (RSUs)** to collect data on traffic flow, road conditions, weather, and air quality in real-time [68]. In addition, CAVs are regarded as an effective solution to alleviate traffic congestion, road safety, and other issues facing the current road traffic system [143].

Connected autonomous vehicles refer to vehicles equipped with intelligent driver assistance systems and communication technologies that interact with the elements of the surrounding driving environment [2, 70]. The application of **vehicle edge computing (VEC)** makes it possible for CAVs to interact with other nodes without barriers, allowing CAVs to expand sensing range, reduce data transmission, and optimize computing latency [197]. The perception system of CAVs shares and fuses information gathered from vehicle sensors and road infrastructure to better understand the surrounding environment [6, 36, 214]. The communication system improves the performance of CAVs to enable collaborative capabilities, namely, collaborative perception and collaborative manipulation [2, 27, 54, 216]. With the assistance of the first two, the computation system completes collaborative computing tasks more efficiently and ultimately brings a safer and more comfortable driving experience [70, 111, 137, 178]. VEC helps to solve problems faced by CAVs in environmental data acquisition, communication, and computation [26, 46, 132, 153], thereby supporting intelligent transportation system services, smart city applications, and urban computing [70].

It is worth mentioning that L2-level autonomous driving generally requires the **end-to-end (E2E)** delay of $20ms \sim 100ms$ for sensing data acquisition, transmission, and data calculation and processing [119, 178]. This is generally considered an acceptable range, and specific latency requirements may vary from application to application. More importantly, CAVs equipped with many high-definition cameras generate 2 GB of data per second, while fast data processing requires 250 trillion computational operations per second [111]. However, performing efficient computation based on such large amounts of data is challenging, because it requires reliable and low-latency methods to accommodate multiple services [184]. How to effectively reduce delay in CAVs is still an open challenge [46, 70, 184, 214].

To comprehensively analyze and understand the delay of CAVs in the VEC environment, it is important to thoroughly study the composition of various structural layers and related problems. For example, in the perception layer, how to improve the speed of object detection; in the communication layer, how to ensure the efficiency of data transmission; in the computation layer, how to choose the appropriate collaborative computing destination. By comprehensively analyzing the delay problem, we can understand the mutual influence and interaction between different layers to make systematic optimization. Only a comprehensive analysis at every layer can find the best integrated solution to improve the performance and efficiency of the entire CAV system. In this survey, we comprehensively analyzed the delay problem of each structural layer of autonomous vehicles, filling in the gap in the existing surveys.

1.1 Scope and Contributions

The section discusses the scope and contributions of this survey. As is shown in Figure 1, the survey is initially selected by searching publications using keywords in recent years on Google Scholar, IEEE Xplore, ACM Digital Library, DBLP, and so on. The types of publications searched consist of two parts: surveys and research papers. Although some research papers are included in the surveys, to cover a comprehensive set of important publications in the field, we search for keywords at each layer of CAVs in a fine-grained manner and select research papers that meet the evaluation metrics. This process continues until no new articles are found. We then analyze the

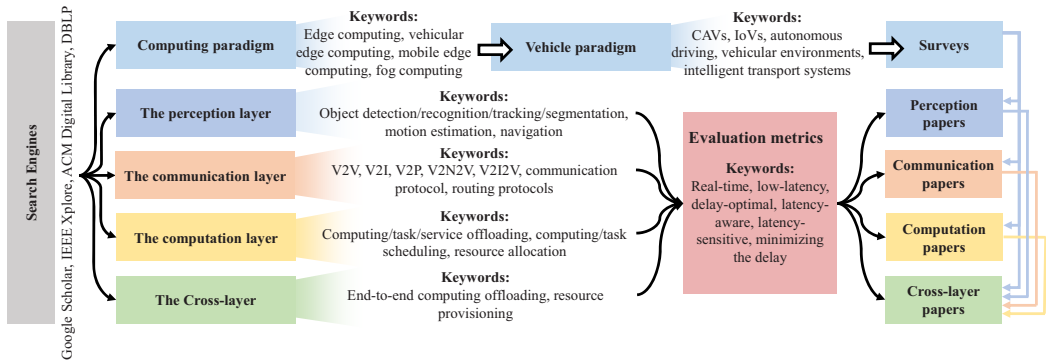


Fig. 1. Road map of systematic surveying.

most relevant and important articles by reading the abstract and the main findings of the paper. In this process, papers that are less relevant to the scope of the investigation are filtered out. The selected papers form the core of this survey, and we have made continuous updates throughout the survey writing process to cover papers published since the beginning of our process.

The scope of this survey. Keywords used to search include two aspects: (1) About computing paradigm. Edge computing, vehicular edge computing, mobile edge computing, fog computing; (2) About vehicle paradigm. CAVs, IoVs, autonomous driving, vehicular environments, intelligent transport systems. We start by looking for a review of computational paradigms on a search engine. Second, we further find a review of vehicle paradigms on this basis. We prioritize review articles from flagship journals due to their significant reference and analytical value to us, such as *ACM Computing Surveys* and *IEEE Communications Surveys & Tutorials*, and so on. To ensure scope focus, this survey does not discuss other related areas as follows: (i) Other paradigms, such as unmanned aerial vehicles [147], robotic [47]; (ii) Referring to the landscape of surveys, including aspects such as vision, challenges, and future work [156].

The scope of research papers. The keywords for each layer of the connected car are shown in Figure 1. We find research papers on search engines by the above keywords. In addition, to avoid keyword matching limitations, we manually searched the publications of the top-tier conferences and journals in the relevant subject areas for the past five years, such as TPDS, TMC, TON, SIGCOMM, MobiCom, CVPR, ICCV, RTSS, and so on. In line with the research theme of this article, we use evaluation criteria to further filter research papers. The evaluation criteria are keywords about delay, such as real-time, low-latency, delay-optimal, and so on. Cross-layer research papers are special and therefore available not only by way of keyword search but also from search results in other layers. What is more, we exclude some early-warning journal articles and keep only high-quality papers.

This survey describes the overall architecture of CAVs and sub-architectures at various layers. In addition, we categorize delay problems in the CAV architecture and review relevant recent solutions. We also shed light on some ideas and point to future research challenges. In summary, the contribution of this work is as follows:

- (1) A comprehensive overview of CAVs, which details the sub-architectures at each layer of CAVs. To the best of our knowledge, this survey is the first comprehensive introduction to the perception layer, communication layer, computation layer, and cross-layer of CAVs;
- (2) We investigate current surveys and evaluate their impact on delay reduction strategies for CAVs. Our survey pioneers a systematic and comprehensive analysis of each layer of CAVs from the delay perspective, synthesizing a holistic approach to delay optimization. This work

Table 1. Comparative Analysis of Surveys on IoV, VEC, and CAV: Coverage of Delay, Architecture, Perception, Communication, Computation, and Application

Survey	Characteristics								
	IoV	VEC	CAV	Delay	Architecture	Perception	Communication	Computation	Application
[36]	✓					✓			
[193]	✓					✓			
[120]	✓					✓	✓		
[35]	✓				✓		✓		
[29]	✓						✓		
[180]	✓						✓		✓
[145]		✓			✓		✓	✓	
[123]		✓			✓			✓	
[15]		✓		✓	✓		✓	✓	
[168]		✓			✓		✓	✓	
[121]		✓					✓		
[222]			✓					✓	✓
[101]			✓				✓		✓
[172]			✓			✓			✓
Ours			✓	✓	✓	✓	✓	✓	

addresses a significant gap in the current literature by providing an in-depth analysis of delay in CAVs.

- (3) To systematically categorize and compare proposed methods with the aim of optimizing delay, we review existing literature and analyze them for insights into how to reduce delay in CAVs. We model and analyze the delay at each layer, identifying areas of concern and neglect that currently affect CAV delay. We conclude with an extensive discussion on unresolved issues and future challenges, providing a clear roadmap for future research.

1.2 Comparison of Existing Surveys

We observe existing definitions and categorical perspectives of CAVs and find that existing reviews do not comprehensively address this area. Current studies mainly investigate CAVs from a specific perspective, such as architecture, **quality of service (QoS)**, **Vehicular Ad Hoc Network (VANET)**, and computing offloading. In summary, these reviews still lack a more comprehensive and detailed study, especially for delay analysis.

First, the subject of our survey is CAV, which is different from the **Internet of Vehicles (IoV)** and VEC. According to the definitions of the three in existing references [68, 70, 145], we determined the unique position of this survey, as shown in Table 1. Next, we divide the architecture of CAVs into the perception layer, communication layer, computation layer, and cross-layer and expand the analysis based on each sub-architecture. Finally, we focus on the delay problem of CAVs, an analysis perspective that is lacking in other surveys.

IoV refers to the connection of vehicles to the Internet and extends the scale, structure, and application of VANET. Cui et al. [36] analyze automatic driving in IoV environment from the perspective of collaborative perception. Combined with the practical application of V2X, the influence of network communication performance on collaborative perception is analyzed. Brumelen et al. [193] focused on summarizing the sensors and partial perception tasks of autonomous vehicles, such as localization and mapping. References [36, 193] focus more on the analysis of the perception layer. References [29, 35, 180] focus on the analysis of the communication layer under the IoV environment. Cheng et al. [29] review routing protocols in the IoV by providing five different routing protocol classifications. Sichitiu et al. [180] analyze existing networking protocols in a bottom-up fashion under inter-vehicle communication systems. In addition to

summarizing the communication protocols used in the IoV environment, Contreras et al. [35] also focus on the IoV architecture. A new seven-layer model architecture for the IoV is proposed, which allows all network components to transparently interconnect and transmit data to the IoV environment. In addition to investigating the perception layer and the communication layer separately, Liu et al. [120] propose novel system classifications and basic performance indicators from the perspective of integrated sensing and communication, and they provide a comprehensive review of the current research progress on their basic limitations.

The IoV focuses on the interaction between vehicles, while the VEC focuses more on the interaction between vehicles and edge nodes. In contrast to IoV, VEC also specifically looks at computing offloading, resource allocation, and resource provisioning for edge servers. References [15, 121, 123, 145, 168] summarize the work related to VEC, including the architecture, communication layer, and computation layer. Considering the computation layer separately, Liu et al. [121] classify task offloading in VEC environment from the perspective of algorithm. In terms of architecture, Meneguet et al. [145] discuss three types of VEC architecture and give an example to illustrate it through hierarchical design. Liu et al. [123] divide the CAV architecture into three layers: vehicular terminal as the user layer, RSU as the mobile edge computing layer, and cloud server as cloud layer, and introduce the three layers. Compared with Reference [123], Boukerche et al. [15] involve the time-delay sensitivity calculation task. Reference [15] classifies scheduling mechanisms as adaptive, social-based, and time-sensitive. The main focus of deadline-sensitive scheduling is to minimize delays to meet the deadlines set for the task.

The application of CAVs cannot be separated from the assistance of VEC. CAVs are regarded as the extension of IoV in VEC environment, which realize intelligent information exchange and sharing between vehicles and people, roads, backgrounds, and so on. References [101, 172, 222] start from the application scenarios of CAVs. Zafar et al. [222] consider the carpooling problem in CAVs environment. Khayatian et al. [101] investigate the intersection management of CAVs. Reyes et al. [172] focus on the interaction between CAVs and vulnerable road users. However, the above studies are limited to considering one specific application scenario of CAVs and do not start from a general architecture.

In a word, the existing surveys still lack a system-level analysis of CAVs. As shown in Table 1, recent investigations related to CAVs focus on specific applications on a single layer. In previous studies, we do not find delay analysis for perception, communication, computation, and cross-layer under CAVs architecture, which adversely affected any discussion of CAVs. Therefore, our survey differs from the existing surveys in that it takes a more comprehensive approach to analyze the delay problem in CAVs.

1.3 Survey Organization

The rest of our work is structured as follows: Section 2 describes the concepts and techniques involved in CAVs. Section 3 introduces the perception layer delay analysis of CAVs. Section 4 introduces the communication layer delay analysis of CAVs. Section 5 introduces the delay analysis at the computation layer. Section 6 describes the cross-layer delay analysis. Section 7 models and analyzes the delay of each layer. Section 8 describes the main challenges and future directions. Finally, Section 9 provides the conclusion of this survey. Figure 2 outlines the organization of the survey.

2 Architecture

We simulate the behavior of CAV in the real world, as shown in Figure 3. CAV relies on its perception for obstacle detection or RSU for collaborative perception. When performing compute-intensive tasks, the CAV performs collaborative computation with other vehicles or edge nodes.

§ 2 Architecture • 2.1 The perception layer • 2.2 The communication layer • 2.3 The computation layer § 3 The perception layer delay analysis in CAVs • 3.1 Single perception ■ 3.1.1 Dynamic object-based ■ 3.1.2 Static object-based ■ 3.1.3 Panoramic object-based ■ 3.1.4 CAV's state-based • 3.2 Collaborative perception ■ 3.2.1 Vehicle-vehicle collaborative perception ■ 3.2.2 Vehicle-edge collaborative perception • 3.3 Conclusion § 4 The communication layer delay analysis in CAVs • 4.1 Direct communication ■ 4.1.1 V2V communication ■ 4.1.2 V2I communication • 4.2 Indirect communication ■ 4.2.1 Complementary V2X-based multi-hop communication ■ 4.2.2 Routing-based multi-hop communication • 4.3 Hybrid communication ■ 4.3.1 V2V or V2I ■ 4.3.2 V2V and V2I • 4.4 Conclusion	§ 5 The computation layer delay analysis in CAVs • 5.1 Vehicle-vehicle collaborative computing ■ 5.1.1 Static scenario ■ 5.1.2 Dynamic scenario • 5.2 Vehicle-edge collaborative computing ■ 5.2.1 Single-edge collaborative computing ■ 5.2.2 Multiple-edges collaborative computing • 5.3 Hybrid computing ■ 5.3.1 Offloading task to vehicle or edge ■ 5.3.2 Offloading task to vehicle and edge • 5.4 Conclusion § 6 Cross-layer delay analysis in CAVs • 6.1 Offloading decision • 6.2 Resource allocation • 6.3 Resource provisioning • 6.4 Conclusion § 7 Delay modeling and discussion • 7.1 The perception layer • 7.2 The communication layer • 7.3 The computation layer § 8 Challenges and future directions § 9 Summary
---	--

From Perception to Computation: Revisiting Delay Optimization for Connected Autonomous Vehicles

Fig. 2. Road map of the survey.

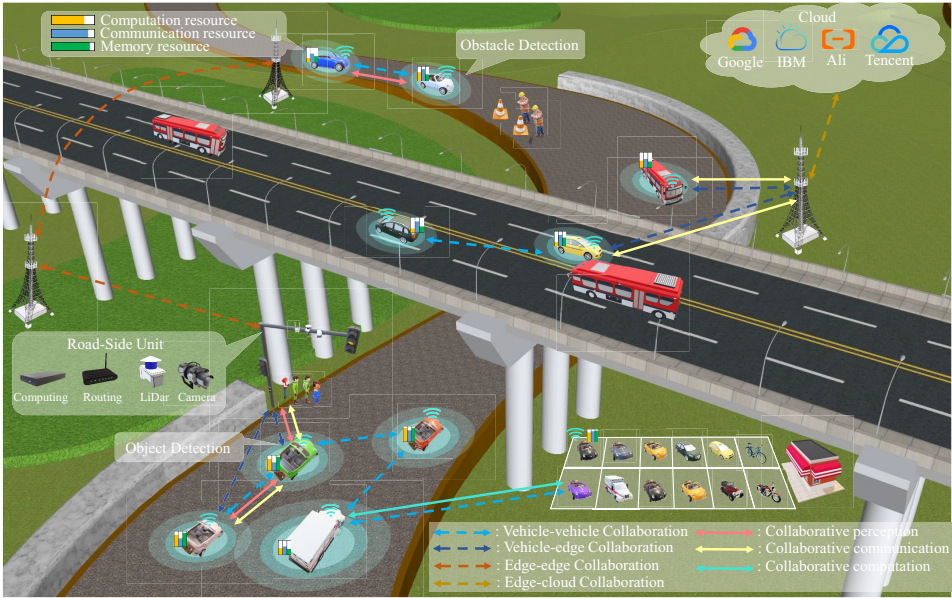


Fig. 3. Autonomous and collaborative capabilities of CAVs in real-world scenarios: self-reliant and cooperative modes in perception, communication, and computation.

Whether it is collaborative perception or collaborative computation, CAV cannot interact with other nodes without the support of the communication layer. Therefore, based on the driving environment of CAV, we divide the overall architecture into a perception layer, a communication layer, and a computation layer.

Table 2. Summary of Acronyms

Acronym	Definition	Acronym	Definition	Acronym	Definition
AVs	Autonomous Vehicles	RL	Reinforcement Learning	V2V	Vehicle-to-Vehicle
mmWave	Millimeter Wave	DRL	Deep Reinforcement Learning	LTE	Long Term Evolution
VEC	Vehicle Edge Computing	SDN	Software Defined Network	V2I	Vehicle-to-Infrastructure
RSU	Road-Side Units	E2E	End-to-End	IoV	Internet of Vehicle
QoS	Quality of Service	V2X	Vehicle-to-Everything	V2P	Vehicle-to-Pedestrian
VANET	Vehicular Ad Hoc Network	MAC	Medium Access Control	MAB	Multi-Armed Bandit
MEC	Mobile Edge Computing	MEC	Multi-access Edge Computing	V2N	Vehicle-to-Network
CNN	Convolutional Neural Network	GPS	Global Positioning System	MDP	Markov Decision Process

Some research [63, 91] also investigates the control layer of the CAVs, the core task of which is to perform the accurate operation and cooperative control of the vehicle according to the output results obtained from the computation layer [196]. Since the VEC environment does not involve the vehicle dynamic models, longitudinal and lateral control of autonomous vehicles, and the vehicle control design, the control layer is outside the scope of this survey. In the perception layer, CAVs decide whether to rely on other devices (other CAVs or edge nodes) to complete the perception task jointly. In the communication layer, CAVs communicate with other high-speed vehicles or edge nodes. In the computation layer, CAVs use computing resources within their communication range for collaborative computing. Given the limitations of space and focus, we chose to focus on these levels to get a more complete understanding of the real-time performance of the entire system. Table 2 lists the acronyms that will be frequently used in the survey.

2.1 The Perception Layer

The perception layer of CAVs is the most basic of autonomous driving technology but an essential part [113, 184, 214]. CAVs obtain the surrounding state data through the perception layer to provide necessary information for subsequent decision-making [6, 36, 213]. According to different collaboration modes of CAVs, we divide the perception layer into single perception and collaborative perception.

Single perception. Single perception refers to capturing environmental data only through its sensors, such as LiDAR [12, 166], **millimeter wave (mmWave)** radar [115, 201, 208], ultrasonic radar [118], camera [166, 199, 213], and so on. Sensors are responsible for collecting information needed by CAVs, such as the CAV's driving state [224, 237] and the surrounding environment [72, 74], to provide a reliable decision-making basis for safe driving. The acquired perception data is processed locally to complete perception tasks such as object detection and tracking, semantic segmentation, vehicle state estimation, position estimation, positioning, and navigation.

Collaborative perception. Single CAV has a blind spot in the sensor field of vision, which makes it impossible to obtain complete information about environment, which may lead to CAVs being unable to detect impending danger [189]. Environmental information acquisition from a single CAV leads to the inability to achieve high-level autonomous driving, so assistance from other intelligent nodes is needed. Collaborative perception technology [127, 204] enables vehicles to obtain sensor information from other edge nodes, which expands the perceiving range of CAVs and increases the integrity of environmental data. The environmental information obtained by edge nodes and local sensing information is interactively fused to improve sensing accuracy, reduce delay, and eliminate sensing blind spots.

2.2 The Communication Layer

V2X technology [27, 155, 191] supports the communication of CAVs, which provides the means of communication between CAVs and other entities in the intelligent transportation system, and

is the basis of cooperation between CAVs and edge nodes [17, 190]. According to the different communication modes of CAVs, we divide the communication layer into direct communication, indirect communication, and hybrid communication.

Direct communication. Compared with cameras or LiDAR, commonly used in autonomous driving, V2X technology gets information through blind spots and over-occluded objects. In addition, V2X technology shares real-time driving status information with other vehicles and facilities [43, 62]. At the same time, V2X is a vehicle sensor communication technology that is not affected by weather, including rain, fog, strong light, and other bad weather [230]. Therefore, V2X technology is widely used in transportation, especially autonomous driving.

V2X integrates four key technologies: V2N [33], V2V [8, 42, 115, 144, 146, 146], V2I [32, 50, 86, 226], and V2P [99, 191]. V2N means vehicle-mounted terminals are connected to the cloud platform through an access network or core network [27, 33]. V2V refers to communication between vehicles through terminals [8, 115, 144]. Vehicle-mounted terminals obtain the speed, location, driving condition, and other information of surrounding vehicles in real-time. V2I refers to vehicles communicating with roadside infrastructure (such as traffic lights [194], traffic cameras [69], and RSUs [50]). V2P means that vulnerable transportation groups (including pedestrians, cyclists, etc.) use user devices (such as mobile phones, laptops, etc.) to communicate with CAVs [99, 191].

Indirect communication. Vehicle applications send messages through direct communication modes of V2N, V2V, V2I, V2P, and indirect communication modes using multi-hop communication [1, 53]. One form of indirect communication is based on complementary V2X multi-hop communication. Complementary V2X multi-hop communication is where CAVs communicate with each other through the network or infrastructure or pedestrians as relay points. V2I2V, V2N2V, and V2P2V are complementary V2X components [219]. In this communication mode, the relay point receives messages from nearby vehicles and organizes and broadcasts messages to surrounding vehicles to support sectional applications.

Another form of indirect communication is routing decision, which uses the least number of hops to obtain the lowest delay [29, 88, 175]. Due to the high-speed movement of CAVs and the complex diversity of urban traffic, the network topology is dynamic and changes frequently [4]. The routing decision principle is to select the best path to forward data. This greatly reduces data transmission delay and provides services for CAVs under the extremely strict delay constraints of autonomous driving applications.

Hybrid communication. Hybrid communication is the coexistence of multiple communication modes in CAV environment. VANET is complex and frequently changing when CAVs are driving on the road at high speed [122, 188]. At this time, communication mode includes both direct communication and indirect communication. Facing a communication environment with high dynamic topology and strict delay requirements, how to select the best communication mode is the key problem to be solved by CAVs.

2.3 The Computation Layer

After acquiring environmental information, CAVs perform real-time tasks such as positioning, object recognition, and tracking to make correct driving decisions. Computationally intensive tasks require a large number of computing resources, and the computing system of CAVs is challenging to complete under the constraint of delay, which requires the assistance of other computing nodes [93, 104, 223]. According to the different objects of computing nodes, we divide the computation layer into vehicle-vehicle cooperative computing, vehicle-edge cooperative computing, and hybrid computing.

Vehicle-vehicle collaborative computing. This collaborative computing paradigm consists of multiple CAVs communicating with each other and sharing computing resources. CAVs use

idle computing resources of different vehicles for collaborative computing through a cooperative strategy between vehicles. For CAVs, other vehicles act as edge servers to collect, process, and store data efficiently [138]. When requiring computation, CAVs offload computationally intensive and delay-sensitive tasks to nearby vehicles to complete the task [223].

Vehicle-edge collaborative computing. Based on vehicle-vehicle collaborative computing, other edge nodes besides CAVs are introduced, such as RSU equipped with edge servers, which significantly expands the computational resources of CAVs [132, 235]. Therefore, vehicle-edge collaborative computing has more communication, computing, and storage resources. Vehicle-edge collaborative computing focuses on a cooperation strategy between vehicles and edge servers, in which vehicles mainly use the computing resources of edge servers for collaborative computing.

Hybrid computing. Collaborative computing technology aims to solve the problem of insufficient computing resources in CAVs by offloading tasks to other edge nodes. The most basic and essential requirement of collaborative computing is to complete the task under delay constraint and return the result to the offloading initiator node to ensure CAVs' safety [75, 225]. Hybrid computing mode includes vehicle-vehicle and vehicle-edge collaborative computing, and even some cloud servers participate in the computing process. However, the use of cloud resources requires long-distance data transmission [79, 149]. Therefore, in the hybrid computing mode, the cloud mainly plays the role of macro-control and optimization of the task offloading process at the edge and also performs delay tolerance tasks.

3 The Perception Layer Delay Analysis in CAVs

The perception layer is a key component of the CAV system, which obtains environmental information and provides key input data for the whole system. This data may be sensing data about surrounding vehicles, pedestrians, road signs, traffic lights, and so on, or it may be state data about the vehicle itself, such as speed, acceleration, attitude, and so on. The task of the perception layer is to collect and predict the vehicle and surrounding environment information, vehicle driving data, and state data through various sensors and terminals.

Research involving the perception layer of CAV focuses on perception algorithms and systems [58, 125, 227]. Researchers improve the performance of perception layer in terms of energy consumption [125], accuracy [102], speed [238], or a balance of speed and accuracy [22, 59] by deploying autonomous driving datasets [16, 57, 126]. We focus on the latter two and exclude algorithms and systems that use other non-autonomous driving datasets, such as sports scenario datasets [112]. According to the different perception forms of CAVs, the perception layer can be divided into single perception and collaborative perception. Single perception refers to the perception ability of a single CAV to acquire and process information about the surrounding environment. Collaborative perception emphasizes information sharing and collaboration between CAVs and other edge nodes to enhance perception ability. As is shown in Tables 3 and 4 (see Appendix A), we summarize the existing papers on the delay analysis of the perception layer of CAVs and conduct fine-grained classification.

3.1 Single Perception

Due to the dynamic and intricate nature of the road environment, CAVs require real-time acquisition of information about various objects in their surroundings. According to the characteristics of different object states in the CAV perception environment, single perception is divided into dynamic object-based perception, static object-based perception, panoramic object-based perception, and CAV's state-based perception.

3.1.1 Dynamic Object-based. The method focuses on the dynamic changes of objects in the surrounding environment. It includes real-time detection and tracking of vehicles, pedestrians,

and other moving objects' location, speed, acceleration, and other parameters. Dynamic object-based perception helps CAVs predict the behavior of other objects, leading to more accurate decisions and planning.

The richer image information and the cheaper price make the camera the most common vision sensor on the CAV. Wu et al. [199] introduce SqueezeDet, a fully convolutional neural network for object detection, to ensure the real-time inference speed of detection tasks. Yang et al. [213] focus on a simple and effective framework for streaming perception. The framework includes dynamic and static flows to capture the moving trend and a basic detection feature for streaming prediction. In addition to improving detection speed, Chen et al. [22] propose a new framework to achieve accuracy-speed tradeoff by using knowledge distillation and hint learning.

The weather environment easily affects cameras, so some papers use LiDAR or mmWave radar to obtain point cloud data to overcome the image of bad weather. Yang et al. [209] utilize the scene from the **BEV (Bird's-Eye View)** and propose a proposal-free, single-stage detector. Simony et al. [182] also use a single-stage YOLOv2 network to infer highly sparse 3D data in real-time. For the shape-aware 2D/3D constraints, Simon et al. [181] present a novel fusion of neural network-based 3D detector and visual semantic segmentation in the context of autonomous driving. Different from Reference [181], Liang et al. [117] present an efficient 3D object detection framework based on 2D convolution, making it possible to have a fast inference speed.

The complex traffic environment promotes the complementary advantages of camera, radar, and LiDAR. Some papers combine the characteristics of these sensors with studying. Rashed et al. [166] propose a robust real-time CNN architecture for moving object detection in low-light conditions. Authors capture motion information and fuse color signals from camera and LiDAR sensors. Liang et al. [116] propose an E2E learnable architecture to reason about 2D and 3D object detection as well as ground estimation and depth completion. Importantly, this approach achieves real-time performance on 2D, 3D, and BEV object detection.

3.1.2 Static Object-based. Many traffic accidents have proved that static object-based detection still needs to be mature enough [221]. Static objects in the CAV environment are divided into two types: the three-dimensional objects in the road, such as road edges, walls, poles, and so on, and the information on road pavement signs, such as lane lines, parking lines, arrows, zebra crossings, and so on.

A few papers focus on reducing the detection delay for road pavement sign information. Garnett et al. [61] propose a novel system to detect candidate regions as maximally stable extremal regions for the recognition of traffic signs. For small-size traffic-signs recognition, Chen et al. [23] use high-level features to construct a better detector head and a data augmentation method to enhance the robustness of the model. At the same time, receptive field blockcross is utilized for capturing contextual information.

General obstacle detection refers to objects on the road, including static objects and dynamic objects. Establishing a real-time detection model for general obstacles on traffic roads is helpful in improving the driving safety of CAVs. Garnett et al. [55] design a single-shot multi-box detector for categorized object detection and pose estimation, with a version of the StixelNet [107] for general obstacles. For reducing E2E real-time obstacle detection delay, Song et al. [183] propose two efficient single-task network architectures and a simple refinement module. This architecture learns and refines semantic segmentation and disparity maps from a stereo RGB pair in a single module. To further improve the detection efficiency, in the real-time detection stage, Prabhakar et al. [161] filter road object detection so the whole system only detects classes corresponding to road objects.

3.1.3 Panoramic Object-based. The CAV system based on a panoramic object is important in object detection, because the system is closer to the reality of autonomous driving [12, 141]. The

panoramic driving perception system has three different tasks: semantic segmentation, instance segmentation, and panoramic segmentation.

The task of semantic segmentation is to assign a category label to each location in the scene (each pixel in the image or each point in the point cloud). Wu et al. [200] formulate the problem of detecting and categorizing instances of interest as a point-wise classification problem. Yu et al. [220] propose bilateral segmentation network-based spatial path and context path to improve speed and accuracy simultaneously. However, adding an extra path to encode spatial information is time-consuming in [220]. To handle these problems, Fan et al. [48] propose a short-term dense concatenate network by removing structure redundancy.

The task of instance segmentation is similar to object detection. But instead of an object box, instance segmentation outputs a category label and an instance label for each point. Perreault et al. [160] use bounding polygons to design instance segmentation framework for detecting road users in dense urban environments. The authors select ground truth vertices by developing a policy and show the importance of a depth map to improve the accuracy of polygons because of the overlap problem.

Panoramic segmentation is a combination of semantic segmentation and instance segmentation. The algorithm needs to distinguish between points on objects (foreground points) and non-objects (background points). Hou et al. [74] propose a new single-shot panoptic segmentation network that leverages dense detection and a global self-attention mechanism to operate in real time. To simplify the pipeline and decrease computation/memory cost caused by a two-stage detection network, Hong et al. [72] propose a one-stage approach, which does not involve a proposal, anchor, or mask head. Li et al. [109] propose a fast and high-performance framework. At the same time, the authors fuse the fine-grained voxel features and 2D BEV features with different receptive fields based on a solid backbone design.

3.1.4 CAV's State-based. Vehicle-guided automation requires accurate and reliable knowledge of the current vehicle position and motion. Based on the perception of its state, the self-operation status of CAVs is described, such as vehicle attitude, self-motion estimation, and positioning.

The ability of CAVs to locate themselves is a crucial capability. The motion estimation itself needs to be performed faster than in real time so the remaining time is spent running other algorithms. Both traditional monocular and stereo approaches compute the motion parameters based on some feature correspondences between successive frames. A method of 3D ego-motion estimation is designed in Reference [85], and the algorithm's parallel feature detection and parallel sparse optical flow are implemented based on GPU. Zindler et al. propose a real-time ego-motion estimation method using LiDAR and an advanced vehicle model based Extended Kalman Filter. Zhang et al. [224] adopt a 3D laser, a camera, and an IMU to estimate ego-motion online. Different from the traditional Kalman Filter adopted in Reference [237], the method employs a sequential, multi-layer processing pipeline, solving for motion from coarse to fine to achieve low-latency ego-motion estimation.

High-precision positioning and navigation technology are still challenging to interact with complex urban environments in real-time. The vehicle navigation system is modeled as a directed graph in Reference [21]. The optimal navigation path from the source point to the destination point is defined as a sequential list of directed links with minimum cost. Different from Reference [21], the real-time navigation system is constructed by formulating the task as a series of decisions in Reference [103]. The motion planning problem of automatic guided vehicles is formulated as a nonlinear optimal control problem in Reference [171]. In addition, a smooth transformation technique is used to obtain the optimal solution.

3.2 Collaborative Perception

The task of collaborative perception is to fuse and process environmental information obtained in various ways so CAVs understand the surrounding environment. The collaborative perception is divided into vehicle-vehicle collaborative perception and vehicle-edge collaborative perception, depending on the object of collaboration. Vehicle-vehicle collaborative perception refers to the cooperative processing of perception data between vehicles through direct communication and information sharing. Vehicle-edge collaborative perception refers to the communication and cooperation between the vehicle and the edge server, and the perception data is uploaded to the edge server for processing and fusion.

3.2.1 Vehicle-vehicle Collaborative Perception. Due to the existence of obstacles and the limitation of sensing range, the sensing ability of a single vehicle cannot meet the requirements. CAVs perceive the surrounding environment more efficiently and robustly by integrating the perceptual content of nearby vehicles through vehicle-vehicle collaborative perception technology. In this mode, vehicles can directly communicate and exchange information and share perception data and results to enhance the perception ability and safety of each vehicle. With the consideration of dynamic perception workloads and channel conditions, Qu et al. [165] propose an adaptive cooperative perception scheme to switch between cooperative perception and stand-alone perception for each CAV pair, achieving high computing efficiency gain.

Allowing vehicles to share raw data in dense urban environments leads to redundant information exchange. For the problem of regional overlap in the process of LiDAR data sharing for multiple vehicles, RECAP [179] adjusts it to a setting with minimal point cloud overlap based on the previous point cloud registration work. In addition, RECAP continuously and opportunistically produces 3D traffic reconstructions from multiple vehicles to minimize error and computation time in multi-way registration. Feature-based data sharing not only meets the training process but also makes use of the small size of features to achieve real-time computation without network congestion. Xiao et al. [203] integrate the information of multiple vehicles into the coordinate system of its vehicle by combining a unified aerial view, including multimedia information based on vision, GPS, and digital map. Chen et al. [26] use feature-level fusion to realize E2E 3D object detection for the first time and support voxel feature fusion and spatial feature fusion. More importantly, the framework dynamically adjusts the feature map size to be transmitted to realize real-time collaborative sensing.

In addition to feature-level sensing data sharing, object-level data sharing is also an important method of collaborative perception. Generally speaking, object-level data sharing has a lower delay than feature-level data sharing. Tsukada et al. [191] propose an open-source experimental and prototype platform called AutoC2X by sharing information about detected objects. In the worst case, AutoC2X delivers co-aware information within 100 ms. Liu et al. [128] propose a real-time dynamic map system to share detected objects and implement an adaptive vehicle scheduling and offloading algorithm to reduce the offloading delay of vehicles.

3.2.2 Vehicle-edge Collaborative Perception. Edge servers have more powerful computing power and storage resources, which can carry out more complex perceptual algorithms and decision processing. Compared with vehicle-vehicle cooperative perception, this model has a larger perception range and more information content. In this mode, the vehicle sends the sensing data to edge server for analysis and receives the results returned by edge server.

Since edge servers have more computing resources than CAVs, vehicles share original sensing data, thus improving object detection accuracy. Zhang et al. [228] propose an edge-assisted sensing system to merge original views from the multiple CAVs to form a complete BEV with a higher

resolution. Yang et al. [216] develop an edge-assisted collaborative perception prototype system. The platform aggregates sensing data from multiple vehicles and edge servers to perform object detection and instance segmentation.

Bandwidth resources often limit the transmission latency of original data. Therefore, the vehicle-edge cooperative perception is further realized by sharing a feature map. Liu et al. [124] design the first collaborative localization and sharing system. In this system, the edge server first merges key frames from the multiple CAVs, generates 3D feature maps, and then sends the feature maps to CAVs. CAVs complete the perception task according to the sharing feature map, which reduces the delay of the feature map generated by CAVs and expands the sensing range of CAVs. For accelerating feature transmission and matching, Sun et al. [185] design a lightweight vehicle re-identification network to adaptively extract the bit features of the perceived vehicle target.

The timeliness of the collected data is essential to achieve precise collaborative perception, and the delay affects the integrity and consistency of sensing data, often quantified in terms of the **Age of Information (AoI)**. To quantify the freshness of transmitted packets, Kaul et al. [97] propose AoI as a performance indicator to measure the latency, since the last packet received at the destination was generated at the source. The concept of AoI is extended in Reference [232] to evaluate the timeliness of status updates by combining time-varying context information. Qin et al. [164] study the AoI optimization problem, taking into account the data sampling of source vehicles and the data forwarding process of multiple information flows across the network. The timeliness of collaborative perception is optimized by carefully designing AoI-sensitive data collection schemes.

3.3 Conclusion

This section summarizes the delay analysis of CAV's perception layer. We divide the perception layer into single perception and collaborative perception according to the different ways of collaboration. Single perception is needed when CAVs are strong enough or when the density of CAVs on the road is not sufficient to support collaborative perception. If the perception ability of CAVs is weak and the density of CAVs on the road allows, then cooperative perception is needed to expand its perception range. If the communication delay between the vehicle and the cooperating vehicle is low and the bandwidth is large, then vehicle-vehicle collaborative perception can be preferred to achieve real-time information sharing and collaborative decision-making. If the communication latency between the vehicle and the edge server is low and the edge server has more computing power, then vehicle-edge collaborative perception can be preferred for more complex perception and decision processing. No matter what kind of collaborative perception method, it aims to improve the vehicle's perception ability and decision-making effect through information sharing and collaboration.

4 The Communication Layer Delay Analysis in CAVs

In CAVs network [155], vehicles connect with many infrastructures, users' equipment, smart devices, and so on. Various services, content, and applications provide information and entertainment content for vehicles and drivers through the communication layer of CAVs. Some papers focus on optimizing network utility, including cost [98], energy efficiency [133], connection stability [92], delay [115, 201], and so on. In this survey, the communication layer delay analysis of CAVs focuses on reducing various delays, so it is not in the scope of this survey to study how to ensure real-time performance by taking the delay as the constraint condition. More importantly, although some papers reduce the delay in the communication layer, they are not related to vehicle paradigm, which is not within the scope of this article.

Depending on the communication method, we classify autonomous vehicles into direct communication, indirect communication, and hybrid communication. Direct communication refers to

how a CAV establishes contact by communicating directly with other entities. Indirect communication means that the CAV communicates with other entities employing multi-hop communication. Hybrid communication is a strategy in which the autonomous vehicle chooses the best mode among various communication modes to achieve low-delay communication. We summarize the delay analysis of the communication layer of CAVs as shown in Tables 5–7 (see Appendix A) and conduct fine-grained classification.

4.1 Direct Communication

This mode means that CAVs establish communication connections through a single communication mode, such as V2V, V2I, V2P, or V2N. The direct communication mode usually has a lower delay than other communication modes. We focus on the first two, because they are more prevalent at the CAV communication layer.

4.1.1 V2V Communication. CAVs use V2V communication to send their driving information, such as speed, position, and heading information, to other CAVs in the communication range. The surrounding vehicles use the received data to build a real-time dynamic map to avoid traffic accidents and effectively alleviate congestion.

The first step in optimizing the delay of CAVs communication is to start with the data itself. Similarly, optimizing the transmitted data packets further reduces latency. Wu et al. [201] use ultra-high-speed directional mmWave communications as V2V links, increasing the amount of data transmitted each time as much as possible to improve the content transmission rate. To further reduce the transmission delay, an iterative algorithm is proposed in Reference [211] to obtain the optimal frame design. Based on the lower bound of signal-to-interference-plus-noise ratio quality, Yang et al. [211] study the optimal frame design to reduce overhead for the short frame.

CAVs use wireless communication to exchange sensor data, however, traditional technologies still cannot support the gigabit-per-second data rate that is required for raw sensor data exchange among autonomous vehicles for emerging applications [18]. Choi et al. [31] prove that mmWave communication is the only feasible method for high-bandwidth connected vehicles. Aiming at the high overhead of mmWave beam training, Graff et al. [60] use passive radar arrays to sense automotive radar transmissions from multiple vehicles on the road. Kamari et al. [94] create a system of 3D reflection profiles from street view images. The system reduces beamforming overhead through material identification, environment-driven ray tracing, and beam search methods.

Millimeter-wave links, 6G mid-high frequency bands, and even terahertz transmissions are severely affected by physical obstacles, which are prone to congestion and reduce system reliability and delay performance. For blockage events in V2V communication, Tunc et al. [192] design a continuous-time Markov chain model to characterize reliability and delay performance. Aiming at the difficulties of beam alignment and routing stability, Rasheed et al. [167] propose a 3D-based position detection scheme for beam alignment to achieve trusted data transmission. Benelmir et al. [11] propose a new beam alignment method to select the optimal starting beam angle of the mmWave base station and further to transmit data directly to the moving vehicle.

The high mobility of the vehicle environment and the dynamic nature of network topology further lead to the continuous switching of communication links during vehicle driving. Bai et al. [8] propose a framework to monitor periodic beacons sent by other distributed and cooperating vehicles to meet deadlines. Bai et al. [8] assume that terminal devices operate under the same operator, but this assumption is unrealistic. Martin et al. [144] propose a fast cross-operator handoff mechanism to support V2V communication and reduce operator communication delays.

Most of the above methods assume that network resources are sufficient and there is no resource competition between CAVs. In reality, however, RSUs cannot process services in real-time

due to resource constraints. Ashraf et al. [7] transform the network transmission minimization problem into the average queue length problem by using Markov inequality in resource allocation. Huang et al. [80] translate the optimal resource allocation scheme into an optimal vehicle selection problem to determine the V2V link and minimize the total delay.

4.1.2 V2I Communication. Due to the scalability issues of VANET, V2I communication will still be required for the safe and reliable operation of autonomous vehicles in the future. V2I communication takes place between the vehicle and edge servers such as base stations and RSUs. The **ETC (electronic toll collection)** system is the most common systems-based V2I communication [99]. ETC systems utilize V2I communications to perform electronic payment transactions when a vehicle passes through a toll booth without requiring the vehicle to slow down.

When a CAV shuttles between multiple RSUs, seamless switching is required to achieve a continuous link and provide uninterrupted service. Jarupan et al. [86] use a delay estimation model to determine the best available path, but applications generally cannot tolerate long access delays, such as traffic conditions and collision warnings. According to this problem, Chung et al. [32] propose an access scheme based on contention-free access and pre-ordered transmission opportunities to improve V2I communication and handover. Also, for applications with high reliability and low latency, Bohm et al. [13] define real-time data as data traffic that depends on the deadline and propose a deterministic MAC method for real-time communication based on V2I. COPILOT [162] addresses the problem that performance in the millimeter-wave band can be severely degraded without line of sight by actively switching to identify the most appropriate road-side units. COPILOT yields improvement in latency and throughput compared to traditional reactive handoff for mmWave networks.

Resource allocation in V2I is slightly different from that in V2V. In V2V scenarios, resources are allocated to RSUs to reduce the latency of V2V communication links. In V2I scenarios, resource allocation focuses on the reliability and delay of V2I networks. Zhang et al. [226] consider potential resource requests in V2I scenarios and focus on wireless resource allocation between vehicle users and RSUs. Reference [226] considers the queueing delay of the MAC layer but does not consider the transmission delay of the physical layer. Fang et al. [50] propose a two-stage radio resource allocation problem for physical layer delay guarantee to optimize the worst-case transmission delay.

4.2 Indirect Communication

4.2.1 Complementary V2X-based Multi-hop Communication. Compared to previous cellular technologies, more powerful 5G capabilities have raised expectations about the possibility of supporting V2X services through **vehicle-to-network-to-vehicle (V2N2V)** communications. In recent years, the use of V2N2V connections rather than direct V2V communications or side chains to support critical services has been experimentally demonstrated in terms of delay performance [146, 174].

The model proposed in Reference [34] is the first E2E 5G delay model. This model quantifies the E2E delay experienced by V2N and V2N2V communications in different 5G network deployments. Authors identified which 5G network deployments and configurations better meet the latency and reliability requirements of V2X services to enable connectivity and autonomous driving. Coll et al. [33] analyze the use of 5G networks and **multi-access edge computing (MEC)** to support V2X services through V2N2V in the deployment of multi-mobile network operators. Analysis shows that the federation of MECs reduces the E2E latency for network deployments and is independent of network load. To address the performance degradation caused by frequent switching in highly mobile V2X networks, Li et al. [114] propose novel frame structures that incorporate **Integrated**

Sensing And Communications (ISAC) signals for three crucial stages in the NR-V2X system: initial access, connected mode, and beam failure and recovery. The structures enable faster beam establishment during initial access and accelerate beam fault detection and recovery.

In existing literature and 3GPP standards, reliability and latency are commonly used to evaluate the performance of in-vehicle communications. However, the requirements of these two metrics do not strike a good balance, and reliability performance can be improved at the cost of increased latency and vice versa. AoI [157] is defined as a measure of state freshness from any other vehicle, a performance that combines reliability and latency. Peng et al. [157] propose a resource allocation scheme based on AoI optimization. The scheme significantly reduces packet collisions and improves the stability of vehicle collaboration. Chen et al. [25] establish an empirical model for AoI in connected vehicles by measuring commercial LTE networks for three months and study the impact of major factors on AoI performance.

Previous research has shown that there can be conflicts between reliability and latency performance of in-vehicle networks. Therefore, it is necessary to study the reliability and delay joint model of vehicle networks. Ge et al. [56] propose a joint reliability and latency function by using Monte Carlo simulations to evaluate the combined effects of reliability and latency in 5G autonomous vehicle networks. Ashraf et al. [7] prove that when the finite-size queue buffer is overloaded, data arrival results in packet drops that incur unreliable communication. Yang et al. [212] discuss the solution for building **URLLC (ultra-reliable and low-latency communications)** VNET at the physical and media access control layers, including an optimization framework for latency and reliability. Guo et al. [67] obtain steady-state reliability and delay expressions for each possible spectral reuse pair of V2N links and V2V links based on queueing analysis.

4.2.2 Routing-based Multi-hop Communication. In the complex urban environment, the communication link of CAVs often leads to route failure due to their high mobility, the distribution of vehicle nodes, the complexity of traffic roads, and the existence of obstacles between vehicles. Compared with traditional wireless sensor networks, network topology in the VANET is dynamic and frequently changes. Therefore, the traditional routing scheme is unsuitable for CAVs [95, 105].

The location-aware routing protocol is considered a very promising routing strategy for inter-vehicular communication systems. Location-aware routing protocol covers vehicle location and traffic road location information. Seet et al. [175] propose anchor-based street- and traffic-aware routing. However, location information used in the routing protocol is not real-time. Jerbi et al. [88] propose an improved greedy traffic-aware routing protocol considering real-time road traffic changes and urban environment characteristics. In addition to considering the location information of traffic roads, Jarupan et al. [87] focus on the location information of vehicles based on GPS to estimate latency and select the best available path. References [30, 96] also consider the location information of traffic roads and vehicles. In Reference [96], vehicle location prediction and electronic map information are used to improve the efficiency of routing protocol in the vehicle network. Cheng et al. [30] propose a hybrid geographic routing solution. By using delay-tolerant forwarding in sparse networks, the packet delivery rate and the routing delay are greatly improved compared to the benchmark algorithm.

4.3 Hybrid Communication

In the real world, the communication network of CAVs is enormous and complex. Vehicles are often faced with a choice of communication mode during driving, that is, they choose to connect to other vehicles, RSUs, or both.

4.3.1 V2V or V2I. Some researchers divide complex and variable network links according to scenarios or stages. In a given scenario or phase, a single communication mode is considered.

The multi-link selection problem is transformed into a multi-scenario or multi-stage single-link selection problem.

The scheduling research of cooperative data dissemination under a hybrid V2I and V2V communication environment is proposed for the first time [122]. This study divides a scheduling cycle into three stages: V2V, V2I, V2I, or V2V. CAVs select a stage based on the scheduling time. Liu et al. [122] consider the hybrid communication mode of V2V and V2I but ignore V2N communication. Zhou et al. [233] study a method that enables each vehicle to determine a transmission mode from V2N, V2I, and V2V. The problem with transmission mode selection is maximizing the total capacity of transmission modes to reduce transmission delay. Yaqoob et al. [218] design a distributed method to continuously update vehicle status through direct or intermediate node communication to minimize congestion and message delivery delay. Huang et al. [80] avoid message delivery delay caused by data congestion by caching service content at the network edge.

4.3.2 V2V and V2I. Vehicles communicate with others not only through one or more wireless hops but also through V2I links. These V2I links help improve communication stability and also reduce driver reaction time. Therefore, through the interconnection of V2V and V2I networks, the efficiency and security of vehicle networks are significantly improved. To explore the potential of infrastructure-assisted autonomous driving, Soar [177] is designed to support autonomous driving systems by combining a new communication architecture and a hierarchical task management framework. The communication architecture comprises a bi-directional multi-hop I2I network and a downlink I2V broadcast service, realizing high-bandwidth and reliable data transmission.

Resource allocation in a hybrid V2V and V2I communication environment aims to maximize the sum ergodic capacity of V2I links and ensure the reliability and delay of V2V links. Guo et al. [65, 66] focus on the spectrum and power allocation by considering each possible spectrum. Yang et al. [210] research jointly communication mode selection and resource block allocation to ensure low-latency communication requirements of V2V links. Gu et al. [62] propose an intelligent algorithm at the reconfigurable intelligent surface to solve the sum ergodic capacity optimization problem of vehicles in V2I communication.

By studying resource scheduling and allocation problems in hybrid communication mode, the access times of CAVs and the additional delay in the large-scale exchange of control information are reduced. Di et al. [43] use non-orthogonal multiple access to reduce access latency and improve packet reception probability. Unlike simply optimizing latency performance, Huang et al. [81] adopt a vehicle-centered method to avoid the large-scale exchange of control information and signaling and reduce its impact on latency.

Due to complex dynamic influencing factors such as RSU coverage, traffic density, and vehicle travel distance during uplink, it remains a challenge for vehicles to select destination uplink nodes among RSUs deployed in the VANET. Therefore, in V2V and V2I hybrid communication modes, selecting the optimal communication link to achieve seamless handoff has been widely studied. Through time delay evaluation in different scenarios, Dey et al. [42] prove the potential of V2V and V2I composed of Wi-Fi, DSRC, and LTE. By predicting the mobility of CAVs in Reference [188], RSU independently makes different routing decisions based on transmission probability and average delay of each vehicle to minimize the overall vehicle service delay. To reduce the delay of dynamic routing path selection, Di et al. [44] use the coordination capability of SDN to determine the optimal V2V multi-hop path and offload the traffic volume of V2I2V to V2V communication. Similarly, to solve the high delay and packet loss rate caused by the dynamic topology of vehicles, Guo et al. [64] establish the benchmark for selecting RSUs in V2V and V2R hybrid communication environment to minimize the delay.

4.4 Conclusion

The communication distance of direct communication is relatively short, and this near-field communication characteristic makes the communication delay low, which can realize real-time information exchange and response. Based on indirect communication, the CAV first accesses the network and then connects to other entities. For example, the indirect communication mode requires V2N2V communication rather than direct V2V communication. The indirect communication mode usually needs to consider the selection of the appropriate routing path in the multi-hop network to further reduce the communication delay. V2V communications are able to wirelessly exchange information about the speed and position of surrounding vehicles, but such interconnection is achieved when vehicles communicate with each other. If only 2 out of 10 vehicles are equipped with V2V technology and the others are not, then it will not be possible to build a communication network, which is of no practical significance [56]. In the real world, CAVs is often in the area where multiple communication modes coexist. In this case, CAVs select the optimal communication mode for low-latency communication.

5 The Computation Layer Delay Analysis in CAVs

The computation layer is responsible for receiving, processing, and analyzing large amounts of data collected from various sensors. By using algorithms and models, the computation layer can extract useful information and features and perform tasks such as scene understanding, object recognition, and trajectory prediction. Based on these calculations, the computation layer can make precise decisions, including vehicle control, path planning, obstacle avoidance, and so on, to ensure safe and efficient driving.

In this section, we focus on analyzing the computation delay of CAVs. We divide the computation layer into vehicle-vehicle cooperative computing, vehicle-edge collaborative computing, and hybrid computing according to different cooperative computing modes. Hybrid computing includes the first two. To be different from collaborative perception, the edge nodes of this layer do not participate in the perception process. The delay analysis of edge nodes participating in the perception process is introduced in Section 3. We summarize the delay analysis of the computation layer as is shown in Tables 8–10 (see Appendix A) and perform fine-grained classification.

5.1 Vehicle-vehicle Collaborative Computing

CAVs-based road network is an extensive transmission system that leverages the mobility of existing vehicles equipped with wireless and storage capabilities and a series of offloading points for vehicle-vehicle collaborative computing. Vehicle-vehicle collaborative computing includes two scenarios: static scenario and dynamic scenario.

5.1.1 Static Scenario. The offloading site of the static scenario is a data storage device located in the usual parking place, such as a parking space located on the street, a large dedicated parking lot, a gas station, and so on. These vehicles stay parked there for long periods to maintain adequate computing power for CAVs.

The main challenges faced by the offloading system are selecting an offloading path that meets data transmission performance requirements and configuring the sequence of offloading points involved in the transmission process. In Reference [9], the SDN paradigm is used to adapt to the computing offloading requirements of each CAV, thereby reducing E2E delay during computing offloading. When a large number of computationally intensive tasks arrive, References [153] and [140] use idle computing resources in the parking lot to realize the computing offloading task. Specifically, Nguyen et al. [153] integrate a container orchestration into the computing offloading framework to run a sufficient number of task replicas. Ma et al. [140] design a task

scheduling algorithm to reasonably allocate parked vehicles and perform computing offloading tasks.

The above studies consider one-hop communication between vehicles. Chen et al. [19] select adjacent vehicles within the range of k -hop wireless communication as candidate vehicles for offloading nodes. The corresponding selection relationship between the task set and the candidate vehicle set is modeled as a many-to-one generalized allocation model to minimize the delay of CAVs in completing tasks.

5.1.2 Dynamic Scenario. The dynamic, random, and time-varying characteristics of vehicle networks make the task offloading of CAVs more complicated. In dynamic scenarios, the offloading site is a high-speed service vehicle aiming to provide efficient computing services in a highly dynamic offloading environment.

Due to the high-speed dynamic nature of service vehicles, it is more challenging to compute offloading difficulty compared with the static scenario. Feng et al. [51] use a decentralized scheduling scheme based on the information of nearby vehicles (including GPS information) to improve the computing ability of vehicles. In addition to using the information of adjacent vehicles, Sun et al. [186] realize the task offloading scheme of adaptive learning based on multi-armed bandit theory. Cui et al. [37] also implement self-updating of offloading strategy based on deep reinforcement learning to reduce the execution delay of tasks. References [37, 51, 186] assume that tasks are mutually independent, such as speech recognition, GPS positioning services, and so on. Hu et al. [77] propose a greedy task scheduling algorithm to minimize total job delay by taking into account dependence between computing tasks.

The above work optimizes the inference delay, and some researchers consider the problem of large training delay in vehicle-vehicle collaborative computing. References [108, 129] conduct data sharing and collaborative analysis through federated learning methods to reduce training delay. Li et al. [108] consider whether vehicles participate in training within each time slice to determine a joint optimization scheme. Liu et al. [129] apply gradient compression and mini-batch federated stochastic gradient descent to reduce local gradient transmission and computation and improve learning efficiency.

5.2 Vehicle-edge Collaborative Computing

In the vehicle-edge collaborative computing scenario, speeding vehicles offload computationally intensive tasks to edge servers with a fixed location. In CAV environment, a fixed edge server is an edge device equipped with computing, storage, and communication resources on the side of the road, such as a base station, RSU, and intelligent street lamp. Vehicle-edge collaborative computing is divided into single-edge collaborative computing and multiple-edge collaborative computing according to the number of edge nodes.

5.2.1 Single-edge Collaborative Computing. Only one edge server is used in the collaborative computing process, that is, a vehicle and an edge server are used to complete collaborative computing tasks. Within the communication range of the edge server, vehicles offload computing tasks to the edge server to realize vehicle-edge collaborative computing.

Although CAVs offload computing tasks to an edge server, edge nodes still have the difficulty of being resource-constrained. According to the limitation of network resources and the dynamic resource availability change, an efficient resource allocation scheme is designed to meet the E2E service delay of delay-sensitive vehicle applications in Reference [104]. Given the frequent switching of multi-edge servers caused by vehicle movement, Zhan et al. [223] assume that the coverage of different RSUs does not overlap. The edge server performs offloading tasks from vehicles within its communication range. The above studies minimize computation delay from the

perspective of computing offloading or resource allocation. Luo et al. [138] consider joint computing offloading decisions and allocating resources to minimize delay and cost under the constraints of resources.

5.2.2 Multiple-edge Collaborative Computing. In the real world, telecom operators typically deploy multiple edge servers in urban core locations. The coverage of different edge servers often overlaps, and CAVs appear in the cross zone. When offloading tasks, CAVs offload to multiple edge servers to achieve lower offloading delay.

When CAVs are within communication coverage of multiple edge servers, CAVs figure out which edge server to offload computing tasks to. Kadhimi et al. [93] use SDN to assign computing tasks to optimal edge servers. Based on local and global load balancing technology, Kadhimi et al. [93] optimize the resource utilization of edge nodes to meet the deadline of tasks. Once the edge server receives the computation task of vehicle offloading, it may partially or entirely allocate the computation workload to another edge server. Li et al. [111] propose a task division and scheduling scheme that distributes computation workload among edge servers and coordinates the execution sequence of tasks assigned to servers. Frequent switching from optimal mapping between multiple vehicles and multiple RSUs is a challenge to meet the delay of computing tasks. Martins et al. [45] model the optimal mapping problem as the uncapacitated facility location problem and propose a biased-randomized algorithm to provide a real-time solution to the problem.

5.3 Hybrid Computing

With the popularization of CAVs and the wisdom of infrastructure, autonomous driving is often within the communication range of multiple offloading nodes. It is a complex problem to select the appropriate offloading node to minimize computation delay.

5.3.1 Offloading Task to Vehicle or Edge. Since the offloading process is quasi-static, CAVs usually need to select an offloading node within a period [138]. The offloading nodes are divided into fixed-edge servers (such as RSUs) and vehicle-edge servers (such as fixed parking lots and moving vehicles).

For reducing offloading nodes' switching delay, Zhou et al. [235] consider vehicle-vehicle offloading with similar moving direction and speed. Zhou et al. [235] consider computing offloading but do not consider the challenge of resource limitation. References [89, 106, 132] jointly consider how to select offloading nodes to reduce delay under computing offloading and resource allocation. Specifically, Liu et al. [132] express the offloading node selection problem as a semi-Markov process and obtained the optimal policy by Q-learning and DRL. Lee et al. [106] consider four offloading nodes: local vehicle fog node, fixed parking lot, slow-moving vehicle node, and remote cloud server. Service delays are minimized by allocating limited fog resources to different vehicle applications with delay requirements.

In addition, Jiang et al. [89] consider resource allocation while offloading computing tasks. When tasks are offloaded to other vehicles, more communication resources are allocated. When tasks are offloaded to edge nodes, more computing resources are allocated. For multi-task offloading scenarios with different parameters, such as task topology, resource requirement and transmission/computation capability, Dai et al. [40] develop a seq2seq-based meta reinforcement learning algorithm to minimize task execution time under different offloading scenarios. MESON [231] considers the data dependence between tasks in offloading, while optimizing the average response time and average energy consumption of the system, and proposes mobility-aware dependent task offloading scheme for urban VEC and develop a DRL-based algorithm to train the offloading strategy.

5.3.2 Offloading Task to Vehicle and Edge. A single computing offloading model is often ineffective in the face of many computationally intensive tasks. Safe and reliable autonomous driving requires effective interaction and collaboration between vehicles, as well as between vehicles and RSUs.

Due to the mobility of the vehicle, the content packets are sent back to the requesting node through different paths of the VANET. As a relay node for content packet transmission, the offloading node is frequently accessed, which easily causes access delay. Yao et al. [217] take mobile vehicles as cache nodes and access delay is reduced by pre-caching popular content on a set of mobile nodes. In addition to moving vehicles that serve as cache nodes, Zhang et al. [225] utilize the synergy between mobile edge computing, multiple base stations, and vehicle caching to ensure low latency performance.

Using a pre-cached content package to reduce access delay effectively improves computing offloading efficiency. Hou et al. [75] use mobile vehicles and RSU infrastructure as partial offloading nodes to provide low-latency computing services cooperatively. This strategy considers partial offloading, task allocation, and reprocessing mechanisms comprehensively, and a low-complexity fault-tolerant particle swarm optimization algorithm is designed. Fan et al. [49] consider multiple devices covered by edge servers, parked vehicles, and moving vehicles as offloading nodes to reduce the total priority-weighted task processing delay of devices.

5.4 Conclusion

Through vehicle-vehicle collaborative computing, resource-constrained CAVs can offload computation-intensive tasks to other vehicles to meet task deadlines. Vehicle-vehicle collaborative computing is more effective when there is a high enough density of autonomous vehicles on the road and the vehicles are within communication range. Vehicle-edge collaborative computing is more worthwhile to try when the service vehicle is insufficient to support collaborative computing tasks and the CAV is within the coverage of the edge server. The hybrid computing mode is more robust than the first two computing modes. When a CAV is within communication range of multiple offload nodes, it can choose to transfer computing tasks to other vehicles, edge servers, or both.

6 Cross-layer Delay Analysis in CAVs

In Sections 3–5, we analyze the delay of the perception layer, communication layer, and computation layer in CAVs, respectively. However, if CAVs adopt a collaborative approach to complete a task, then optimizing the delay of any layer alone is not enough. There is a tight coupling between each layer, and optimizing latency for only one layer can cause performance bottlenecks in other parts of the system.

In this section, we mainly discuss the cross-layer optimization of the perception layer, communication layer, and computation layer. Cross-layer optimization involves different aspects of edge computing resource scheduling and is particularly suitable for applications such as CAVs. There are three directions of edge computing resource scheduling that are involved in CAV application scenarios: offloading decision, resource allocation, and resource provisioning. Therefore, we analyze the cross-layer delay from the above three perspectives. We summarize the delay analysis of CAVs across the communication layer and computation layer as shown in Tables 11–13 (see Appendix A) and carry out fine-grained classification.

6.1 Offloading Decision

The offloading decision studies whether CAVs choose to offload and how many tasks to offload. The offloading decision is divided into 0/1 offloading and partial offloading. In the computation

layer, we have analyzed the delay of 0/1 offloading. In this part, we focus on partial offloading. Partial offloading means that tasks are split, with one part being processed locally and the other part being offloaded elsewhere.

References [131, 139, 198] use game theory to find the optimal offloading decision. Liu et al. [131] describe the multi-vehicle computing offloading problem as a multi-user game problem and prove that computing offloading game reaches Nash equilibrium. Considering the resource competition of edge servers, Wang et al. [198] propose a multi-user non-cooperative computing offloading game and prove that the strategy converges to a unique stable equilibrium under certain conditions. Luo et al. [139] propose a self-learning-based distributed computing offloading scheme to minimize the joint cost of delay and offloading and prove the existence of Nash equilibrium. In addition to using game theory to find Nash equilibrium for offloading decisions, reinforcement learning also is used. Chen et al. [20] propose a distributed computing offloading strategy based on a deep Q-learning network to find the best offloading method to minimize the execution time of composite tasks. Zhu et al. [236] design the optimal offloading decision of the MEC server based on multi-agent deep reinforcement learning to minimize the total task processing delay in the long term.

In applications where multiple entities interact, keeping latency low is critical to the overall reliability of the system. If there is a significant delay in communication or computation between different entities, then the reliability of the CAV system can be compromised. Zhou et al. [234] explore the most reliable way to collaborate in vehicle-road systems. By developing a reliability-oriented collaborative computational optimization analysis framework, vehicle users can optimally divide, offload, and execute application workloads between vehicles and infrastructure. The framework combines modeling communication reliability (V2V and V2I) and computational reliability (local execution) to formulate constraint optimization problems. Optimization aims to divide workloads to maximize the overall reliability of vehicle collaborative computing. In addition, to meet the computation deadlines during offloading, Ndikuman et al. [152] develop a collaboration space of edge clouds to process data as closely as possible to the vehicles and design an Age of Processing-aware algorithm to compute the solution in near real-time.

6.2 Resource Allocation

Resource allocation studies how to reasonably allocate resource offloading and task processing in edge servers during computing offloading. The efficient resource allocation scheme improves the network throughput and resource utilization of the VEC system and further reduces the delay of CAVs.

References [149] and [79] optimize communication resources from the transmitted data itself to reduce data transmission scheduling. To cope with highly random and noisy network conditions, Naeem et al. [149] design an intelligent agent that schedules packet transmission for ultra-reliable low-latency communication. Huang et al. [79] use graph-theoretic method and heuristic algorithm to find low-complexity approximate optimal solutions, which greatly reduce the amount of data transmitted to the server.

Some studies consider offloading decisions from the perspective of resource allocation. Dai et al. [41] model the problem of CAVs selection server as a binary decision. Through the joint optimization of offloading decisions and computing resource allocation, the optimal policy of system utility maximization is obtained. The task offloading problem is divided into two subproblems in Reference [24]: task placement subproblem and resource allocation subproblem, and mixed-integer nonlinear programming is used to minimize task offloading delay. Ibrar et al. [83] design software-defined fog computing for time-critical CAV applications to attain adaptive resource scaling and minimize E2E delay.

6.3 Resource Provisioning

Because the request load of CAV varies over time, edge nodes constantly experience workload fluctuations. Resource provisioning mainly refers to how to provide enough computing and storage resources based on the dynamic needs of users and system load. Resource provisioning includes task allocation and resource placement. Task allocation mainly studies how to allocate resources to tasks to achieve ideal QoS with the goal of task delay when edge resources are limited. Resource placement focuses on how edge services or edge servers are placed. The location and number of edge services critically impact the cost and latency of CAV networks.

Due to the heterogeneity of computing resources, high mobility of vehicles, and uneven distribution of workload, it is a complex problem to select the best edge nodes and improve the resource utilization of edge nodes. Dai et al. [39] propose a MAB learning algorithm to realize workload balance among MEC servers by learning real-time workload distribution online. Xu et al. [207] design an adaptive computing offloading method to optimize the task offloading delay of the edge system by migrating the offloading position. Parked vehicles are selected as an auxiliary computing node in Reference [158]. Based on the prediction probability, the availability of parked vehicles is evaluated to select the location of the parked node. Peng et al. [158] add transfer delay and computing delay into cost function and use Stackelberg game and a ternary search-based algorithm to minimize the cost.

References [73, 154, 163] take dynamic resource provisioning from the perspective of resource providers and reduce content transfer latency by caching services at edge nodes. In Reference [73], service content is cached in the RSU to provide a fast caching service for mobile vehicles of content providers. In Reference [163], collaborative edge caching between macro cell stations, RSUs, and intelligent vehicles minimizes content access cost and reduces content transmission delay. Ning et al. [154] solve a mixed integer non-linear programming problem to minimize the total network delay, including communication delay, computing delay, network congestion delay, and content download delay of users. To solve the problem that frequent changes in the topology of vehicle networks increase the communication link switching delay, Jin et al. [90] propose an adaptive cooperative caching strategy and described the optimization problem of minimizing the average transmission delay.

6.4 Conclusion

Computationally intensive tasks are entirely or partially offloaded to places with sufficient resources for processing, because CAVs are mostly resource-constrained. If CAVs decide to offload partially, then the delay model includes local computation delay, transmission delay, and computation delay at offloading destination. If CAVs decide to offload all, then the delay model includes transmission delay and computation delay of the offloading destination. The resource allocation problem is often modeled jointly with the offloading decision problem. By considering communication, computing, and storage resources, a better offloading decision is achieved to reduce the system delay. Resource provisioning to CAVs is greater than the workload required, and edge nodes may pay unnecessary costs. Conversely, if resource provisioning to CAVs is smaller than the workload needed, then it results in poor QoS or even failure to complete tasks before the deadline. Therefore, designing a reasonable resource provisioning scheme effectively reduces task delay.

7 Delay Modeling and Discussion

In this section, we provide an overview of the delay modeling and discussion for each layer of CAVs, incorporating the fine-grained classification of research directions presented in Sections 3–6. We also examine the factors that influence the delay in each layer and highlight the key concerns

addressed by current researchers. Additionally, we identify other factors affecting delay that have not been thoroughly considered.

Since the fine-grained modeling of delay varies across different papers, we adopt a high-level perspective to analyze and model the delay of the perception layer, communication layer, and computation layer. We acknowledge that the cross-layer delay modeling and analysis overlap to some extent with other layers. Therefore, we consolidate the cross-layer content and summarize it within the context of other layers.

7.1 The Perception Layer

The delay in the perception layer of CAVs refers to the time taken for environmental information to be received from sensors and made available for subsequent processing and decision-making. In cases where the vehicles have limited perception capabilities, collaborative perception techniques can be employed, leveraging sensors from other vehicles or edge nodes [26, 38, 124, 128, 203, 228]. Consequently, collaborative perception involves the joint modeling of the perception layer and the communication layer [36, 202]. Tables 3 and 4 highlight that the delay in the perception layer of CAVs primarily revolves around task inference delay. Task inference delay pertains to the time required for task analysis and inference based on the sensing data. The modeling of task inference delay typically takes into account the following factors: data preprocessing [116, 166, 183, 200], feature extraction [23, 26, 61, 124, 203, 213, 220], and model inference [21, 22, 48, 55, 72, 103, 109, 117, 134, 161, 182, 199, 209].

Data preprocessing delay. The delay in the perception layer refers to the time required such as data parsing, data format conversion, and data fusion of the sensing data [84]. The perception of various data types in CAVs has been extensively studied. Researchers commonly collect sensing data using camera [22, 199, 213], radar [117, 134, 181, 182, 209], radar-assisted camera [116, 166], **IMU (Inertial Measurement Unit)** [85, 224, 237], and GPS [21, 103, 171]. However, the challenge lies in developing real-time and scalable multi-sensor fusion algorithms that effectively process large volumes of sensor data and provide accurate environment perception results within the required time frame [202]. There is still a significant research gap in improving the real-time performance and scalability of data fusion algorithms in CAVs.

Feature extraction delay. The delay refers to the time spent extracting useful information or features to solve specific problems [142]. Most researchers use feature extraction methods such as CNN [22, 48, 72, 166, 209, 213], **RNN (Recurrent Neural Network)** [110, 142], and GAN [205] to extract advanced features of data effectively. It is worth noting that compared with ResNet [71, 176], DenseNet [78], and InceptionNet [187], lightweight feature extraction networks such as MobileNet [76, 173], SqueezeNet [82], and ShuffleNet [229] have been widely used in CAVs because of their lower computational complexity and efficiency.

Model inference delay. The delay refers to the time of calculation and inference in the task execution phase. Researchers usually adopt lightweight inference models to ensure the completion efficiency of tasks, such as fast object detection based on YoLo [169] and SSD [130] algorithms and real-time instance segmentation based on BiSeNet [220], Yolact [14], and DeepSnake [159] models. While significant research has been dedicated to reducing the complexity of inference models for improved real-time performance, challenges arise when deploying these models in practical application scenarios [148, 170]. Limited computing power, memory, and storage capacity in CAVs leads to resource competition during model reasoning, impacting real-time performance. To address this, performance analysis tools [10] can be utilized to monitor performance bottlenecks and identify time-consuming operations of the model in actual scenarios. Targeted optimization measures, such as asynchronous computing and module parallelism, can then be implemented to enhance overall performance and real-time capabilities.

7.2 The Communication Layer

The delay in the communication layer of CAVs refers to the delay experienced during communication between vehicles or between vehicles and infrastructure. As shown in Tables 5–7, the communication latency in CAVs primarily focuses on factors such as data transmission delay [50, 79, 80, 115, 122, 149, 154, 201, 211, 218, 233], communication switching delay [8, 13, 44, 62, 64–66, 86, 144, 188, 210], and end-to-end delay [30, 33, 34, 87, 88, 96, 135, 136, 151, 175, 215].

Data transmission delay. The delay refers to the time elapsed from the start of data transmission until the receiving party successfully receives the data. Modeling data transmission delay typically takes into account factors such as transmission distance [80, 115], transmission rate [11, 60, 94, 115, 201], packet quality [79, 122, 149, 211], and channel quality [50, 154, 218, 233]. In addition to optimizing these factors to reduce data transmission delay, parallel transmission and multi-path transmission techniques can be further employed to improve real-time data transmission. Additionally, optimizing the communication system of CAVs, including hardware acceleration and network stack optimization, can also help reduce data transmission delay.

Communication switching delay. The delay is the time that occurs when a vehicle switches from one communication link to another. The modeling of communication switching delay considers factors such as the availability of communication links [8, 44, 86, 144], the switching decision [13, 64, 188, 210], and the allocation of communication resources [62, 65, 66]. However, there is still a lack of consideration for data and resource caching and prefetching when CAVs seamlessly switch with other nodes. For example, leveraging local caching and prefetching technologies to store a portion of data before network switching can enable uninterrupted service provision during the transition, thereby reducing interruptions and delays.

End-to-end delay. The delay in the communication layer refers to the total time delay experienced when sending data from the source node to the target node for data reception [28, 150]. It encompasses various components, including sender delay, transmission delay, routing delay, and receiver delay [195]. End-to-end delay aims to comprehensively consider the delays introduced by each link in the communication layer, and the modeling typically involves summing up the delays of each link. However, there is a limited amount of research focused on simulating and measuring end-to-end communication delay [3]. To address this, network simulation tools or real measurement platforms can be utilized to evaluate the communication process of intelligent connected vehicles. These tools and platforms enable the assessment of system performance using test and evaluation data. Additionally, such approaches can be employed to validate the accuracy and applicability of theoretical models.

7.3 The Computation Layer

The computation layer delay refers to the time between receiving the sensing data and producing the decision. When dealing with compute-intensive tasks, due to the limited computing capacity of vehicles, it is usually necessary to use the resources of other vehicles or edge nodes for collaborative computing [5]. Therefore, collaborative computing usually involves the joint modeling of the communication layer and the computation layer to achieve efficient task processing and reduce delay [206, 234]. As shown in Tables 8–10, the delay of the computation layer of CAVs mainly focuses on the computing offloading delay [9, 19, 89, 93, 111, 131, 138, 140, 153, 186, 198, 235] and the task execution delay [20, 24, 39, 41, 49, 51, 77, 104, 106, 132, 163, 223, 225, 236].

Computing offloading delay. Modeling the delay of computing offloading usually considers several factors. First, the offloading path is one of the key factors [19, 131, 140, 153, 186, 198]. Different paths may have different network delays and bandwidths. Second, the processing capacity of the offloading destination also needs to be considered [89, 138, 140, 235]. Offloading

tasks to a destination with high computing power can effectively improve computing efficiency. In addition, load balancing is considered during task offloading [93, 111], which can overload or waste resources between nodes. At the same time, the transmission time of computing tasks [9, 19, 138, 140] should also be taken into consideration in modeling, in which task size and transmission distance have an impact on delay. In addition, it is feasible to further optimize the offloading delay according to the characteristics and priority of the task. Different types of tasks may have different characteristics and real-time requirements; for example, object detection tasks usually have higher priority and real-time requirements.

Task execution delay. The delay refers to the time during which computing tasks are completed on cloud servers or edge nodes. The modeling of task execution delay usually considers the following factors: First, different scheduling policies [20, 41, 49, 77, 83, 223, 236] determine the order in which tasks are executed and how resources are allocated on computing nodes. Choosing an appropriate scheduling policy improves resource utilization and task execution efficiency. Second, efficient resource allocation supports smooth task execution, including CPU, memory, and storage [24, 41, 49, 83, 104, 106, 132]. In addition, different tasks may have varying computational loads [39, 51, 163, 225]. Considering the complexity of each task, the task execution sequence and resource allocation can be optimized to reduce task execution delay. Task segmentation involves breaking down large tasks into smaller subtasks, while parallel execution utilizes multiple compute nodes simultaneously. These techniques enhance efficiency and parallelism in task execution, leading to reduced delay.

8 Challenges and Future Directions

Although significant progress has been made in the field of CAV delay reduction, important research challenges and open questions remain. Next, the challenges we have studied in depth in the literature are presented. These topics can guide future research in this area. These topics only address issues other than the direction of CAVs' delay, such as security and privacy, incentives, datasets, and evaluation criteria. These open challenges and issues require further investigation and have been organized around four key topics:

Real-time collaborative sensing between CAVs. The effect of multi-modal and multi-view image-point cloud fusion is more evident than that of the traditional perception algorithm. Moreover, the fused data significantly reduce the amount of data transmitted and the network communication load to improve the real-time data transmission performance. Therefore, to ensure object detection performance, the transmission of redundant data should be reduced as much as possible to enhance collaborative sensing results' accuracy and real-time performance. Good redundant data sharing helps improve the accuracy of perception, but too much redundant information is not good. The communication resources of vehicle networks are very precious, and it is necessary to schedule tasks and allocate resources to edge nodes reasonably. The information on the vehicle's **region of interest (ROI)** is selectively shared, and data redundancy is minimized to meet the requirements of timeliness and accuracy. Appropriate network technology deployment effectively reduces the delay.

Seamless communication. The vehicle communicates with one roadside unit and uploads its mission within a specific area. Once the vehicle is out of the coverage area of the roadside device, the transmission process will be interrupted. Therefore, the high mobility of vehicles leads to frequent switching between servers at the edge of small coverage areas, causing vehicles to move out of communication range with each other, and the connection time is short. This is very complicated and unreliable. In addition, since the network may have a few nodes or a large number of nodes in a few seconds or minutes, scalability issues may make it challenging to maintain good performance regardless of the number of nodes in the network. Therefore, it is crucial to understand the

spatial and temporal mobility of CAVs. A more effective routing link scheme can be proposed by understanding the spatio-temporal characteristics of the considered mobile entities. For example, an RSU used to perform a mobile user task is connected based on the historical trajectory of the vehicle. Maintaining QoS in the vehicle environment is a very complex and delicate task to obtain a timely and reliable communication transmission link between the vehicle and edge servers.

Efficient parallel processing of task computation. How dispatching incoming tasks to different cells at run time is a challenge to achieve optimal energy efficiency and performance. This is achieved through the run-time layer. Designing run times for heterogeneous autonomous driving systems is an entirely new research area with great potential, as most existing run-time designs focus on mapping an algorithm to one type of accelerator or focus on scheduling homogeneous systems or heterogeneous systems with a single accelerator. To ensure real-time performance, GPUs are generally responsible for running certain DNN tasks, such as image recognition. However, in future CAVs, multiple hybrid critical tasks must be run on GPUs in parallel. For example, GPUs simultaneously process various DNNs in parallel inference tasks (image recognition, speech recognition, natural language processing), DNN online training tasks, fault-tolerant backup tasks, and other vital tasks. At the same time, information display tasks of the dashboard, 2D/3D multimedia, or game system should be performed. This kind of multi-task requirement increases the complexity of task scheduling and resource management in the real-time embedded operating system.

Large language model-driven CAV. Over the past few years, we have witnessed the power of **large language models (LLMs)** to process and understand natural language. The application of LLMs in the field of CAVs is increasingly becoming an important direction of future development, especially in the enhancement of human-computer interaction, driver assistance systems, and fully autonomous driving technology. First, LLMs can make the voice assistant not only respond to simple instructions through deeper semantic understanding and situational adaptability but also participate in the daily conversation of the driver, providing a more humane interactive experience. Second, LLMs provide drivers with the latest road conditions, weather forecasts, and warnings of possible traffic risks by analyzing large amounts of text information such as social media, news reports and traffic databases in real-time. Finally, through real-time data exchange between vehicles and vehicles and between vehicles and the road network, LLMs can help build a more efficient connected vehicle ecosystem. Vehicles can not only communicate with each other and exchange driving data but also analyze this information to optimize traffic flow, reduce congestion, and improve energy efficiency. In the future, with the further enhancement of the capability of LLMs and the continuous expansion of application scenarios, CAVs will be more intelligent, safe, and efficient.

9 Summary

In this survey, we conduct a systematic and comprehensive review of delay optimization in CAVs. First, we introduce the importance of delay problems and divide the architecture of CAVs into the perception layer, communication layer, computation layer, and cross-layer. Second, we outline the meaning of each layer in CAV application scenarios and divide each layer according to the different ways of collaboration. Third, the research questions and techniques of each layer of delay analysis are outlined in depth, which is the outstanding result of this survey. Fourth, we model the delay of each layer and discuss the factors that affect the delay. Finally, through extensive and deep research on delay analysis in CAVs, we find out the bottlenecks and challenges of current research and put forward a valuable prospect.

Acknowledgments

We thank all the reviewers for their insightful feedback to improve this article.

References

- [1] Mohamad Abou Houran, Maimoona Asad, Gautam Srivastava, Jawad Mirza, Ali Ranjha, Muhammad Awais Javed, and Xu Yang. 2024. Intelligent reflecting surfaces assisted cellular V2X based open RAN communications. *IEEE Trans. Vehic. Technol.* 73, 7 (2024), 9226–9233.
- [2] Elijah I. Adegoke, Jasmine Zidane, Erik Kampert, Col R. Ford, Stewart A. Birrell, and Matthew D. Higgins. 2019. Infrastructure Wi-Fi for connected autonomous vehicle positioning: A review of the state-of-the-art. *Vehic. Commun.* 20 (2019), 100185.
- [3] M. Nadeem Ahangar, Qasim Z. Ahmed, Fahd A. Khan, and Maryam Hafeez. 2021. A survey of autonomous vehicles: Enabling communication technologies and challenges. *Sensors* 21, 3 (2021), 706.
- [4] Manzoor Ahmed, Muhammad Ayyed Mirza, Salman Raza, Haseeb Ahmad, Fang Xu, Wali Ullah Khan, Qing Lin, and Zhu Han. 2023. Vehicular communication network enabled CAV data offloading: A review. *IEEE Trans. Intell. Transport. Syst.* 24, 8 (2023), 7869–7897.
- [5] Julio C. S. Dos Anjos, Kassiano J. Matteussi, Fernanda C. Orlandi, Jorge L. V. Barbosa, Jorge Sá Silva, Luiz F. Bittencourt, and Cláudio F. R. Geyer. 2023. A survey on collaborative learning for intelligent autonomous systems. *Comput. Surv.* 56, 4 (2023), 1–37.
- [6] Shunsuke Aoki, Takuro Yonezawa, Nobuo Kawaguchi, Peter Steenkiste, and Ragunathan Raj Rajkumar. 2022. Time-sensitive cooperative perception for real-time data sharing over vehicular communications: Overview, challenges, and future directions. *IEEE Internet Things Mag.* 5, 2 (2022), 108–113.
- [7] Muhammad Ikram Ashraf, Chen-Feng Liu, Mehdi Bennis, and Walid Saad. 2017. Towards low-latency and ultra-reliable vehicle-to-vehicle communication. In *European Conference on Networks and Communications (EuCNC'17)*. IEEE, 1–5.
- [8] Yunhao Bai, Kuangyu Zheng, Zejiang Wang, Xiaorui Wang, and Junmin Wang. 2020. MC-safe: Multi-channel real-time V2V communication for enhancing driving safety. *ACM Trans. Cyber-phys. Syst.* 4, 4 (2020), 1–27.
- [9] Benjamin Baron, Prométhée Spathis, Hervé Rivano, Marcelo Dias de Amorim, Yannis Viniotis, and Mostafa H. Ammar. 2017. Centrally controlled mass data offloading using vehicular traffic. *IEEE Trans. Netw. Serv. Manag.* 14, 2 (2017), 401–415.
- [10] Shajulin Benedict. 2013. Performance issues and performance analysis tools for HPC cloud applications: A survey. *Computing* 95, 2 (2013), 89–108.
- [11] Rima Benelmir, Salim Bitam, Scott Fowler, and Abdelhamid Mellouk. 2023. A novel MmWave beam alignment approach for beyond 5G autonomous vehicle networks. *IEEE Trans. Vehic. Technol.* 73, 2 (2023), 1597–1610.
- [12] Jose Luis Blanco, Francisco Moreno, and Javier González-Jiménez. 2014. The Málaga urban dataset: High-rate stereo and LiDAR in a realistic urban scenario. *Int. J. Robot. Res.* 33 (02 2014), 207–214. DOI: <https://doi.org/10.1177/0278364913507326>
- [13] Annette Böhm and Magnus Jonsson. 2011. Real-time communication support for cooperative, infrastructure-based traffic safety applications. *Int. J. Vehic. Technol.* 2011 (2011).
- [14] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. 2019. YOLACT: Real-time instance segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV'19)*. 9157–9166.
- [15] Azzedine Boukerche and Victor Soto. 2020. Computation offloading and retrieval for vehicular edge computing. *ACM Comput. Surv.* 6 (2020), 442–457.
- [16] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuScenes: A multimodal dataset for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20)*. 11621–11631.
- [17] Zekun Cai and Aiping Xiong. 2023. Understand users' privacy perception and decision of V2X communication in connected autonomous vehicles. In *32nd USENIX Security Symposium (USENIX Security'23)*. 2975–2992.
- [18] Yongxu Cao, Xinxin He, Congqing Yu, and Changchuan Yin. 2021. Reliability and latency of MmWave communications based on blockage avoidance in internet of vehicles. In *13th International Conference on Wireless Communications and Signal Processing (WCSP'21)*. IEEE, 1–5.
- [19] Chen Chen, Yini Zeng, Huan Li, Yangyang Liu, and Shaohua Wan. 2022. A multi-hop task offloading decision model in MEC-enabled internet of vehicles. *IEEE Internet Things J.* 10, 4 (2022), 3215–3230. DOI: <https://doi.org/10.1109/JIOT.2022.3143529>
- [20] Chen Chen, Yuru Zhang, Zheng Wang, Shaohua Wan, and Qingqi Pei. 2021. Distributed computation offloading method based on deep reinforcement learning in ICV. *Appl. Soft Comput.* 103 (2021), 107108. DOI: <https://doi.org/10.1016/j.asoc.2021.107108>
- [21] C. L. Philip Chen, Jin Zhou, and Wei Zhao. 2012. A real-time vehicle navigation algorithm in sensor network environments. *IEEE Trans. Intell. Transport. Syst.* 13, 4 (2012), 1657–1666.
- [22] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. 2017. Learning efficient object detection models with knowledge distillation. *Advan. Neural Inf. Process. Syst.* 30 (2017), 742–751.

- [23] Junzhou Chen, Kunkun Jia, Wenquan Chen, Zhihan Lv, and Ronghui Zhang. 2022. A real-time and high-precision method for small traffic-signs recognition. *Neural Comput. Applic.* 34, 3 (2022), 2233–2245.
- [24] Min Chen and Yixue Hao. 2018. Task offloading for mobile edge computing in software defined ultra-dense network. *IEEE J. Select. Areas Commun.* 36, 3 (2018), 587–597. DOI: <https://doi.org/10.1109/JSAC.2018.2815360>
- [25] Maohong Chen, Yong Xiao, Qiang Li, and Kwang-cheng Chen. 2020. Minimizing age-of-information for fog computing-supported vehicular networks with deep Q-learning. In *IEEE International Conference on Communications (ICC'20)*. IEEE, 1–6.
- [26] Qi Chen, Xu Ma, Sihai Tang, Jingda Guo, Qing Yang, and Song Fu. 2019. F-Cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3D point clouds. In *4th ACM/IEEE Symposium on Edge Computing*. 88–100.
- [27] Shanzhi Chen, Jinling Hu, Yan Shi, Ying Peng, Jiayi Fang, Rui Zhao, and Li Zhao. 2017. Vehicle-to-everything (V2X) services supported by LTE-based systems and 5G. *IEEE Commun. Stand. Mag.* 1, 2 (2017), 70–76.
- [28] Xiaosha Chen, Supeng Leng, Jianhua He, Longyu Zhou, and Hao Liu. 2023. The upper bounds of cellular vehicle-to-vehicle communication latency for platoon-based autonomous driving. *IEEE Trans. Intell. Transport. Syst.* 24, 7 (2023), 6874–6887.
- [29] JiuJun Cheng, JunLu Cheng, MengChu Zhou, FuQiang Liu, ShangCe Gao, and Cong Liu. 2015. Routing in internet of vehicles: A review. *IEEE Trans. Intell. Transport. Syst.* 16, 5 (2015), 2339–2352.
- [30] Pei-Chun Cheng, Kevin C. Lee, Mario Gerla, and Jérôme Harri. 2010. GeoDTN+ Nav: Geographic DTN routing with navigator prediction for urban vehicular environments. *Mob. Netw. Applic.* 15, 1 (2010), 61–82.
- [31] Junil Choi, Vutha Va, Nuria Gonzalez-Prelcic, Robert Daniels, Chandra R. Bhat, and Robert W. Heath. 2016. Millimeter-wave vehicular communication to support massive automotive sensing. *IEEE Commun. Mag.* 54, 12 (2016), 160–167.
- [32] Jong-Moon Chung, Minseok Kim, Yong-Suk Park, Myungjun Choi, Sangwoo Lee, and Hyun Seo Oh. 2011. Time coordinated V2I communications and handover for WAVE networks. *IEEE J. Select. Areas Commun.* 29, 3 (2011), 545–558.
- [33] B. Coll-Perales, M. C. Lucas-Estañ, T. Shimizu, J. Gozálvez, T. Higuchi, S. Avedisov, O. Altintas, and M. Sepulcre. 2022. Improving the latency of 5G V2N2V communications in multi-MNO scenarios using MEC federation. In *IEEE 95th Vehicular Technology Conference (VTC'22-Spring)*. IEEE, 1–5.
- [34] Baldomero Coll-Perales, M. Carmen Lucas-Estañ, T. Shimizu, Javier Gozálvez, T. Higuchi, S. Avedisov, Onur Altintas, and Miguel Sepulcre. 2022. End-to-end V2X latency modeling and analysis in 5G networks. *arXiv preprint arXiv:2201.06082* (2022).
- [35] Juan Contreras-Castillo, Sherali Zeadally, and Juan Antonio Guerrero-Ibañez. 2017. Internet of vehicles: Architecture, protocols, and security. *IEEE Internet Things J.* 5, 5 (2017), 3701–3709.
- [36] Guangzhen Cui, Weili Zhang, Yanqiu Xiao, Lei Yao, and Zhanpeng Fang. 2022. Cooperative perception technology of autonomous driving in the internet of vehicles environment: A review. *Sensors* 22, 15 (2022), 5535.
- [37] Yaping Cui, Lijuan Du, Peng He, Dapeng Wu, and Ruyan Wang. 2022. Cooperative vehicles-assisted task offloading in vehicular networks. *Trans. Emerg. Telecommun. Technol.* 33, 7 (2022). DOI: <https://doi.org/10.1002/ett.4472>
- [38] Bin Dai, Fanglin Xu, Yuanyuan Cao, and Yang Xu. 2020. Hybrid sensing data fusion of cooperative perception for autonomous driving with augmented vehicular reality. *IEEE Syst. J.* 15, 1 (2020), 1413–1422.
- [39] Penglin Dai, Zihua Hang, Kai Liu, Xiao Wu, Huanlai Xing, Zhaoifei Yu, and Victor Chung Sing Lee. 2020. Multi-armed bandit learning for computation-intensive services in MEC-empowered vehicular networks. *IEEE Trans. Vehic. Technol.* 69, 7 (2020), 7821–7834. DOI: <https://doi.org/10.1109/TVT.2020.2991641>
- [40] Penglin Dai, Yaorong Huang, Kaiwen Hu, Xiao Wu, Huanlai Xing, and Zhaoifei Yu. 2024. Meta reinforcement learning for multi-task offloading in vehicular edge computing. *IEEE Trans. Mob. Comput.* 23, 3 (2024), 2123–2138.
- [41] Yueyue Dai, Du Xu, Sabita Maharjan, and Yan Zhang. 2018. Joint offloading and resource allocation in vehicular edge computing and networks. In *IEEE Global Communications Conference (GLOBECOM'18)*. IEEE, 1–7. DOI: <https://doi.org/10.1109/GLOCOM.2018.8648004>
- [42] Kakan Chandra Dey, Anjan Rayamajhi, Mashrur Chowdhury, Parth Bhavsar, and James Martin. 2016. Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication in a heterogeneous wireless network—Performance evaluation. *Transport. Res. Part C: Emerg. Technol.* 68 (2016), 168–184.
- [43] Boya Di, Lingyang Song, Yonghui Li, and Geoffrey Ye Li. 2017. Non-orthogonal multiple access for high-reliable and low-latency V2X communications in 5G systems. *IEEE J. Select. Areas Commun.* 35, 10 (2017), 2383–2397.
- [44] Antonio Di Maio, Maria Rita Palattella, and Thomas Engel. 2019. Multi-flow congestion-aware routing in software-defined vehicular networks. In *IEEE 90th Vehicular Technology Conference (VTC'19-Fall)*. IEEE, 1–6.
- [45] Leandro do C. Martins, Daniele Tarchi, Angel A. Juan, and Alessandro Fusco. 2022. Agile optimization for a real-time facility location problem in internet of vehicles networks. *Networks* 79, 4 (2022), 501–514. DOI: <https://doi.org/10.1002/net.22067>

- [46] Zheng Dong, Weisong Shi, Guangmo Tong, and Kecheng Yang. 2020. Collaborative autonomous driving: Vision and challenges. In *International Conference on Connected and Autonomous Driving (MetroCAD'20)*. IEEE, 17–26.
- [47] Leonidas Droukas, Zoe Doulgeri, Nikolaos L. Tsakiridis, Dimitra Triantafyllou, Ioannis Kleitsiotis, Ioannis Mariolis, Dimitrios Giakoumis, Dimitrios Tzovaras, Dimitrios Kateris, and Dionysis Bochtis. 2023. A survey of robotic harvesting systems and enabling technologies. *J. Intell. Robot. Syst.* 107, 2 (2023), 21.
- [48] Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, and Xiaolin Wei. 2021. Rethinking BiSeNet for real-time semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'21)*. 9716–9725.
- [49] Wenhao Fan, Jie Liu, Mingyu Hua, Fan Wu, and Yuan'an Liu. 2022. Joint task offloading and resource allocation for multi-access edge computing assisted by parked and moving vehicles. *IEEE Trans. Vehic. Technol.* 71, 5 (2022), 5314–5330.
- [50] Chun-Hao Fang, Kai-Ten Feng, and Lie-Liang Yang. 2022. Resource allocation for URLLC service in in-band full-duplex-based V2I networks. *IEEE Trans. Commun.* 70, 5 (2022), 3266–3281.
- [51] Jingyun Feng, Zhi Liu, Celimuge Wu, and Yusheng Ji. 2017. AVE: Autonomous vehicular edge computing framework with ACO-based scheduling. *IEEE Trans. Vehic. Technol.* 66, 12 (2017), 10660–10675. DOI : <https://doi.org/10.1109/TVT.2017.2714704>
- [52] Shuo Feng, Haowei Sun, Xintao Yan, Haojie Zhu, Zhengxia Zou, Shengyin Shen, and Henry X. Liu. 2023. Dense reinforcement learning for safety validation of autonomous vehicles. *Nature* 615, 7953 (2023), 620–627.
- [53] Qiang Fu and Jiajia Liu. 2023. Generalized multi-hop NR sidelink relay for future V2X communication. *IEEE/ACM Trans. Netw.* 32, 2 (2023), 1691–1706.
- [54] Nihal Gaouar and Mohamed Lehsaini. 2021. Toward vehicular cloud/fog communication: A survey on data dissemination in vehicular ad hoc networks using vehicular cloud/fog computing. *Int. J. Commun. Syst.* 34, 13 (2021), e4906.
- [55] Noa Garnett, Shai Silberstein, Shaul Oron, Ethan Fetaya, Uri Verner, Ariel Ayash, Vlad Goldner, Rafi Cohen, Kobi Horn, and Dan Levi. 2017. Real-time category-based and general obstacle detection for autonomous driving. In *IEEE International Conference on Computer Vision Workshops*. 198–205.
- [56] Xiaohu Ge. 2019. Ultra-reliable low-latency communications in autonomous vehicular networks. *IEEE Trans. Vehic. Technol.* 68, 5 (2019), 5005–5016.
- [57] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. 2013. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* 32, 11 (2013), 1231–1237.
- [58] Ionel Gog, Sukrit Kalra, Peter Schafhalter, Joseph E. Gonzalez, and Ion Stoica. 2022. D3: A dynamic deadline-driven approach for building autonomous vehicles. In *17th European Conference on Computer Systems*. 453–471.
- [59] Ionel Gog, Sukrit Kalra, Peter Schafhalter, Matthew A. Wright, Joseph E. Gonzalez, and Ion Stoica. 2021. Pylot: A modular platform for exploring latency-accuracy tradeoffs in autonomous vehicles. In *IEEE International Conference on Robotics and Automation (ICRA'21)*. IEEE, 8806–8813.
- [60] Andrew Graff, Yun Chen, Nuria González-Prelcic, and Takayuki Shimizu. 2023. Deep learning-based link configuration for radar-aided multiuser mmWave vehicle-to-infrastructure communication. *IEEE Trans. Vehic. Technol.* 72, 6 (2023), 7454–7468.
- [61] Jack Greenhalgh and Majid Mirmehdi. 2012. Real-time detection and recognition of road traffic signs. *IEEE Trans. Intell. Transport. Syst.* 13, 4 (2012), 1498–1506.
- [62] Xiaohui Gu, Guoan Zhang, Yancheng Ji, Wei Duan, Miaowen Wen, Zhiguo Ding, and Pin-Han Ho. 2022. Intelligent surface aided D2D-V2X system for low-latency and high-reliability communications. *IEEE Trans. Vehic. Technol.* 71, 11 (2022), 11624–11636.
- [63] Jacopo Guanetti, Yeojun Kim, and Francesco Borrelli. 2018. Control of connected and automated vehicles: State of the art and future challenges. *Ann. Rev. Contr.* 45 (2018), 18–40.
- [64] Chang Guo, Demin Li, Xuemin Chen, and Guanglin Zhang. 2022. An adaptive V2R communication strategy based on data delivery delay estimation in VANETs. *Vehic. Commun.* 34 (2022), 100444.
- [65] Chongtao Guo, Le Liang, and Geoffrey Ye Li. 2019. Resource allocation for high-reliability low-latency vehicular communications with packet retransmission. *IEEE Trans. Vehic. Technol.* 68, 7 (2019), 6219–6230.
- [66] Chongtao Guo, Le Liang, and Geoffrey Ye Li. 2019. Resource allocation for low-latency vehicular communications: An effective capacity perspective. *IEEE J. Select. Areas Commun.* 37, 4 (2019), 905–917.
- [67] Chongtao Guo, Le Liang, and Geoffrey Ye Li. 2019. Resource allocation for vehicular communications with low latency and high reliability. *IEEE Trans. Wirel. Commun.* 18, 8 (2019), 3887–3902.
- [68] Umar Zakir Abdul Hamid, Hairi Zamzuri, and Dilip Kumar Limbu. 2019. Internet of vehicle (IoV) applications in expediting the implementation of smart highway of autonomous vehicle: A survey. In *Performability in Internet of Things*. Springer, 137–157.
- [69] Muhammad Hasanujjaman, Mostafa Zaman Chowdhury, and Yeong Min Jang. 2023. Sensor fusion in autonomous vehicle with traffic surveillance camera system: Detection, localization, and AI networking. *Sensors* 23, 6 (2023), 3335.

- [70] Jianhua He, Zuoyin Tang, Xiaoming Fu, Supeng Leng, Fan Wu, Kaisheng Huang, Jianye Huang, Jie Zhang, Yan Zhang, Andrew Radford et al. 2019. Cooperative connected autonomous vehicles (CAV): Research, applications and challenges. In *IEEE 27th International Conference on Network Protocols (ICNP'19)*. IEEE, 1–6.
- [71] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*. 770–778.
- [72] Weixiang Hong, Qingpei Guo, Wei Zhang, Jingdong Chen, and Wei Chu. 2021. LPSNet: A lightweight solution for fast panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'21)*. 16746–16754.
- [73] Lu Hou, Lei Lei, Kan Zheng, and Xianbin Wang. 2019. A Q-learning-based proactive caching strategy for non-safety related services in vehicular networks. *IEEE Internet Things J.* 6, 3 (2019), 4512–4520. DOI : <https://doi.org/10.1109/JIOT.2018.2883762>
- [74] Rui Hou, Jie Li, Arjun Bhargava, Allan Raventos, Vitor Guizilini, Chao Fang, Jerome Lynch, and Adrien Gaidon. 2020. Real-time panoptic segmentation from dense detections. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20)*. 8523–8532.
- [75] Xiangwang Hou, Zhiyuan Ren, Jingjing Wang, Wenchi Cheng, Yong Ren, Kwang-Cheng Chen, and Hailin Zhang. 2020. Reliable computation offloading for edge-computing-enabled software-defined IoV. *IEEE Internet Things J.* 7, 8 (2020), 7097–7111. DOI : <https://doi.org/10.1109/JIOT.2020.2982292>
- [76] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan et al. 2019. Searching for MobileNetV3. In *IEEE/CVF International Conference on Computer Vision (ICCV'19)*. 1314–1324.
- [77] Shihong Hu, Guanghui Li, and Weisong Shi. 2021. LARS: A latency-aware and real-time scheduling framework for edge-enabled internet of vehicles. *IEEE Trans. Serv. Comput.* 16, 1 (2021), 398–411.
- [78] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. 2017. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*. 4700–4708.
- [79] Jiaqi Huang, Yi Qian, and Rose Qingyang Hu. 2019. A vehicle-assisted data offloading in mobile edge computing enabled vehicular networks. In *IEEE Global Communications Conference (GLOBECOM'19)*. IEEE, 1–6. DOI : <https://doi.org/10.1109/GLOBECOM38437.2019.9013760>
- [80] Xiaoge Huang, Ke Xu, Qianbin Chen, and Jie Zhang. 2021. Delay-aware caching in internet-of-vehicles networks. *IEEE Internet Things J.* 8, 13 (2021), 10911–10921.
- [81] Shao-Chou Hung, Xin Zhang, Andreas Festag, Kwang-Cheng Chen, and Gerhard Fettweis. 2019. Vehicle-centric network association in heterogeneous vehicle-to-vehicle networks. *IEEE Trans. Vehic. Technol.* 68, 6 (2019), 5981–5996.
- [82] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. *arXiv preprint arXiv:1602.07360* (2016).
- [83] Muhammad Ibrar, Aamir Akbar, Syed Rooh Ullah Jan, Mian Ahmad Jan, Lei Wang, Houbing Song, and Nadir Shah. 2022. ARTNet: AI-based resource allocation and task offloading in a reconfigurable internet of vehicular networks. *IEEE Trans. Netw. Sci. Eng.* 9, 1 (2022), 67–77. DOI : <https://doi.org/10.1109/TNSE.2020.3047454>
- [84] Henry Alexander Ignatious, Hesham El-Sayed, Manzoor Ahmed Khan, and Bassem Mahmoud Mokhtar. 2023. Analyzing factors influencing situation awareness in autonomous vehicles—A survey. *Sensors* 23, 8 (2023), 4075.
- [85] Szakats Istvan, Cătălin Golban, and Sergiu Nedevschi. 2011. Fast vision based ego-motion estimation from stereo sequences—A GPU approach. In *14th International IEEE Conference on Intelligent Transportation Systems (ITSC'11)*. IEEE, 538–543.
- [86] Boangoat Jarupan and Eylem Ekici. 2009. Location- and delay-aware cross-layer communication in V2I multihop vehicular networks. *IEEE Commun. Mag.* 47, 11 (2009), 112–118.
- [87] Boangoat Jarupan and Eylem Ekici. 2010. PROMPT: A cross-layer position-based communication protocol for delay-aware vehicular access networks. *Ad Hoc Netw.* 8, 5 (2010), 489–505.
- [88] Moez Jerbi, S.-M. Senouci, Rabah Meraihi, and Yacine Ghamri-Doudane. 2007. An improved vehicular ad hoc routing protocol for city environments. In *IEEE International Conference on Communications*. IEEE, 3972–3979.
- [89] Fan Jiang, Wei Liu, Junxuan Wang, and Xinying Liu. 2020. Q-learning based task offloading and resource allocation scheme for internet of vehicles. In *IEEE/CIC International Conference on Communications in China (ICCC'20)*. IEEE, 460–465.
- [90] Zhu Jin, Tiecheng Song, and Wen-Kang Jia. 2024. An adaptive cooperative caching strategy for vehicular networks. *IEEE Trans. Mob. Comput.* 23, 10 (2024), 9502–9517.
- [91] Zhiyang Ju, Hui Zhang, Xiang Li, Xiaoguang Chen, Jinpeng Han, and Manzhi Yang. 2022. A survey on attack detection and resilience for connected and automated vehicles: From vehicle dynamics and control perspective. *IEEE Trans. Intell. Vehic.* 7, 4 (2022), 815–837.

- [92] Chanyoung Jung, Daegyu Lee, Seungwook Lee, and David Hyunchul Shim. 2020. V2X-communication-aided autonomous driving: System design and experimental validation. *Sensors* 20, 10 (2020), 2903.
- [93] Ahmed Jawad Kadhim and Seyed-Amin Hosseini-Seno. 2019. Maximizing the utilization of fog computing in internet of vehicle using SDN. *IEEE Commun. Lett.* 23, 1 (2019), 140–143. DOI : <https://doi.org/10.1109/LCOMM.2018.2878710>
- [94] Ahmad Kamari, Yoon Chae, and Parth Pathak. 2023. mmSV: mmWave vehicular networking using street view imagery in urban environments. In *29th Annual International Conference on Mobile Computing and Networking (MobiCom'23)*. 1–16.
- [95] Brad Karp and Hsiang-Tsung Kung. 2000. GPSR: Greedy perimeter stateless routing for wireless networks. In *6th Annual International Conference on Mobile Computing and Networking (MobiCom'00)*. 243–254.
- [96] Konstantinos Katsaros, Mehrdad Dianati, Rahim Tafazolli, and Ralf Kernchen. 2011. CLWPR–A novel cross-layer optimized position based routing protocol for VANETS. In *IEEE Vehicular Networking Conference (VNC'11)*. IEEE, 139–146.
- [97] Sanjit Kaul, Roy Yates, and Marco Gruteser. 2012. Real-time status: How often should one update? In *IEEE Conference on Computer Communications (INFOCOM'12)*. IEEE, 2731–2735.
- [98] Tobias Kessler and Alois Knoll. 2019. Cooperative multi-vehicle behavior coordination for autonomous driving. In *IEEE Intelligent Vehicles Symposium (IV'19)*. IEEE, 1953–1960.
- [99] Aidil Redza Khan, Mohd Faizal Jamlos, Nurmadiha Osman, Muhammad Izhar Ishak, Fatimah Dzaharudin, You Kok Yeow, and Khairil Anuar Khairi. 2022. DSRC technology in Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) IoT system for Intelligent Transportation System (ITS): A review. *Recent Trends Mechatron. Industr. 4.0* 730 (2022), 97–106.
- [100] Manzoor Ahmed Khan, Hesham El Sayed, Sumbal Malik, Talha Zia, Jalal Khan, Najla Alkaabi, and Henry Ignatious. 2022. Level-5 autonomous driving–Are we there yet? A review of research literature. *ACM Comput. Surv.* 55, 2 (2022), 1–38.
- [101] Mohammad Khayatian, Mohammadreza Mehrabian, Edward Andert, Rachel Dedinsky, Sarthake Choudhary, Yingyan Lou, and Aviral Shirvastava. 2020. A survey on intersection management of connected autonomous vehicles. *ACM Trans. Cyber-phys. Syst.* 4, 4 (2020), 1–27.
- [102] Jeong-ah Kim, Ju-Yeong Sung, and Se-ho Park. 2020. Comparison of Faster-RCNN, YOLO, and SSD for real-time vehicle type recognition. In *IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia'20)*. IEEE, 1–4.
- [103] Songsang Koh, Bo Zhou, Hui Fang, Po Yang, Zaili Yang, Qiang Yang, Lin Guan, and Zhigang Ji. 2020. Real-time deep reinforcement learning based vehicle navigation. *Appl. Soft Comput.* 96 (2020), 106694.
- [104] Seung-Seob Lee and Sukyoung Lee. 2019. Poster abstract: Deep reinforcement learning-based resource allocation in vehicular fog computing. In *IEEE Conference on Computer Communications Workshops (INFOCOM Workshops'19)*. IEEE, 1029–1030. DOI : <https://doi.org/10.1109/INFCOMW.2019.8845095>
- [105] Sung-Ju Lee, William Su, and Mario Gerla. 2002. On-demand multicast routing protocol in multihop wireless mobile networks. *Mob. Netw. Applic.* 7, 6 (2002), 441–453.
- [106] Seung-Seob Lee and Sukyoung Lee. 2020. Resource allocation for vehicular fog computing using reinforcement learning combined with heuristic information. *IEEE Internet Things J.* 7, 10 (2020), 10450–10464. DOI : <https://doi.org/10.1109/JIOT.2020.2996213>
- [107] Dan Levi, Noa Garnett, Ethan Fetaya, and Israel Herzlyia. 2015. StixelNet: A deep convolutional network for obstacle detection and road segmentation. In *British Machine Vision Conference (BMVC'15)*. 4.
- [108] Chunlin Li, Yong Zhang, and Youlong Luo. 2022. A federated learning-based edge caching approach for mobile edge computing-enabled intelligent connected vehicles. *IEEE Trans. Intell. Transport. Syst.* 24, 3 (2022), 3360–3369.
- [109] Jinke Li, Xiao He, Yang Wen, Yuan Gao, Xiaoqiang Cheng, and Dan Zhang. 2022. Panoptic-PHNet: Towards real-time and high-precision LiDAR panoptic segmentation via clustering pseudo heatmap. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*. 11809–11818.
- [110] Lin Li, Wanzhong Zhao, Can Xu, Chunyan Wang, Qingyun Chen, and Shijuan Dai. 2021. Lane-change intention inference based on RNN for autonomous driving on highways. *IEEE Trans. Vehic. Technol.* 70, 6 (2021), 5499–5510.
- [111] Mushu Li, Jie Gao, Lian Zhao, and Xuemin Shen. 2020. Deep reinforcement learning for collaborative edge computing in vehicular networks. *IEEE Trans. Cogn. Commun. Netw.* 6, 4 (2020), 1122–1135.
- [112] Yixuan Li, Lei Chen, Runyu He, Zhenzhi Wang, Gangshan Wu, and Limin Wang. 2021. Multisports: A multi-person video dataset of spatio-temporally localized sports actions. In *IEEE/CVF International Conference on Computer Vision (ICCV'21)*. 13536–13545.
- [113] You Li and Javier Ibanez-Guzman. 2020. LiDAR for autonomous driving: The principles, challenges, and trends for automotive LiDAR and perception systems. *IEEE Sig. Process. Mag.* 37, 4 (2020), 50–61.
- [114] Yunxin Li, Fan Liu, Zhen Du, Weijie Yuan, Qingjiang Shi, and Christos Masouros. 2024. Frame structure and protocol design for sensing-assisted NR-V2X communications. *IEEE Trans. Mob. Comput.* 23, 12 (2024), 11045–11060. DOI : <https://doi.org/10.1109/TMC.2024.3389697>

- [115] Zipeng Li, Lin Xiang, Xiaohu Ge, Guoqiang Mao, and Han-Chieh Chao. 2020. Latency and reliability of mmWave multi-hop V2V communications under relay selections. *IEEE Trans. Vehic. Technol.* 69, 9 (2020), 9807–9821.
- [116] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. 2019. Multi-task multi-sensor fusion for 3D object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19)*. 7345–7353.
- [117] Zhidong Liang, Zehan Zhang, Ming Zhang, Xian Zhao, and Shiliang Pu. 2021. RangeIoUDet: Range image based real-time 3D object detector optimized by intersection over union. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'21)*. 7140–7149.
- [118] Bing Shun Lim, Sye Loong Keoh, and Vrizlynn L. L. Thing. 2018. Autonomous vehicle ultrasonic sensor vulnerability and impact assessment. In *IEEE 4th World Forum on Internet of Things (WF-IoT'18)*. IEEE, 231–236.
- [119] Shih-Chieh Lin, Yunqi Zhang, Chang-Hong Hsu, Matt Skach, Md E. Haque, Lingjia Tang, and Jason Mars. 2018. The architectural implications of autonomous driving: Constraints and acceleration. In *23rd International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'18)*. 751–766.
- [120] An Liu, Zhe Huang, Min Li, Yubo Wan, Wenrui Li, Tony Xiao Han, Chenchen Liu, Rui Du, Danny Kai Pin Tan, Jianmin Lu et al. 2022. A survey on fundamental limits of integrated sensing and communication. *IEEE Commun. Surv. Tutor.* 24, 2 (2022), 994–1034.
- [121] Jinshi Liu, Manzoor Ahmed, Muhammad Ayzed Mirza, Wali Ullah Khan, Dianlei Xu, Jianbo Li, Abdul Aziz, and Zhu Han. 2022. RL/DRL meets vehicular task offloading using edge and vehicular cloudlet: A survey. *IEEE Internet Things J.* 9, 11 (2022), 8315–8338.
- [122] Kai Liu, Joseph K. Y. Ng, Victor C. S. Lee, Sang H. Son, and Ivan Stojmenovic. 2015. Cooperative data scheduling in hybrid vehicular ad hoc networks: VANET as a software defined network. *IEEE/ACM Trans. Netw.* 24, 3 (2015), 1759–1773.
- [123] Lei Liu, Chen Chen, Qingqi Pei, Sabita Maharjan, and Yan Zhang. 2021. Vehicular edge computing and networking: A survey. *Mob. Netw. Applic.* 26, 3 (2021), 1145–1168.
- [124] Luyang Liu and Marco Gruteser. 2021. EdgeSharing: Edge assisted real-time localization and object sharing in urban streets. In *IEEE Conference on Computer Communications (INFOCOM'21)*. IEEE, 1–10.
- [125] Liangkai Liu, Ren Zhong, Aaron Willcock, Nathan Fisher, and Weisong Shi. 2023. An open approach to energy-efficient autonomous mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA'23)*. IEEE, 11569–11575.
- [126] Mingyu Liu, Ekim Yurtsever, Jonathan Fossaert, Xingcheng Zhou, Walter Zimmer, Yuning Cui, Bare Luka Zagar, and Alois C. Knoll. 2024. A survey on autonomous driving datasets: Statistics, annotation quality, and a future outlook. *IEEE Trans. Intell. Vehic.* (2024).
- [127] Pengfei Liu, Zhangyu Wang, Guizhen Yu, Bin Zhou, and Peng Chen. 2023. Region-based hybrid collaborative perception for connected autonomous vehicles. *IEEE Trans. Vehic. Technol.* 37, 4 (2023), 273–281.
- [128] Qiang Liu, Tao Han, Jiang Linda Xie, and BaekGyu Kim. 2021. LiveMap: Real-time dynamic map in automotive edge computing. In *IEEE Conference on Computer Communications (INFOCOM'21)*. 1–10. DOI: <https://doi.org/10.1109/INFOCOM42981.2021.9488872>
- [129] Shengli Liu, Guanding Yu, Rui Yin, Jiantao Yuan, and Fengzhong Qu. 2023. Communication and computation efficient federated learning for internet of vehicles with a constrained latency. *IEEE Trans. Vehic. Technol.* 73, 1 (2023), 1038–1052.
- [130] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. 2016. SSD: Single shot multibox detector. In *14th European Conference on Computer Vision (ECCV'16)*. Springer, 21–37.
- [131] Yujiong Liu, Shangguang Wang, Jie Huang, and Fangchun Yang. 2018. A computation offloading algorithm based on game theory for vehicular edge networks. In *IEEE International Conference on Communications (ICC'18)*. IEEE, 1–6. DOI: <https://doi.org/10.1109/ICC.2018.8422240>
- [132] Yi Liu, Huimin Yu, Shengli Xie, and Yan Zhang. 2019. Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks. *IEEE Trans. Vehic. Technol.* 68, 11 (2019), 11158–11168. DOI: <https://doi.org/10.1109/TVT.2019.2935450>
- [133] Yupei Liu, Haijun Zhang, Keping Long, Arumugam Nallanathan, and Victor C. M. Leung. 2019. Energy-efficient subchannel matching and power allocation in NOMA autonomous driving vehicular networks. *IEEE Wirel. Commun.* 26, 4 (2019), 88–93.
- [134] Zongdai Liu, Dingfu Zhou, Feixiang Lu, Jin Fang, and Liangjun Zhang. 2021. AutoShape: Real-time shape-aware monocular 3D object detection. In *IEEE/CVF International Conference on Computer Vision (ICCV'21)*. 15641–15650.
- [135] Christian Lochert, Hannes Hartenstein, Jing Tian, Holger Fussler, Dagmar Hermann, and Martin Mauve. 2003. A routing strategy for vehicular ad hoc networks in city environments. In *IEEE Intelligent Vehicles Symposium (IV'03)*. IEEE, 156–161.
- [136] Christian Lochert, Martin Mauve, Holger Füßler, and Hannes Hartenstein. 2005. Geographic routing in city scenarios. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* 9, 1 (2005), 69–72.

- [137] Sidi Lu and Weisong Shi. 2021. The emergence of vehicle computing. *IEEE Internet Comput.* 25, 3 (2021), 18–22.
- [138] Qu Yuan Luo, Changle Li, Tom H. Luan, and Weisong Shi. 2022. Minimizing the delay and cost of computation offloading for vehicular edge computing. *IEEE Trans. Serv. Comput.* 15, 5 (2022), 2897–2909. DOI: <https://doi.org/10.1109/TSC.2021.3064579>
- [139] Qu Yuan Luo, Changle Li, Tom H. Luan, Weisong Shi, and Weigang Wu. 2021. Self-learning based computation offloading for internet of vehicles: Model and algorithm. *IEEE Trans. Wirel. Commun.* 20, 9 (2021), 5913–5925. DOI: <https://doi.org/10.1109/TWC.2021.3071248>
- [140] Chunmei Ma, Jinqi Zhu, Ming Liu, Hui Zhao, Nianbo Liu, and Xinyu Zou. 2021. Parking edge computing: Parked-vehicle-assisted task offloading for urban VANETs. *IEEE Internet Things J.* 8, 11 (2021), 9344–9358. DOI: <https://doi.org/10.1109/JIOT.2021.3056396>
- [141] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 2017. 1 Year, 1000km: The Oxford RobotCar dataset. *Int. J. Robot. Res.* 36, 1 (2017), 3–15. DOI: <https://doi.org/10.1177/0278364916679498>
- [142] Pegah Mansourian, Ning Zhang, Arunita Jaekel, and Marc Kneppers. 2023. Deep learning-based anomaly detection for connected autonomous vehicles using spatiotemporal information. *IEEE Trans. Intell. Transport. Syst.* 24, 12 (2023), 16006–16017.
- [143] Stefano Mariani, Giacomo Cabri, and Franco Zambonelli. 2021. Coordination of autonomous vehicles: Taxonomy and survey. *ACM Comput. Surv.* 54, 1 (2021), 1–33.
- [144] David Martín-Sacristán, Sandra Roger, David Garcia-Roger, Jose F. Monserrat, Panagiotis Spapis, Chan Zhou, and Alexandros Kaloxilos. 2020. Low-latency infrastructure-based cellular V2V communications for multi-operator environments with regional split. *IEEE Trans. Intell. Transport. Syst.* 22, 2 (2020), 1052–1067.
- [145] Rodolfo Meneguette, Robson De Grande, Jo Ueyama, Geraldo P. Rocha Filho, and Edmundo Madeira. 2021. Vehicular edge computing: Architecture, resource management, security, and challenges. *ACM Comput. Surv.* 55, 1 (2021), 1–46.
- [146] Manabu Mikami, Koichi Serizawa, Yoshikazu Ishida, Hideya Nishiyori, Kohei Moto, and Hitoshi Yoshino. 2020. Field experimental evaluation on latency and reliability performance of 5G NR V2V direct communication in real express highway environment. In *IEEE 91st Vehicular Technology Conference (VTC'20-Spring)*. IEEE, 1–5.
- [147] Syed Agha Hassnain Mohsan, Nawaf Qasem Hamood Othman, Yanlong Li, Mohammed H. Alsharif, and Muhammad Asghar Khan. 2023. Unmanned aerial vehicles (UAVs): Practical aspects, applications, open challenges, security issues, and future trends. *Intell. Serv. Robot.* 16, 1 (2023), 109–137.
- [148] M. G. Sarwar Murshed, Christopher Murphy, Daqing Hou, Nazar Khan, Ganesh Ananthanarayanan, and Faraz Hus-sain. 2021. Machine learning at the network edge: A survey. *ACM Comput. Surv.* 54, 8 (2021), 1–37.
- [149] Faisal Naeem, Sattar Seifollahi, Zhenyu Zhou, and Muhammad Tariq. 2021. A generative adversarial network enabled deep distributional reinforcement learning for transmission scheduling in internet of vehicles. *IEEE Trans. Intell. Transport. Syst.* 22, 7 (2021), 4550–4559. DOI: <https://doi.org/10.1109/TITS.2020.3033577>
- [150] Ashish Nanda, Deepak Puthal, Joel J. P. C. Rodrigues, and Sergei A. Kozlov. 2019. Internet of autonomous vehicles communications security: Overview, issues, and directions. *IEEE Wirel. Commun.* 26, 4 (2019), 60–65.
- [151] Valery Naumov and Thomas R. Gross. 2007. Connectivity-aware routing (CAR) in vehicular ad-hoc networks. In *26th IEEE International Conference on Computer Communications (INFOCOM'07)*. IEEE, 1919–1927.
- [152] Anselme Ndikumana, Kim Khoa Nguyen, and Mohamed Cheriet. 2024. Age of processing-aware offloading decision for autonomous vehicles in 5G open RAN environment. *IEEE Trans. Mob. Comput.* 23, 7 (2024), 7865–7877.
- [153] Khoa Nguyen, Steve Drew, Changcheng Huang, and Jiayu Zhou. 2020. Collaborative container-based parked vehicle edge computing framework for online task offloading. In *IEEE 9th International Conference on Cloud Networking (CloudNet'20)*. 1–6. DOI: <https://doi.org/10.1109/CloudNet51028.2020.9335809>
- [154] Zhaolong Ning, Kaiyuan Zhang, Xiaojie Wang, Lei Guo, Xiping Hu, Jun Huang, Bin Hu, and Ricky Yu-Kwong Kwok. 2021. Intelligent edge computing in internet of vehicles: A joint computation offloading and caching solution. *IEEE Trans. Intell. Transport. Syst.* 22, 4 (2021), 2212–2225. DOI: <https://doi.org/10.1109/TITS.2020.2997832>
- [155] Md Noor-A-Rahim, Zilong Liu, Haeyoung Lee, Mohammad Omar Khyam, Jianhua He, Dirk Pesch, Klaus Moessner, Walid Saad, and H. Vincent Poor. 2022. 6G for vehicle-to-everything (V2X) communications: Enabling technologies, challenges, and opportunities. *Proc. IEEE* 110, 6 (2022), 712–734.
- [156] Stephan Olariu. 2019. A survey of vehicular cloud research: Trends, applications and challenges. *IEEE Trans. Intell. Transport. Syst.* 21, 6 (2019), 2648–2663.
- [157] Fei Peng, Zhiyuan Jiang, Shunqing Zhang, and Shugong Xu. 2020. Age of information optimized MAC in V2X sidelink via piggyback-based collaboration. *IEEE Trans. Wirel. Commun.* 20, 1 (2020), 607–622.
- [158] Qingxia Peng, Yunjian Jia, Liang Liang, and Zhengchuan Chen. 2021. A task assignment scheme for parked-vehicle assisted edge computing in IoV. In *93rd IEEE Vehicular Technology Conference (VTC-Spring'21)*. IEEE, 1–5. DOI: <https://doi.org/10.1109/VTC2021-Spring51267.2021.9448735>
- [159] Sida Peng, Wen Jiang, Huaijin Pi, Xiuli Li, Hujun Bao, and Xiaowei Zhou. 2020. Deep snake for real-time instance segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20)*. 8533–8542.

- [160] Hughes Perreault, Guillaume-Alexandre Bilodeau, Nicolas Saunier, and Maguelonne Héritier. 2021. CenterPoly: Real-time instance segmentation using bounding polygons. In *IEEE/CVF International Conference on Computer Vision (ICCV'21)*. 2982–2991.
- [161] Gowdham Prabhakar, Binsu Kailath, Sudha Natarajan, and Rajesh Kumar. 2017. Obstacle detection and classification using deep learning for tracking in high-speed autonomous driving. In *IEEE Region 10 Symposium (TENSYP'17)*. IEEE, 1–6.
- [162] Suyash Pradhan, Debashri Roy, Batool Salehi, and Kaushik Chowdhury. 2024. COPILOT: Cooperative perception using LiDAR for handoffs between road side units. In *IEEE Conference on Computer Communications (INFOCOM'24)*. 1301–1310. DOI: <https://doi.org/10.1109/INFOCOM52122.2024.10621174>
- [163] Guanhua Qiao, Supeng Leng, Sabita Maharjan, Yan Zhang, and Nirwan Ansari. 2020. Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks. *IEEE Internet Things J.* 7, 1 (2020), 247–257. DOI: <https://doi.org/10.1109/JIOT.2019.2945640>
- [164] Xiaqi Qin, Yangyang Xia, Hang Li, Zhiyong Feng, and Ping Zhang. 2021. Distributed data collection in age-aware vehicular participatory sensing networks. *IEEE Internet Things J.* 8, 19 (2021), 14501–14513.
- [165] Kaige Qu, Weihua Zhuang, Qiang Ye, Wen Wu, and Xuemin Shen. 2024. Model-assisted learning for adaptive cooperative perception of connected autonomous vehicles. *IEEE Trans. Wirel. Commun.* 23, 8 (2024), 8820–8835. DOI: <https://doi.org/10.1109/TWC.2024.3354507>
- [166] Hazem Rashed, Mohamed Ramzy, Victor Vaquero, Ahmad El Sallab, Ganesh Sistu, and Senthil Yogamani. 2019. Fuse-MODNet: Real-time camera and LiDAR based moving object detection for robust low-light autonomous driving. In *IEEE/CVF International Conference on Computer Vision Workshops*.
- [167] Ifthikhar Rasheed, Fei Hu, Yang-Ki Hong, and Bharat Balasubramanian. 2020. Intelligent vehicle network routing with adaptive 3D beam alignment for mmWave 5G-based V2X communications. *IEEE Trans. Intell. Transport. Syst.* 22, 5 (2020), 2706–2718.
- [168] Salman Raza, Shanguang Wang, Manzoor Ahmed, and Muhammad Rizwan Anwar. 2019. A survey on vehicular edge computing: Architecture, applications, technical issues, and future directions. *Wirel. Commun. Mob. Comput.* 2019 (2019).
- [169] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*. 779–788.
- [170] Ju Ren, Yundi Guo, Deyu Zhang, Qingqing Liu, and Yaoxue Zhang. 2018. Distributed and efficient object detection in edge computing: Challenges and solutions. *IEEE Netw.* 32, 6 (2018), 137–143.
- [171] Zhigang Ren, Jialun Lai, Zongze Wu, and Shengli Xie. 2021. Deep neural networks-based real-time optimal navigation for an automatic guided vehicle with static and dynamic obstacles. *Neurocomputing* 443 (2021), 329–344.
- [172] Angélica Reyes-Muñoz and Juan Guerrero-Ibáñez. 2022. Vulnerable road users and connected autonomous vehicles interaction: A survey. *Sensors* 22, 12 (2022), 4614.
- [173] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. MobileNetV2: Inverted residuals and linear bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*. 4510–4520.
- [174] Roshan Sedar, Francisco Vázquez-Gallego, Ramon Casellas, Ricard Vilalta, Raul Muñoz, Rodrigo Silva, Laurent Dizambourg, Antonio Eduardo Fernández Barciela, Xavier Vilajosana, Soumya Kanti Datta et al. 2021. Standards-compliant multi-protocol on-board unit for the evaluation of connected and automated mobility services in multi-vendor environments. *Sensors* 21, 6 (2021), 2090.
- [175] Boon-Chong Seet, Genping Liu, Bu-Sung Lee, Chuan-Heng Foh, Kai-Juan Wong, and Keok-Kee Lee. 2004. A-STAR: A mobile ad hoc routing strategy for metropolis vehicular communications. In *International Conference on Research in Networking*. Springer, 989–999.
- [176] Muhammad Shafiq and Zhaoquan Gu. 2022. Deep residual learning for image recognition: A survey. *Appl. Sci.* 12, 18 (2022), 8972.
- [177] Shuyao Shi, Neiwen Ling, Zehao Jiang, Xuan Huang, Yuze He, Xiaoguang Zhao, Bufang Yang, Chen Bian, Jingfei Xia, Zhenyu Yan et al. 2024. Soar: Design and deployment of a smart roadside infrastructure system for autonomous driving. In *30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom'24)*. Association for Computing Machinery, New York, NY, USA, 139–154. DOI: <https://doi.org/10.1145/3636534.3649352>
- [178] Weisong Shi and Liangkai Liu. 2021. *Computing Systems for Autonomous Driving*. Springer.
- [179] Christina Suyong Shin, Weiwu Pang, Chuan Li, Fan Bai, Fawad Ahmad, Jeongyeup Paek, and Ramesh Govindan. 2024. RECAP: 3D traffic reconstruction. In *30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom'24)*. Association for Computing Machinery, New York, NY, USA, 1252–1267. DOI: <https://doi.org/10.1145/3636534.3690691>
- [180] Mihail L. Sichitiu and Maria Kihl. 2008. Inter-vehicle communication systems: A survey. *IEEE Commun. Surv. Tutor.* 10, 2 (2008), 88–105.

- [181] Martin Simon, Karl Amende, Andrea Kraus, Jens Honer, Timo Samann, Hauke Kaulbersch, Stefan Milz, and Horst Michael Gross. 2019. Complexer-YOLO: Real-time 3D object detection and tracking on semantic point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19) Workshops*.
- [182] Martin Simon, Stefan Milz, Karl Amende, and Horst-Michael Gross. 2018. Complex-YOLO: An Euler-region-proposal for real-time 3D object detection on point clouds. In *European Conference on Computer Vision (ECCV'18) Workshops*.
- [183] Taek-Jin Song, Jongoh Jeong, and Jong-Hwan Kim. 2022. End-to-end real-time obstacle detection network for safe self-driving via multi-task learning. *IEEE Trans. Intell. Transport. Syst.* 23, 9 (2022), 16318–16329.
- [184] Kieran Strobel, Sibo Zhu, Raphael Chang, and Skanda Koppula. 2020. Accurate, low-latency visual perception for autonomous racing: Challenges, mechanisms, and practical solutions. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'20)*. IEEE, 1969–1975.
- [185] Chenyang Sun, Yang Wang, Yanfei Deng, Huaifu Li, Rundong Zhou, and Junqi Guo. 2024. Efficient vehicle-infrastructure collaborative perception based on vehicle re-identification and mini-ICP algorithm. *IEEE Trans. Intell. Transport. Syst.* 25, 7 (2024), 6580–6593. DOI: <https://doi.org/10.1109/TITS.2023.3346214>
- [186] Yuxuan Sun, Xueying Guo, Jinhui Song, Sheng Zhou, Zhiyuan Jiang, Xin Liu, and Zhisheng Niu. 2019. Adaptive learning-based task offloading for vehicular edge computing systems. *IEEE Trans. Vehic. Technol.* 68, 4 (2019), 3061–3074. DOI: <https://doi.org/10.1109/TVT.2019.2895593>
- [187] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*. 1–9.
- [188] Yujie Tang, Nan Cheng, Wen Wu, Miao Wang, Yanpeng Dai, and Xuemin Shen. 2019. Delay-minimization routing for heterogeneous VANETs with machine learning based mobility prediction. *IEEE Trans. Vehic. Technol.* 68, 4 (2019), 3967–3979.
- [189] Hrishikesh M. Thakurdesai and Jagannath V. Aghav. 2021. Autonomous cars: Technical challenges and a solution to blind spot. In *Conference on Advances in Computational Intelligence and Communication Technology (CICIT'19)*. Springer, 533–547.
- [190] Arun Tigadi and M. Chandra. 2023. Literature review on the future of V2X communication in connected and autonomous vehicles. *Int. J. Res. Analyt. Rev.* 10 (2023), 5.
- [191] Manabu Tsukada, Takaharu Oi, Akihito Ito, Mai Hirata, and Hiroshi Esaki. 2020. AutoC2X: Open-source software to realize V2X cooperative perception among autonomous vehicles. In *IEEE 92nd Vehicular Technology Conference (VTC'20-Fall)*. IEEE, 1–6.
- [192] Caglar Tunc, Mustafa F. Özkoç, Fraida Fund, and Shivendra S. Panwar. 2020. The blind side: Latency challenges in millimeter wave networks for connected vehicle applications. *IEEE Trans. Vehic. Technol.* 70, 1 (2020), 529–542.
- [193] Jessica Van Brummelen, Marie O'Brien, Dominique Gruyer, and Homayoun Najjaran. 2018. Autonomous vehicle perception: The technology of today and tomorrow. *Transport. Res. Part C: Emerg. Technol.* 89 (2018), 384–406.
- [194] Tamás Wágner, Tamás Ormándi, Tamás Tettamanti, and István Varga. 2023. SPaT/MAP V2X communication between traffic light and vehicles and a realization with digital twin. *Comput. Electric. Eng.* 106 (2023), 108560.
- [195] Jiadai Wang, Jiajia Liu, and Nei Kato. 2018. Networking and communications in autonomous driving: A survey. *IEEE Commun. Surv. Tutor.* 21, 2 (2018), 1243–1274.
- [196] Jiawei Wang, Yang Zheng, Qing Xu, and Keqiang Li. 2022. Data-driven predictive control for connected and autonomous vehicles in mixed traffic. In *American Control Conference (ACC'22)*. IEEE, 4739–4745.
- [197] Xiaojie Wang, Zhaolong Ning, Xiping Hu, Lei Wang, Bin Hu, Jun Cheng, and Victor C. M. Leung. 2018. Optimizing content dissemination for real-time traffic management in large-scale internet of vehicle systems. *IEEE Trans. Vehic. Technol.* 68, 2 (2018), 1093–1105.
- [198] Yunpeng Wang, Ping Lang, Daxin Tian, Jianshan Zhou, Xuting Duan, Yue Cao, and Dezong Zhao. 2020. A game-based computation offloading method in vehicular multiaccess edge computing networks. *IEEE Internet Things J.* 7, 6 (2020), 4987–4996. DOI: <https://doi.org/10.1109/JIOT.2020.2972061>
- [199] Bichen Wu, Forrest Iandola, Peter H. Jin, and Kurt Keutzer. 2017. SqueezeDet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17) Workshops*. 129–137.
- [200] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. 2018. SqueezeSeg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud. In *IEEE International Conference on Robotics and Automation (ICRA'18)*. IEEE, 1887–1893.
- [201] Yu Wu, Li Yan, and Xuming Fang. 2019. A low-latency content dissemination scheme for mmWave vehicular networks. *IEEE Internet Things J.* 6, 5 (2019), 7921–7933.
- [202] Chao Xiang, Chen Feng, Xiaopo Xie, Botian Shi, Hao Lu, Yisheng Lv, Mingchuan Yang, and Zhendong Niu. 2023. Multi-sensor fusion and cooperative perception for autonomous driving: A review. *IEEE Intell. Transport. Syst. Mag.* 15, 5 (2023), 36–58.

- [203] Zhongyang Xiao, Zhaobin Mo, Kun Jiang, and Diange Yang. 2018. Multimedia fusion at semantic level in vehicle cooperative perception. In *IEEE International Conference on Multimedia & Expo Workshops (ICMEW'18)*. IEEE, 1–6.
- [204] Zhu Xiao, Jinmei Shu, Hongbo Jiang, Geyong Min, Jinwen Liang, and Arun Iyengar. 2023. Toward collaborative occlusion-free perception in connected autonomous vehicles. *IEEE Trans. Mob. Comput.* 23, 5 (2023), 4918–4929.
- [205] Guoqi Xie, Laurence T. Yang, Yuanda Yang, Haibo Luo, Renfa Li, and Mamoun Alazab. 2021. Threat analysis for automotive CAN networks: A GAN model-based intrusion detection technique. *IEEE Trans. Intell. Transport. Syst.* 22, 7 (2021), 4467–4477.
- [206] Kai Xiong, Supeng Leng, Xiaosha Chen, Chongwen Huang, Chau Yuen, and Yong Liang Guan. 2020. Communication and computing resource optimization for connected autonomous driving. *IEEE Trans. Vehic. Technol.* 69, 11 (2020), 12652–12663.
- [207] Xiaolong Xu, Xing Zhang, Xihua Liu, Jielin Jiang, Lianyong Qi, and Md. Zakirul Alam Bhuiyan. 2021. Adaptive computation offloading with edge for 5G-envisioned internet of connected vehicles. *IEEE Trans. Intell. Transport. Syst.* 22, 8 (2021), 5213–5222. DOI: <https://doi.org/10.1109/TITS.2020.2982186>
- [208] Dong Yan, Ke Guan, Danping He, Junhyeong Kim, Heesang Chung, Dao Tian, and Zhangdui Zhong. 2023. Blockage effects of road bridge on mmWave channels for intelligent autonomous vehicles. *IEEE Trans. Intell. Transport. Syst.* 25, 3 (2023), 2908–2919.
- [209] Bin Yang, Wenjie Luo, and Raquel Urtasun. 2018. PIXOR: Real-time 3D object detection from point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*.
- [210] Helin Yang, Xianzhong Xie, and Michel Kadoch. 2019. Intelligent resource management based on reinforcement learning for ultra-reliable and low-latency IoT communication networks. *IEEE Trans. Vehic. Technol.* 68, 5 (2019), 4157–4169.
- [211] Haojun Yang, Kuan Zhang, Kan Zheng, and Yi Qian. 2020. Joint frame design and resource allocation for ultra-reliable and low-latency vehicular networks. *IEEE Trans. Wirel. Commun.* 19, 5 (2020), 3607–3622.
- [212] Haojun Yang, Kan Zheng, Kuan Zhang, Jie Mei, and Yi Qian. 2020. Ultra-reliable and low-latency communications for connected vehicles: Challenges and solutions. *IEEE Netw.* 34, 3 (2020), 92–100.
- [213] Jinrong Yang, Songtao Liu, Zeming Li, Xiaoping Li, and Jian Sun. 2022. Real-time object detection for streaming perception. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*. 5385–5395.
- [214] Qing Yang, Song Fu, Honggang Wang, and Hua Fang. 2021. Machine-learning-enabled cooperative perception for connected autonomous vehicles: Challenges and opportunities. *IEEE Netw.* 35, 3 (2021), 96–101.
- [215] Qing Yang, Alvin Lim, Shuang Li, Jian Fang, and Prathima Agrawal. 2008. ACAR: Adaptive connectivity aware routing protocol for vehicular ad hoc networks. In *17th International Conference on Computer Communications and Networks*. IEEE, 1–6.
- [216] Shiyun Yang, Emily Bailey, Zhengye Yang, Jonatan Ostrometzky, Gil Zussman, Ivan Seskar, and Zoran Kostic. 2020. Cosmos smart intersection: Edge compute and communications for bird's eye object tracking. In *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops'20)*. IEEE, 1–7.
- [217] Lin Yao, Ailun Chen, Jing Deng, Jianbang Wang, and Guowei Wu. 2018. A cooperative caching scheme based on mobility prediction in vehicular content centric networks. *IEEE Trans. Vehic. Technol.* 67, 6 (2018), 5435–5444. DOI: <https://doi.org/10.1109/TVT.2017.2784562>
- [218] Shumayla Yaqoob, Ata Ullah, Muhammad Akbar, Muhammad Imran, and Mohsen Guizani. 2018. Fog-assisted congestion avoidance scheme for internet of vehicles. In *14th International Wireless Communications & Mobile Computing Conference (IWCMC'18)*. IEEE, 618–622.
- [219] Takahito Yoshizawa, Dave Singelée, Jan Tobias Muehlberg, Stephane Delbruel, Amir Taherkordi, Danny Hughes, and Bart Preneel. 2023. A survey of security and privacy issues in V2X communication systems. *Comput. Surv.* 55, 9 (2023), 1–36.
- [220] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. 2018. BisSeNet: Bilateral segmentation network for real-time semantic segmentation. In *European Conference on Computer Vision (ECCV'18)*. 325–341.
- [221] Xiaoyan Yu and Marin Marinov. 2020. A study on recent developments and issues with obstacle detection systems for automated vehicles. *Sustainability* 12, 8 (2020), 3281.
- [222] Farkhanda Zafar, Hasan Ali Khattak, Moayad Aloqaily, and Rasheed Hussain. 2022. Carpooling in connected and autonomous vehicles: Current solutions and future directions. *ACM Comput. Surv.* 54, 10s (2022), 1–36.
- [223] Wenhan Zhan, Chunbo Luo, Jin Wang, Chao Wang, Geyong Min, Hancong Duan, and Qingxin Zhu. 2020. Deep-reinforcement-learning-based offloading scheduling for vehicular edge computing. *IEEE Internet Things J.* 7, 6 (2020), 5449–5465. DOI: <https://doi.org/10.1109/JIOT.2020.2978830>
- [224] Ji Zhang and Sanjiv Singh. 2017. Enabling aggressive motion estimation at low-drift and accurate mapping in real-time. In *IEEE International Conference on Robotics and Automation (ICRA'17)*. IEEE, 5051–5058.
- [225] Ke Zhang, Supeng Leng, Yejun He, Sabita Maharjan, and Yan Zhang. 2018. Cooperative content caching in 5G networks with mobile edge computing. *IEEE Wirel. Commun.* 25, 3 (2018), 80–87. DOI: <https://doi.org/10.1109/MWC.2018.1700303>

- [226] Mingyu Zhang, Jianxin Zhang, Xing Tang, and Yuliang Tang. 2020. Wireless resource pre-allocation for cellular V2I low-latency communications. In *IEEE 91st Vehicular Technology Conference (VTC'20-Spring)*. IEEE, 1–5.
- [227] Qingwen Zhang, Mingkai Tang, Ruoyu Geng, Feiyi Chen, Ren Xin, and Lujia Wang. 2022. MMFN: Multi-modal-fusion-net for end-to-end driving. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'22)*. IEEE, 8638–8643.
- [228] Xumiao Zhang, Anlan Zhang, Jiachen Sun, Xiao Zhu, Y. Ethan Guo, Feng Qian, and Z. Morley Mao. 2021. EMP: Edge-assisted multi-vehicle perception. In *27th Annual International Conference on Mobile Computing and Networking (ACM MobiCom'21)*. 545–558.
- [229] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. 2018. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*. 6848–6856.
- [230] Yuxiao Zhang, Alexander Carballo, Hanting Yang, and Kazuya Takeda. 2023. Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. *ISPRS J. Photogram. Rem. Sens.* 196 (2023), 146–177.
- [231] Liang Zhao, Enchao Zhang, Shaohua Wan, Ammar Hawbani, Ahmed Y. Al-Dubai, Geyong Min, and Albert Y. Zomaya. 2024. MESON: A mobility-aware dependent task offloading scheme for urban vehicular edge computing. *IEEE Trans. Mob. Comput.* 23, 5 (2024), 4259–4272.
- [232] Xi Zheng, Sheng Zhou, and Zhisheng Niu. 2020. Beyond age: Urgency of information for timeliness guarantee in status update systems. In *2nd 6G Wireless Summit (6G SUMMIT'20)*. IEEE, 1–5.
- [233] Fanqin Zhou, Lei Feng, Peng Yu, Wenjing Li, Xiaoyu Que, and Luoming Meng. 2021. DRL-based low-latency content delivery for 6G massive vehicular IoT. *IEEE Internet Things J.* 9, 16 (2021), 14551–14562.
- [234] Jianshan Zhou, Daxin Tian, Yunpeng Wang, Zhengguo Sheng, Xuting Duan, and Victor C. M. Leung. 2019. Reliability-optimal cooperative communication and computing in connected vehicle systems. *IEEE Trans. Mob. Comput.* 19, 5 (2019), 1216–1232.
- [235] Sheng Zhou, Yuxuan Sun, Zhiyuan Jiang, and Zhisheng Niu. 2019. Exploiting moving intelligence: Delay-optimized computation offloading in vehicular fog networks. *IEEE Commun. Mag.* 57, 5 (2019), 49–55. DOI: <https://doi.org/10.1109/MCOM.2019.1800230>
- [236] Xiaoyu Zhu, Yueyi Luo, Anfeng Liu, Md. Zakirul Alam Bhuiyan, and Shaobo Zhang. 2021. Multiagent deep reinforcement learning for vehicular computation offloading in IoT. *IEEE Internet Things J.* 8, 12 (2021), 9763–9773. DOI: <https://doi.org/10.1109/JIOT.2020.3040768>
- [237] Klaus Zindler, Niklas Geiß, Konrad Doll, and Sven Heinlein. 2014. Real-time ego-motion estimation using LiDAR and a vehicle model based extended Kalman filter. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC'14)*. IEEE, 431–438.
- [238] Zhengxia Zou, Rusheng Zhang, Shengyin Shen, Gaurav Pandey, Punarjay Chakravarty, Armin Parchami, and Henry X. Liu. 2022. Real-time full-stack traffic scene perception for autonomous driving with roadside cameras. In *International Conference on Robotics and Automation (ICRA'22)*. IEEE, 890–896.

A Appendix A

The Appendix summarizes most of the papers mentioned in the survey in tabular form, with fine-grained classifications for each layer. The classification attributes include the direction of the paper, the time delay of optimization, proposal, optimization methods, and experiment.

Tables 3 and 4 summarize the existing papers on the delay analysis of the perception layer of CAVs. Tables 5–7 summarize the delay analysis of the communication layer of CAVs. Tables 8–10 summarize the delay analysis of the computation layer of CAVs. Tables 11–13 summarize the delay analysis of CAVs across the communication layer and computation layer.

Table 3. Comparison of Papers Focusing on Single Perception Delay Analysis

Direction	Category	Proposal	Characteristics	Experiment
Dynamic object-based	camera	[199]	Detection pipeline only contains a single forward pass of a neural network	57.2 FPS on NVIDIA TITAN X GPUs in the KITTI dataset
	camera	[22]	Using knowledge distillation and hint learning to balance accuracy-speed	Accuracy-speed tradeoff in the hybrid datasets
	camera	[213]	Dynamic and static flows to capture the moving trend and basic detection feature for streaming prediction	0.8 <i>ms</i> delay on Tesla V100 GPU in Argoverse dataset
	radar	[182]	A specific complex regression strategy to estimate multi-class 3D boxes in Cartesian space	50 FPS on NVIDIA Titan X in the KITTI benchmark
	radar	[134]	Incorporating the shape-aware 2D/3D constraints into 3D detection framework	25 FPS on an NVIDIA V100 GPU in the KITTI dataset
	radar	[181]	A fast parameterizable evaluation metric by designing rotation-translation score to reduce inference time	100 FPS in the KITTI dataset
	radar	[117]	An efficient and effective 3D object detection framework based on 2D convolution	45 FPS on a NVIDIA Tesla V100 in the KITTI dataset
	camera + radar	[166]	A novel CNN architecture fusing both RGB and LiDAR information	18 FPS on a standard desktop GPU in the KITTI dataset
	camera + radar	[116]	An E2E learnable architecture to reason about 2D and 3D object detection	Real-time performance in the hybrid datasets
Static object-based	General obstacle	[55]	A unified deep convolutional network to detect relevant static and dynamic obstacles	33 <i>ms</i> on Nvidia M6000 GPU in the KITTI dataset
	General obstacle	[183]	Learning and refining semantic segmentation and disparity maps from a stereo RGB pair	14.5 FPS on a Nvidia Titan X GPU in fused datasets
	Traffic sign	[61]	A novel system to detect candidate regions as maximally stable extremal regions	20 FPS on Intel Core i5 in an online database
	Traffic sign	[23]	Using high-level features and utilizing receptive field block cross to capture contextual information	> 30 FPS on Tesla V100 GPUs in the TT100K dataset
Panoramic object-based	Semantic segmentation	[200]	Formulating point-wise classification and inputting LiDAR point cloud and outputting a point-wise label map	8.7 $\pm$ 0.5 <i>ms</i> on TITAN X GPUs in the KITTI dataset
	Semantic segmentation	[220]	A novel bilateral segmentation network-based spatial path and context path	105 FPS on NVIDIA Titan XP in the Cityscapes dataset
	Panoramic segmentation	[74]	A new single-shot panoptic segmentation network	30 FPS on Tesla V100 GPU in the hybrid datasets
	Panoramic segmentation	[72]	Designing a parameter-free head to merge the per-pixel bbox and semantic prediction from feature map	Real-time performance in the hybrid datasets
	Instance segmentation	[160]	A novel real-time instance segmentation method that segments objects using bounding polygons	Real-time performance on GTX 1080Ti
	Semantic segmentation	[48]	Reducing the dimension of feature maps and using the aggregation of them for image representation	250.4 FPS on GTX 1080Ti in the Cityscapes dataset
	Panoramic segmentation	[109]	A fast and high-performance LiDAR by introducing a clustering pseudo heatmap and a KNN-transformer module	6 times faster than baselines in the KITTI dataset
CAV's state-based	motion estimation	[224]	A sequential, multi-layer processing pipeline, solving for low-latency ego-motion estimation from coarse to fine	Real-time performance on a laptop computer
	motion estimation	[237]	A method for the real-time estimation of vehicle position and of further longitudinal and lateral dynamic state variables	Almost no time delay
	motion estimation	[85]	A GPU-based 3D ego-motion estimation method	40 FPS on GeForce GTX 480 GPU
	navigation	[171]	Formulating motion planning problem as a nonlinear optimal control problem and obtaining the optimal solution	Better real-time performance than baselines on GTX750Ti
	navigation	[103]	Constructing a real-time navigation system-based DRL by formulating tasks as a series of decisions	Improving 30.4% for traveling time
	navigation	[21]	Defining the optimal navigation path as a sequential list of directed links with minimum cost	Less navigation time than baselines

Table 4. Comparison of Papers Focusing on Collaborative Perception Delay Analysis

Direction	Category	Proposal	Characteristics	Experiment
Vehicle-vehicle collaborative perception	Object sharing	[128]	Using neural network pruning technique to reduce the detection time and a novel variational autoencoder to reduce object feature size in feature extraction	Reducing 34.1% average latency than baselines
	Feature sharing	[203]	Merging semantic bird's-eye view of multiple vehicles	Less data transfer
	Feature sharing	[26]	Using feature level fusion to realize E2E 3D object detection and supporting voxel and spatial feature fusion	Achieving a 30%–40% detection precision increase
	Data sharing	[179]	Settings with minimal point cloud overlap (both in the spatial and temporal sense) and developing techniques to minimize error and computation time in multi-way registration	Generating highly accurate reconstructions within minutes
Vehicle-edge collaborative perception	Data sharing	[228]	Edge server merges views from multiple vehicles to form a more complete bird's-eye view with a higher resolution	Achieving 24 FPS and E2E latency of 93 <i>ms</i> on average
	Data sharing	[38]	Proposing an algorithm of computing offloading and resource allocation based on sensing data fusion to deal with real-time data of edge servers	Less task completion delay
	Feature sharing	[124]	Edge server merges key frames from multiple vehicles and generates 3D feature maps and then sends to the vehicles	Reducing E2E latency by 40.09%
	Feature sharing	[185]	A mini-iterative closest point algorithm that can automatically select feature points and perform point-cloud registration	Feature matching time could be accelerated by 13.94%–72.32%

Table 5. Comparison of Papers Focusing on Direct Communication Delay Analysis

Direction	Category	Proposal	Characteristics	Experiment
V2V communication	Content retrieval latency	[201]	Increasing the amount of data transmitted each time based on mmWave	Average content retrieval latency less than baselines
	Data transmission delay	[211]	The optimal frame design to maximize the information transmitted between all V2V pairs	0.332 <i>ms</i> optimal transmission delay
	Communication latency	[8]	Dynamically selecting optimal transmission channel by estimating communication delay	23.4% lower packet delay than baselines
	Communication switching delay	[144]	Avoiding major delay components by pre-registering vehicles with multiple carriers	20 <i>ms</i> in the majority of locations
	Queueing delay	[7]	Using Lyapunov stochastic optimization to optimize resource block in each region	Average queueing latency reduction up to 97%
	Resource contention delay	[80]	Selecting the optimal receiving vehicle to determine the V2V link	Total delay less than baselines
	Beamforming overhead	[94]	A system that creates 3D reflection profiles from street view images	Lower beamforming overhead than baseline
	Beam training overhead	[60]	Using a passive radar array to sense automotive radar transmissions coming from multiple vehicles on the road	A significant improvement for beam training overhead
	Beam sweeping overhead	[11]	A new beam alignment method to select the optimal beam angle of departure from the mmWave base station	Minimal computational complexity
	Blockage durations	[192]	Establishing a continuous-time Markov chain model of the blockage events	Eliminating long blockages and improving latency performance
V2I communication	Communication switching delay	[86]	Using a delay estimation model to determine the best available path	Average latency less than baselines
	Access latency	[32]	A time-coordinated multiple access scheme to improve V2I communication and handoff	Service interruption time less 10%–28.3% than baselines
	Communication switching delay	[13]	A deterministic MAC method for real-time communication based on V2I	Total latency less than baselines
	Queueing delay	[226]	Low latency resource preallocation algorithm based on Markov prediction model	User rejection rate less than baselines
	Transmission delay	[50]	Two-phase radio resource allocation problem for physical layer delay guarantee	The tradeoff among rate, latency, and reliability
	Transmission delay	[162]	Extracting and transmitting low-dimensional intermediate features to lower the overhead of inter-vehicle messaging	Yielding up to 69.8% and 20.42% improvement in latency and throughput

Table 6. Comparison of Papers Focusing on Indirect Communication Delay Analysis

Direction	Category	Proposal	Characteristics	Experiment
Complementary V2X-based multi-hop communication	E2E latency	[34]	Quantifying the E2E delay experienced by V2N and V2N2V communications in different 5G network deployments	E2E latency less than baselines
	E2E latency	[33]	Reducing E2E latency under network deployments via mobile edge computing federation	E2E latency less than baselines
	Communication link switching latency	[114]	Incorporating integrated sensing and communications signals in the NR-V2X system	Employing 75% fewer pilots and reduce reference signals by 43.24%
Routing-based multi-hop communication	E2E latency	[175]	Anchor-based street- and traffic-aware routing	E2E latency less than baselines
	E2E latency	[88]	Improved greedy traffic-aware routing protocol	E2E latency less than baselines
	E2E latency	[87]	Estimating the delay and select the best available path by obtaining vehicle data flow statistics	E2E latency less than baselines
	E2E latency	[96]	Routing by using weights associated with neighboring nodes	E2E latency less than baselines
	E2E latency	[30]	A hybrid geographic routing solution by using delay-tolerant forwarding	E2E latency less than baselines
	E2E latency	[135]	Designing a truly two-dimensional system area by GPS	Less latency of first packet per 'connection' than baselines
	E2E latency	[136]	A greedy perimeter coordinator routing algorithm without using any global or external information	E2E latency less than baselines
	E2E latency	[151]	Integrating the connection path between the integration destination and the source destination and dynamic adjustment	Less the average data packet delays than baselines
	E2E latency	[215]	Adaptive selecting the optimal route with the best network transmission quality	Less network delay than baselines

Table 7. Comparison of Papers Focusing on Hybrid Communication Delay Analysis

Direction	Category	Proposal	Characteristics	Experiment
V2V or V2I	Transmission delay	[122]	Dividing a scheduling cycle into three phases: V2I, V2V, V2I, or V2V	Less the mean service delay than baselines
	Transmission delay	[233]	Selecting one of the three communication modes: V2N, V2I, V2V	The performance supremacy of transmission delay
	Transmission delay	[218]	A distributed approach by continuously updating vehicle status	0.1305 s E2E message transmission delay
	Transmission delay	[80]	Pre-caching some services on the upcoming RSU but not the current RSU	The performance supremacy of transmission delay
V2V and V2I	Communication delay	[65]	Reusing the optimal power allocation of a pair of V2I links and a V2V link by considering each possible spectrum	Guaranteeing the V2V link latency requirement
	Communication delay	[66]	Using the effective capacity theory and the delay index to express the delay constraint explicitly	Guaranteeing the V2V link latency requirement
	Communication delay	[210]	Formulating the joint communication mode selection problem as a decentralized RL framework	Achieving the fast convergence speed
	Communication delay	[62]	An intelligent algorithm for the beamforming design at the reconfigurable intelligent surface	Guaranteeing the V2V link latency requirement
	Access delay	[43]	Using non-orthogonal multiple access to reduce access latency	Less access delay than the traditional orthogonal multiple access scheme
	Access delay	[81]	Avoiding large-scale exchange of control information and signaling based on the vehicle-centered method	Guaranteeing the upper bound of the latency
	Link switching delay	[188]	Making further routing decisions independently based on the successful transmission probability and average delay of each vehicle request	Less average delay than baselines
	Link switching delay	[44]	Using the coordination capability of SDN to determine the optimal V2V multi-hop path	Less average packet delay than centralized routing algorithms
	Link switching delay	[64]	Analyzing quantitatively the data transmission delay caused by dynamic factors	Less data delivery delay than baselines

Table 8. Comparison of Papers Focusing on Vehicle-vehicle Collaborative Computing Delay Analysis

Direction	Category	Proposal	Characteristics	Experiment
Static scenario	E2E delay	[9]	The weighted round robin scheduler to overcome bufferbloat and distribute the data better	0.822 data cargo/s and 3 times than other aggregated throughput
	Offloading delay	[153]	Integrating a container orchestration into a computational offloading framework to run more task replicas	64% improvement in terms of average time cost
	Offloading delay	[140]	Jointing task scheduling and resource allocation	Lower service delay than baselines
	Total computing delay	[19]	Modeling the corresponding selection relationship between task set and candidate vehicle set as a generalized allocation model	0.2 s-2.4 s latency reduction than local computing and 0.16 s-2.3 s latency reduction than random offloading
Dynamic scenario	System delay	[51]	Efficient job caching based on adjacent vehicles	95% of the utility of the optimal solution within fewer than 10 iterations
	Average offloading delay	[186]	Adaptive learning of offloading delay performance of adjacent vehicles	Decreasing the average delay up to 30% compared with baselines
	The job total latency	[77]	A greedy-based task scheduling algorithm for offloading jobs	Lower the response time of jobs than baselines
	Task execution delay	[37]	Realizing self-update of offloading strategy based on DRL according to previous experience	Reducing the total system delay up to 69.4% compared with local execution strategy
	Learning latency	[108]	Constructing Markov decision process to jointly optimize participant selection and resource allocation problems	Lower the training delay than baselines
	Learning latency	[129]	Jointly optimize compression ratio, batch size, and spectrum allocation	Improving the learning performance by maximizing the convergence rate under a constrained latency

Table 9. Comparison of Papers Focusing on Vehicle-edge Collaborative Computing Delay Analysis

Direction	Category	Proposal	Characteristics	Experiment
Single-edge collaborative computing	E2E service delay	[104]	Prediction of resource availability and service requirements of edge nodes based on reinforcement learning	Higher user satisfaction than baselines
	Average task delay	[223]	Markov decision process based on proximal policy optimization	Relatively small delays than baselines
	Offloading delay	[138]	Solving a multi-objective optimization problem to minimize both the delay and cost by jointly considering the offloading decision, allocation of communication and computation resources	A optimal tradeoff between delay and cost
Multiple-edge collaborative computing	Offloading delay	[93]	Optimizing the resource utilization of edge nodes based on local and global load-balancing technology	Lower average response times than baselines
	Average service delay	[111]	A task partition and scheduling scheme based on Markov decision process to divide the computing workload among edge servers	A near-optimal task scheduling solution with low time-complexity
	System delay	[45]	A biased-randomized algorithm to solve optimal mapping between vehicles and multiple RSUs	Feasible and high-quality solutions in real-time

Table 10. Comparison of Papers Focusing on Hybrid Computing Delay Analysis

Direction	Category	Proposal	Characteristics	Experiment
Offload task to vehicle or edge	Average offloading delay	[235]	Mobility prediction based on model or reinforcement learning	Low complexity and fast convergence
	Task computing delay	[132]	Obtaining the optimal strategy of computing offloading and resource allocation based on Q-learning and DRL	Less delay than baselines
	Service latency	[106]	Obtaining an efficient solution of resource allocation decision problem based on heuristic algorithm	Less delay than baselines
	Offloading system delay	[89]	Formulating offloading and resource allocation scheme as a mixed-integer non-linear problem	Delay within the maximum allowable latency requirement
	Task execution delay	[40]	Designing a unified solution to minimize task execution time under different multi-task offloading scenarios	Reducing task execution time by 11.36%, on average
	Average response delay	[231]	The minimization of the average response time and average energy consumption of the system is defined as a combinatorial optimization problem	Superior average response time compared to baselines
Offload task to vehicle and edge	Access delay	[217]	Caching popular content on a set of mobile nodes that may repeatedly visit the same hotspot	Less average access delay than baselines
	Service delay	[225]	The two-sided matching method of computation task assignment	Less formation delay and perception reaction time than baselines
	Service delay	[75]	Designing a low complexity fault-tolerant particle swarm optimization algorithm by considering the mechanism of partial offloading, task allocation, and reprocessing	Less service than baselines
	Task processing delay	[49]	Designing a generalized benders decomposition and reformulation linearization-based iterative algorithm	Near-optimal solutions with low computational complexity

Table 11. Comparison of Papers Focusing on Offloading Decision

Direction	Category	Proposal	Characteristics	Experiment
Offloading decision	Computing offloading delay	[131]	Describing the multi-vehicle computing offloading problem as a multi-user computing offloading game problem	Less computing overhead than baselines
	Computing offloading delay	[198]	Proposing a multi-user non-cooperative computing offloading game to obtain the best response of each vehicle	Meeting the requirement of task execution latency
	Offloading decision delay	[139]	A distributed computing offloading scheme based on self-learning to minimize the joint cost of delay and offloading	Achieving at least an order-of-magnitude improvement on time overhead and 88% performance gain on message overhead
	Execution delay of composite tasks	[20]	A distributed computing offloading strategy based on deep Q-learning network to find the best offloading method	Less execution time of the computing tasks than baselines
	Total task processing delay	[236]	A centralized training algorithm and distributed execution algorithm based on multi-agent DRL to select the appropriate MEC server to execute the task of multiple vehicles	Decreasing delay 10.2%–21.7% compared with baselines
	Task processing delay	[152]	Proposing an Age of Processing (AoP)-aware offloading mechanism for autonomous vehicles and formulating an optimization problem that minimizes AoP	Meeting computation deadlines while minimizing AoP

Table 12. Comparison of Papers Focusing on Resource Allocation

Direction	Category	Proposal	Characteristics	Experiment
Resources allocation	Packet transmission delay	[149]	Deep distributed Q network based on generative adversarial network to learn the action value distribution of intelligent transmission scheduling	Higher packet transmission rate and lower transmission delay than baselines
	Packet transmission delay	[79]	A method-based graph theory and heuristics to minimize data transfer to server	Obtaining the near-optimal solution
	Task processing delay	[41]	Developing a low-complexity algorithm to jointly optimize offloading decision and resource allocation	Better performance than baselines
	Task offloading delay	[24]	Using mixed integer nonlinear programming to solve task placement and resource allocation subproblems	Reducing 20% of task duration
	E2E delay	[83]	Software-defined CAVs based on horizontal and vertical traffic offloading	Less average latency than baselines

Table 13. Comparison of Papers Focusing on Resource Provisioning

Direction	Category	Proposal	Characteristics	Experiment
Resource Provisioning	Task completion delay	[39]	A multi-armed bandit learning algorithm to learn real-time workload distribution online	Better performance than baselines
	Task transmission delay	[207]	Designing an adaptive computing offloading method based on decomposition multi-objective evolutionary algorithm	Less average transmission time than baselines
	Computing offloading delay	[158]	Evaluating the availability of parked vehicle based on the prediction probability to select computing offloading location	Less computing offloading delay than baselines
	Content access delay	[73]	Caching service content in the RSUs	Better performance than baselines
	Content access delay	[163]	Cooperative caching in a macro-cell station, roadside units, and smart vehicles	Suboptimal solution with low computational complexity
	Content access delay	[154]	Caching service content in RSUs and vehicles	Lower network delay in diverse traffic scenarios
	Average transmission delay	[90]	Proposing an adaptive cooperative caching strategy to adapt the frequent network changes and describing an optimization problem to minimize the average transmission delay	Less the average transmission delay than baselines

Received 1 January 2023; revised 22 December 2024; accepted 22 January 2025