UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Road-Assisted Cooperative Model Training and Inference for Perception in Intelligent Networked Vehicular Systems

by Jianjun Chen

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of A/Prof. Haiyan Lu, Prof. Qi Hao, A/Prof. Shuai Wang, A/Prof. Elvis Liu

University of Technology Sydney
Faculty of Engineering and Information Technology

Mar 2025

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Jianjun Chen, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the *Faculty of Engineering and Information Technology* at the *University of Technology Sydney*.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree at any other academic institution except as fully acknowledged within the text. This thesis is the result of a Collaborative Doctoral Research Degree program with the Southern University of Science and Technology.

This research is supported by the Australian Government Research Training Program.

SIGNATURE: Production Note: Signature removed prior to publication.

DATE: 10 Feb, 2025

Dedication

To my lovely family

my parents

my relatives

my friends

For their love and support

ACKNOWLEDGMENTS

To everyone who has helped, supported, encouraged, accompanied, motivated, inspired, loved, challenged, and questioned me during my Ph.D., this thesis is dedicated to you all.

First and foremost, I would like to express my sincere gratitude and appreciation to my supervisors, A/Prof. Haiyan Lu, and my co-supervisor, Prof. Qi Hao, A/Prof. Elvis Liu. Over the years, their unwavering support, expert guidance, and constant encouragement have been instrumental to my academic and personal growth. Prof. Haiyan Lu's remarkable attention to detail, rigorous approach to research, and insightful guidance have been a source of constant inspiration. Prof. Qi Hao's profound expertise, innovative thinking, and ability to connect theory to practical applications have broadened my horizons and enriched my research pursuits. A/Prof. Elvis Liu's supportive mentorship, pragmatic advice, and encouragement to think critically have motivated me to explore challenging problems and persist through obstacles. Their dedication to excellence, patience in mentoring, and selfless support have left an indelible mark on my Ph.D. journey, for which I am deeply grateful.

Additionally, I wish to convey my gratitude to my co-supervisor, A/Prof. Shuai Wang, for his valuable feedback, intellectual rigor, and guidance during the final stages of my research. His insights and collaborative spirit have been pivotal in shaping the outcomes of this thesis. I also extend my heartfelt thanks to the Southern University of Science and Technology for providing the full scholarship, which made my academic journey possible.

Secondly, I would like to extend my heartfelt thanks to the colleagues of my research group, as well as collaborators and friends who have provided technical support, insightful discussions, and camaraderie. They are, but not limited to, Dr. Shuaijun Wang, Dr.

Chenguang Liu, Dr. Ruihua Han, Dr. Shuai Zhang, Rui Gao, Dr. Jie Ma, Dr. Adi Lin, Pengjie Liu. Special thanks go to Dr. Chenguang Liu for his mentorship and guidance in both academic and professional matters, as well as Dr. Shuaijun Wang for his collaborative efforts and valuable insights. It has been a privilege to work alongside such talented, kind-hearted, and inspiring individuals. Their unwavering support, positivity, and encouragement have made my Ph.D. experience truly fulfilling and enjoyable.

Additionally, I would like to acknowledge the teams at UTS, including staff at the Australian Artificial Intelligence Institute, School of Computer Science, Faculty of Engineering and Information Technology, iHPC, GRS, and the Library, for their efficient support and dedication to creating an environment conducive to learning and research.

Last but most importantly, I express my deepest gratitude to my family for their unconditional love, unwavering support, and belief in me. To my wife, who has stood by my side through every challenge, your patience, encouragement, and sacrifices have been the foundation of my strength and perseverance. To my parents, whose dedication and love have shaped who I am today, I am forever indebted. To my extended family, especially my father-in-law, your wisdom, kindness, and unwavering faith in me have been a constant source of motivation.

To all who have been part of this journey, from mentors to friends to family, thank you from the bottom of my heart. This milestone would not have been possible without you.

ABSTRACT

The perception module in an autonomous driving (AD) system strives to accurately represent the surrounding environment. This component plays a crucial role in the realization of autonomous vehicles (AVs). By fusing data from various AV on-board sensors, it facilitates a more accurate and comprehensive representation of surrounding driving environments, overcoming the limitations of relying on a single sensor. However, ensuring accurate and reliable perception in dynamic and complex traffic scenarios remains a significant challenge. First, model uncertainty, stemming from the vast and infinite range of traffic scenarios, must be addressed from both short-term and long-term perspectives to enhance robustness. Second, perception uncertainty, caused by occlusions and long distance objects, must be handled in time to help AVs adapt to the dynamic and densely clustered driving environments. Finally, as connected autonomous vehicles (CAVs) and intelligent roadside units (IRSUs) collaborate via wireless networks to share knowledge for robust perception, it is also critical to account for wireless communication constraints, such as latency, bandwidth limitations, and signal distortion.

To address these challenges, this thesis develops a series of methods for robust perception in intelligent networked vehicular systems, focusing on road-assisted cooperative model training and inference.

To mitigate uncertainty within perception models, a road-assisted cooperative training framework is introduced, along with a road supervised data annotation algorithm for newly collected out-of-distribution (OOD) data. Additionally, a roadside sensor placement

algorithm is developed to facilitate optimal knowledge sharing between CAVs and IRSUs based on learning requirements.

To reduce the communication latency among CAVs and IRSUs during the cooperative training process, a network topology optimization algorithm is devised to minimize latency under varying network conditions.

For improving perception robustness, this thesis incorporates diverse V2V communication channel models, including Rician fading, WINNER II, and non-stationary time-varying V2V channels, into the cooperative inference system, providing a systematic analysis of performance degradation due to communication impairments. Building upon this analysis, a joint weighting and denoising framework is developed to correct both CAV-level and pixel-level feature distortions, thereby enhancing the resilience of shared intermediate features in cooperative perception.

Extensive evaluations using the CARLA (Dosovitskiy et al., 2017) simulator and real-world datasets demonstrate that the proposed cooperative learning solutions along with the roadside sensor placement algorithm consistently outperform baseline approaches, achieving up to a 16% improvement in perception accuracy, and the proposed joint weighting and denoising algorithm for cooperative inference achieves at least 38% robustness gain under challenging wireless channel conditions. These findings contribute to the advancement of efficient cooperative training and inference in AD systems, providing a scalable and adaptable framework for improving perception reliability in real-world traffic environments.

List of Publications

Journal

[J1]. **J, Chen.,** S, Wang., C, Liu., D, Ng., C, Xu., Q, Hao., H, Lu. (2024). Road Supervised Federated Learning with Bug-Aware Sensor Placement. *IEEE Transactions on Vehicular Technology*, vol. 73, no. 12, pp. 19762-19767.

[J2]. **J, Chen.,** C, Liu., W, Shuai., Y, He., Z, Wei., Y, Chen., H, Sun., Q, Hao., H, Lu. Enhancing Cooperative Perception with Robust V2V Communications: A Joint Weighting and Denoising Approach. (*Under revision of TWC*).

[J3]. **J, Chen.,** C, Liu., H, Tang., Z, Wei., H, Lu., Q, Hao., H, Sun. Road Assisted Federated Learning For Cooperative Perception With Topology Optimization In V2X Communication. (*Under revision of WCL*).

[J4]. C, Liu., **J, Chen.,** Y, Chen., R, Payton., M, Riley., S, Yang. (2024). Self-supervised Adaptive Weighting for Cooperative Perception in V2V Communications. *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 2, pp. 3569-3580.

[J5]. C, Liu., Y, Chen., **J, Chen.,** R, Payton., M, Riley., S, Yang. (2023). Cooperative Perception with Learning-Based V2V Communications. *IEEE Wireless Communications Letters*, vol. 12, no. 11, pp. 1831-1835.

[J6]. S, Wang., R, Gao., R, Han., **J, Chen.,** Z, Zhao., Z, Lyu., Q, Hao. (2024). Active Scene Flow Estimation for Autonomous Driving via Real-Time Scene Prediction and Optimal Decision. *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 6, pp. 5997-6012, 2024

[J7]. R, Han., S, Wang., S, Wang., Z, Zhang., **J, Chen.,** S, Lin., C, Li., C, Xu., Y, Eldar., Q, Hao., J, Pan. (2025). NeuPAN: Direct Point Robot Navigation with End-to-End Model-based Learning. *IEEE Transactions on Robotics*.

[J8]. G, Li., W, Ni., M, Ding., Y, Qu., **J, Chen.,** D, Smith., W, Zhang., T, Rakotoarivelo. (2024). Decentralized Privacy Preservation for Critical Connections in Graphs. *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 5911-5925.

Conference

[C1]. Z, Xie., **J, Chen.,** G, Li., W, Shuai., K, Ye., Y, Eldar., C, Xu. (2025) Clutter Resilient Occlusion Avoidance for Tight-Couple Motion-Assisted Detection. *IEEE International Conference on Acoustics, Speech, and Signal Processing*.

TABLE OF CONTENTS

List of Publications **vii**

List of Figures **xii**

List of Tables **xiv**

Abbreviation **xv**

1 Introduction **1**

1.1 Background 1

1.2 Research Motivations and Contributions 5

1.2.1 Road Supervised Federated Learning with Bug-Aware Sensor Placement 5

1.2.2 Communication Topology Optimization for FLCAV 7

1.2.3 Enhancing Cooperative Inference with practical V2V communications: A Joint Weighting and Denoising Approach 8

1.3 Thesis Outline 10

2 Literature Review **12**

2.1 Architectures for the Vehicular Networks 12

2.1.1 Mobile Cloud Computing 13

2.1.2 Mobile Edge Computing 13

2.2 Machine Learning Enabled Perception for Autonomous Driving 14

2.2.1 3D Object Detection for Autonomous Driving 14

2.2.2 Simulator for Traffic Scenarios Generation and Data Collection . . 18
2.2.3 Public Dataset for Perception 19
2.3 Cooperative Model Training for Perception in Networked Vehicular Systems 22
2.3.1 Federated Learning for CAVs 23
2.3.2 Knowledge Distillation among CAVs and IRSUs 26
2.3.3 Task Offloading for Wireless CAV Systems 27
2.4 Cooperative Model Inference for Perception in Networked Vehicular Systems 28
2.4.1 Fusion Schemes for Cooperative Inference 29
2.4.2 Cooperative inference with IRSUs assistant 32
2.4.3 Cooperative inference with V2V communication 35
2.5 Research Gap 39

3 Road Assisted Cooperative Model Training for Perception 43
3.1 Introduction 43
3.2 RSFL System Model 46
3.3 Proposed RSFL Approach 48
3.3.1 BARL: Bug Aware Road Labeling: 48
3.3.2 Information Gain of BARL: 50
3.4 BASP: Bug-Aware Sensor Placement: 51
3.5 Experiment Analysis 55
3.6 Summary 57

4 Cooperative Model Training for Perception in Networked Vehicular Systems 59
4.1 Introduction 59
4.2 System Model 61
4.2.1 Dataset Collection and Processing 62
4.2.2 Federated Learning Assisted Cooperative Training Scheme 63
4.3 Dynamic Communication Topology Optimization 66
4.3.1 Model Aggregation and Communication Model 66

4.3.2 Topology Optimization Algorithm 69
4.4 Simulation and Numerical results 70
4.5 Summary 74

5 Cooperative Model Inference for Perception in Networked Vehicular Systems 75

5.1 Introduction 75
5.2 System Model 78
5.2.1 V2V Communication Models 79
5.3 Cooperative inference with End-to-end Communicaiton 86
5.3.1 Conventional Fusion Schemes with V2V Communication 87
5.3.2 Numerical Results and Discussion 89
5.4 Joint Adaptive Weighting and Denoising Approach 91
5.4.1 CAV-level Weighting 92
5.4.2 Pixel-level Denoise 94
5.4.3 Joint Training Scheme for Weighting and Denoising 99
5.4.4 Numerical Results and Discussion 101
5.5 Summary 109

6 Conclusion and Future Work 110

6.1 Conclusion 110
6.2 Limitations and Future Work 111
6.2.1 Real-world Roadside Infrastructures Deployment and Validation . 112
6.2.2 Task Offloading and Model Partition for Cooperative Inference . . 113
6.2.3 Multi-modality Cooperative Perception 114
6.2.4 Active Perception 115
6.2.5 Privacy Preserving for Cooperative Inference 115
6.2.6 V2X Communication Channel Models for Cooperative Perception . 116
6.2.7 Physical Experiment in Real-world Devices 116

Bibliography 118

List of Figures

Figure **Page**

1.1 The collaboration among CAVs and IRSUs in the ITS. 2

2.1 Visualization of point cloud and detected bounding boxes of surrounding objects. . . 15

3.1 RSFL framework, which consists of the BASP and BARL modules. Road-detected boxes are marked in red. Vehicle-detected boxes are marked as cyan. 48

3.2 Comparison between BASP and TSP in 7 different Scenarios. Scenarios BEV Images, bug data distributions, Sensor placements of BASP, and sensor placements of TSP. . 55

3.3 Qualitative comparison of different schemes. Red, pink, yellow, and cyan boxes represent results obtained from ground truth, pre-trained, TSP-RSFL, and BASP-RSFL schemes, respectively. 57

4.1 System framework of road-assisted topology optimization for FLCAV. 61

4.2 Visualization of point clouds and front view of RGB image from CAVs perspective. . 62

4.3 BEV of point clouds and RGB image from IRSUs perspective. 63

4.4 An illustration of the communication topology between the IRSU and CAVs for FLCAV in the networked vehicular system 67

4.5 Birds-Eye-Views of the three traffic scenarios (i.e., T-junction, roundabout and crossroad). 71

4.6 An illustration of the adaptive time-varying communication topology between the IRSU and CAVs for FLCAV in the networked vehicular system 72

4.7 Qualitative comparison of different cooperative training schemes. Red, pink, cyan, and yellow boxes represent results obtained from ground truth, pre-trained, FTFL, and DTFL schemes, respectively. 73

5.1 Cooperative inference via V2V communication. 78
5.2 System architecture of cooperative inference with v2v communication. 86
5.3 Cooperative inference with V2V communications: (a) Raw-data-level fusion, (b) Intermediate-level fusion, (c) Object-level fusion. 88
5.4 CAV-level weighting processes. 92
5.5 Pixel-level diffusion and denoising processes. 94
5.6 The proposed joint weighting and denoising algorithm pipeline. 100
5.7 Average precision under various channels. (a) Rician fading. (b) WINNER II. (c) Non-stationary V2V channel. 104
5.8 Average precision with imperfect CSI. 105
5.9 Average precision with different path loss factors. 106
5.10 Performance under time-varying disturbances, simulating disturbances with a fixed time duration following Gaussian distribution. (a) Time-varying noise levels (σ_{SNR}). (b) Time-varying CSI errors(σ_{CSI}). 107
5.11 Visualization examples of Coop-D (top) and the Coop-WD (bottom). (a) False positive correction. (b) False negative correction. 107
5.12 Visualization examples of Coop-W (top) and Coop-WD (bottom). (a) False positive correction. (b) False negative correction. 108

List of Tables

Table **Page**

2.1 Comparisons between CAV and IRSU 33

3.1 Comparison of federated learning methods for autonomous driving. 45
3.2 Comparison of mAPs for different schemes. 58

4.1 Comparison of mAP for different cooperative learning schemes 74

5.1 Average precision under fading and noise conditions. 90
5.2 Average precision under path loss, fading, and noise conditions. 90
5.3 Comparison of runtime and size of the baseline without weighting and denoising, Coop-W, Coop-D, and Coop-WD. 109

Abbreviation

ITS	Intelligent transportation system
AD	Autonomouse driving
AV	Autonomous vehicle
CAV	Connected autonomous vehicle
IRSU	Intelligent roadside unit
RSI	Roadside infrastructure
V2V	Vehicle-to-vehicle
V2I	Vehicle-to-infrastructure
V2X	Vehicle-to-everything
DNN	Deep neural network
FL	Federated learning
FEEL	Federated edge learning
FLAD	FL assisted AD
FLCAV	FL assisted CAV
RSFL	Road supervised FL
OOD	Out-of-distribution
BASP	Bug aware sensor placement
TSP	Topology sensor placement
SISO	single-input single-output
MIMO	Multi-input multi-output
ML	Machine learning
DL	Deep learning

MCC	Mobile cloud computing
MEC	Mobile edge computing
LiDAR	Light detection and range
BEV	Birds eye view
VFE	Voxel feature extractor
PFE	Pillar feature extractor
RPN	Region proposal net
SSD	Single shot detector
ConvGRU	Convolutionall gated recurrent unit
SGD	Stochastic gradient descent
ReLU	Rectified linear unit
GNN	Graph neural network
KD	Knowledge distillation
EAD	Ensemble attention distillation
EMA	Ensemble moving average
MVFD	Multi-view fusion distillation
BARL	Bug aware road labeling
FPC	False positive correction
FNC	False negative correction
IBC	Inaccurate box correction
FTFL	Fixed topology FL
DTFL	Dynamic topology FL
AWGN	Additional white Gaussian noise
CSI	Channel state information
FLOPs	Floating points operations
IoU	Intersection of union
LoS	Line-of-sight
NLoS	Non-line-of-sight
NMS	Non-maximum suppression

SNR	Signal to noise ratio
PDF	Probability density function
VM	von Mise distribution
DDPM	Denoising diffusion probabilistic model
KL	Kullback-Leibler divergence
ELBO	Evidence lower bound
AP	Average Precision

CHAPTER 1

INTRODUCTION

1.1 Background

Intelligent transportation systems (ITSs) (Kuo & Choi, 2024), which aim to enhance road safety and optimize traffic flow and efficiency, have garnered significant attention in both academic and industrial communities in recent years. A key enabler of the ITS is the wireless vehicular network (Abdelkader et al., 2021), which consists of two primary components: connected autonomous vehicles (CAVs) (Tsukada et al., 2020) and intelligent roadside units (IRSUs) (Shan et al., 2020). CAVs and IRSUs are equipped with multi-modal sensors, including LiDAR, radar, and cameras; mobile computing units for real-time data analysis; and advanced communication units, which enable both CAVs and IRSUs to connect and share information through Vehicle-to-Vehicle (V2V) or Vehicle-to-Everything (V2X) communication (Bréhon–Grataloup et al., 2022). The IRSUs act as infrastructure-based sensors and communication hubs, while the CAVs utilize both on-board sensors and received information from other road participants to make decisions. For instance, Figure 1.1 illustrates a collaboration scenario involving multiple CAVs and IRSUs. The left subfigure shows a top-down view of a T-junction scenario where CAVs and IRSUs collaborate by sharing perception information via wireless communication.

The right subfigures provide detailed views of the 2D sensors used by the CAVs and IRSUs. This collaboration enables CAVs to receive information beyond their Field of Views (FoVs), reducing blind spots and enhancing situational awareness. As a result, it improves road safety and promotes more efficient traffic management.

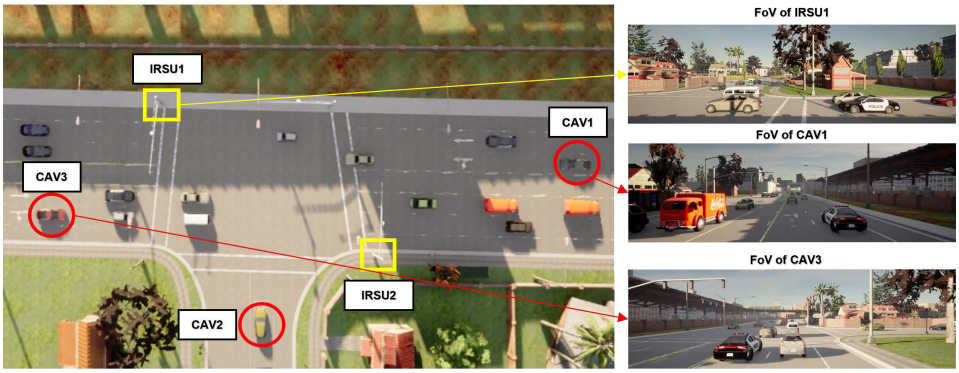


Figure 1.1: The collaboration among CAVs and IRSUs in the ITS.

Built upon the hardware devices on CAVs and IRSUs, the interaction between CAVs and their surrounding environment comprises three fundamental functionalities: 1) *cooperative perception*, which aims to observe the surrounding environment and convert it into machine-readable information; 2) *CAV navigation*, which generates target routes and collision-free motion plans based on perception data; and 3) *CAV control*, which generates driving signals such as throttle and brake. Among these functionalities, cooperative perception is crucial to the safety of autonomous driving (AD), as uncertainties in perception results can be amplified and transformed into control uncertainties, potentially leading to catastrophic outcomes (Eskandarian et al., 2019). Thus, it is imperative to develop robust and adaptive perception models that can handle the complexities of real-world traffic scenarios.

Given the importance of perceptions, centralized training methods have been widely used, following a typical three-step process, for training deep neural network (DNN)-based perception models: 1) data collected from CAVs is first uploaded to a cloud server; 2) the detection model is then trained centrally; 3) finally, the trained model is deployed back to the CAVs for inference. Although this centralized training scheme has achieved good performances in driving scenarios with deterministic data distribution, it struggles

to accommodate open driving scenarios with a wide variety of traffic scenes, as collected datasets often lack representation of rare cases, such as adverse weather conditions or traffic accidents. To overcome this limitation, federated learning (FL) (J. Wen et al., 2023), combined with wireless edge computing, has emerged as a promising distributed learning paradigm that enables efficient cooperative model training and continuous updates.

However, applying FL directly to domain-specific autonomous driving areas presents unique challenges. Many existing FL applications, such as image classification (S. Wang, Li, et al., 2022), rely on manually annotated data. In contrast, FL for AD often needs to operate on newly collected unlabeled data due to its distributed and high-mobility nature, making the conventional FL approach less effective. Moreover, IRSUs play crucial roles in extracting knowledge from diverse traffic scenarios, while existing roadside sensor placement algorithms primarily focus on monitoring traffic flow (Vijay et al., 2021), and overlooking the learning requirements and knowledge sharing necessary for FL among CAVs and IRSUs. As a result, roadside-involved FL remains an unexplored area.

In addition to the challenges in training perception models, new issues arise during the inference stage due to the complexities of communication. Cooperative inference techniques have been developed to enable CAVs to leverage information from other road participants, enhancing their perception capabilities. However, collaboration between CAVs and IRSUs depends heavily on wireless communication, which introduces several challenges, such as latency, bandwidth limitations, and channel distortion. The non-stationarity of V2X communication channels, caused by moving scatterers (i.e., other vehicles and pedestrians), further complicates real-time information sharing, highlighting the need for robust and adaptive communication strategies to ensure reliable and efficient cooperative inference.

Based on our literature review, we identifies the following areas that present challenges on achieving efficient cooperative training and inference among CAVs and IRSUs:

- Roadside sensor placement: Road-assisted cooperative model training among CAVs and IRSUs requires careful design for roadside sensor placement, as these sensors

significantly affects both the quality of data collected for model refinement and the effectiveness of cooperative learning. Traditional roadside sensor deployment strategies typically focus on maximizing coverage, often overlooking the specific learning requirements needed to assist CAVs in acquiring roadside knowledge for model updates. This critical pain point has not been well investigated. Addressing this gap is essential to ensure that roadside infrastructure effectively supports the dynamic learning and continuous refinement needs of CAVs. Annotation for newly collected data: Existing FL models typically rely on manually annotated datasets to facilitate cooperative model training. However, FL for AD often needs to operate on newly collected, unlabeled data to upgrade the perception models, This poses a significant challenge that remains largely underexplored. Wireless communication constrains for cooperative training: Developing an efficient cooperative training scheme among CAVs and IRSUs with wireless communication requires careful consideration of dynamic V2X communication conditions. Existing FL with wireless communication primarily focus on optimizing the communication protocols but often overlook the specific perception tasks required for AD. Additionally, the dynamic nature of real-world network topology and the heterogeneity of local CAV resources must be carefully considered to ensure the practicality and effectiveness of road-assisted cooperative training in real-world scenarios.

- Channel impairment analysis for cooperative inference: Existing research on cooperative model inference often operates under the assumption of ideal V2V channel conditions. However, distortions introduced by various channel impairments have the potential to compromise the effectiveness of wireless-connected cooperative inference systems, which has not been well investigated. In addition, a realistic non-stationary V2V communication channel needs to account for non-stationary and time-varying CAV-related parameters, including CAVs' velocity, acceleration, trajectory, and time-variant distances. Building such a realistic V2V communication model and applying it to real-world scenarios is an urgent yet challenging task that remains underexplored.

- Robust cooperative inference under V2V communication: Given the performance degradation caused by channel impairment for cooperative inference under various V2V channel conditions, it is imperative to find an efficient solution to mitigate the adverse effects of channel impairments, which has yet to be investigated.

1.2 Research Motivations and Contributions

To overcome the aforementioned challenges, this study firstly develops innovative frameworks for enabling efficient collaboration among CAVs and IRSUs. Additionally, we designs optimization algorithms that are tailored to the deployment of roadside infrastructure and the management of distributed learning and inference processes over wireless communication. All these novel techniques are tested across various traffic scenarios through three focuses. In this section, we first presents motivations for these three focuses. Then we summarize our research contributions.

1.2.1 Road Supervised Federated Learning with Bug-Aware Sensor Placement

To achieve effective road-assisted cooperative training, the position of IRSUs' sensors should be carefully designed. Existing work on 2D sensor deployments cannot be efficiently extended to 3D sensor deployments because of fundamental differences in data structures and spatial resolution. While 2D sensors capture planar information, 3D sensors (e.g., LiDAR, radar) generate volumetric point clouds with depth, orientation information that fundamentally alter coverage modeling. Besides, most existing 3D sensor deployment focuses on maximizing the coverage of the traffic areas, or maximizing the number of detected objects. Apart from providing additional sensing coverage for CAVs, roadside sensors also aim to assist CAVs in conducting the cooperative learning process. In 3D object detections, the pose of roadside sensors significantly impacts the geometric features of detected objects, as the reflected point clouds vary across different parts of the objects' surface. This variation further affects the learning process by influencing

the knowledge extracted from the point cloud data. When designing roadside sensors placement for cooperative training, it is essential to balance both coverage optimization and effective knowledge sharing to enhance model performance.

In addition, existing literature on training of perception models primarily focuses on centralized paradigms, where the model is trained once and then deployed on CAVs for inference. However, those trained models may not generalize well in real-world AD scenarios. Two key factors contribute to this limited generalizability: 1) *Sensing uncertainty* – The sensing data collected by each CAV may be sparse, noisy, or missing due to hardware limitations and environmental complexity (such as occlusion or long distance); 2) *Parameter uncertainty* – The downloaded model may be compressed or pruned to accommodate the limited computing, communication, and storage resources available on CAVs, further affecting its performance. To overcome the sensing and parameter uncertainties, a direct approach would be to collect large-scale datasets and adopt a perfectly optimized model. However, this solution not only incurs unbearable communication overhead but also requires high-cost labor for accurate annotation. A more practical and promising solution is to leverage newly collected data from CAVs and IRSUs, combined with a FL learning scheme, to fine-tune perception models – particularly when CAVs encounter out-of-distribution (OOD) data in open traffic scenarios. However, vanilla FL approaches rely on manually annotated datasets and are typically based on supervised learning, which makes them ineffective in AD contexts, where most of the newly collected data is unlabeled. Therefore, it is critical to develop an efficient road-assisted FL scheme tailored for AD systems to enable cooperative model training without the need for the manual data annotation.

Motivated by this, we develop an efficient cooperative training scheme between CAVs and IRSUs by utilizing shared information across various traffic scenarios. Specifically, we devise road-supervised FL (RSFL), which leverages the perception results from roadside sensors to annotate data collected by CAV sensors. It offers a novel perspective on data annotation for FLAD systems. To gain deeper insights into RSFL, we derive the information gain of object annotation through roadside sensors by leveraging the *expected*

entropy reduction, providing a theoretical foundation for understanding the benefits of road-assisted annotations. Furthermore, we develop a bug-aware sensor placement (BASP) algorithm, where the bug data (representing perception errors) from each traffic scenario is extracted to optimize both the number and placement of roadside sensors. BASP strategically reduces (increases) the number of sensors in low (high) complexity scenarios. Upon comparisons with traditional sensor placement strategies that primarily focus on sensing coverage or road geometry, we demonstrate that BASP approximately maximizes the information gain achieved through road supervision. Experimental results confirm the superiority of the proposed RSFL framework and BASP algorithm in enhancing cooperative training performance.

1.2.2 Communication Topology Optimization for FLCAV

V2X communication facilitates collaboration by enabling CAVs and IRSUs to share and fuse multi-modal data, with central nodes on IRSUs playing a key role in processing and aggregating information. However, the effectiveness of information sharing in such systems heavily depends on the characteristics of the wireless V2X communication channels, which are subject to variations in latency, bandwidth, and channel reliability. For the FL process in AD, CAVs and IRSUs collaboratively upgrade the global model without transmitting private local data. However, during the cooperative training process, uploading local model parameters or gradients from individual CAVs to the IRSU incurs significant communication overhead, resulting in high communication latency. Additionally, the heterogeneity of wireless channels and CAV on-board resources further complicates the cooperative FL training process. The presence of stragglers – slower or less reliable CAVs – and the mobility of CAVs in dynamic traffic environments further slow down the convergence of the global model. This delay impacts the overall efficiency and performance of cooperative training.

Motivated by this, we focus on how to achieve efficient cooperative training among IRSUs and CAVs in considering the practical V2X communication and dynamic network topology in the wireless V2X communication settings. We propose a topology optimization

algorithm to reduce communication latency and straggler issues and accelerate the cooperative model training and upgrading process. Experiment results show the proposed algorithm accelerates the model upgrade and improves the perception performance with the help of IRSUs.

1.2.3 Enhancing Cooperative Inference with practical V2V communications: A Joint Weighting and Denoising Approach

Different from cooperative training, aiming to upgrade the perception model by collaborative training processes, cooperative inference aims to improve the perception accuracy by leveraging shared information from other road participants in the inference stage. Cooperative inference (i.e., cooperative perception) among CAVs has been widely studied to alleviate the inherent limitation of single vehicle perception. Existing research on cooperative perception primarily focuses on improving final perception accuracy by fusing information obtained from other vehicles, typically under the assumption of ideal communication conditions. In real-world cooperative perception scenarios, data transmission and sharing among CAVs take place via V2V communication. This communication is subject to dynamic time-variant channel conditions, which can significantly have negative affect on the transmitted messages and subsequently impact the final perception results. Therefore, how to incorporate vehicular communications in cooperative perception is vital for ensuring system robustness, which has not been adequately investigated. Furthermore, a more sophisticated channel model that accounts for the non-stationarity of V2V channels, particularly the time-varying distortions caused by vehicle speed and moving scatterers, has yet to be investigated.

Motivated by this, we propose a joint weighting and denoising framework, named **Coop-WD**, to enhance cooperative perception in the presence of V2V communications impairments, where the self-supervised contrastive model and the conditional diffusion probabilistic model are adopted hierarchically for feature enhancement on CAV and

pixel level, respectively. Simulated Rician fading, multipath, non-stationarity, and time-varying distortion of V2V characteristics are considered. Numerical results demonstrate that the proposed Coop-WD outperforms conventional benchmarks under all types of channels. It is also validated that the proposed algorithm could adaptively mitigate the negative effects according to the level of channel impairments and improve the performance when there is mild distortion. Qualitative analysis with visual examples further proves the superiority of the proposed method.
To summarize, we have made the following research contributions:

- To contribute to road supervised federated learning, we propose an effective road-side sensor placement algorithm, a novel labelling algorithm for unlabeled newly collected data, and a road-supervised federated learning scheme to enable efficient collaboration between IRSUs-CAVs. These methods are designed under the assumption of ideal communication conditions while addressing the challenges of handling unlabeled data and ensuring continuous upgrades to the perception models.

- To contribute to communication topology optimization for FLCAV, we develop a flexible FL for CAVs (FLCAV) framework and propose an efficient topology optimization algorithm to facilitate cooperative training between CAVs and IRSUs within wireless networked vehicular systems. The proposed framework has demonstrated significance reduction in the communication latency, mitigate perception uncertainty and improve the models' generalizability.

- To contribute to enhancing cooperative inference with practical V2V communications, we first develop a practical non-stationary V2V channel model that considers time-varying CAV parameters, such as velocity, acceleration, location, and orientation. This model fully accounts for the impact of these parameters on channel conditions. Furthermore, we propose a joint weighting and denoising framework to recover noisy information received by the ego-CAV and mitigate the adverse effects of channel impairments during cooperative inference.

1.3 Thesis Outline

Remaining of the thesis is organized as follows:

- *Chapter 2*: This chapter presents the review of relevant literature on ITS architectures, perception-related DNN models, public dataset, simulators for AD, cooperative model training and inference schemes, and research gaps.
- *Chapter 3*: This chapter details our exploration of the architectures, simulators, datasets, and DNN models to build the cooperative model training system among CAVs and IRSU. A BASP algorithm based on integer programming is derived to solve the roadside sensor placement. In addition, a RSFL framework equipped with bug aware road labelling (BARL) algorithm is devised for the AD system.
- *Chapter 4*: This chapter illustrates the research of cooperative model training with V2X communications. We first explore the integration of the perception task for FLCAV with V2X communication under practical communication topology among CAVs and IRSUs. A multi-layer communication topology optimization algorithm is then proposed to reduce the communication delay among CAVs and IRSUs and improve the communication efficiency of the cooperative training process.
- *Chapter 5*: This chapter illustrate the research of cooperative inference with V2V communication. A joint weighting and denoising framework for cooperative inference (Coop-WD), has been proposed to enhance cooperative inference under V2V communication impairments. The framework employs a hierarchical approach, integrating a self-supervised contrastive model for feature enhancement at the CAV level and a conditional diffusion probabilistic model for pixel-level refinement. Simulated scenarios include Rician fading, multipath effects, non-stationarity, and time-related distortions, capturing the key characteristics of V2V communication challenges. The proposed algorithm can effectively adapt to varying levels of channel impairments, mitigating their negative effects and improving performance, particularly in cases of mild distortion.

- *Chapter 6*: This chapter concludes this thesis and discusses the potential future works.

CHAPTER 2

LITERATURE REVIEW

This chapter presents the literature review about efficient cooperative training and inference for networked vehicular systems. It includes architecture design, advanced 3D object detection models, public datasets, simulators for AD, cooperative schemes for CAVs and IRSUs and the identified research gaps which underpin the research questions of this study.

2.1 Architectures for the Vehicular Networks

The rapid development of wireless communication and machine learning(ML)-enabled applications have facilitated the advances in AD systems. However, the substantial resource demands of these ML-based applications, particularly for computationally intensive and delay-sensitive tasks, often exceed the capabilities of vehicles with limited on-board processing and communication resources. To this end, mobile cloud computing (MCC) (Asghari & Sohrabi, 2024) and mobile edge computing (MEC) (Loutfi et al., 2024) are widely acting as promising architectures for the networked vehicular system.

2.1.1 Mobile Cloud Computing

Originally, the vehicular network architecture was based on MCC (Qureshi et al., 2018). In MCC, cloud servers are located far from end devices but provide integrated resource allocation and mobility management for vehicles to address the needs of mobile end users (Bréhon–Grataloup et al., 2022). The integration of vehicular networks with MCC led to the emergence of vehicular cloud computing (VCC) (Saleem et al., 2024), a paradigm that promises to address the computational limitations of individual vehicles by leveraging the collective resource of the network. In this paradigm, computation-extensive applications are offloaded to a cloud server, and large amounts of data are stored in the cloud. In this way, the paradigm can reduce the computation and storage burden of vehicles. However, the cloud server is far from the end vehicles, which leads to high transmission latency. Furthermore, the volume of data transmission and retrieval will place considerable pressure on backhaul network resources.

2.1.2 Mobile Edge Computing

Although the VCC brought advancement for the architecture of a vehicular network, the communication latency issue impedes its use of delay-sensitive applications for vehicular networks like object detection, lane recognition, and traffic sign detection. Thus, MEC (X. Wang et al., 2023) is envisioned as a promising paradigm to alleviate such issues. In MEC, the cloud service is distributed to the edge servers which are generally located in the vicinity of end users. In this regard, the communication latency can be reduced remarkably. By integrating a vehicular network into MEC, a paradigm, named vehicular edge computing (VEC) (Hasan et al., 2024), was proposed in recent years. In VEC, vehicles have limited communication and computation resources. Roadside infrastructures can act as edge servers distributed along the road and are in charge of data collection and processing.

2.2 Machine Learning Enabled Perception for Autonomous Driving

Deep neural networks (DNN) have revolutionized every branch of AD. The perception module is the visual system of a CAV. Through the perception module, CAVs can detect, track and classify nearby objects to reason about the surrounding environment. Failure to detect and classify important objects could lead to tragic accidents. As the core task of the perception module, object detection has greatly benefited from the progress in deep learning (DL) technologies. Numerous DL-based object detection models have been developed and deployed for perception in AD. This section reviews the techniques used in 3D object detection and the public datasets for 3D object detection.

2.2.1 3D Object Detection for Autonomous Driving

Generally, 2D object detection (Vijayakumar & Vairavasundaram, 2024) leverages cameras to collect 2D images, but this approach cannot provide precise depth information. Although the techniques for 3D depth estimation on 2D images (Y. Li, Bao, et al., 2023) have been improved with the advancement of the DL-based vision algorithms, the estimations are still far from precise and reliable. Besides, 2D object detection tends to have poor performance in low-light scenarios and a high computational cost of processing high-resolution images, so it is not robust enough to be used in resource-constrained vehicles.

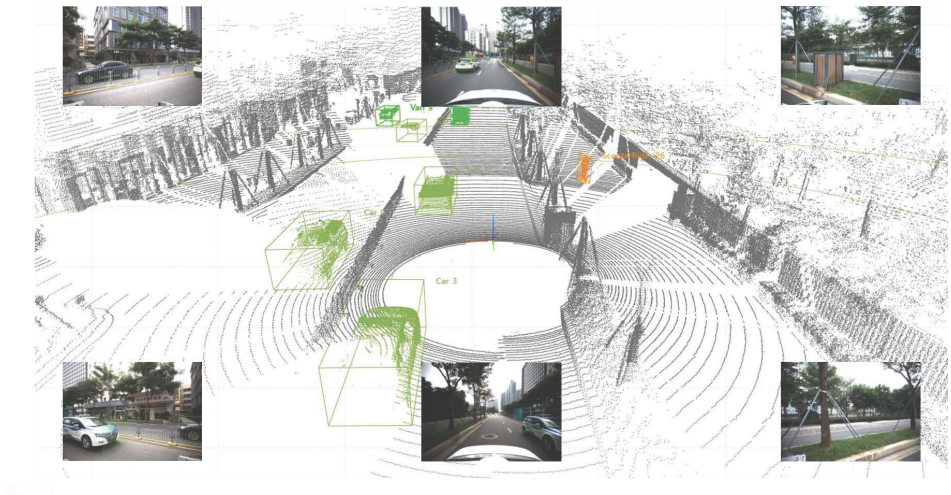


Figure 2.1: Visualization of point cloud and detected bounding boxes of surrounding objects.

Thanks to the advancement of sensor technologies, 3D scanners like LiDAR are becoming more and more affordable and available. LiDAR, which stands for light detection and ranging, can precisely measure the distance between the sensor and surrounding obstacles while providing rich geometry, shape, and scale information (Fernandes et al., 2021). It has been widely adopted in CAV applications. In recent years, DL-based point cloud processing approaches have been attracting attention in both academic and industrial communities. Various 3D detection models have been proposed to address problems related to the point cloud data process. LiDAR-based 3D object detection can get the distance from the LiDAR to the surface of objects through real-time LiDAR sweep, then feed into 3D object detection models, and output a list of 3D bounding boxes. As shown in Figure 2.1, CAVs exploit LiDAR to capture point cloud data, subsequently transforming it into 3D bouding boxes. These boxes encapsulate the pose (position and orientation), dimensions, and classification of detected objects in the surrounding environment. This sub-section reviews existing popular DL-based point cloud processing models for 3D object detection and recent progress in this research area.

Existing DL-based 3D point cloud processing approaches can be classified into four categories: (i) raw point-based methods; (ii) 3D voxelization-based methods; (iii) Birds eye view (BEV)-based methods; and (iv) range-view-based methods.

2.2.1.1 Raw-Point-Cloud-based Perception Approaches

In terms of raw point-based detection models, pioneer work includes PointNet (Qi, Su, et al., 2017) and its variants PointNet++ (Qi, Yi, et al., 2017). PointNet proposed a DNN that can directly exploit point cloud data to perform 3D object classification, segmentation, and scene semantic parsing. Firstly, it adopts several multi-layer perception modules to extract key point features and uses the max pooling layer as a symmetric function to aggregate the information from all points. A segmentation network is then used to concatenate the global feature with each point feature. Finally, a joint alignment network is applied to align the point feature and extracted features for prediction. The drawback of PointNet is that it is a point-wise process, so it fails to capture the local structure among the points, which can contain important features. To solve this problem, PointNet++ proposed an approach to process point clouds hierarchically. It first builds the distance metric of the space and partitions the set of points into several overlapped local regions. Then, local features are extracted from a small neighborhood. These local features are subsequently grouped into larger units and processed to produce higher levels of features. Lastly, this process is repeated until the features of the whole point set were obtained. Besides, several other similar works like PointRCNN (S. Shi et al., 2019), PointRGCN (Zarzar et al., 2019), SVGA-Net (He et al., 2022) and PG-RCNN (Koo et al., 2023) can be categorized under this scheme of point cloud representation.

2.2.1.2 3D Voxelization-based Perception Approaches

A 3D voxelization-based method first segments point clouds into 3D voxels with equal volumetric scales in the 3D Cartesian coordinate system. Features are then extracted from a set of points within each voxel, which are subsequently aggregated and concatenated to expand the receptive field and incorporate additional context into the extracted features. This approach effectively reduces the dimensionality of point clouds, saving memory resources. The pioneering work in this domain is VoxelNet (Y. Zhou & Tuzel, 2018). It first utilizes a voxel feature encoding (VFE) layer to generate unified feature representations for groups of points within 3D voxels, which are created by partitioning

the input point cloud into a regular grid and enabled the conversion of raw 3D points into voxel-based representations while simultaneously learning 3D geometric features within each voxel. Following this, a 3D convolutions is exploited to further extract voxel-wise features through a region proposal network (RPN), ultimately generating the final outputs. Other notable voxelization-based methods include SECOND (Yan et al., 2018), Part-A2 (S. Shi et al., 2020), Voxel R-CNN (Deng et al., 2021), FSD (Fan et al., 2022) and LION (Z. Liu et al., 2025).

2.2.1.3 BEV-based Perception Approaches

For BEV-based methods, notable examples include PIXOR (B. Yang et al., 2018) and 3D-CVF (Yoo et al., 2020). By assuming that all the objects are on the ground and cannot fly, it becomes reasonable to exploit the BEV representation, which is more computationally friendly than a 3D voxel grid. PIXOR proposes a proposal-free, one-stage detector, in which the scene is represented in a BEV map. 2D convolutions are applied to learn pixel-wise features of the BEV map. Additionally, 3D-CVF proposes a LiDAR-camera fusion architecture that transforms the camera view's feature map into a calibrated and interpolated feature map in BEV. This approach leverages the complementary strengths of LiDAR and camera data to enhance perception performance. Other notable BEV-based methods include PillarNeXt (J. Li, Luo, & Yang, 2023), Graphbev (Song et al., 2024) and Bevnext (Z. Li et al., 2024).

2.2.1.4 Range-View-based Perception Approaches

Range view representation, which aligns with the native view of LiDAR sensors, is used to construct a dense input image. This representation is inherently more compact than BEV, resulting in significant computational efficiency gains.

For range-view-based methods, notable examples include LaserNet (Meyer et al., 2019) and VeloFCN (B. Li et al., 2016). In LaserNet, the method first predicts a class probability for each LiDAR point within the range-view image. It then regresses a probability distribution over bounding boxes in the top-down view. To refine these

predictions, the per-point distributions are combined using mean shift clustering, which helps reduce noise in individual predictions. Finally, a novel adaptive non-maximum suppression algorithm was applied to remove duplicate bounding box distributions, further enhancing the detection accuracy and efficiency.

To summarize, among DL-based 3D point cloud approaches, the raw data-based methods can fully exploit point-wise information from the point cloud, while it lacks spatial prior. The 3D voxelization-based methods and 2D BEV-based methods are typically more straightforward; however, due to the inherent sparsity of point clouds, these methods often result in a large number of empty voxels or pixels, leading to inefficiencies in computation and memory usage. The range view-based methods are more compact compared to 3D voxelization-based methods and 2D BEV-based methods, as they align naturally with the native sensor view, reducing redundancy. However, the notable challenge with range-view representation is that objects appear at varying scales depending on their distance from the sensor, complicating consistent object detection (S. Chen et al., 2020).

2.2.2 Simulator for Traffic Scenarios Generation and Data Collection

Traffic simulators allow researchers to evaluate algorithms in controlled virtual environments, mimicking real-world traffic scenarios such as road configurations, traffic flow patterns, and vehicle interactions. This simulation capability enables the generation of large-scale datasets for training and testing algorithms across varied conditions, eliminating the need for costly and time-intensive physical testing on actual roadways.

In (Dosovitskiy et al., 2017), CARLA was introduced as an open-source urban driving simulator, particularly designed for AD research. CARLA provides open digital assets such as urban layouts, buildings, and vehicles, all created for simulation purposes and freely usable. Additionally, CARLA simulates a variety of sensors, including RGB cameras and pseudo-sensors for ground-truth depth and semantic segmentation. This could provide detailed data for training and testing perception systems. CARLA also allows users to configure a broad spectrum of environmental conditions, such as weather

and time of day, enabling robust testing of AD systems' robustness under different scenarios. Furthermore, CARLA facilitates the collection of driving data and supports driving policy analysis through performance metrics such as the distance travelled between infractions and other performance indicators.

Another widely used open-source simulator is SUMO (Krajzewicz et al., 2012), which was particularly designed to handle complex city road networks. SUMO could be used for various applications, including transportation planning, traffic management, and the evaluation of AD systems. It could generate traffic demand based on predefined routes or more sophisticated models considering trip purposes and origins-destinations. Additionally, SUMO also include a built-in editor for creating and editing road networks, which could be customized to match real-world scenarios or hypothetical layouts. It also provides various output options, including vehicle trajectories, traffic densities, and visualizations, enabling detailed analysis and a better understanding of the simulation results.

2.2.3 Public Dataset for Perception

Research in 3D object detection for AD has been greatly accelerated by the availability of high-quality annotated datasets. Current mainstream public datasets explored for 3D object detection include KITTI (Geiger et al., 2013), ApolloScape (Huang et al., 2019), nuScenes (Caesar et al., 2020), Lyft Level 5 AV dataset (Houston et al., 2021)), and Waymo (Sun et al., 2020).

As the pioneering dataset in the AD field, KITTI has been widely used to benchmark various visual tasks. The dataset was collected using a vehicle equipped with four video cameras, a 3D laser scanner, and GPS/IMU inertial navigation system. KITTI also include benchmarks for 3D object detection and tracking to promote its usage (Geiger et al., 2013). ApolloScape is a large AV dataset developed by Baidu, which consists of over $140,000$ video frames with point-wise semantic annotation for 28 classes in 3D. The nuScenes dataset is the first dataset with a fully autonomous vehicle sensor suit, including 6 cameras, 5 radars, and 1 LiDAR. It provides full surround sensor coverage

consisting of 1,000 scenes, each 20 seconds long and fully annotated with 3D bounding boxes for 23 classes, and 8 attributes. The Lyft dataset consist of 170,000 scenes, each lasting 20 seconds. It provide extensive coverage for 3D perception tasks. The Waymo dataset consist of 1,150 scenes spanning 20 seconds each, with over 12 million 3D bounding box annotations on the LiDAR point cloud.

While these datasets have played a crucial role in advancing 3D object detection, they all have significant limitations. That is, they were generated using data from a single vehicle in different scenes, making them unsuitable for cooperative perception evaluation. Moreover, most works in cooperative perception rely on synthetic datasets that simulate multi-vehicle data by duplicating a single vehicle's point cloud at different timestamps to mimic multiple vehicles at the same timestamp. For instance, the authors (Q. Chen, Ma, et al., 2019) adopted the KITTI dataset and practical vehicle-to-vehicle setting by leveraging one vehicle's data at different timestamps as multiple CAVs.

However, such synthetic scenes are unrealistic and introduce spatial and temporal inconsistency. The shortage of cooperative driving datasets makes it difficult to benchmark comprehensively the cooperative perception algorithms. Efforts have been made to develop datasets specifically tailored for cooperative perception in V2V and V2X communication scenarios. Based on their data sources, these datasets can be classified as either simulation-based or real-world datasets.

Acquiring real-world V2V or V2X cooperative perception datasets poses significant challenges due to high costs and labor demands. Consequently, most existing cooperative perception datasets are generated using simulators, such as CARLA (Dosovitskiy et al., 2017) and SUMO (Krajzewicz et al., 2012). In (Y. Li et al., 2022), the authors introduced V2X-Sim, a synthetic, multi-agent, and collaborative perception dataset. Specifically, it was designed to support AD research, focusing on V2X scenarios. The V2X-Sim is a combination of CARLA and SUMO, where CARLA is in charge of driving scenarios simulation and generates high-fidelity sensor data, and SUMO provides realistic traffic flow. In which, CAVs and an RSU are equipped with a range of sensors, including RGB cameras, LiDAR, GPS, and IMU, allowing multi-modality data collected in a 360-degree

view. Each scene represents a multi-agent environment, where multiple vehicles and an RSU collaboratively share and exchange perception data. The dataset comprises 100 scenes, totaling 10,000 frames. Each scene in V2X-Sim represents 20 seconds of traffic flow data captured at a specific intersection. In (R. Mao et al., 2022), the DOLPHINS dataset was introduced to address the lack of multi-view and multi-modality data by offering a large-scale and diverse dataset designed for collaborative perception, enabling more comprehensive testing and development of the AD algorithm. It encompasses 42,376 frames in total, with annotations for 292,549 objects featuring 3D bounding boxes, geo-positions, and calibration information. Objects are classified into three difficulty levels (easy, moderate, and hard) based on occlusion levels, allowing for diverse testing conditions. Meanwhile, in (Xu, Xiang, Tu, et al., 2022), the authors introduced the V2XSet dataset, which was generated using CARLA and OpenCDA (Xu et al., 2021) simulators. This dataset captures 11,447 frames (or 33,081 samples when considering frames from each agent) across a variety of scenarios. It includes 55 scenes representing different driving situations, such as intersections, midblock, and ramps, to support comprehensive evaluations of cooperative perception systems.

On the other hand, real-world sensors also encounter challenges like sensor drift, signal interference, and wear over time, which simulators typically cannot replicate accurately. Consequently, constructing datasets from real-world environments has gained increasing attention in recent years. The DAIR-V2X (Yu et al., 2022) dataset was developed to address the limitations of real-world cooperative perception datasets, particularly in the domain of Vehicle-Infrastructure cooperative AD. The dataset comprises data collected from 28 intersections in Beijing. Each intersection is equipped with infrastructure sensors, including 300-beam LiDAR and high-resolution RGB cameras, capturing a 100^o horizontal and 40^o vertical FoV. The CAVs used in the dataset is equipped with 40-beam LiDAR (providing 360^o coverage) and RGB cameras with a resolution of 1920x1080. The sensors capture data at 10 Hz for LiDAR and $20-25$ Hz for cameras, ensuring temporal alignment between frames. It consists of three parts: (i) DAIR-V2X-C (Cooperative data) includes 39,000 LiDAR and camera frames from synchronized vehicle-infrastructure

views, facilitating vehicle-roadside 3D object detection; (ii) DAIR-V2X-V (Vehicle-only data) contains 22,000 frames, capturing diverse scenarios exclusively from the vehicle's perspective; and (iii) DAIR-V2X-I (Infrastructure-only data) comprises 10,000 frames, focusing on scenarios exclusively from the infrastructure's perspective.

2.3 Cooperative Model Training for Perception in Networked Vehicular Systems

Generally, a centralized optimization scheme is employed to accelerate the model training and upgrade process. In a conventional centralized learning process, all users' data are collected and transmitted to the cloud server, where it is partitioned and distributed to nodes for parallel processing. This is often done within a cluster, which is a group of interconnected computers that work together as a single system. The cluster provides the necessary computing power and resources to handle the massive amounts of data and complex computations involved in training perception networks. Additionally, DNN-based models are usually trained in a cluster environment, which relies on a stable network connection and ample bandwidth for efficient data transfer and communication between nodes. However, in multi-agent cooperative training systems, IRSUs and CAVs are generally connected via wireless vehicular networks, which often leads to network instability and limited bandwidth. Moreover, CAVs must navigate in diverse and constantly changing traffic environments that often include corner cases arising from the infinite scenario spaces. As a result, perception models need to be continuously retrained and updated whenever rare cases are encountered to further improve their accuracy and generalization. To achieve efficient cooperative training among CAVs and IRSUs in networked vehicular systems, several promising techniques are explored. Specifically, we delve into federated learning, which enables collaborative model training without centralizing data; knowledge distillation, which facilitates knowledge transfer from powerful models to smaller, more efficient ones; and task offloading, which strategically distributes computational burdens across the network.

2.3.1 Federated Learning for CAVs

The model training for CAV perception typically followed a centralized scheme (S. Shi et al., 2019, 2020), where training datasets were collected from all CAVs, the detection model was trained in a cluster, and the trained model was subsequently deployed on CAVs for inference. However, this scheme incured significant communication overhead and raised concerns about potential information leakage. Since the collected datasets were typically large in volume and often contained human-related privacy-sensitive information. It is always unrealistic to collect all users' data and send collected data to the server. Moreover, not all users are willing to participate in the model training process. To address these challenges, a distributed optimization paradigm known as federated learning (McMahan et al., 2017) was proposed in 2017. In FL, each user trained a local model with its own generated data, which remained private and was not shared with others. To leverage knowledge from other users, the local model parameters were periodically uploaded to a central server for aggregation, resulting in a global model. This aggregated global model was broadcast to all users for further local updates (S. Wang, Hong, et al., 2022). Through its training procedure, FL demonstrated its ability to reduce network load while safeguarding the privacy of local data. As such, FL emerged as a promising solution for the networked vehicular systems to achieve cooperative model training amongst CAVs and IRSUs.

By combining FL and edge computing, a widely adopted framework known as federated edge learning (FEEL) was proposed (G. Zhu et al., 2020). In FEEL, each end device repeatedly transmitted its locally learned model to the edge server over a throughput-limited uplink channel (Chellapandi et al., 2023). All end devices shared the same wireless network for uploading their locally update models, enabling collaborative learning while optimizing network resources. As the number of end devices increases, the limited bandwidth faces significant pressure, leading to communication bottlenecks. Extensive research has been conducted to mitigate the communication overhead of distributed stochastic gradient descent (SGD) algorithms, taking into account both noiseless and noisy network conditions. In terms of model training without considering wireless

noise, approaches can be grouped into three categories: (i) large mini-batch size (Goyal et al., 2017; B. Yang et al., 2018) and factorization (Guo et al., 2024; W. Jeong & Hwang, 2022) without loss accuracy of gradient descent. This category focuses on increasing the efficiency of gradient descent by using larger mini-batches during training or by factorizing the model or data to reduce computational complexity; (ii) Gradient sparsification (Alistarh et al., 2018; Beitollahi et al., 2022; X. Lin et al., 2023). This category aims to reduce the communication overhead by transmitting only the most important gradients during training. By selectively choosing and transmitting a subset of the gradients, these techniques can save bandwidth and reduce training time; (iii) Quantization (Y. Mao et al., 2022; Shlezinger et al., 2020). This category aims to reduce the number of bits used to represent model parameters and gradients. By compressing the data in this way, these methods can significantly reduce the communication overhead and accelerate the training process. As FEEL relies on wireless networks, the quality of network connection and wireless channel impairments also need to be considered. The authors (G. Zhu et al., 2020) proposed a one-bit digital aggregation scheme for FEEL, integrating over-the-air aggregation with one-bit gradient quantization to reduce communication overhead. They also provided theoretical analytics of the effects of channel noise, fading, and estimation error on convergence rate. In (M. Chen et al., 2020), the authors proposed a closed-form expression for the convergence rate of FL considering the impact of wireless factors, such as package error and resource limitations on FL. According to the expression, they proposed a resource allocation and a user selection method to further minimize the FL loss function.

The data distribution in each end device is generally non-independent and identically distributed (non-IID) and often unbalanced, which can significantly slow down the convergent rate or even prevent convergence when using traditional distributed machine learning approaches. To overcome these statistical challenges in FL, the authors in (F. Chen et al., 2018) introduced a novel framework, named FedMeta, which incorporated meta-learning techniques into FL. This framework was designed to improve the learning process under non-IID and unbalanced data conditions, enhancing the robustness and ef-

fectiveness of FL models. In addition, in (Smith et al., 2017), the authors proposed a novel framework, named MOCHA, for FL. This framework employed multi-task learning to learn the data personalization at each node and used a shared representative to express the global model. Furthermore, in (Zhao et al., 2018), the authors also explored the non-IID data in FL. They investigated the discrepancy in weights between the global model and each local model during each communication round to quantify this divergence, they employed the Earth Mover's Distance to measure the discrepancy between the overall data distribution and the data distribution at each node. The results demonstrated that training local models using a combination of shared data and locally generated data significantly improved performance and alleviated the statistical challenges inherent in FL settings. In addition, in (Zhuang et al., 2020), the authors proposed a new algorithm to cope with the statistical heterogeneity in person re-identification. They leveraged Cosine Distance Weight (CDW) to measure the changes in local models between each communication round and dynamically allocated the aggregated weight in the global aggregation phase.

Due to the inherent heterogeneity among devices in FEEL, such as computational power, storage, and limited capacity of communication, it is not feasible to treat all devices equally during model training. One way to mitigate this issue is to select devices that have better network connections, higher computational powers, and more storage capacity to participate in the training process, and avoid those devices with limited resources from participating in the training process. In (Nishio & Yonetani, 2019), the authors proposed a FedCS protocol, an FL approach designed for mobile edge computing (MEC) systems. This protocol actively selected edge devices for participation in the training process based on their resource conditions. By prioritizing devices with better resources, FedCS achieved better acceleration in the training process compared to the original FL protocol, while optimizing resource utilization.

In general, the effectiveness of the FL training scheme highly depends on resource optimization and device scheduling during the pretraining, update, and aggregation procedure of DNN. These aspects have been widely discussed in (M. Chen et al., 2021;

Du et al., 2023; W. Wen et al., 2022). However, the conventional FL technologies are not well-suited to tackle the unique challenges when deploying cooperative model training among CAV and IRSUs perception systems. These challenges include the high mobility of CAVs, limited bandwidth between CAVs and IRSUs, and the collaboration of all road participants. Thus, optimizing and tailoring the FL scheme for the cooperative model training among CAVs and IRSUs poses a significant challenge. Developing efficient FL strategies to address complexities like CAVs' data diversity, communication efficiency and model convergence, is crucial for advancing the potential of FL in enhancing cooperative perceptions for networked vehicular systems.

2.3.2 Knowledge Distillation among CAVs and IRSUs

In a networked vehicular system, CAVs leverage vehicle-to-infrastructure (V2I) communication that enables CAVs to work with IRSUs for collaborative analysis. It is often assumed that learning models implemented on CAVs and IRSUs are well-trained and that the collected dataset has been properly annotated. However, in the real world, it is unrealistic to pre-train a detection model and expect it to obtain high accuracy for all complicated scenarios (e.g., crossroads, T-junctions and roundabouts) due to the infinite traffic scenario space. Thus, the detection model needs to be continuously updated using newly collected data. Moreover, the raw data collected by CAVs and IRSUs lacks annotation. Traditional methods for annotating raw data are manual, labour-intensive, and non-trivial, which significantly slows down the process of updating detection models. This creates a bottleneck in ensuring the effectiveness of models in dynamic and diverse traffic environments.

To address this challenge, knowledge distillation (Hinton et al., 2015) was originally proposed to transfer the generalization ability of a cumbersome model to a small model. It was organized in a teacher-student architecture, where the large model (acted as the "teacher") transferred knowledge into the small model (acted as the "student"). Recent knowledge distillation methods have been extended to mutual learning (C. Wu et al., 2022), assistant teaching (Mirzadeh et al., 2020), self-learning (Vu et al., 2021) and

Co-Distillation (Y. Liu et al., 2022). However, knowledge distillation required common training samples to be observed by both the student and teacher models (Seo et al., 2020). Consequently, all models must share the same samples during each loss computation. This required data exchanges among all workers, which might violate local data privacy. In this regard, federated distillation (FD) (E. Jeong et al., 2018) was proposed to eliminate the dependency on common data observations by grouping data according to labels. The FD follows an online version of knowledge distillation and each device in the FD stores per-label mean logit vectors and uploads these local average logits to a server periodically. For each label, the uploaded local average logits are averaged into a global average logit per label which is then broadcast to each device to calculate the distillation regularizer. Since FD involves transmitting logit vectors instead of gradient updates or model parameters, it requires less communication payload. However, this make it more vulnerable to the non-IID data issues. To mitigate this, the authors (Itahara et al., 2021) constructed a common sub-dataset that could be accessed by all devices to avoid coarse sampling in the original FD. Besides, in Mix2FLD (Seo et al., 2020), the authors uploaded local model output as in FD and downloaded model parameters as in FL to cope with uplink-downlink channel asymmetry.

Most FD-related works focus on 2D visual tasks, but for cooperative perception in 3D object detection, it is highly desirable to leverage the knowledge from both IRSUs and individual CAVs for online label generation.

2.3.3 Task Offloading for Wireless CAV Systems

With the rapid deployment of roadside infrastructure and smart vehicles, various types of AD-related applications arise. Many of these applications, such as object detection and motion planning, are delay-sensitive and have a stringent demand for computation power, which imposes a burden on CAVs. Despite the abundant resources, the cloud server is not suitable for delay-sensitive applications due to the long distance between the cloud and CAVs. Moreover, the large amount of data generated by those applications will impose a burden on the network bandwidth (L. Liu et al., 2021). One main advantage

of CAVs and IRSUs in ITS is that CAVs are encouraged to leverage resources of their nearby devices including other CAVs and roadside infrastructures by task offloading.

Extensive research has been conducted to explore offloading computation-intensive tasks of CAVs to the edge server to reduce the computation overhead of CAVs. In (Y. Liu et al., 2018), the authors explored the problem of offloading computing tasks from multi-vehicles to an edge server based on a game theory. They formulated the task offloading decision-making as a game, analyzed the interaction among all vehicles, and proposed a distributed incentive algorithm to optimally select suitable channels and make the decision to decrease computation overhead. The authors (Du et al., 2018) extended prior work by taking into account the profits of both the vehicle and the edge server. They proposed a dual-side cost minimization framework for computation offloading and resource allocation in vehicular networks. Instead of only considering single-edge servers, the authors (Tareq et al., 2018) proposed a highly reliable and low latency V2I communication architecture to jointly optimize the association of vehicles with small-base-stations (SBSs) and the allocation of wireless resources.

2.4 Cooperative Model Inference for Perception in Networked Vehicular Systems

Cooperative inference has emerged as a promising approach to overcoming the inherent limitations of the individual AV in understanding their environment. As the individual AV detect surrounding objects only by their onboard sensors, they can only perceive the driving environment from a single point of view and have limited capability of perception. Consequently, they always suffer from some inherent limitations (Arnold et al., 2020). For instance, some moving objects are always heavily occluded by other static or dynamic objects, while far-away objects always reflect low-point density. Cooperative inference has emerged as a promising approach to overcoming the inherent limitations of the individual AV in understanding their environment.

In CAV systems, wireless networks provide possibilities for sharing perceived data

among CAVs and IRSUs. Specifically, CAVs in proximity share their detected information and fuse received data to cooperatively perceive the driving environment, enlarging sensing ranges and improving sensing accuracy.

2.4.1 Fusion Schemes for Cooperative Inference

An important aspect of cooperative inference (i.e., cooperative perception) in CAV systems is determining how to effectively fuse shared information from other participants to enhance overall perception performance. To address this challenge, various approaches have been proposed in the literature. According to the type of shared data, current cooperative inference methods fall into three categories: (i) raw-data-level fusion scheme, (ii) feature-level fusion scheme, and (iii) object-level fusion scheme, which will be explained in the following sub-sections.

2.4.1.1 Raw-data-level Fusion

With the raw-data-level fusion scheme, CAVs share raw sensor data, which tends to yield high accuracy. For example, the authors (Q. Chen, Tang, et al., 2019) proposed a data fusion scheme, named Cooper, for 3D point clouds and conducted a study on raw-data-level cooperative inference for enhancing detection ability. They also introduced a sparse point cloud object detection (SPOD) method to overcome the sparsity of point clouds. This method focused on extracting only positional coordinates and reflection values of point clouds, effectively compressing the data into acceptable sizes for transmission. Additionally, to achieve data reconstruction, the system encapsulates additional GPS and IMU readings into the exchange packages, helping merge the received data into the physical position of each vehicle. Similarly, in (Arnold et al., 2020), the authors proposed a central system to fuse point cloud data from multiple infrastructure sensors. In the central fusion system, each respective point cloud data was concatenated into a single point cloud which was then fed to the 3D object detection model. The detected results were then disseminated to vehicles in the systems.

2.4.1.2 Object-level Fusion

The object-level fusion scheme has gained attention due to the high communication overhead and computational cost associated with transmitting and merging raw point cloud data in the previous scheme. Instead of sharing raw data, this scheme focuses on transmitting processed perception results to reduce communication costs and eliminate data merging complexities. For instance, in (Ambrosin et al., 2019), the authors proposed a two-layer architecture for object detection and tracking. In the system, dynamic obstacle information was shared via V2V messages among vehicles, and the information matrix fusion (IMF) algorithm was then adopted to fuse the received information. Similarly, in (Arnold et al., 2020), the authors proposed an object-level cooperative perception model by sharing the outputs of the locally trained 3D object detection models. In the system, the point cloud was firstly pre-processed and then fed into the detection model of each CAV, and the output (a list of objects represented by 3D bounding boxes) was then transmitted to the central fusion system, in which non-maximum suppression (NMS) was used to mitigate the detection noise of sensors to generate the final object list. The bounding box list was finally broadcast to all vehicles for the next actions.

2.4.1.3 Intermediate-level Fusion

The Intermediate-level fusion scheme tends to share feature maps or intermediate representations from perception models of individual vehicles to achieve cooperative inference. As the intermediate representation tends to have a smaller size than raw data and contains valuable features of scene context, sharing the intermediate representation could not only reduce bandwidth requirement but also keep high performance. For instance, in (T.-H. Wang et al., 2020), the authors proposed a V2VNet to achieve cooperative inference and motion prediction by incorporating V2V communication. In the system, CAVs share their compressed intermediate representations with the ego vehicle. Then, a spatially aware graph neural network (GNN) is utilized to aggregate and combine the information received from all nearby vehicles. Finally, an output network is applied to compute the final perception and prediction results. Besides, to cope with limited

bandwidth and stringent real-time constrain of CAV issues, authors in (Q. Chen, Ma, et al., 2019) proposed a fusion method, named F-Cooper, to achieve high and real-time object detection precision. In F-Copper, different levels of features of other CAVs were shared with the ego vehicle (i.e., the CAV making decisions based its on-board sensors and information received from other cooperative participants, such as other CAVs or IRSUs) to achieve cooperative detection, as feature-based data was sufficient for the training process, and the intrinsic small size of feature data could also benefit real-time edge computing. The F-Cooper integrated edge computing and CAV to analyse efficiently massive amounts of data in real-time under limited network bandwidth. It also proposed two feature fusion paradigms: voxel feature fusion and spatial feature fusion. In voxel feature fusion, point cloud data was first processed in each vehicle by the voxel feature encoding layer to get voxel features. Each vehicle's voxel features were then transmitted to the edge server to do voxel feature fusion. A sparse convolutional layer was applied subsequently to generate the spatial features. A region proposal network was finally used to output detection results. In spatial feature fusion, spatial features were first obtained locally on each vehicle. The spatial features were then transmitted to the edge server to do spatial feature fusion. The fusion result was then fed into region proposal network and detection headers to get the detected results. To further reduce communication overhead, in (Y.-C. Liu, Tian, Ma, et al., 2020), the authors proposed Who2com, a learnable handshake communication mechanism that enabled CAVs to selectively communicate based on relevance scores. By establishing connections solely with agents that provided valuable perceptual data, this method effectively reduced bandwidth usage while maintaining efficient and meaningful communication within the system. In contrast to these feature-based approaches, in (Hu et al., 2022), a spatial confidence map was proposed to identify perceptually critical regions, allowing CAVs to communicate only in areas with high spatial importance. The fusion scheme employed confidence-aware multi-head attention to prioritize spatially critical regions for feature sharing, enhancing detection while conserving bandwidth. To address localization errors caused by dynamic environments, V2X-ViT (Xu, Xiang, Tu, et al., 2022) utilized

a transformer-based architecture, combining heterogeneous multi-agent self-attention (HMSA) and multi-scale window attention (MSwin). HMSA facilitated the fusion of data from vehicles and infrastructure, while MSwin handled localization errors by applying attention across different spatial scales.

2.4.1.4 Summary of Fusion Schemes

To summarize, all cooperative data fusion paradigms can extend the FoV of a single AV and the accuracy of detected results. In raw-data-level fusion, it can most effectively exploit complementary information obtained from raw sensor observation. However, sharing all collected raw point clouds seems to be impractical due to the limitation of bandwidth and latency of networks. Also, the reconstruction of received data is not trivial, as it is taken from different positions and angles. Object-level fusion schemes are less complex and straightforward and require less bandwidth. However, the detected results from other CAVs are hard to authenticate and trust issues further complicate this matter (Q. Chen, Ma, et al., 2019). In addition, this approach only works when both vehicles share a reference object in their detection. This does not resolve the issue of previously undetected objects as they will remain undetected even after fusion. Intermediate data fusion is sufficient for the training process, and the small size of feature-based data makes it more efficient to achieve real-time edge computing without considering the risk of congesting the network. However, most of these works aim to optimize the trade-offs between bandwidth utilization and detection accuracy under the assumption of ideal communications and they ignores the realistic V2V communication caused distortions during the cooperative inference process.

2.4.2 Cooperative inference with IRSUs assistant

Different from V2V communication, which only involves CAVs collaborating with each other, roadside infrastructures offer unique advantages due to their static nature. Being stationary, they can provide continuous observation and perform long-term reasoning in complex traffic flow areas such as intersections, roundabouts, and T-junctions. Addition-

ally, roadside infrastructures allow for flexible sensor deployment and always leverage the power grid to support operation, which means they have more computing and communication capabilities than CAVs. Furthermore, roadside sensors play a fundamental role in overcoming the occlusion problem. With absolute ego-position, broader FoV, and highly optimized hardware units, roadside sensors can generate accurate detections and transfer their knowledge to CAVs. As shown in Table 2.1, CAVs and IRSUs offer distinct yet complementary capabilities across spatial, temporal, hardware, and software domains. This synergy, particularly with IRSUs providing a wider, continuous perspective, is crucial for overcoming the limitations of individual AVs and enabling comprehensive environment perception.

Domain	**Sub-Domain**	**CAVs**	**IRSUs**
Spatial	**Sensing Range**	Local, short-range, dense sensor deployment	Global multi-spot, long-range, flexible deployment
	FoV	90° – 120°	360° multiple views with cross coverage
	Blind Areas	Static and dynamic blind areas	Multi-sensor overlap eliminates blind areas
Temporal	**Time Range**	Dynamic and real-time perception, decision, control	Static, continuous observation, long-term reasoning
Hardware	**Data Acquisition**	On-board sensors	Sensors, traffic lights, traffic control systems
	Communication	Cellular network	NR-V2X and wired network
	Power	On-board battery	Power grid
Software	**Algorithms**	Object detection, segmentation, planning	Cooperation (cooperative perception, decision-making)

Table 2.1: Comparisons between CAV and IRSU

Combining CAVs and roadside infrastructure such as IRSUs to advance road safety and efficiency has become an active research topic in recent years (Tihanyi et al., 2021).

In (Krämmer et al., 2019), the authors proposed an ITS system, named Providentia, which was an intelligent infrastructure system that consisted of several gantries bridges with cameras and radars to provide additional information for vehicles to extend their perception range and help vehicles perceive blind spots. In "Providential", the roadside infrastructure acted as an edge computing node and oversaw the creation of a digital twin of the highway to detect all vehicles in the sensing areas. Data connected by radars and cameras were processed and fused before being shared with consumer vehicles via 5G. Similarly, in (Gabb et al., 2019), the authors applied mobile edge computing (MEC) to structure their system, in which they have shown a hybrid vehicular perception system. On the roadside, a central communication node, named the MEC server, was connected to a base station to process and fuse multi-model data collected from sensors attached to roadside infrastructures. The processed results were then distributed to intelligent vehicles close by to improve their perception capabilities. The University of Tokyo also contributed to this field by releasing an open-source project, AutoC2X (Tsukada et al., 2020), for cooperative perception between CAVs and roadside infrastructures. AutoC2X combined robot operating system (ROS)-based Autoware and OpenC2X to construct an infrastructure-based cooperative perception system. In their system, several roadside perception units (RSPUs) were connected via a wired network, and RSPUs shared their object information with nearby vehicles through V2X communication. Further advancing the concept of cooperative perception, the authors of (L. Wang et al., 2020) exploited the concept of digital twin and advanced driver assistance system (ADAS) to build a vehicle-to-cloud-based ADAS use case. In their system, intelligent vehicles were connected to the cloud server via vehicle-to-cloud (V2C) communication to upload perceived data to the cloud server. At the cloud server end, shared data were used to create a virtual world and sent back to connected vehicles to help better understand surrounding environments. In another notable effort, researchers from the University of Sydney demonstrated a cooperative perception system for ITS in (Shan et al., 2020). In their system, sensor-less vehicles were connected to well-equipped IRSUs and communicated with each other with messages that obeyed European Telecommunications Standard Institute (ETSI)

Collective Perception Messages (CPMs) standards. Their experiments demonstrated that CAVs with fewer sensors could observe vulnerable road users through the assistance of IRSUs.

Despite these advancements, the strategies of roadside sensor deployment in the real world are still challenging. The amount, locations, and orientation of sensors would affect the coverage of traffic areas and the overall cost of the system. There are two groups of sensor deployment strategies: 2D design (Senouci & Lehtihet, 2018) and 3D design (Saad et al., 2020). A significant portion of existing literature focuses on 2D sensor deployment strategies, as deploying sensors in 2D environments is generally simpler compared to 3D deployment. However, due to the widely use of 3D sensors like LiDAR in recent years, 3D sensor deployment has recently attracted increasing attention. Besides, 3D sensor deployments cannot be solved by simply extending or generalizing 2D methods, so new deployment approaches must be developed for 3D lidar sensor deployment. All the aforementioned work in the literature focused on the dissemination of IRSUs information to CAVs to help CAVs enlarge CAVs sensing range and alleviate occlusion problems. However, **these works largely overlooked leveraging the full potential of IRSUs' resources to facilitate a collaborative learning process, particularly for applications like object detection.** Utilizing IRSUs for collaborative learning could fundamentally enhance the performance of detection models, enabling more robust and accurate perception in complex traffic environments.

2.4.3 Cooperative inference with V2V communication

Cooperative perception with V2V communication involves information exchange among CAVs via wireless channels. Thus, factors like latency, bandwidth, and channel distortion must to be taken into account to maintain reliable and effective perception.

To address the challenges posed by latency, in SyncNet (Lei et al., 2022), a latency-aware collaborative perception system was proposed to actively synchronize the perceptual features shared by multiple CAVs. The system incorporated a feature attention synbiotic estimation component, leveraging a dual-branch pyramid LSTM network to

mitigate cascading estimation errors. Additionally, a time modulation mechanism was designed to attentively merge real-time and estimate features based on latency duration. This mechanism ensured that the estimation adapted according to the communication delay, and balanced the weight between real-time and historical data. To alleviate the communication overhead for cooperative perception with V2V communication, FPV-RCNN (Yuan et al., 2022) introduced a keypoints-based deep feature fusion framework using Furthest Point Sampling (FPS) to select keypoints and a Voxel Set Abstraction (VSA) module to aggregate multi-scale features around these keypoints. To further reduce the transmitted data volume, only features within the bounding box proposals were selected for sharing. In contrast to feature-based sharing approaches, where2comm (Hu et al., 2022), introduced a spatial confidence map to optimize bandwidth usage by identifying perceptually critical regions. This method allowed CAVs to communicate only in regions of high spatial importance, avoiding the transmission of irrelevant information. By employing confidence-aware multi-head attention, the system prioritized these critical regions, achieving a better trade-off between perception performance and communication efficiency. To enhance robustness against adversarial attacks and communication noise, ROBOSAC (Y. Li, Fang, et al., 2023) took a different approach by focusing on a defense mechanism. Unlike traditional methods that depend on prior knowledge of attack patterns, ROBOSAC employed an attacker-agnostic, sampling-based defense mechanism. This innovative approach ensured reliable cooperative perception even under unpredictable and hostile conditions, demonstrating its effectiveness in safeguarding perception systems.

While most research efforts concentrate on reducing bandwidth usage and communication delay under idealized information exchange assumptions, only a few studies have explored realistic V2V communication. For instance, authors of (C. Liu et al., 2023) incorporated a learning-based communication channel to account for Rician fading and path loss in cooperative perception. Unlike traditional approaches that rely on fixed channel models, this method adaptively modeled wireless communication dynamics, enabling more accurate cooperative perception under varying channel conditions. Building on this,

authors of (C. Liu et al., 2024) developed a self-supervised adaptive weighting model designed to mitigate the adverse effects of channel distortion in scenarios involving Rician fading and imperfect channel state information. To further addressing the challenges, in (J. Li, Xu, et al., 2023), a Lossy Communication-aware Repair Network and a V2V Attention Module is developed to improve the resilience of V2V cooperative perception systems under non-ideal communication conditions.

In the context of real-world CAVs' cooperative perception with V2V communication, it is imperative to acknowledge that CAVs exhibit non-stationary behavior and possess time-variant statistical properties under complex traffic conditions. Specifically, parameters such as velocity, acceleration, inter-vehicle spacing, location, and orientation are subject to continuous change. These dynamic attributes of CAVs exert a significant influence on the V2V communication channels, attributable to the mobile nature of the transmitters and receivers integrated into the vehicles. However, most of these works rely solely on free-space or long-distance path loss assumptions, which fail to capture the dynamic characteristics of vehicular communication channels. Additionally, V2V communication models are prone to various factors that can impair the shared data, such as Doppler shifts (Zhao & Haggman, 2001) from moving CAVs, multi-path effects (Dahech et al., 2017) from static or dynamic objects, and changes in topology (Schwartz & Stern, 1980) due to routing failures. As a result, the communication channels experience rapid fluctuations, necessitating careful consideration in the modeling and analysis of V2V communication systems. Therefore, identifying effective methods to counteract performance degradation due to channel distortion remains a critical challenge.

Diffusion models have received unprecedented success in artificial intelligence generated content (AIGC), such as image super-resolution (Saharia et al., 2022), image recovery (Xia et al., 2023), audio enhancement (Lu et al., 2022), and speech synthesis (Kang et al., 2023). There are also some works applying the diffusion models to wireless communications, where the diffusion model acts as a generative model to simulate the wireless communication channel by progressively adding noise to data samples (the forward diffusion process) and then learning to reverse this noise (the denoising process).

In (T. Wu et al., 2024), a channel denoise diffusion model was devised to mitigate the noise in wireless semantic communication, specifically in the context of wireless image transmission over the additive white Gaussian noise (AWGN) and Rayleigh fading. In (Choukroun & Wolf, 2022), the authors proposed a Denoising Diffusion Error Correction Code (DDECC) where the diffusion model was adopted to provide a reliable error correction for wireless communication over noisy channels. The reverse process decoded and recovered the codeword conditioned on the number of parity check errors to provide information about the noise level. In (M. Kim et al., 2023), denoising diffusion probabilistic models (DDPM) were employed to generate channels to simulate real-world wireless channels. The diffusion model enhanced wireless systems by accurately modelling channel conditions and denoising transmitted data, achieving a symbol error rate (SER) close to the channel-aware framework. Moreover, in semantic communications, (Grassucci et al., 2023) proposed a novel generative diffusion-guided framework for a noise-resilient semantic communication system capable of synthesizing high-quality images while preserving semantic content under wireless conditions. In (J. Chen et al., 2024), the CommIN was devised that combined Ivertible neural networks (INN) with diffusion models to redefine joint source-channel coding (JSCC) for wireless image transmission, enhancing the semantic and perceptual fidelity of image in noisy wireless communication settings. A diffusion model-based generative audio semantic communication framework was proposed in (Grassucci et al., 2024), which using latent representations and semantic guidance to restore both corrupted and missing parts of the audio, achieving a semantically accurate audio reconstruction that was robust to transmission noise and data loss. However, existing research on diffusion model-based methods in wireless communication primarily focused on utilizing diffusion models as decoders to generate images or audio, while overlooking their potential ability for denoising and enhancing the quality of received signal. Moreover, these diffusion model-based methods were tailored to particular channel conditions (e.g., AWGN and Rayleigh fading), which could make it less adaptable to dynamic or multi-path channels without retraining or additional tuning. For instance, in practical V2V communication scenarios, obtaining precise chan-

nel state information (CSI) can be challenging, especially considering the high-mobility nature of CAVs in wireless traffic scenarios where channel estimation errors are frequent. Inaccurate CSI can adversely affect the model's denoising capability and degrade the quality of the resulting transmissions. Furthermore, deploying diffusion models in diverse real-world V2V scenarios often require model reconfiguration or adjustments to accommodate specific channel characteristics. This requirement can complicate both deployment and maintenance, reducing scalability. While diffusion models demonstrate robustness in moderate noise conditions, they may struggle in high-noise environments where signal degradation is severe. The iterative nature of diffusion could amplify certain noise patterns or introduce artifacts, impacting model stability and performance.

2.5 Research Gap

Through the reviews of the existing literature, the following research gaps are identified.

1. Lack of Efficient approaches of roadside sensors deployment for road-assisted cooperative training of perception models.

To achieve effective road-assisted cooperative training, the position of sensors should be carefully designed. Existing literature on 2D sensor deployment cannot be efficiently extended to 3D sensor deployment, as they have different sensor data structures. In addition, 3D sensor deployments in literature only focuses on maximizing the coverage of the traffic areas or maximizing the visibility of detected objects. Apart from providing additional sensing information for CAVs, roadside sensors also aim to assist CAVs in conducting the learning process and update the pretrained models. In 3D object detection, the pose of roadside sensors also determines the geometric features of detected objects due to the reflected point clouds on the different parts of the object's surface, which further leads to different learning knowledge extracted from point clouds. To this end, conventional roadside sensor deployment strategies only consider the sensing requirement for perceptions while ignoring the learning requirement that assists CAVs in learning roadside knowledge for model refinements. It is highly desirable to lever-

age IRSUs and prioritize knowledge sharing among CAVs and IRSUs when designing roadside sensors placements in cooperative training schemes.

2. An Efficient road-assisted cooperative training scheme among CAVs and IRSUs to update pre-trained models and gain the models' generalization when CAVs encounter out-of-distribution data in open traffic scenarios. Existing works on training perception models often adopt centralized approaches, where the DNN-based models only train one time and deploy on CAVs for inference. However, those pretrained models may not generalize well enough. There are two reasons for the poor generalisability: 1) sensing uncertainty, the sensing data at each CAV may be sparse, noisy, or missing due to the limitation of hardware and environmental complexity such as occlusion or long distance; 2) parameter uncertainty, the downloaded model may be compressed or pruned due to the limited computing, communication, and resources at CAVs. To overcome both uncertainty problems, a direct approach is to collect large-scale datasets and adopt a perfectly optimized model. However, this would not only cause unbearable communication overhead but also require high-cost labor for accurate annotation. A more promising solution for addressing sensing and parameter uncertainties is to leverage newly collected data from CAVs and IRSUs in combination with a FL scheme. This approach is particularly effective when CAVs encounter out-of-distribution (OOD) data in open traffic scenarios, as it enables the detection models to be fine-tuned collaboratively and dynamically, ensuring improved adaptability and accuracy in diverse real-world environments. However, in vanilla FL approaches, manual data annotation is available, and those approaches are always based on supervised learning. Thus, the vanilla FL approaches become ineffective for FL for CAVs in the AD context, where annotating large-scale, real-time data is impractical. It is urgent to find an efficient road-assisted FL scheme for AD context to tackle the cooperative model training with newly collected unlabeled data.

3. Lack of practical V2X communication analysis for CAVs-IRSUs collaboration in the network vehicular system.

Existing research on cooperative training and inference either under the assumption

of ideal channel conditions or under a simple V2V communication model that only considers CAV distance-related large-scaling fading (LSF) factors, which is not practical in real-world networked vehicular systems. Realistic V2V communication among CAVs also needs to consider time-variant parameters, including CAVs' velocity, acceleration, trajectory, and time-variant distances, which makes them not practical in real-world traffic scenarios. It is urgent to develop realistic V2V communication channel models that capture the uncertainty and stochastic variability of real-world V2V communication. This will enable accurate analysis of communication-induced impairments for both cooperative model training and inference.

4. Lack of robust cooperative learning scheme for the domain specific FLCAV with V2V communications.

In domain-specific FL among CAVs and IRSUs, each CAV uploads its local model to the IRSUs. However, transmitting these high-dimensional perception models creates significant communication overhead, especially given the heterogeneous network conditions and varying on-board resources of CAVs. This results in substantial communication latency. Existing works of cooperative training primarily focus on simple simulated tasks, such as image classifications and digital recognitions. There is an urgent need to investigate communication protocol optimization techniques that minimize communication delay and resource consumption, thereby accelerating convergence speed for domain-specific FLCAVs.

5. Lack of robust methods to mitigate channel impairment for cooperative inference with V2V communication.

For cooperative inference, CAVs can share their sensing information through V2V communication channels and fuse the received messages, allowing the ego vehicle to utilize this collective knowledge to improve the performance of inference. A significant area of ongoing research focuses on designing fusion methods to enhance perception accuracy or developing communication-efficient algorithms to reduce bandwidth usage, often under the assumption of ideal channel conditions. However, these studies have not thoroughly explored communication channel models to investigate the effects of channel

impairments. It is crucial to thoroughly investigate channel impairments in cooperative inference under realistic V2V communication models and develop robust methods to mitigate their negative impacts.

CHAPTER 3

ROAD ASSISTED COOPERATIVE MODEL TRAINING FOR PERCEPTION

3.1 Introduction

This chapter focuses on the design of a system to achieve efficient road-assisted cooperative training for percetion among CAVs and IRSUs. The objective is to enhance the DNN-based perception model when encountering OOD data in open driving scenarios.

Detecting OOD data in open driving scenarios holds significant importance for AD since OOD data encapsulates rare cases in driving scenarios and contributes to the enhancement of model generalization. Federated Learning in AD (FLAD), which updates DNNs in a distributed manner whenever OOD data is encountered, is an effective solution to enhance and robustify DNNs against OOD data. However, in contrast to existing FL applications (e.g., image classification (M. Chen et al., 2021)) where manual data annotation is available, FL between CAVs and IRSUs often needs to operate on newly collected, unlabeled data samples, due to their high-privacy and high mobility nature. Consequently, the vanilla FL approaches (Q. Wu et al., 2020) based on supervised learning become ineffective. Several studies have explored FL for AD (as illustrated in

Table 3.1). To address the above problem and reduce labor cost for manual annotation for the newly collected data, emerging FL approaches proposed model average (e.g., ensemble moving average (EMA)) (T. Kim et al., 2024) and logit average (e.g., ensemble attention distillation (EAD)) (Gong et al., 2021) to generate pseudo labels. However, these works fail to exploit the geometric relationship among different data frames. To this end, multi-view fusion distillation (MVFD) has been proposed for generating pseudo labels, e.g., feature-level MVFD (Zheng et al., 2023), and box-level MVFD (Z. Zhang et al., 2021). Nonetheless, leveraging consensus between CAVs could potentially lead to higher bias.

Besides, when IRSUs operate within the cooperative training loop, the performance of FL is highly dependent on the associated road sensor placements. Most existing sensor placement approaches follow the road topology (i.e., topology-based approaches) (Arnold et al., 2020; Cai et al., 2023) or maximize the sensing coverage (Gonzalez-Barbosa et al., 2009) (i.e., coverage-based approaches). While these methods are effective for traffic flow monitoring in the ITS system, they are inefficient for the road-assisted cooperative learning process as they ignore the specific learning requirements of FL process. Specifically, road assisted FL needs to integrate road annotation/placement and FL features for joint optimization, for which the existing MVFD FL algorithm become inefficient, as they ignore the inter-dependency between low-level road designs and the high-level FL requirement. There are also multi-view fusion methods and IRSUs sensor placement strategies developed for non-FL scenarios. While these methods are effective for traffic flow monitoring, they are inefficient for RSFL as they ignore the requriement of FL services.

To address these problems, in this chapter, we propose a road supervised FL (RSFL) framework, categorized under the MVFD-type solution, to overcome the bias issue by introducing IRSUs. In addition, to satisfy the new requirement of RSFL, we propose a bug-aware road labeling (BARL) algorithm that leverage roadside knowledge to generate pseudo labels (automatically generated by the BARL algorithm) for correcting false positives (i.e., the model predict an object, but there is no ground truth object at that location), false negatives (i.e., there is a ground truth object, but the model fails to

Type	Reference	Method	FL	w/o Full Label	MVFD	Road Labeling	Labeling Rule	Road Placement	Placement Rule
2D	(Posner et al., 2021)	FedAvg	✔	✗	✗	✗	manual	✗	✗
	(Papernot et al., 2022)	PATE	✗	✔	✗	✗	model-fusion	✗	✗
	(T. Kim et al., 2024)	EMA	✔	✔	✗	✗	model-fusion	✗	✗
	(Gong et al., 2021)	EAD	✔	✔	✗	✗	logits-fusion	✗	✗
	(Gonzalez-Barbosa et al., 2009)	CameraPlace	✗	✗	✗	✗	manual	✔	max. coverage
3D	(S. Wang, Hong, et al., 2022)	EdgeFL	✔	✗	✗	✗	manual	✗	✗
	(J. Yang et al., 2022)	ST3D++	✗	✔	✗	✗	feature-fusion	✗	✗
	(Zheng et al., 2023)	AutoFed	✔	✔	✔	✗	feature-fusion	✗	✗
	(Z. Zhang et al., 2021)	EDFL	✔	✔	✔	✗	box-fusion	✗	✗
	(Arnold et al., 2020; Cai et al., 2023; Vijay et al., 2021)	LiDARPlace	✗	✗	✗	✗	manual	✔	topology-based
	Ours	RSFL	✔	✔	✔	✔	point-box fusion	✔	max. information gain

Table 3.1: Comparison of federated learning methods for autonomous driving.

predict it), and inaccurate bounding boxes produced by local models running on CAVs. Specifically, these pseudo labels are generated by aggregating information from multiple IRSUs and CAVs, which provide a broader and more accurate contextual understanding of the environment. They serve as corrective signals, enabling the system to adjust and refine erroneous predictions made by individual CAVs, which are limited by their own sensor perspectives and occlusions. Unlike conventional manual labeling methods that require time-consuming human annotations, pseudo labeling allows CAVs to continuously adapt to new and unseen scenarios in real time, without waiting for external annotation processes. To gain insights into RSFL with BARL, We derive the information gain of annotating objects with a road sensor, with respect to its placement location, by leveraging the *expected entropy reduction*. As such, the problem of maximizing the information gain under road placement constraints is formulated, and a bug-aware sensor placement (BASP) algorithm is proposed based on integer programming. It is shown that BASP is equivalent to optimizing a surrogate function that approximates the

total information gain in Bayesian optimization. The main contributions are summarized below:

- Propose and verify an RSFL framework with BARL for AD with high-fidelity experiments, and derive the information gain brought by road supervision.
- Derive a bug-aware sensor placement (BASP) algorithm based on integer programming and verify its superiority.
- Prove that BASP is equivalent to optimizing an approximate function of the total information gain.

3.2 RSFL System Model

We consider a FLAD system with K CAVs. The LiDAR data at the k-th CAV ($k \in \{1,\dots,K\}$) at the t-th ($t \in \{1,\dots,T\}$) LiDAR time frame, is denoted as ${}^{V}\mathscr{D}_k = \{{}^{V}\mathbf{d}_{k,1}, {}^{V}\mathbf{d}_{k,2}, \cdots\}$, where ${}^{V}\mathbf{d}_{k,t} \in \mathbb{R}^{D_k \times 3}$ is the vector concatenating the coordinates of all points, with D_k being the number of points in each cloud. The DNN parameter vector at the k-th vehicle is $\mathbf{w}_k \in \mathbb{R}^{W_k \times 1}$ with W_k being the dimension of each DNN. The k-th DNN maps ${}^{V}\mathbf{d}_{k,t}$ into a set of bounding boxes

$$ {}^{V}\mathscr{B}_{k,t} = \left\{ {}^{V}\mathbf{b}_{k,t}^{[1]}, {}^{V}\mathbf{b}_{k,t}^{[2]}, \cdots \right\}, \tag{3.1} $$

where ${}^{V}\mathbf{b}_{k,t}^{[n]}$ represents the n-th object at the t-th LiDAR frame in the k-th vehicle coordinate system, and its label format is given by

$$ \mathbf{b} = [c, x, y, z, l, w, h, \theta]^T, \tag{3.2} $$

where c is the category, (x,y,z) is the center position, (l,w,h) is the tuple of length, width, and height, and θ denotes the yaw rotation, respectively. DNN inference at the k-th vehicle can be written as ${}^{V}\mathscr{B}_{k,t} = {}^{V}\Phi_k({}^{V}\mathbf{d}_{k,t}|\mathbf{w}_k)$, where ${}^{V}\Phi_k$ represents the DNN inference function.

The goal of AD perception is to make ${}^{V}\mathscr{B}_{k,t}$ an accurate structured representation of ${}^{V}\mathbf{d}_{k,t}$. However, ${}^{V}\mathscr{B}_{k,t}$ cannot match ${}^{V}\mathbf{d}_{k,t}$ if there exist OOD points in ${}^{V}\mathbf{d}_{k,t}$. Consequently,

we need the FLAD procedure for fine-tuning $\mathbf{w}_k$ in a distributed manner. Specifically, the FLAD aims to acquire a global DNN with parameter vector $\mathbf{g}$, by solving the following loss minimization problem:

$$\min_{\{\mathbf{w}_k\},\mathbf{g}} \underbrace{\frac{1}{\sum_k |{}^V\mathscr{D}_k^\diamond|} \sum_k \sum_{({}^V\mathbf{d}_{k,t},{}^V\mathscr{B}_{k,t}^\diamond)\in {}^V\mathscr{D}_k^\diamond} \Theta(\mathbf{g};{}^V\mathbf{d}_{k,t},{}^V\mathscr{B}_{k,t}^\diamond)}_{:=\Lambda(\mathbf{g})}$$
$$\text{s.t.} \quad \mathbf{w}_1 = \cdots = \mathbf{w}_K = \mathbf{g}, \tag{3.3}$$

where ${}^V\mathscr{B}_{k,t}^\diamond$ is the set of pseudo labels obtained from a *teacher model* Φ[1], $\Theta(\mathbf{g};{}^V\mathbf{d}_{k,t},{}^V\mathscr{B}_{k,t}^\diamond)$ is the loss function corresponding to a single sample $({}^V\mathbf{d}_{k,t},{}^V\mathscr{B}_{k,t}^\diamond)$ $(1 \le t \le |{}^V\mathscr{D}_k^\diamond|)$ in ${}^V\mathscr{D}_k^\diamond = \{({}^V\mathbf{d}_{k,t},{}^V\mathscr{B}_{k,t}^\diamond)\}_t$ with respect to parameter vector $\mathbf{g}$, and $\Lambda(\mathbf{g})$ denotes the global loss function to be minimized.

Now let $\mathbf{g}^{[0]}$ denote the pre-trained DNN at the cloud, and let $\mathbf{w}_k^{[i]}(0)$ denote the local DNN parameters at AV k at the beginning of the i-th iteration ($i \ge 0$ and $\mathbf{w}_k^{[i]}(0) = \mathbf{g}^{[i]}$). The FL training of model parameters (i.e., solving Eq. (3.3)) is a distributed and iterative procedure, where each iteration involves the following two steps:

1) The k-th vehicle first minimizes the loss function via the gradient descent approach[2] as

$$\mathbf{w}_k^{[i]}(\tau+1) = \mathbf{w}_k^{[i]}(\tau) - \frac{\varepsilon}{|{}^V\mathscr{D}_k^\diamond|} \sum_{({}^V\mathbf{d}_{k,t},{}^V\mathscr{B}_{k,t}^\diamond)\in {}^V\mathscr{D}_k^\diamond} \nabla\Theta(\mathbf{w}_k^{[i]}(\tau);{}^V\mathbf{d}_{k,t},{}^V\mathscr{B}_{k,t}^\diamond), \tag{3.4}$$

where ε is the step-size, $0 \le \tau \le E-1$ (E is the number of local updates) and $\nabla\Theta$ denotes the gradient of Θ.

2) All CAVs upload $\{\mathbf{w}_k^{[i]}(E)|\forall k\}$ to the server, which computes an average model

$$\mathbf{g}^{[i+1]} = \frac{1}{K}\sum_{k=1}^K \mathbf{w}_k^{[i]}(E), \tag{3.5}$$

and then broadcast to the CAVs for next-round updates.

[1]The set of ground truth labels is denoted as ${}^V\mathscr{B}_{k,t}^*$.

[2]If $|\mathscr{D}_k|$ is large, stochastic gradient descent can be adopted to accelerate the training speed.

This completes one FL round and we set $i \leftarrow i+1$. The entire procedure stops when $i = I_{\mathrm{FL}}$ with I_{FL} being the number of federated learning rounds

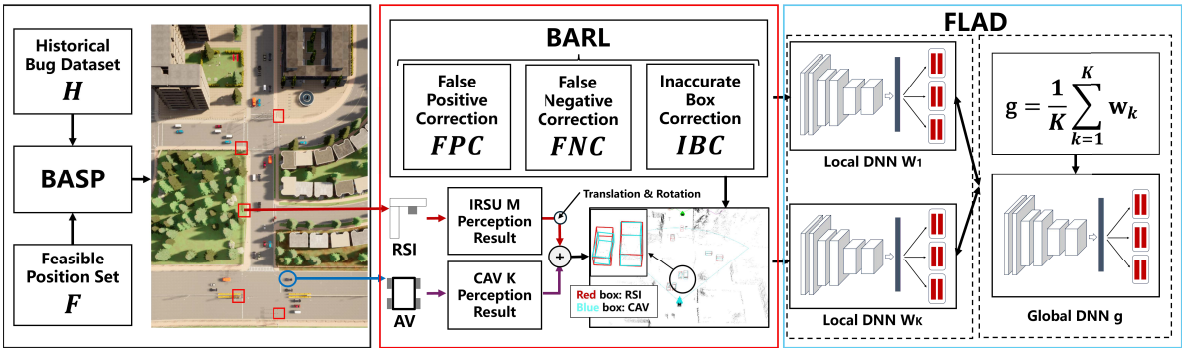


Figure 3.1: RSFL framework, which consists of the BASP and BARL modules. Road-detected boxes are marked in red. Vehicle-detected boxes are marked as cyan.

3.3 Proposed RSFL Approach

The key to FLAD is to find a proper teacher model Φ such that the pseudo labels in ${}^{V}\mathscr{B}^{\diamond}_{k,t}$ is closer to ground truth. To this end, we propose the RSFL framework shown in Figure 3.1, which consists of the BARL and BASP modules for the FLAD system presented in Section 3.2. The BARL module leverages IRSUs to annotate data at CAVs. The BASP module determines the positions of IRSUs using bug database and integer programming. Below we present the details of the two newly developed modules.

3.3.1 BARL: Bug Aware Road Labeling:

Denote the LiDAR data at IRSU m ($1 \le m \le M$) as ${}^{R}\mathscr{D}_m = \{{}^{R}\mathbf{d}_{m,1}, {}^{R}\mathbf{d}_{m,2}, \cdots\}$, where ${}^{R}\mathbf{d}_{m,t}$ represents a frame of road point cloud. The DNN inference at IRSU m at LiDAR time t is ${}^{R}\mathscr{B}_{m,t} = {}^{R}\Phi_m({}^{R}\mathbf{d}_{m,t}|\mathbf{r}_m)$, where $\mathbf{r}_m \in \mathbb{R}^{R_m \times 1}$ is the DNN parameter vector at IRSU m.

Now, a direct approach is to label ${}^{V}\mathscr{D}_k$ with ${}^{R}\mathscr{B}_{m,t}$ via the associated rotation and translation matrices between CAV k and IRSU m. However, IRSU m also suffers from occlusion, and its perception performance of a certain object could be even worse than that at CAV k. To improve the quality of pseudo labels, we propose to merge the perception results of all IRSUs into a global object list ${}^{G}\mathscr{B}_t = \{{}^{G}\mathbf{b}_t^{[1]}, {}^{G}\mathbf{b}_t^{[2]}, \cdots\}$. This is realized

Algorithm 1 RSFL with BARL

Input: ${}^{V}\mathscr{D}_k$, ${}^{G}\mathscr{B}_t$, ${}^{V}\mathscr{B}_{k,t}$, $F_{G\to k}$, $F_{k\to G}$, $\mathbf{g}^{[0]}$, I_{FL}, E
Output: $\mathbf{g}^{[I_{\mathrm{FL}}]}$
Function RS(${}^{G}\mathscr{B}_t$, ${}^{V}\mathscr{B}_{k,t}$)**:**
  Map ${}^{G}\mathscr{B}_t$ to AV k as $\mathscr{C}_{k,t} = F_{G\to k}({}^{G}\mathscr{B}_t)$ Remove hidden points from $\mathscr{C}_{k,t}$ to acquire $\mathscr{C}^{\diamond}_{k,t}$
  for $\mathbf{c} \in \mathscr{C}^{\diamond}_{k,t}$ **do**
    if $\mathrm{IoU}(\mathbf{c}, {}^{V}\mathbf{b}^{[n]}_{k,t}) = 0,\ \forall n$ **then**
      Update ${}^{V}\mathscr{B}^{\diamond}_{k,t} \leftarrow {}^{V}\mathscr{B}^{\diamond}_{k,t} \cup \{\mathbf{c}\}$
    if $\delta \le \mathrm{IoU}(\mathbf{c}, {}^{V}\mathbf{b}^{[n]}_{k,t}) \le \alpha$ **then**
      Update ${}^{V}\mathscr{B}^{\diamond}_{k,t} \leftarrow {}^{V}\mathscr{B}^{\diamond}_{k,t} \setminus \{{}^{V}\mathbf{b}^{[n]}_{k,t}\} \cup \{\mathbf{c}\}$
  Map ${}^{V}\mathscr{B}_{k,t}$ to global as $\mathscr{A}_{k,t} = F_{k\to G}({}^{V}\mathscr{B}_{k,t})$ Remove hidden points from $\mathscr{A}_{k,t}$ to acquire $\mathscr{A}^{\diamond}_{k,t}$ **for** $\mathbf{c} \in \mathscr{A}^{\diamond}_{k,t}$ **do**
    if $\mathrm{IoU}(\mathbf{c}, {}^{G}\mathbf{b}^{[n]}_{t}) = 0$ *for all* n **then**
      Update ${}^{V}\mathscr{B}^{\diamond}_{k,t} \leftarrow {}^{V}\mathscr{B}^{\diamond}_{k,t} \setminus \{F_{G\to k}(\mathbf{c})\}$
Function FL**:**
  for $k \leftarrow 1$ **to** K **do**
    Execute RS(${}^{G}\mathscr{B}_t$, ${}^{V}\mathscr{B}_{k,t}$) to obtain RS dataset ${}^{V}\mathscr{D}^{\diamond}_k = \{({}^{V}\mathbf{d}_{k,1}, {}^{V}\mathscr{B}^{\diamond}_{k,1}), ({}^{V}\mathbf{d}_{k,2}, {}^{V}\mathscr{B}^{\diamond}_{k,2}), \cdots\}$
  for $i \leftarrow 0$ **to** $I_{\mathrm{FL}} - 1$ **do**
    for $k \leftarrow 1$ **to** K **do**
      for $\tau \leftarrow 0$ **to** $E - 1$ **do**
        Update local model using Eq. (3.4)
    Update global model using Eq. (3.5)

by exploiting the late fusion cooperative perception algorithm developed in (Z. Zhang et al., 2021). After fusion, the road server executes BARL(${}^{G}\mathscr{B}_t$, ${}^{V}\mathscr{B}_{k,t}$) to generate the pseudo label dataset ${}^{V}\mathscr{D}^{\diamond}_k = \{({}^{V}\mathbf{d}_{k,1}, {}^{V}\mathscr{B}^{\diamond}_{k,1}), ({}^{V}\mathbf{d}_{k,2}, {}^{V}\mathscr{B}^{\diamond}_{k,2}), \cdots\}$, which is fed to the FLAD. The entire procedure of BARL is summarized in Algorithm 1.

In particular, for the k-th vehicle at time t, we first initialize ${}^{V}\mathscr{B}^{\diamond}_{k,t} = {}^{V}\mathscr{B}_{k,t}$ and then consider three cases to update ${}^{V}\mathscr{B}^{\diamond}_{k,t}$.

1) **False Negative Correction (FNC)**. We map ${}^{G}\mathscr{B}_t$ to the vehicle coordinate system as $\mathscr{C}_{k,t} = F_{G\to k}({}^{G}\mathscr{B}_t)$, where $F_{G\to k}$ is the function mapping coordinates from global frame to local frame. We then perform view frustum culling and hidden point

removal on $\mathscr{C}_{k,t}$ with respect to the FoV of the k-th vehicle, which yields $\mathscr{C}^{\diamond}_{k,t}$. For any box $\mathbf{c} \in \mathscr{C}^{\diamond}_{k,t}$, if $\text{IoU}(\mathbf{c}, {}^{V}\mathbf{b}^{[n]}_{k,t}) = 0$ for all n and $\mathbf{c}$ is learnable for the CAV[3], then we update ${}^{V}\mathscr{B}^{\diamond}_{k,t} = {}^{V}\mathscr{B}^{\diamond}_{k,t} \cup \{\mathbf{c}\}$, where IoU is intersection over union function (Arnold et al., 2020).

2) **False Positive Correction (FPC)**. We map ${}^{V}\mathscr{B}_{k,t}$ to the global coordinate system as $\mathscr{A}_{k,t} = F_{k\to G}({}^{V}\mathscr{B}_{k,t})$, where $F_{k\to G}$ is a reverse mapping of $F_{G\to k}$. We then perform view frustum culling and hidden point removal on $\mathscr{A}_{k,t}$ with respect to the FoV of RSIs, which yields $\mathscr{A}^{\diamond}_{k,t}$. For any box $\mathbf{c} \in \mathscr{A}^{\diamond}_{k,t}$, if $\text{IoU}(\mathbf{c}, {}^{G}\mathbf{b}^{[n]}_{t}) = 0$ for all n, then we update ${}^{V}\mathscr{B}^{\diamond}_{k,t} = {}^{V}\mathscr{B}^{\diamond}_{k,t} \setminus \{F_{G\to k}(\mathbf{c})\}$.

3) **Inaccurate Box Correction (IBC)**. For any box $\mathbf{c} \in \mathscr{C}^{\diamond}_{k,t}$, if there exists some n such that $\delta \le \text{IoU}(\mathbf{c}, {}^{V}\mathbf{b}^{[n]}_{k,t}) < \alpha$, where (δ, α) are predefined thresholds, e.g., we set $(\delta, \alpha) = (0.05, 0.7)$, then we update ${}^{V}\mathscr{B}^{\diamond}_{k,t} = {}^{V}\mathscr{B}^{\diamond}_{k,t} \setminus \{{}^{V}\mathbf{b}^{[n]}_{k,t}\} \cup \{\mathbf{c}\}$.

After iterating all the elements in ${}^{V}\mathscr{B}_{k,t}$ and ${}^{G}\mathscr{B}_{t}$, the resultant ${}^{V}\mathscr{B}^{\diamond}_{k,t}$ is our desired pseudo label set. By concatenating the results of all time frames, the RS dataset ${}^{V}\mathscr{D}^{\diamond}_{k}$ at the k-th vehicle is obtained.

3.3.2 Information Gain of BARL:

To quantify the benefit brought by BARL, we first consider the single-IRSU case and evaluate the information gain of annotating the objects with a roadside sensor. Specifically, consider the i-th object at the k-th vehicle at the t-th frame, and denote its points as $\mathbf{p}_i \in {}^{V}\mathbf{d}_{k,t}$. The associated ground truth box is $\mathbf{b}^{*}_{i}$, with its format given by Eq. (3.2). However, the generated box $\hat{\mathbf{b}}$ at the CAV and box $\mathbf{b}^{\diamond}$ at the IRSU involve uncertainties. To quantify their uncertainties, we model the elements in $\hat{\mathbf{b}}$ and $\mathbf{b}^{\diamond}$ as random variables and compute the associated entropy.

Let $\hat{\mathbf{b}}_i = [\hat{c}_i, \hat{x}_i, \hat{y}_i, \hat{z}_i, \hat{l}_i, \hat{w}_i, \hat{h}_i, \hat{\theta}_i]^T$, which contains 1 discrete variable (i.e., category $\hat{c}_i$) and 7 continuous variables (i.e., position $(\hat{y}_i, \hat{z}_i, \hat{l}_i)$, size $(\hat{l}_i, \hat{w}_i, \hat{h}_i)$, and orientation $\hat{\theta}_i$). The probability mass function (pmf) of $\hat{c}_i$ is $p_{\hat{c}_i}$. For instance, in a binary classification,

[3]A learnable object should contain a sufficient number of points observed by the CAV.

$\hat{c}_i = 0$ represents car and $\hat{c}_i = 1$ represents person. Then, the pmf $[p_{\hat{c}_i}(\hat{c}_i = 0), p_{\hat{c}_i}(\hat{c}_i = 1)] = [0.2, 0.8]$ means that the object is classified as a person with a probability of 0.8. On the other hand, since $(\hat{x}_i, \hat{y}_i, \hat{z}_i, \hat{l}_i, \hat{w}_i, \hat{h}_i, \hat{\theta}_i)$ are continuous random variables, we model their uncertainties using probability density functions (pdfs), which are given by $(p_{\hat{x}_i}, p_{\hat{y}_i}, p_{\hat{z}_i}, p_{\hat{l}_i}, p_{\hat{w}_i}, p_{\hat{h}_i}, p_{\hat{\theta}_i})$. For instance, for a box position, the mean of $(p_{\hat{x}_i}, p_{\hat{y}_i}, p_{\hat{z}_i})$ is the output box and the variance of $(p_{\hat{x}_i}, p_{\hat{y}_i}, p_{\hat{z}_i})$ is the perturbation. Based on the above pmf and pdf models, the entropy of $\hat{\mathbf{b}}_i$ is given by

$$H(\hat{\mathbf{b}}_i) = -\sum_t p_{\hat{c}_i}(t)\log(p_{\hat{c}_i}(t)) - \sum_{u\in\{\hat{x}_i,\hat{y}_i,\hat{z}_i,\hat{l}_i,\hat{w}_i,\hat{h}_i,\hat{\theta}_i\}} \int p_u(t)\log(p_u(t))\,dt. \tag{3.6}$$

Similarly, the entropy of $\mathbf{b}_i^{\diamond}$ is $H(\mathbf{b}_i^{\diamond})$.

Now we consider two cases.

- If the IoU between $\hat{\mathbf{b}}_i$ and $\mathbf{b}_i^{\diamond}$ is larger than α, then the IRSU would agree with the ego-vehicle and would not change the bounding box. In this case, the entropy is unchanged and we denote this event as $\mathbf{b}_i^{+}$.
- If the IoU between $\hat{\mathbf{b}}_i$ and $\mathbf{b}_i^{\diamond}$ is smaller than α, then the IRSU would change the bounding box. In this case, the entropy is reduced and we denote this event as $\mathbf{b}_i^{-}$.

According to (W. Zhang et al., 2022), the expected information gain (IG) of annotating $\mathbf{p}_i$ by IRSU is given by

$$IG_i = P\left(\mathbf{b}_i^{-}\right)\left[H(\hat{\mathbf{b}}_i) - H\left(\mathbf{b}_i^{\diamond}\right)\right]. \tag{3.7}$$

The total IG of annotating all data is given by

$$IG_{\text{sum}} = \sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{\mathbf{p}_i\in{}^{V}\mathbf{d}_{k,t}} P\left(\mathbf{b}_i^{-}\right)\left[H(\hat{\mathbf{b}}_i) - H\left(\mathbf{b}_i^{\diamond}\right)\right]. \tag{3.8}$$

3.4 BASP: Bug-Aware Sensor Placement:

The IG_{sum} in Eq. (3.8) is defined with respect to a single IRSU. However, in the multi-IRSU case, IG_{sum} would also depend on the sensor placement vector $\mathbf{v} = [v_1, \cdots, v_M]^T \in$

$\{0,1\}^M$ (M is the number of candidate locations), where $v_i = 1$ represents that the i-th position is selected as a placement site and $v_i = 0$ denotes that the i-th position is abandoned. The number of deployed sensors should satisfy $\sum_{m=1}^{M} v_m = L$. Furthermore, since roadside sensors can only be attached to utility poles, the set of candidate locations, denoted as $\mathscr{F}$ (with $|\mathscr{F}| = M$), is known, and we denote $\mathbf{r}_m = [r_{m,x}, r_{m,y}, r_{m,z}]^T \in \mathscr{F}$ as the location of the m-th feasible position to attach sensors.

Based on the above sensor placement model, we rewrite Eq. (3.8) into its multi-IRSU form:

$$\Psi(\mathbf{v}) = \sum_{k=1}^{K} \sum_{t=1}^{T} \sum_{\mathbf{p}_i \in {}^{V}\mathbf{d}_{k,t}} P\left(\mathbf{b}_i^- | \mathbf{v}\right) \left[H(\hat{\mathbf{b}}_i) - H\left(\mathbf{b}_i^{\diamond} | \mathbf{v}\right) \right], \tag{3.9}$$

where the labeling probability $P\left(\mathbf{b}_i^- | \mathbf{v}\right)$ and entropy of IRSU's detection $H\left(\mathbf{b}_i^{\diamond} | \mathbf{v}\right)$ are now dependent on $\mathbf{v}$. The problem of maximizing IG for all data with respect to the sensor placement $\mathbf{v}$ is

$$(\mathbf{P0}) \quad \max_{\mathbf{v}} \quad \Psi(\mathbf{v}) \tag{3.10a}$$

$$\text{s.t.} \quad \sum_{m=1}^{M} v_m = L, \quad v_m \in \{0,1\}, \ \forall m. \tag{3.10b}$$

In practice, however, it is challenging solve P0 since $H(\hat{\mathbf{b}}_i)$ and $H\left(\mathbf{b}_i^{\diamond} | \mathbf{v}\right)$ have no explicit forms. To this end, we propose to approximate $\Psi(\mathbf{v})$ in Eq. (3.10) using a surrogate function $\Psi'(\mathbf{v})$. In particular, no matter what values $H(\hat{\mathbf{b}}_i)$ and $H\left(\mathbf{b}_i^{\diamond} | \mathbf{v}\right)$ take, we always have $H(\hat{\mathbf{b}}_i) - H\left(\mathbf{b}_i^{\diamond} | \mathbf{v}\right) > 0$ if $\mathbf{b}_i^-$ happens and $H(\hat{\mathbf{b}}_i) - H\left(\mathbf{b}_i^{\diamond} | \mathbf{v}\right) = 0$ otherwise. By setting $H(\hat{\mathbf{b}}_i) - H\left(\mathbf{b}_i^{\diamond} | \mathbf{v}\right) = G > 0$ for all i, P0 is approximated as

$$(\mathbf{P1}) \quad \max_{\mathbf{v}} \underbrace{G \sum_{i} P\left(\mathbf{d}_i^- | \mathbf{v}\right)}_{:=\Psi'(\mathbf{v})}, \quad \text{s.t. (3.10b).} \tag{3.11}$$

It can be seen that P1 is equivalent to maximizing the expected number of pseudo labels, which can be estimated by Monte-Carlo sampling. Specifically, we deploy the pre-trained DNN $\mathbf{g}^{[0]}$ on CAVs and test them inside our region of interest. If any one

of CAV generates a bug object (including the inaccurate box, false negative, and false positive) within its FoV, the 3D position of this bug data $\mathbf{e}_j$ is registered into a database (i.e., bug database). After a sufficiently long simulation time, the database becomes $\mathscr{H} = \{\mathbf{e}_1, \mathbf{e}_2, \cdots\}$, which contains all registered error items. Therefore, maximizing the expected number of pseudo labels is equivalent to maximizing the cardinality $|\mathscr{Z}|$, where $\mathscr{Z} = \{\mathbf{e} \in \mathscr{H} | \exists m : v_m = 1, \|\mathbf{e} - \mathbf{r}_m\| \leq R\}$ and R in meter is the detection range of IRSUs.

To further derive $\mathscr{Z}$, consider a bipartite graph $(\mathscr{V}_1, \mathscr{V}_2, \mathscr{E})$, where $\mathscr{V}_1 = \{a_1, \cdots, a_M\}$ is the set of M positions, $\mathscr{V}_2 = \{b_1, \cdots, b_J\}$ is the set of J bug data, and $\mathscr{E}$ is the set of edges with $E_{m,j}$ representing the observability of bug data j from position m. Specifically, if $\|\mathbf{e}_j - \mathbf{r}_m\| \leq R$, we set $E_{m,j} = 1$; otherwise, we set $E_{m,j} = 0$. Weights $\{E_{m,j}\}$ are stacked into a matrix $\mathbf{E} = [E_{1,1}, \cdots, E_{1,j}; \cdots; E_{M,1}, \cdots, E_{M,J}] \in \{0,1\}^{M \times J}$. The nonzero elements in the vector $\mathbf{v}^T\mathbf{E}$ represents the expected number of false detections at CAVs that could be supervised by the IRSUs. That is, $\mathscr{Z} = \|\mathbf{v}^T\mathbf{E}\|_0$. Based on the above derivation, P1 is reformulated as

$$\max_{\mathbf{v}} \quad \|\mathbf{v}^T\mathbf{E}\|_0, \quad \text{s.t. (3.10b).} \tag{3.12}$$

The above problem is a nonconvex integer programming problem due to the nonconvexity of l_0 norm and discontinuity of $\mathbf{v}$. To this end, a slack variable $\mathbf{q} \in \{0,1\}^J$ with $\mathbf{q} \preceq \mathbf{v}^T\mathbf{E}$ is introduced, and the following surrogate problem of Eq. (3.12) is considered:

$$\max_{\mathbf{v},\mathbf{q}} \sum_{j=1}^{J} q_j, \text{ s.t. (3.10b)}, \ \mathbf{v}^T\mathbf{E} \succeq \mathbf{q}, \ q_j \in \{0,1\}, \ \forall j. \tag{3.13}$$

Denoting the optimal solutions to problem Eq. (3.12) and Eq. (3.13) as $\mathbf{v}^*$ and $(\mathbf{v}^\diamond, \mathbf{q}^\diamond)$, respectively, the following proposition is established.

Proposition 3.4.1. *The optimal* $\mathbf{v}^*$ *to problem Eq.* (3.12) *and the optimal* $\mathbf{v}^\diamond$ *to problem Eq.* (3.13) *satisfies* $\|\mathbf{v}^{\diamond T}\mathbf{E}\|_0 = \|\mathbf{v}^{* T}\mathbf{E}\|_0$.

Proof. Assume that $\|\mathbf{v}^{\diamond T}\mathbf{E}\|_0 \neq \|\mathbf{v}^{* T}\mathbf{E}\|_0$. Then we can always find a feasible solution $\mathbf{v}'$ to Eq. (3.12) such that $\|\mathbf{v}'^T\mathbf{E}\|_0 < \|\mathbf{v}'^T\mathbf{E}\|_0 \leq \|\mathbf{v}^{* T}\mathbf{E}\|_0$. Construct $(\mathbf{v}', \mathbf{q}')$ with $[\mathbf{q}']_j = 0$ if $[\mathbf{v}'^T\mathbf{E}]_j = 0$ and $q'_j = 1$ if $[\mathbf{v}'^T\mathbf{E}]_j \neq 0$, where $[\mathbf{a}]_j$ represents the j-th element of vector $\mathbf{a}$.

Algorithm 2 BSAP

Input: $\mathscr{F}$ - Feasible set of positions for IRSUs
$\mathscr{H}$ - Historical bug database
L - Maximum number of IRSUs to be deployed under limited financial budget

Output: Optimal sensor placement vector $\mathbf{v}^*$

Function:

Construct bipartite graph $(\mathscr{V}_1, \mathscr{V}_2, \mathscr{E})$ based on candidate positions $\mathscr{F}$ and bug data $\mathscr{H}$

Calculate $E_{m,j}$ based on distance $\|\mathbf{e}_j - \mathbf{r}_m\|$, detection range R, and occlusion O Form the observability matrix $\mathbf{E} \in \{0,1\}^{M \times J}$

Formulate the BASP problem as:

$$\begin{aligned} \max_{\mathbf{v}} \quad & \|\mathbf{v}^T \mathbf{E}\|_0 \\ \text{s.t.} \quad & \sum_{m=1}^{M} v_m = L, \quad v_m \in \{0,1\},\ m = 1, \cdots, M. \end{aligned}$$

Introduce slack variable $\mathbf{q} \in \{0,1\}^J$ with $\mathbf{q} \preceq \mathbf{v}^T \mathbf{E}$

Solve the surrogate problem (proved equivalent to primal problem in Proposition 1):

$$\begin{aligned} \max_{\mathbf{v},\mathbf{q}} \sum_{j=1}^{J} q_j, \quad \text{s.t.} \ & \sum_{m=1}^{M} v_m = L,\ \mathbf{v}^T \mathbf{E} \succeq \mathbf{q}, \\ & v_m \in \{0,1\},\ \forall m,\ q_j \in \{0,1\},\ \forall j. \end{aligned}$$

Output the solution of the surrogate problem $\mathbf{v}^\diamond$ and $\mathbf{q}^\diamond$

It can be shown that $(\mathbf{v}', \mathbf{q}')$ is a feasible solution to Eq. (3.13). Furthermore, $\sum_{j=1}^{J} q'_j = \|\mathbf{v}'^T \mathbf{E}\|_0 > \|\mathbf{v}'^T \mathbf{E}\|_0 = \sum_{j=1}^{J} q_j^\diamond$. This contradicts to the optimality of $(\mathbf{v}^\diamond, \mathbf{q}^\diamond)$. ■

Based on **Proposition 3.4.1**, we can solve Eq. (3.13) instead of Eq. (3.12). Moreover, problem Eq. (3.13) is an integer linear programming (ILP) problem, which can be optimally solved by off-the-shelf software packages, such as CVXPY (Agrawal et al., 2018). The entire procedure of BASP is summarized in Algorithm 2.

3.5 Experiment Analysis

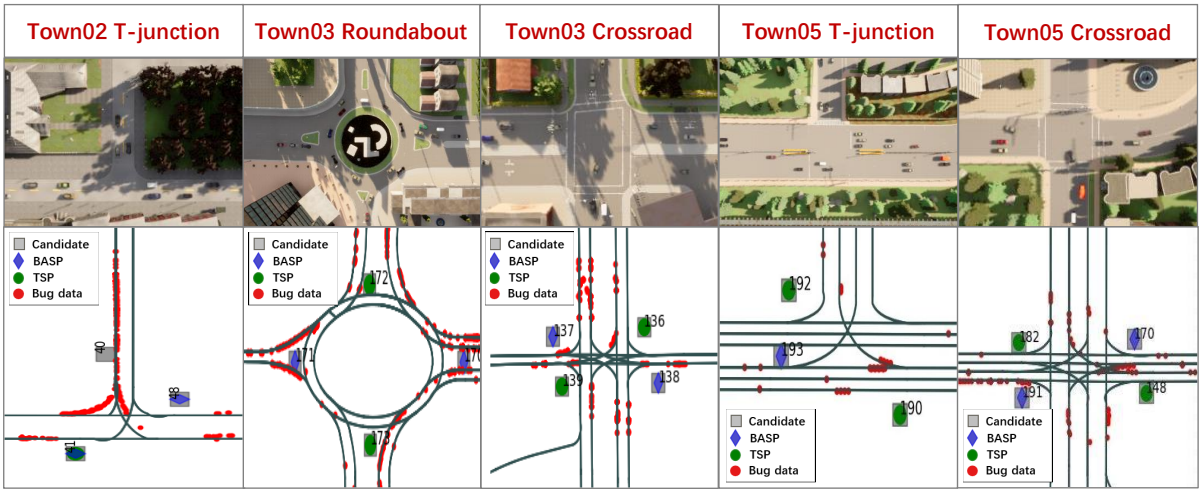


Figure 3.2: Comparison between BASP and TSP in 7 different Scenarios. Scenarios BEV Images, bug data distributions, Sensor placements of BASP, and sensor placements of TSP.

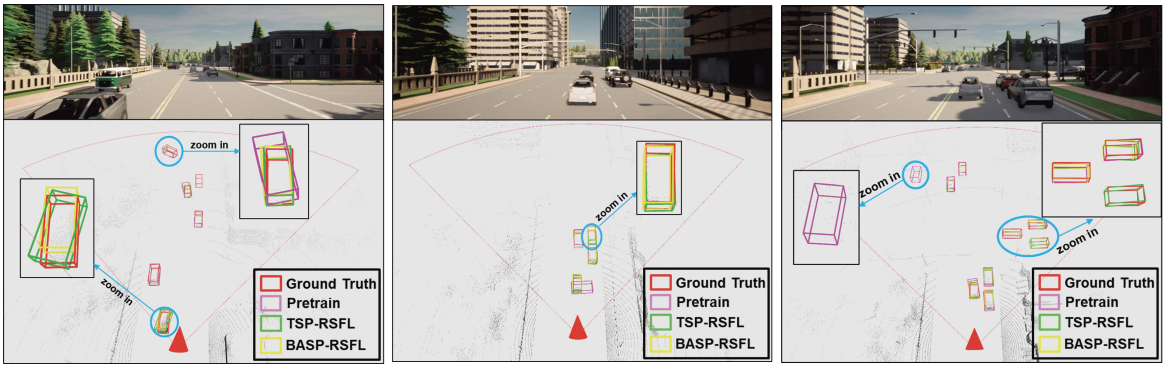
We implement the proposed RSFL system using Python in the high-fidelity CARLA simulator on a Ubuntu workstation with a 3.7 GHz AMD Ryzen 9 CPU and an NVIDIA 3090 Ti GPU. We simulate 5 scenarios (i.e., 2 T-junctions, 2 crossroads, and 1 roundabout), and their bird-eye-view pictures are shown in the first row of Figure 3.2. In the considered scenarios, we have $M = 18$ feasible positions and $L = 9$ IRSUs, where the candidate positions are marked as squares in the second row of Figure 3.2. To collect the RSFL dataset, we generate several CAVs in each scenario, and the CAVs are equipped with a 64-line LiDAR and a SECOND DNN model (Yan et al., 2018) for 3D object detection, to navigate in all these scenarios. Each CAV collects 7000 frames of point cloud data at a frequency of 10 Hz, where objects within the range of any IRSUs are labelled with road-detected bounding boxes and are unlabelled otherwise. Note that the pre-trained SECOND at all CAVs is obtained by training SECOND with 9000 frames in CARLA Town02, Town03, and Town05 maps for 50 epochs. All the DNN models are tested on a common dataset with 4000 samples collected in CARLA Town02, Town03, and Town05 maps.

We compare the following schemes: 1) **Pretrain**, which directly adopts the pre-

trained SECOND; 2) **RSFL with TSP** (Cai et al., 2023), which places IRSUs based on the complexity of road topology; 3) **RSFL with BASP**, which places the IRSUs by solving Eq. (3.13) based on historical bug data. First, as seen from the second row of Figure 3.2, in contrast to existing TSP that places more IRSUs (marked as circles) at roads with more lanes, the proposed RSFL automatically identifies the critical scenarios and places more IRSUs (marked as diamonds) in the areas with more bug data. As such, the number of false vehicle detections (marked as red circles) that can be calibrated by the accurate road detections under the BASP scheme in the third row of Figure 3.2 is larger than that by TSP in the fourth row of Figure 3.2. For instance, the BASP places more IRSUs at narrow T-junctions (i.e., the first scenario) instead of wide T-junctions (i.e., the forth scenario), which is in shapely contrast to TSP. This is because a narrow road, despite its simple topology, would lead to a high occlusion probability. As for the roundabout scenario (i.e., the second scenario), the positions generated by BASP and TSP are also different, where BASP places two LiDARs at west and east sites, but TSP places two LiDARs at north and south sites. Indeed, we assign a higher traffic density to the traffic flow from west to east, and the proposed BASP automatically recognizes such patterns. The above observations imply that the proposed BASP method serves as a better sensor placement strategy for FL than existing methods, producing automatic annotations with a higher probability.

Furthermore, the performance of SECOND models trained by different schemes is presented in Figure 3.3. The detection results of ground truth, pre-trained model, RSFL model and RSFL+BASP model are marked as red, pink, green and yellow, respectively. First, it can be seen from Figure 3.3 that the pre-trained scheme without FL (marked in pink) generates inaccurate boxes in Figure 3.3(a), false negatives in Figure 3.3b, and false positives in Figure 3.3(c). Next, by employing RSFL, the bounding boxes (marked in yellow) become closer to the ground truth in Figure 3.3(a). The RSFL also corrects the false negative in Figure 3.3(b), and recovers a false positive in Figure 3.3(c). Lastly, with the proposed RSFL with BASP, all the objects (marked in red) in all scenarios are successfully detected. This demonstrates the benefits brought by the bug-awareness

feature and corroborates the entropy reduction theory.



(a) Case 1. (b) Case 2. (c) Case 3.

Figure 3.3: Qualitative comparison of different schemes. Red, pink, yellow, and cyan boxes represent results obtained from ground truth, pre-trained, TSP-RSFL, and BASP-RSFL schemes, respectively.

Finally, we compare the mAPs of different schemes. Table 3.2 summarises the mAPs at IoU = 0.5 and IoU = 0.7 in the crossroad, T-junction, and roundabout scenarios of Town03 and Town05. The experimental results show that even with TSP-RSFL, the mAP is significantly increased compared to the pre-trained SECOND, i.e., with up to 5.74% and 4.00% improvements at IoU= 0.7 and IoU= 0.5, respectively. This is because the RSFL fine-tunes the pre-trained model using the OOD data and the road supervision at the adversarial scenarios, thus enhancing the model generalization capability. Furthermore, with BASP, the mAP performance is further boosted. The mAP of RSFL with BASP outperforms RSFL with TSP in almost all test scenarios and leads to an mAP improvement of more than 15% at IoU= 0.7 in the crossroad scenario of Town05. This is because, under BASP, the roadside sensors pose is determined not only by the road topology or traffic flows but also by the bug data distribution, which accounts for the learning requirements at AVs in complex urban scenarios.

3.6 Summary

In this chapter, we have presented RSFL, a cooperative model training framework for CAVs to learn from newly collected, unlabeled data in open traffic scenarios. To

		mAP	
		IoU=0.7	IoU=0.5
Crossroad (Town03)	pre-trained	48.87%	68.25%
	RSFL+TSP	51.92% (↑ 3.05%)	68.3% (↑ 0.11%)
	RSFL+BASP	51.09% (↑ 2.21%)	70.96% (↑ 2.71%)
T-junction (Town03)	pre-trained	49.75%	71.02%
	RSFL+TSP	52.61% (↑ 2.86%)	72.03% (↑ 1.02%)
	RSFL+BASP	52.94% (↑ 3.19%)	76.33% (↑ 5.31%)
Roundabout (Town03)	pre-trained	43.07%	71.74%
	RSFL+TSP	47.36% (↑ 4.29%)	73.89% (↑ 2.15%)
	RSFL+BASP	49.38% (↑ 6.31%)	75.69% (↑ 3.95%)
Crossroad (Town05)	pre-trained	46.65%	69.73%
	RSFL+TSP	47.12% (↑ 0.47%)	73.73% (↑ 4.00%)
	RSFL+BASP	63.09% (↑ 16.43%)	76.30% (↑ 6.57%)
T-junction (Town05)	pre-trained	43.87%	71.13%
	RSFL+TSP	49.61% (↑ 5.74%)	74.73% (↑ 3.60%)
	RSFL+BASP	51.94% (↑ 8.07%)	76.54% (↑ 5.41%)

Table 3.2: Comparison of mAPs for different schemes.

maximize the expected amount of road-supervised OOD data, we further proposed a BASP algorithm based on graph models and integer optimization. Various simulations and experiments have shown that the proposed RSFL fine-tunes the pre-trained model toward better generalization. Moreover, the BARL algorithm can enhance automatic road annotations for FLAD.

CHAPTER 4

COOPERATIVE MODEL TRAINING FOR PERCEPTION IN NETWORKED VEHICULAR SYSTEMS

4.1 Introduction

As addressed in chapter 3, we have explored to apply FL to achieve efficient road assisted cooperative training among CAVs and IRSUs. However, in the networked vehicular systems, the performance of practical cooperative training heavily depended on the characteristics of the wireless V2X channel conditions, as the shared information and the FLCAV training process are subject to the wireless variation including latency, bandwidth, and channel reliabilities. To this end, this chapter focuses on how to achieve efficient cooperative training among CAVs and IRSUs in considering the practical V2X communications and the dynamic network topology.

As mentioned in chapter 1, effective environmental perception is fundamental to AD systems, where individual AVs equipped with sensors like LiDAR, cameras, and radar can access a wealth of data. Recent advancements in DNN have greatly accelerated the AD perception tasks such as object detection (Lang et al., 2019) (S. Shi et al., 2019), tracking (Cheng et al., 2023), and segmentation (J. Li, Dai, et al., 2023), to a great extent. To keep

the perception model effectively adaptive to the dynamic and vast traffic environments, FL is adopted to upgrade the pretrained model and enhance models' adaptive capabilities. Existing works of FL could be categorized into two areas: (i) communication strategies and (ii) application strategies. Communication strategies focused on the design of the communication protocols by considering the communication-related variables, such as throughput (Lee et al., 2024), latency (Su et al., 2023), device scheduling (Z. Chen et al., 2023), and resource allocation (X. Zhou et al., 2023). However, these approaches did not adequately address FL in domain-specific CAV scenarios, datasets, and tasks. Application strategies for FLCAV concentrated on shared information fusion (Z. Zhang et al., 2021) and reduction techniques, such as quantization (Y. Lin et al., 2017), sparsification (M. Chen et al., 2020), to reduce communication loads and latency caused by reducing transmission cost of the large size of shared DNN models. However, these approaches assumed ideal communication conditions for cooperative training between CAVs and IRSUs, which was impractical for real-world FLCAV systems.

Motivated by these observations, in this chapter, we focus on exploring the integration of FL among CAVs and IRSUs with V2X communications. We first present a system-level design for a road-assisted FLCAV framework to achieve cooperative training among CAVs and IRSUs, and then we devise a communication topology optimization algorithm to reduce the communication delay among CAVs and IRSUs and improve the communication efficiency of the cooperative training process. The main contributions are summarized below:

- Proposed a tailored FLCAV framework to explore the interdependency between the specific CAVs perception domain of cooperative training process under wireless communication.
- Derived a communication topology optimization algorithm for FLCAV under V2X communication to reduce communication latency and accelerate the cooperative training process.

- Conducted high-fidelity experiments to verify the framework and the superiority of the proposed communication topology optimization algorithm for FLCAV.

4.2 System Model

Simulation systems have become one of the key components of the development and validation of AD scenarios. Since there is no simulator to simulate road-assisted cooperative learning and inference process in networked vehicular systems, we leverage CARLA (driven by the Unreal engine) (Dosovitskiy et al., 2017) to build our simulation and learning platform for the design and verification of the interdependency between specific CAVs perception domain of cooperative training under V2X wireless communications. The system framework is illustrated in Figure 4.1.

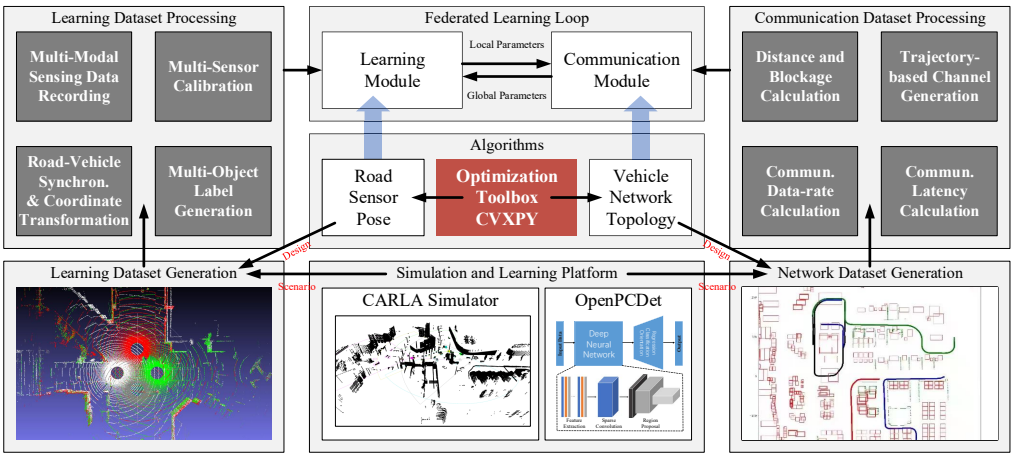


Figure 4.1: System framework of road-assisted topology optimization for FLCAV.

The system framework mainly contains three modules:

- **Dataset collection and processing**: this module is responsible for preparing high-fidelity annotated datasets from multi-modal sensors.
- **FLCAV loop**: this module is an FL-involved cooperative training process that facilitates decentralized model training across CAVs and IRSUs.

- **Communication process**: this module simulates the realistic V2X communication conditions essential for evaluating FLCAV under various channel dynamics.

4.2.1 Dataset Collection and Processing

To simulate the real-world cooperative training for the CAVs' perception task, a new dataset has been developed leveraging the CARLA simulator. Specifically, the dataset is generated under various traffic scenarios with a variety of CAVs and roadside infrastructures to cover several challenging driving environments. In the simulation, each CAV and IRSU is equipped with a camera, LiDAR, and GPS/IMU sensors. The camera captures RGB images with a resolution of 960 x 640 pixels and a horizontal FoV of 90 degrees. The LiDAR generates point clouds of surrounding objects and GPS/IMU is for self-localization. All sensor data is streamed at 20Hz. The visualization of a CAV's point clouds and RGB images in the CARLA simulator are illustrated in Figure 4.2. At the same tiem, the visualization of the roadside sensor's point clouds and BEV images for the roundabout traffic scenario is illustrated in Figure 4.3. Additionally, Multi-sensor calibration is achieved by leveraging sensors' intrinsic and extrinsic parameters to ensure spatial alignment across different sensors. Furthermore, frame stamping is adopted for the collected data samples as the reference to ensure the synchronization among CAVs and IRSUs.

Figure 4.2: Visualization of point clouds and front view of RGB image from CAVs perspective.

We use the SECOND (Yan et al., 2018) detector as the perception backbone to process the collected point cloud data. SECOND is a DNN-based object detection model to process 3D point cloud data, and it leverages the sparsely embedded convolutional layers to reduce the computational cost and accelerate the processing of point cloud data.

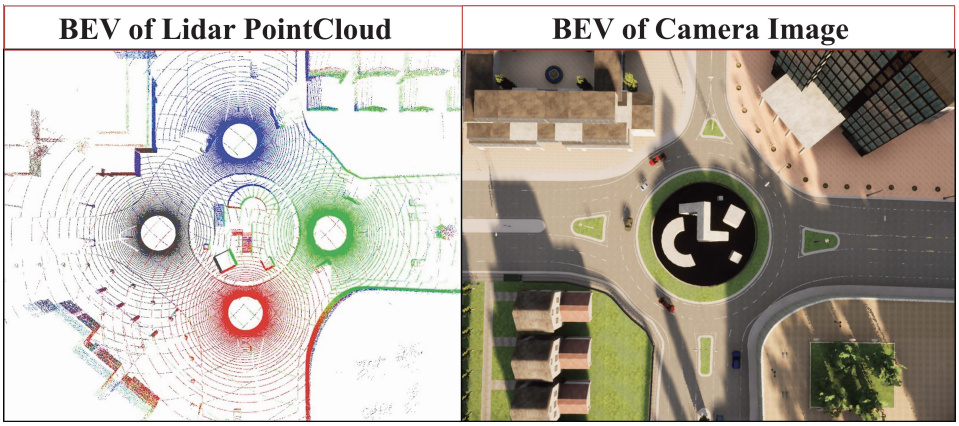


Figure 4.3: BEV of point clouds and RGB image from IRSUs perspective.

The SECOND detector consists of three components: (i) a voxel feature extractor (VFE), (ii) a sparse convolution layer, and (iii) a region proposal network (RPN). In SECOND, Raw point cloud data are first converted into voxel representations. Subsequently, a voxel-wise feature encoding layer extracts point-specific features within each voxel. The sparse convolution layer is then designed to learn 3D voxel features. The learned 3D voxel features are reshaped into 2D (image-like data) by applying convolutions only to non-zero inputs within the sparse voxel grid, enabling efficient processing for downstream tasks. In this way, we can significantly reduce computational cost. As for the RPN, it processes sparse 3D convolution features through two levels of downsampling and upsampling, concatenating the outputs into a feature map. The downsampling uses layers of 2D convolution, coupled with BN and ReLU activation, to extract and compress features. The upsampling employs a deconvolution layer to restore the data to its original size. In the final stage, a single-shot detector (SSD) outputs both object classifications and box localization. Unlike traditional methods, SSD generates detection results in a single feed-forward pass, significantly enhancing computational efficiency while maintaining high detection accuracy.

4.2.2 Federated Learning Assisted Cooperative Training Scheme

Several studies have been effectively modelled for wireless FL learning schemes. However, domain-specific FLCAV is different from the conventional FL that only focuses on some

simple tasks, like handwritten digit recognition, and image classification with small datasets. FLCAV has multiple CAVs with their own private datasets and the channel conditions vary as the dynamic features of moving CAVs. In this section, we present the problem statement of FLCAV in cooperative model training for 3D object detection tasks.

The tailored FL system for the perception of CAVs consists of K CAVs acting as clients and one IRSU acting as the edge server. The lidar data at the k-th CAV ($k \in \{1,\dots,K\}$) is denoted as ${}^{V}\mathscr{D}_k = \{{}^{V}\mathbf{d}_{k,1}, {}^{V}\mathbf{d}_{k,2}, \cdots\}$, where ${}^{V}\mathbf{d}_{k,t} \in \mathbb{R}^{D_k \times 3}$ is the vector concatenating the coordinates of all points at time step t, with D_k being the number of points in each cloud. The DNN parameter vector at the k-th CAV is ${}^{V}\mathbf{w}_k \in \mathbb{R}^{W_k \times 1}$ with W_k being the dimension of each DNN-based perception model. At the t-th lidar time frame, the k-th perception model maps ${}^{V}\mathbf{d}_{k,t}$ into a set of bounding boxes

$$ {}^{V}\mathscr{B}_{k,t} = \left\{ {}^{V}\mathbf{b}_{k,t}^{[1]}, {}^{V}\mathbf{b}_{k,t}^{[2]}, \cdots \right\}, \tag{4.1} $$

where ${}^{V}\mathbf{b}_{k,t}^{[n]}$ represents the n-th object at the t-th lidar frame in the k-th vehicle coordinate system, and is given by

$$ {}^{V}\mathbf{b}_{k,t}^{[n]} = [{}^{V}c_{k,t}^{[n]}, {}^{V}x_{k,t}^{[n]}, {}^{V}y_{k,t}^{[n]}, {}^{V}z_{k,t}^{[n]}, {}^{V}l_{k,t}^{[n]}, {}^{V}w_{k,t}^{[n]}, {}^{V}h_{k,t}^{[n]}, {}^{V}\theta_{k,t}^{[n]}]^T, \tag{4.2} $$

where ${}^{V}c_{k,t}^{[n]}$ is the category, $({}^{V}x_{k,t}^{[n]}, {}^{V}y_{k,t}^{[n]}, {}^{V}z_{k,t}^{[n]})$ is the center position, $({}^{V}l_{k,t}^{[n]}, {}^{V}w_{k,t}^{[n]}, {}^{V}h_{k,t}^{[n]})$ stands for the tuple of length, width, and height, and ${}^{V}\theta_{k,t}^{[n]}$ represents the yaw rotation, respectively. The inference process at the k-th CAV can be written as ${}^{V}\mathscr{B}_{k,t} = {}^{V}\Phi_k({}^{V}\mathbf{d}_{k,t}|\mathbf{w}_k)$, where ${}^{V}\Phi_k$ represents the DNN inference function.

The goal of the FLCAV scheme is to update the perception model $\mathbf{w}_k$ to make ${}^{V}\mathscr{B}_{k,t}$ as an accurate structured representation of ${}^{V}\mathbf{d}_{k,t}$. However, ${}^{V}\mathscr{B}_{k,t}$ cannot match ${}^{V}\mathbf{d}_{k,t}$, due to the uncertainty within the pre-trained perception model. Consequently, we need the FLCAV procedure to fine-tune $\mathbf{w}_k$ in a distributed manner, as it enables the pre-trained model to be fine-tuned collaboratively and dynamically. Specifically, the FLCAV aims to update the pre-trained model and acquire a global model with parameter vector $\mathbf{g}$, by solving the following loss minimization problem:

as it enables the detection models to be fine-tuned collaboratively and dynamically, ensuring improved adaptability and accuracy in diverse real-world environments.

$$\min_{\{\mathbf{w}_k\},\mathbf{g}} \underbrace{\frac{1}{\sum_k |{}^V\mathscr{D}_k|}\sum_k \sum_{({}^V\mathbf{d}_{k,t},{}^V\mathscr{B}_{k,t})\in {}^V\mathscr{D}_k} \Theta(\mathbf{g};{}^V\mathbf{d}_{k,t},{}^V\mathscr{B}_{k,t})}_{:=\Lambda(\mathbf{g})}$$
$$\text{s.t.} \quad \mathbf{w}_1 = \cdots = \mathbf{w}_K = \mathbf{g}, \tag{4.3}$$

where $\Theta(\mathbf{g};{}^V\mathbf{d}_{k,t},{}^V\mathscr{B}_{k,t})$ is the per-sample loss $({}^V\mathbf{d}_{k,t},{}^V\mathscr{B}_{k,t})$ $(1 \le t \le |{}^V\mathscr{D}_k|)$ in ${}^V\mathscr{D}_k$ with respect to parameter vector $\mathbf{g}$, and $\Lambda(\mathbf{g})$ denotes the global loss function to be minimized.

Let $\mathbf{g}^{[0]}$ be the pre-trained perception model at the cloud, and let $\mathbf{w}_k^{[i]}(0)$ be the local DNN parameters at kth CAV at the beginning of the i-th iteration ($i \ge 0$ and $\mathbf{w}_k^{[i]}(0) = \mathbf{g}^{[i]}$). The FL training of model parameters (i.e., solving Eq. (4.3)) is a distributed and iterative procedure, where each iteration involves two steps:

1) **Local update:** The k-th CAV minimizes the loss function via the gradient descent approach as

$$\mathbf{w}_k^{[i]}(\tau+1) = \mathbf{w}_k^{[i]}(\tau) - \frac{\varepsilon}{|{}^V\mathscr{D}_k|}\sum_{({}^V\mathbf{d}_{k,t},{}^V\mathscr{B}_{k,t})\in {}^V\mathscr{D}_k} \nabla\Theta(\mathbf{w}_k^{[i]}(\tau);{}^V\mathbf{d}_{k,t},{}^V\mathscr{B}_{k,t}), \tag{4.4}$$

where ε is the step-size, $0 \le \tau \le E-1$ (E is the number of local updates) and $\nabla\Theta$ denotes the gradient of Θ.

2) **Global averaging:** All CAVs upload $\{\mathbf{w}_k^{[i]}(E)|\forall k\}$ to the IRSU, which computes an average model

$$\mathbf{g}^{[i+1]} = \frac{1}{K}\sum_{k=1}^{K} \mathbf{w}_k^{[i]}(E), \tag{4.5}$$

which is broadcast to the CAVs for next-round updates.

This completes one FLCAV round and we set $i \leftarrow i+1$. The entire procedure stops when $i = I_{\mathrm{FL}}$, with I_{FL} being the number of federated learning rounds.

4.3 Dynamic Communication Topology Optimization

Most existing works of FL in wireless environments assume that end devices, such as cellphones, directly upload their local parameters or gradients to an edge server or via fixed relays, such as base station, due to the limited on-board resources of the end devices. This approach typically follows a fixed topology FL (FTFL) scheme, where the communication topology and the central aggregation process remain static throughout the learning process. However, in FLCAV with wireless communication, when the wireless channel between a particular CAV and the IRSU is poor, it can significantly delay the aggregation process, as the IRSU server must wait to receive the local DNN parameters from all participating CAVs before proceeding with the global aggregation. Therefore, the communication round latency is governed by the CAV with the longest transmission and local computation time. Moreover, CAVs generally have more powerful on-board computation and communication units, enabling greater flexibility. In our wireless networked vehicular system, CAVs and IRSUs can communicate with each other via V2V or Vehicle-to-Infrastructure (V2I) communication. This setup facilitates the adoption of a dynamic topology FL (DTFL) scheme. This allows a CAV to send its local parameters to nearby CAVs or the IRSU instead of only communicating with the IRSU server, which means CAVs can act as a relay in the wireless network. After receiving other CAVs parameters, the relay CAV aggregates it local model parameters with the received parameters and subsequently transmit the aggregated results to other CAVs or IRSUs for further processing, thereby mitigating the straggler problem.

4.3.1 Model Aggregation and Communication Model

The communication between the IRSU and CAVs in a traffic scenario is shown in Figure 4.4, where CAVs communicate with each other through V2V communication,

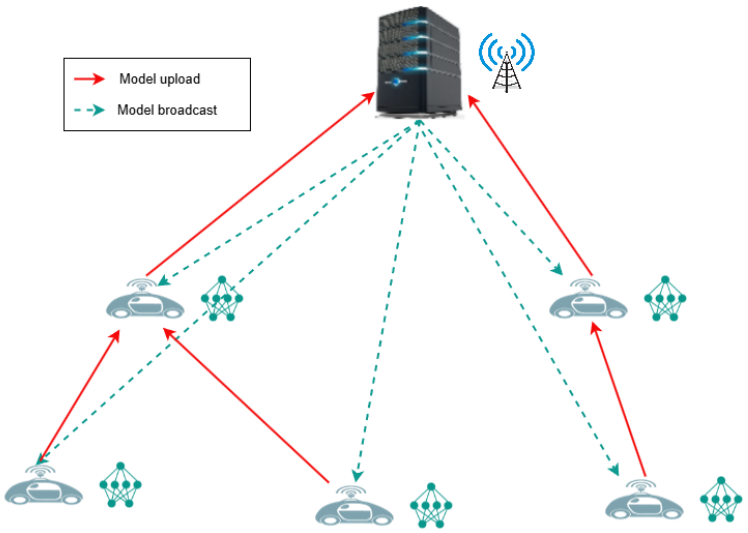


Figure 4.4: An illustration of the communication topology between the IRSU and CAVs for FLCAV in the networked vehicular system

while the communication between the IRSU and a CAV is through V2I communication.

During the training process, CAVs that received parameters from other nearby CAVs are referred to as relay CAVs. The set of all training participants (i.e., CAVs and the IRSU) are named as nodes and the set of all nodes is denoted as $C^0 = C \cup R$ where R represents the IRSU. Then, the communication topology matrix I is defined as $I = \{I_{i,j} | i \in C, j \in C^0\}$, which establishes a unique network topology among CAVs and IRSUs for the perception model parameter transmission and aggregation. $I_{i,j} = 1$ only when node i connects to the node j; otherwise $I_{i,j} = 0$.

If a CAV acts as a relay, the aggregated parameters at this CAV, resulting from combining local and received parameters, are given by

$$\tilde{\mathbf{w}}_\mathbf{j} = \frac{\sum_{i \in K} I_{i,j} \tilde{D}_i \tilde{\mathbf{w}}_\mathbf{i} + D_j \mathbf{w}_\mathbf{j}}{\sum_{i \in C} I_{i,j} \tilde{D}_i + D_j}, \tag{4.6}$$

where $\tilde{D}_i$ corresponds to the total number of data samples residing at i. Since each CAV can transmit its local models to only one other participants (i.e., other CAVs or the IRSU), and at least one CAV with a direct connection to the IRSU, the constraints on the communication matrix I are

$$I_{i,i} = 0, \sum_{j \in C^0} I_{i,j} = 1, \forall i \in C \tag{4.7}$$

$$\sum_{i \in C} I_{i,R} \geq 1, \tag{4.8}$$

As such, the communication topology will be formulated into a tree topology without a ring, with the IRSU as the root node. CAVs serve as both relay and source nodes, forwarding the aggregated parameters $\tilde{\mathbf{w}}_\mathbf{j}$ to their parent node without increasing the transmitted data size.

To evaluate the latency during the FL process, we consider the local computation and wireless communication time, accounting for the heterogeneous computational power of CAVs and channel conditions. Let N_{FLOP} denote the number of floating point operations per training data sample and f_c denote the computation speed of the c-th CAV, the local computation time for the c-th CAV is

$$t_c^{cmp} = \frac{D_c N_{FLOP}}{f_c}. \tag{4.9}$$

In our setting, CAVs transmit model parameters via wireless links. Thus, CAV i can transmit its parameters to the IRSU or other j-th CAV ($j \in C^0, j \neq i$) via V2V or V2I communication. Then the delay between i and j is

$$t_c^{comm} = \frac{B}{\sum_{j \in C^0} I_{i,j} r_{i,j}}, \tag{4.10}$$

where B represents bit count required to transmit the perception model parameters. $r_{i,j}$ is the rate of data transmission from i-th node to j-th node. The transmission rate $r_{i,j}$ depends on the bandwidth w_i for node i, the distance $d_{i,j}$ between the node i and j, transmit power P, and channel noise δ^2, as shown below:

$$r_{i,j} = w_i log_2(1 + \frac{P H_{i,j}}{\delta^2}). \tag{4.11}$$

$H_{i,j} = g_0(d_0/d_{i,j})^{\alpha}|h_{i,j}|$ is the channel gain from the i-th node to the j-th node, with g_0 as the path loss constant, d_0 as the reference distance, α as the path loss coefficient, and $h_{i,j} \sim \mathcal{CN}(0,1)$ representing small-scale fading.

4.3.2 Topology Optimization Algorithm

To accelerate the cooperative training process among CAVs and the IRSU in wireless communications, we are to minimize the latency for each round by optimizing the communication topology matrix I, considering CAVs' heterogeneous computational power and channel conditions. The j-th node aggregates and forwards perception model parameters from nearby nodes only after receiving all parameters from node i with $I_{i,j} = 1$. If the time required for local computation at the j-th node is longer the combined local computation and transmission times of i-th node, j-th node must wait. Thus the communication delay constraints for the connected node pair are

$$\frac{D_i N_{FLOP}}{f_i} + \frac{B}{\sum_{j \in C^0} I_{i,j} r_{i,j}} I_{i,j} < \frac{D_j N_{FLOP}}{f_j}. \tag{4.12}$$

With this communication topology, the cooperative training latency is limited by the slowest IRSU-connected CAV. It is given by

$$t^{round} = \min_{i \in C_R} \left\{ \frac{D_i N_{FLOP}}{f_i} + \frac{B}{\sum_{j \in C^0} I_{i,j} r_{i,j}} \right\}, \tag{4.13}$$

$$= \min_{i \in C} \left\{ \frac{D_i N_{FLOP}}{f_i} + \frac{B}{\sum_{j \in C^0} I_{i,j} r_{i,j}} \right\}, \tag{4.14}$$

where $C_R = \left\{ i | I_{i,K+1} = 1, i \in K \right\}$ denotes the set of CAVs that connect to the IRSU. Finally, the optimization problem that minimizes the communication delay of each round in FLCAV can be formulated as follows

$$
\begin{aligned}
&\min_{i \in C} \left\{ \frac{D_i N_{FLOP}}{f_i} + \frac{B}{\sum_{j \in C^0} I_{i,j} r_{i,j}} \right\} \\
&\text{s.t.} (\frac{D_i N_{FLOP}}{f_i}) + \frac{B}{\sum_{j \in C^0} I_{i,j} r_{i,j}} I_{i,j} < \frac{D_j N_{FLOP}}{f_j} \\
&I_{i,i} = 0, \sum_{j \in C^0} I_{i,j} = 1, \forall i \in C \\
&\sum_{i \in C} I_{i,R} \geq 1 \\
&I_{i,j} \in \{0,1\}, i \in C, j \in C^0.
\end{aligned} \tag{4.15}
$$

The above problem is a mixed-integer nonlinear program (MINLP) problem, which is genrally NP-hard and non-convex due to the binary communication matrix $I_{i,j} \in \{0,1\}$ and the non-linear reciprocal term in the objective. To make the problem solvable by convex optimization frameworks, we relax the binary constraints $I_{i,j} \in \{0,1\}$ to continuous constraints $I_{i,j} \in [0,1]$, efficiently treating $I_{i,j}$ as a soft weight. The reciprocal term is also approximated by exploiting a convex surrogate, such as the negative logarithm function, which preserves monotonicity and penalizes small denominators while maintaining convexity. Additionally, to promote sparsity and encourage binary-like solution, we introduce addition L_1 regularization term in the objective function. The resulting relaxed problem is a continuous, convex optimization problem that can be solved using off-the-shelf software packages such as CVXPY (Agrawal et al., 2018).

4.4 Simulation and Numerical results

To explore the interdependency between the specific FLCAV perception domain under V2X wireless communication and verify the effectiveness of the communication topology optimization algorithm, we conduct a series of experiments by leveraging the CARLA simulator and generate data from three complex traffic scenarios, i.e., a crossroad, a T-junction and a roundabout. The performance evaluation is conducted using an independent test dataset for each scenario. Specifically, each scenario included 5 CAVs and an IRSU to collect the sensing dataset. Each CAV is equipped with a camera-LiDAR

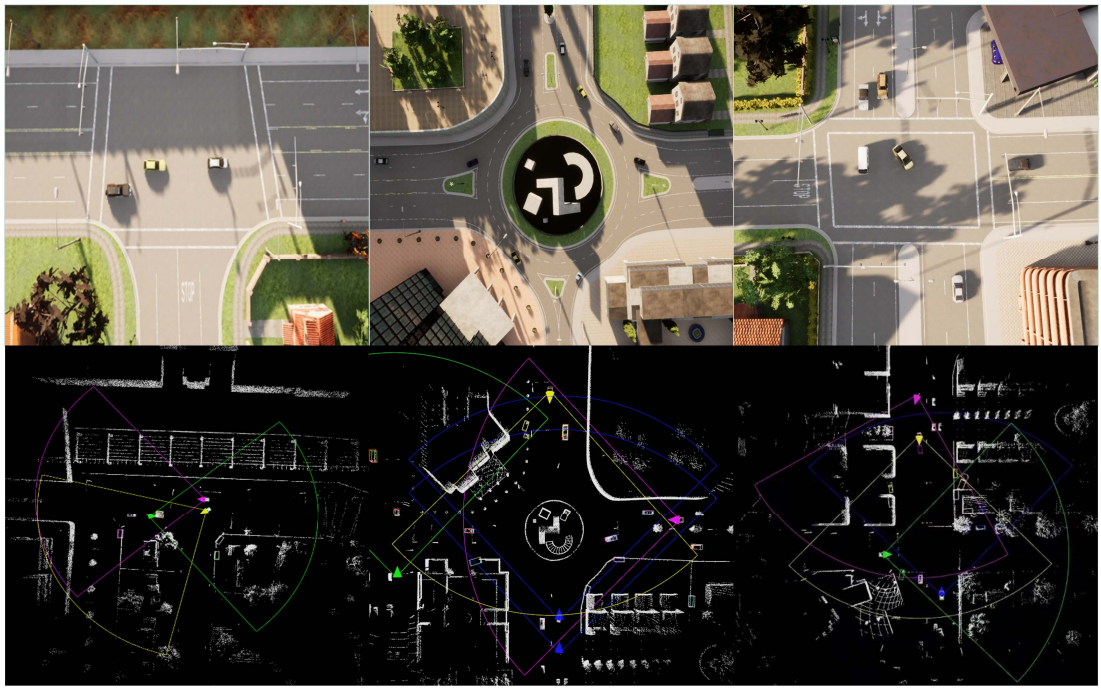


Figure 4.5: Birds-Eye-Views of the three traffic scenarios (i.e., T-junction, roundabout and crossroad).

pair to gather RGB images and point cloud data. The LiDAR is a 64-line model operating at 20Hz, with a default range and FoV set to 100 meters and 90 degrees, respectively. Additionally, each CAV is fitted with a GPS device for self-localization. For the IRSU, in each scenario, sensor configurations are experimentally determined and optimized through a road sensor optimization algorithm to ensure comprehensive coverage of the traffic scenarios and help CAVs for the new data annotation. Figure 4.5 illustrates the roadside birds-eye-views (BEV) of the three traffic scenarios (a T-junction, a roundabout and a crossroad).

In the DTFL scheme, the network topology optimization algorithm optimize the communication topology for the learning process. Each CAV's pose information (location and rotation) is extracted for every frame, and factors such as DNN model size and communication delay are considered during FL. In our experiment, we exploit the SECOND (Yan et al., 2018) as our perception backbone for 3D object detection, which comprises approximately 5 million parameters (model size 64 MB; sample size 1.6 MB). The local computing speed varies between 1 and 10 GFLOPs, with the transmit power

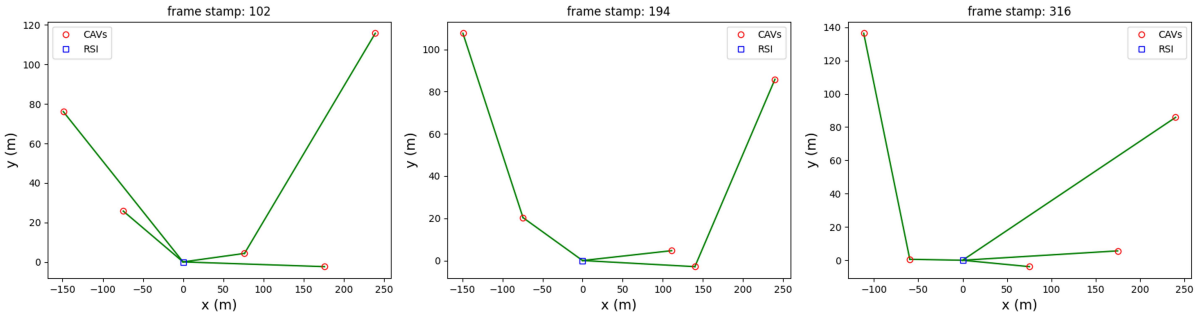


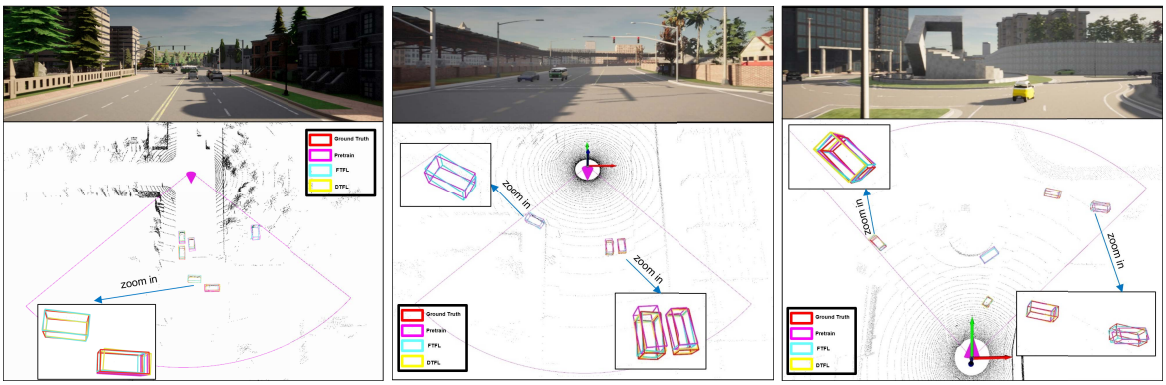
Figure 4.6: An illustration of the adaptive time-varying communication topology between the IRSU and CAVs for FLCAV in the networked vehicular system

$P = 1W$ and a bandwidth of 5 MHz per CAV.

Figure 4.6 visualizes selected communication topologies after optimization. Unlike the fixed topology in conventional FL, where all CAVs send their parameters directly to the edge server, our approach adaptively adjusts communication routes based on the relative positions of road participants and wireless communication conditions. This flexibility allows the FLCAV scheme to better account for time-varying transmission latencies and local CAV resource availability, enhancing the efficiency of the networked vehicular system.

Furthermore, we compare the following schemes: 1) **Pretrain**, which directly adopts the pre-trained SECOND; 2) **FTFL without topology optimization**, where all CAVs directly transmit parameters to the IRSU; 3) **DTFL with topology optimization**, in which the topology optimization algorithm is applied for DTFL during the learning process. The qualitative comparison of the perception model trained by different schemes is presented in Figure 4.7. The detection results of ground truth, pre-trained model, FTFL model, and DTFL model are marked as red, pink, cyan, and yellow, respectively. We can see that with the assistant knowledge of IRSUs after the FL schemes, CAVs can generate more accurate predictions. Moreover, the pretrained baseline method without FL (marked in pink) generates false negatives in Figure 4.7(a), false positives in Figure 4.7(b) and inaccurate boxes in Figure 4.7(c). After employing the FTFL method, the bounding boxes (marked in cyan) become closer to the ground truth in Figure 4.7(b). However, DTFL outperforms FTFL in Figure 4.7(b), as the FTFL method fails to correct the false positive.

Furthermore, the proposed DTFL with the topology optimization algorithm effectively detects all the objects (marked in red) in all scenarios, showing superiority than other benchmarks. This improvement is attributed to the reduced communication latency achieved through the dynamic network optimization algorithm. As a result, within a fixed FL training duration, more training rounds can be executed, thereby enhancing learning accuracy.



(a) Case I. (b) Case II. (c) Case III.

Figure 4.7: Qualitative comparison of different cooperative training schemes. Red, pink, cyan, and yellow boxes represent results obtained from ground truth, pre-trained, FTFL, and DTFL schemes, respectively.

Finally, we compare the mAPs of different methods. Table 4.1 summarises the mAPs at IoU = 0.5 and IoU = 0.7 in cross-road, T-juction, and roundabout scenarios. The experimental results show that, the mAP of FTFL is significantly increased compared to the pre-trained SECOND, i.e., with up to 6.13% and 4.05% improvements at IoU= 0.7 and IoU= 0.5, respectively. This is because the FL scheme fine-tunes the pre-trained model and enhances the model generalization capability. Furthermore, the mAP performance is further boosted by the DTFL scheme with the topology optimization algorithm. The mAP of DTFL with topology optimization outperforms FTFL with fixed topology in almost all test scenarios. It leads to an mAP improvement of more than 8% at IoU= 0.5 in the test scenarios.

		mAP	
		IoU=0.7	IoU=0.5
Crossroad	pretrained	48.87%	68.25%
	FTFL	53.09% (↑ 4.21%)	72.30% (↑ 4.05%)
	DTFL	57.12% (↑ 8.25%)	74.73% (↑ 6.48%)
T-junction	pretrained	49.75%	71.02%
	FTFL	55.94% (↑ 6.13%)	72.33% (↑ 1.31%)
	DTFL	56.61% (↑ 6.86%)	74.03% (↑ 3.02%)
Roundabout	pretrained	43.07%	71.74%
	FTFL	48.38% (↑ 5.31%)	73.69% (↑ 1.91%)
	DTFL	49.36% (↑ 6.29%)	75.22% (↑ 3.48%)

Table 4.1: Comparison of mAP for different cooperative learning schemes

4.5 Summary

In this chapter, we present the exploration of the interdependency between the specific FLCAV perception domain under V2X wireless communication among CAVs and IRSUs. Additionally, a communication-efficient topology optimization algorithm was proposed to consider the geometric relationship between CAVs and IRSUs, enhancing cooperative training efficiency. Numerical results have demonstrated that our proposed framework significantly outperforms the benchmark models in terms of improving detection precision, reducing latency and accelerating the training process. Qualitative analysis with visual examples further validated its effectiveness.

CHAPTER 5

COOPERATIVE MODEL INFERENCE FOR PERCEPTION IN NETWORKED VEHICULAR SYSTEMS

5.1 Introduction

In Chapter 3 and Chapter 4, we have achieved effective cooperative training for perception model upgrade. Different from cooperative training which aims to upgrade and enhance the perception models' adaptive capabilities, cooperative inference aims to reduce the perception uncertainty by sharing perceived information from multiple CAVs in the inference stage via V2V communications. However, when inevitable channel impairments occur in V2V communications, the shared information can compromise the detection performance due to the distortions. To this end, this chapter focuses on improving the robustness of the cooperative inference in the presence of V2V communication impairments.

Emerging technologies, such as LiDAR and artificial intelligence algorithms, have facilitated the development of 3D object detection. However, due to these sensors' physical limitations, including restricted detection range and limited FoV, it may prevent an individual AV from accurately detecting occluded or distant objects. To address this,

cooperative inference enabled by vehicular communications has been widely proposed, which uses shared information from multiple CAVs to obtain wider viewpoints and enhance detection in the inference stage (Y.-C. Liu, Tian, Ma, et al., 2020; T.-H. Wang et al., 2020; Xu, Tu, et al., 2023; Xu, Xiang, Tu, et al., 2022; Xu, Xiang, Xia, et al., 2022). These collaborative frameworks, nevertheless, heavily relies on the reliability of V2V communications, as the non-stationarity time-varying characteristics of V2V channels, caused by moving scatters (i.e., other vehicles and pedestrians), can impact the real-time information sharing among CAVs. Therefore, incorporating vehicular communications in cooperative inference is vital for ensuring system robustness, and this has attracted significant research interests.

Based on the advancements of effective 3D detection backbones(Lang et al., 2019; Yan et al., 2018), several studies have explored various collaborative fusion schemes for cooperative inference, including raw-data-level fusion (sharing raw point clouds) (Arnold et al., 2020; Q. Chen, Tang, et al., 2019; Gao et al., 2018), intermediate-level fusion (sharing intermediate feature representations) (Q. Chen, Ma, et al., 2019; Hu et al., 2022; Y. Li et al., 2021; Y.-C. Liu, Tian, Glaser, & Kira, 2020; Xu, Tu, et al., 2023; Xu, Xiang, Xia, et al., 2022), and object-level fusion (sharing detection outcome) (Ambrosin et al., 2019; D. Chen & Krähenbühl, 2022; Glaser & Kira, 2023; S. Shi et al., 2022; Z. Zhang et al., 2021). These works aim to optimize the trade-offs between bandwidth utilization and detection accuracy, often assuming ideal communication. In order to consider the effects of realistic V2V communications, recent works have shifted toward more realistic scenarios, considering various factors, such as communication delays (Xu, Xiang, Tu, et al., 2022), information loss (J. Li, Xu, et al., 2023), and pose inaccuracies (Y.-C. Liu, Tian, Ma, et al., 2020; T.-H. Wang et al., 2020). However, a more sophisticated channel model that accounts for the non-stationarity and time-varying of V2V channels, particularly the time-varying distortions caused by vehicle speed and moving scatterers, has yet to be investigated. Furthermore, some previous works have utilized both supervised end-to-end training (C. Liu et al., 2023) and self-supervised learning (C. Liu et al., 2024) to mitigate the adverse effects of channel impairments. Nevertheless, challenges

remain. Specifically, supervised distortion-in-the-loop training allows fusion networks, such as attentive fusion and V2VNet, to learn from the distorted shared information, thereby improving detection performance under channel impairments. However, it lacks generalization to varying communication environments, as it struggles to function with varying noise levels, path loss factors, and other dynamic channel conditions. On the other hand, self-supervised learning offers a flexible approach to filtering out distorted information due to severe channel impairments. However, the existing methods operate only at the CAV level and cannot address pixel-level feature maps. Therefore, developing an approach that can flexibly adapt to varying V2V communication channels is essential to enhance the robustness of cooperative inference.

Motivated by these existing challenges, this chapter focuses on improving the robustness of the cooperative inference in the presence of V2V communication impairments. The detailed contributions can be summarized by:

- Develop a new method to incorporate practical communication channels in cooperative inference systems to evaluate the performance degradation caused by communication channel impairments for different fusion schemes.

- To alleviate the performance degradation due to V2V communication impairments, we propose a joint weighting and denoising framework, *Coop-WD*, to improve the robustness of cooperative inference using intermediate feature sharing, where A CAV-level weighting algorithm based on self-supervised learning, combining with a denoising algorithm based on a diffusion probabilistic model, is developed to tackle the CAV-level and pixel-level feature distortion. Besides, generative and self-supervised learning enhances the model's flexibility and adaptivity to randomness and stochastic variations of V2V channels.

- Unlike the previous works that neglected uncertainty and stochastic variability of V2V communication, a non-stationary V2V communication channel model is used to consider more V2V factors, including velocity, acceleration, inter-vehicle distance, geographic positioning, and rotational orientation. In addition to different

noise levels and path loss factors using Rician fading, multi-path, time-varying distortions is also considered to validate the robustness and generalization of our approach.

- Numerical results demonstrate that the proposed *Coop-WD* outperforms conventional benchmarks under all channel conditions across different distortion levels, including simulated Rician fading, realistic WINNER II, and non-stationary V2V channels. This effectiveness is also tested and validated under various communication factors, including path loss, imperfect CSI, and time-related distortions. Qualitative analysis is conducted to further prove that the proposed approach performs better than the benchmark with only a weighting or denoising module.

5.2 System Model

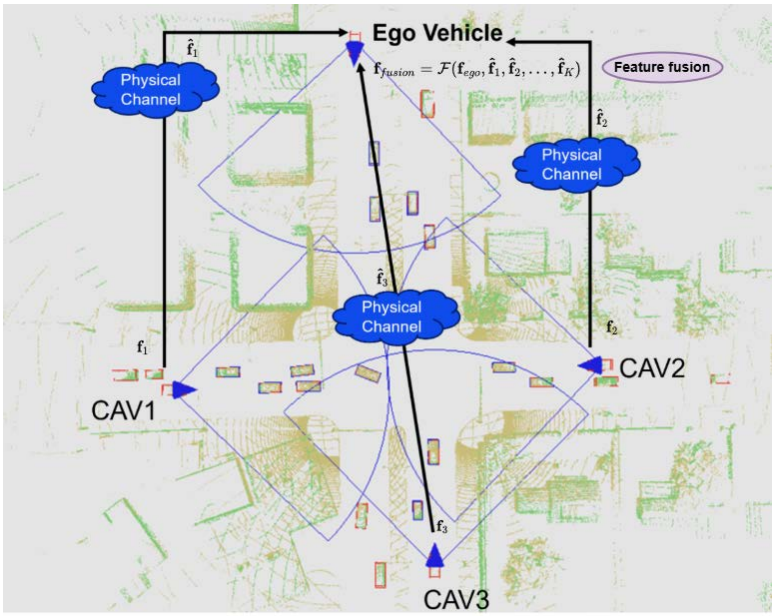


Figure 5.1: Cooperative inference via V2V communication.

5.2.1 V2V Communication Models

The system model for cooperative inference in the V2V networked vehicular system is illustrated in Figure 5.1. In this model, the ego vehicle, connecting with other CAVs, collaborates with other CAVs by sharing information through V2V communication channels. The fusion process of information shared from multiple CAVs at ego vehicle can be expressed as

$$\mathbf{f}_{fusion} = \mathscr{F}(\mathbf{f}_{ego}, \hat{\mathbf{f}}_1, \hat{\mathbf{f}}_2, \ldots, \hat{\mathbf{f}}_K), \tag{5.1}$$

where $\mathbf{f}_{ego}$ denotes the data sensed by the ego vehicle, while $\hat{\mathbf{f}}_k$ represents the shared data (i.e., raw point clouds, intermediate feature maps or model outputs) from the k-th CAV. $\mathscr{F}(\cdot)$ denotes the fusion function employed to combine the shared data transmitted by other CAVs.

The information provided by the CAVs offers diverse viewpoints that enhance both the precision and range of perception. However, the shared information relies on V2V communication to reach the ego vehicle. Due to the inherent uncertainty and stochastic variability in V2V channels, factors such as channel fading, noise levels, path loss, and time-varying distortions can distort the transmitted information. This, in turn, compromises the accuracy of the fused information used in cooperative inference. To investigate how these channel impairments can affect the quality of shared information, three communication models are considered: (i) the Rician fading channel incorporating free-space path loss; (ii) the WINNER II channel model (Bultitude & Rautiainen, 2007), which incorporates multi-path fading; and (iii) a practical time-varying non-stationary V2V channel. Furthermore, to simulate the practical channels, we account for imperfect channel state information (CSI) affected by Gaussian noise and use a pilot-based least-squares method for channel estimation, which introduces CSI estimation errors. The signal $\hat{\mathbf{f}}_k$ is recovered from the received signals using a zero-forcing detector.

5.2.1.1 Rician Fading

In this channel, a single-input single-output (SISO) configuration with Rician fading and free-space path loss is considered. The information sent to the ego vehicle from the k-th CAV via the channel is denoted as

$$\mathbf{y}_k = \sqrt{\frac{p_0}{d_k^n}} h_k \mathbf{x}_k + \mathbf{w}_k, \tag{5.2}$$

where the $\frac{p_0}{d_k^n}$ denotes the path loss with p_0 representing the power loss at 1 meter. d_k denotes the spatial range between mobile transmitter (Tx) and the mobile receiver (Rx), with n denoting the path loss exponent. The variable h_k characterizes the Rician fading channel, modeled as a complex normal distribution $\mathscr{CN}(\mu, \sigma_h^2)$. The vector $\mathbf{y}_k \in \mathbb{C}^{L\times 1}$ represents the received signal from the k-th CAV, with $\mathbf{x}_k \in \mathbb{C}^{L\times 1}$ indicating the transmitted signal. The noise term $\mathbf{w}_k$ corresponds to the additive white Gaussian noise (AWGN), following a distribution $\mathscr{CN}(0, \sigma^2)$.

5.2.1.2 Multi-Path

In practical CAVs' cooperative inference of V2V communication scenarios, the channel may also be subject to multi-path fading. Beyond the communication model present in (Eq. 5.2), a multi-path channel model employing orthogonal frequency-division multiplexing (OFDM) is also taken into account. Under this model, the symbols received at the i-th sub-carrier from the k-th CAV are given by

$$\boldsymbol{Y}_k[i] = \boldsymbol{H}_k[i]\boldsymbol{X}_k[i] + \boldsymbol{W}_k[i], \tag{5.3}$$

where $\boldsymbol{H}_k[i]$ represents the channel frequency response, $\boldsymbol{W}_k[i]$ corresponds to the AWGN, and $\boldsymbol{X}_k[i]$ denotes the transmitted symbol.

5.2.1.3 Non-stationarity

In the practical CAVs' cooperative inference with V2V communications, it is imperative to acknowledge that CAVs exhibit non-stationary behavior and possess time-variant

statistical properties under complex traffic conditions. Specifically, dynamic factors such as CAVs' velocity, acceleration, inter-vehicle spacing, geolocation, and angular orientation are subject to continuous change. These dynamic characteristics of CAVs exert a significant influence on the V2V communication channels, attributable to the mobile nature of the Tx and Rx integrated into the vehicles. Consequently, this results in the channels exhibiting rapidly fluctuating characteristics, which must be meticulously accounted for in the modeling and analysis of V2V networked communication systems.

To demonstrate this characteristic of V2V communication, a practical time-varying non-stationary V2V communication channel model (W. Li et al., 2020) is tailored for our work to better reflect real-world V2V communication scenarios by accounting for the mobility characteristics of CAVs, which introduce time-varying channel parameters, leading to multipath effects and Doppler effects.

Firstly, CAVs attached with Tx and Rx are moving within a traffic scene in arbitrary trajectories with varying velocities as $\mathbf{v}^o(t)$, $o \in \{\text{Tx, Rx}\}$ representing Tx or Rx in brief. In cooperative inference with the V2V communication system, the location of the CAVs' Tx and Rx antenna elements are denoted as $\mathbf{d}^o = \left[d_x^o, d_y^o\right]^T$, $o \in \{\text{Tx,Rx}\}$ where the coordinates d_x^o and ,d_y^o are the position of the antenna along the x axis and y axis in each local coordinate system, respectively. In the V2V communication model, multiple-bounced propagation paths contain Line-of-Sight (LoS) and Non-Line-of-Sight (NLoS) by considering if there are clusters in a certain path. In a twin-cluster model under NLoS scenario, the clusters can be grouped into three separate segments: i.e., (i) the first cluster S_n^{Tx}, (ii) the last cluster S_n^{Rx}, and (iii) the rest between S_n^{Tx} and S_n^{Rx}. The S_n^{Tx} and S_n^{Rx} can be modelled by velocities and locations of CAVs, and the remaining segment can be represented through a virtual link (Q. Zhu, Li, et al., 2018). Under general V2V traffic scenarios, the V2V channel between the Tx and Rx can be expressed as a $P \times Q$ complex matrix $\mathbf{H}$, where P and Q represent the number of antennas at the Tx and R, respectively. We have

$$\mathbf{H}(t,\tau) = \left[h_{p,q}(t,\tau)\right]_{P\times Q}, \tag{5.4}$$

where $h_{p,q}(t,\tau)$ represents the channel impulse response (CIR) between the Tx p-th antenna and Rx q-th antenna at time instant t. According to (Q. Zhu, Yang, et al., 2018), it can be further express as

$$h_{p,q}(t,\tau) = \sum_{n=1}^{N} P_n(t)\tilde{h}_{p,q,n}(t)\delta(\tau - \tau_n(t)), \tag{5.5}$$

where N indicates the total number of propagation paths, $P_n(t)$ refers the normalized power of n-th path, $\tau_n(t)$ corresponds to the time delay of n-th path, and $\tilde{h}_{p,q,n}(t)$ denotes the channel gain and it can be express as

$$\tilde{h}_{p,q,n}(t,\tau) = \frac{1}{\sqrt{M}} \sum_{m=1}^{M} e^{j\cdot(2\pi\cdot\int_0^t f_{n,m}(t')dt' + \Phi_{n,m}(t) + \theta_{n,m})}, \tag{5.6}$$

where $\Phi_{n,m}(t)$ denotes the phase-related parameters induced by the relative spatial position relationship between the scatterers and the antenna elements in the Tx and Rx array. It can be expressed as

$$\Phi_{n,m}(t) = \tilde{\mathbf{s}}^{\mathrm{Tx}}(t)\cdot\mathbf{d}_p^{\mathrm{Tx}} + \tilde{\mathbf{s}}^{\mathrm{Rx}}(t)\cdot\mathbf{d}_q^{\mathrm{Rx}}. \tag{5.7}$$

We have

$$\tilde{\mathbf{s}}^{o}(t) = \left[\cos(\alpha_{n,m}^{o}(t)), \sin(\alpha_{n,m}^{o}(t))\right], o \in \{\mathrm{Tx}, \mathrm{Rx}\}, \tag{5.8}$$

$$\mathbf{d}_p^{Tx} = \left[d_{p,x}^{Tx}, d_{p,y}^{Tx}\right]^{\mathrm{T}}, \mathbf{d}_q^{Rx} = \left[d_{q,x}^{Rx}, d_{q,y}^{Rx}\right]^{\mathrm{T}}. \tag{5.9}$$

where $\tilde{\mathbf{s}}^{o}(t)$ denotes the unit direction vectors for the arrival or departure of the signal, while $\alpha_{n,m}^{o}(t)$ denotes the angle of arrival (AoA) or angle of departure (AoD). Additionally, $\mathbf{d}_p^{Tx}$ and $\mathbf{d}_q^{Rx}$ correspond to the spatial coordinate of p-th Tx antenna or q-th Rx antenna, respectively.

In Eq.(5.6), $\theta_{n,m}$ represents the initial random phase and distributed uniformly over $(0, 2\pi]$, $f_{n,m}(t)$ represents the Doppler Frequency. As the CAV are usually independent from other CAVs, the total Doppler frequency can be calculated as

$$f_{n,m}(t) = f_{n,m}^{Tx}(t) + f_{n,m}^{Rx}(t), \tag{5.10}$$

where the Doppler frequency of the Tx or Rx can be further expressed by speed $v^o(t)$ and moving direction $\alpha_v^o(t)$ of Tx or Rx. That is

$$f_{n,m}^o(t) = \frac{v^o(t)}{\lambda} \cdot \cos(\alpha_{n,m}^o(t) - \alpha_v^o(t)), \tag{5.11}$$

In the context of cooperative inference over wireless communication, we model the time-varying speed and direction during the model feature sharing as linear functions. Consequently, the speed $v^o(t)$ and moving direction $\alpha_v^o(t)$ can be expressed as:

$$\begin{cases} v^o(t) = v_0^o + a_0^o \cdot t, \\ \alpha_v^o(t) = \alpha_v^o + \beta_0^o, \end{cases} \tag{5.12}$$

where v_0^o and α_v^o are the initial speed and orientation of the MT/MR, respectively, and a_0^o and β_0^o represent their respective accelerations. To analyse the feature sharing in each batch of data, we expand the $\cos(\alpha_{n,m}^o(t) - \alpha_v^o(t))$ by Taylor series expansion at the start of each batch of data sharing and retain the first two terms of the expansion provides an approximation. That is

$$\cos(\alpha_{n,m}^o(t) - \alpha_v^o(t)) = \cos(\alpha_{n,m}^o - \alpha_v^o) + k_0^o \cdot t, \tag{5.13}$$

where k_0^o, the linear coefficient in the Taylor expansion, is defined as

$$k_0^o = -\frac{v_0^o \cdot \sin^2(\alpha_{n,m}^o - \alpha_v^o)}{d_n^o} + \beta_0^o \cdot \sin(\alpha_{n,m}^o - \alpha_v^o), \tag{5.14}$$

where d_n^o epresents the initial distance between the Tx/Rx and S_n^{Tx}/S_n^{Rx}, respectively. Thus, $f_{n,m}^o(t)$ is given by

$$f_{n,m}^o(t) \approx \frac{a_0^o {k_0}^o}{\lambda} t^2 + \frac{a_0^o \cdot \cos(\alpha_{n,m}^o - \alpha_v^o) + v_0^o k_0^o}{\lambda} t + \frac{v_0^o \cos(\alpha_{n,m}^o - \alpha_v^o)}{\lambda}. \tag{5.15}$$

Furthermore, the phase parameter can be estimated by $\Phi_{n,m}(t) = \Phi^{Tx}_{n,m}(t) + \Phi^{Rx}_{n,m}(t)$, Let $\Phi^{o}_{n,m}(t)$, $o \in (Tx, Rx)$ as the phase at the Tx/Rx. Similarly, via Taylor expansion, the phase $\Phi^{o}_{n,m}(t)$ can be obtained as

$$\Phi^{o}_{n,m}(t) \approx \frac{2\pi}{\lambda}(k^{o}_{1} d^{o}_{u,x} + k^{o}_{2} d^{o}_{u,y})t + \frac{2\pi}{\lambda}(\cos(\alpha^{o}_{n,m}) \cdot d^{o}_{u,x} + \sin(\alpha^{o}_{n,m}) \cdot d^{o}_{u,y}), \tag{5.16}$$

where $u \in \{p, q\}$ indexes the p-th Tx or q-th Rx, k^{o}_{1} and k^{o}_{2} are then given by

$$\begin{aligned} k^{o}_{1} &= -\frac{v^{o}_{0}}{d^{o}_{n}} \cdot \sin^{2}(\alpha^{o}_{n,m}) + \beta^{o}_{0} \cdot \sin(\alpha^{o}_{n,m}), \\ k^{o}_{2} &= -\frac{v^{o}_{0}}{d^{o}_{n}} \cdot \cos^{2}(\alpha^{i}_{n,m}) + \beta^{o}_{0} \cdot \cos(\alpha^{o}_{n,m}). \end{aligned} \tag{5.17}$$

Upon substitution of the above equations in Eq. (5.6), $\tilde{h}_{p,q,n}(t, \tau)$ can be expressed as

$$\tilde{h}_{p,q,n}(t, \tau) = \frac{1}{\sqrt{M}} \sum_{m=1}^{M} \mathrm{e}^{\mathrm{j} \cdot (A \cdot t^{3} + B \cdot t^{2} + C \cdot t + D + \theta_{n,m})}, \tag{5.18}$$

where

$$\begin{aligned} A &= \frac{2\pi}{3\lambda} a^{o}_{0} k^{o}_{0}, \\ B &= \frac{\pi}{\lambda}(a^{o}_{0} \cdot \cos(\alpha^{o}_{n,m} - \alpha^{o}_{v}) + v^{o}_{0} k^{o}_{0}), \\ C &= \frac{2\pi}{\lambda}(v^{o}_{0} \cdot \cos(\alpha^{o}_{n,m} - \alpha^{o}_{v}) + k^{o}_{1} d^{o}_{u,x} + k^{o}_{2} d^{o}_{u,y}), \\ D &= \frac{2\pi}{\lambda}(\cos(\alpha^{o}_{n,m}) \cdot d^{o}_{u,x} + \sin(\alpha^{o}_{n,m}) \cdot d^{o}_{u,y}). \end{aligned} \tag{5.19}$$

The distance $d^{o}_{n}(t)$ between CAVs and clusters, is approximately expressed as

$$d^{o}_{n}(t) \approx d^{o}_{n} - v^{o}_{0} \cdot \cos(\bar{\alpha}^{o}_{n} - \alpha^{o}_{v}) \cdot t, \tag{5.20}$$

where $\bar{\alpha}^{o}_{n}$ denotes the initial mean orientation of AoA or AoD. The total delay of the n-th path is the sum of the Tx delay, the virtual link delay, and the Rx delay. It can be derived by

$$\tau_{n}(t) \approx \frac{d^{Tx}_{n} + d^{Rx}_{n} - (v^{Tx}_{0} \cdot \cos(\bar{\alpha}^{Tx}_{n} - \alpha^{Tx}_{v}) + v^{Rx}_{0} \cdot \cos(\bar{\alpha}^{Rx}_{n} - \alpha^{Rx}_{v})) \cdot t}{c} + \tilde{\tau}_{n}(t). \tag{5.21}$$

where c denotes the light speed, and $\tilde{\tau}_n(t)$ denotes the equivalent virtual link delay, updated using a first-order filtering method (Q. Zhu, Yang, et al., 2018).

The nth path power is determined by

$$P'_n(t) = \mathrm{e}^{-\tau_n(t)\cdot\frac{r_\tau - 1}{r_\tau \cdot \sigma_\tau}} \cdot 10^{-\frac{Z_n}{10}}, \tag{5.22}$$

where r_τ denotes the shadow term, σ_τ represents the delay distribution, and Z_n represents the spread.

Please note that the AoA and AoD can be characterized by a certain probability density function, and several previous works (Pedersen et al., 2000; Zajic & Stuber, 2008) have demonstrated the versatility of the von Mises distribution in approximating these distributions. Therefore, we assume that the AoA and AoD follow a von Mises distribution.

$$p_{\alpha^i_{n,m}}(\alpha^i_n(t)) = \frac{\exp(\kappa^i \cdot \cos(\alpha^i_{n,m}(t) - \bar{\alpha}^i_n(t)))}{2\pi \cdot \mathrm{I}_0(\kappa^i)}, \tag{5.23}$$

where κ^i denotes the concentration parameter, I_0 denotes the zeroth-order modified Bessel function of the first kind, and $\bar{\alpha}^i_n(t)$ represents the mean AoA/AoD.

5.2.1.4 Time-varying Distortion

In order to consider the effects of time-varying distortions in V2V channel, we model the noise level and the estimation error of the channel state information (CSI) as random processes. Their variations can be simulated as a Gaussian process. In this case, the signal-to-noise ratio (SNR) at time t can be expressed as

$$\mathrm{SNR}(t) = \mu_{\mathrm{SNR}} + \epsilon_{\mathrm{SNR}}(t), \tag{5.24}$$

where $\epsilon_{\mathrm{SNR}}(t) \sim \mathcal{N}(0, \sigma^2_{\mathrm{SNR}})$ with the standard deviation σ_{SNR} characterizing the extent of noise fluctuation around the mean SNR as μ_{SNR}.

Similarly, the time-varying CSI estimation error at time t can be modelled as

$$\hat{h}(t) = \tilde{h}(t) + \epsilon_{\mathrm{CSI}}(t), \tag{5.25}$$

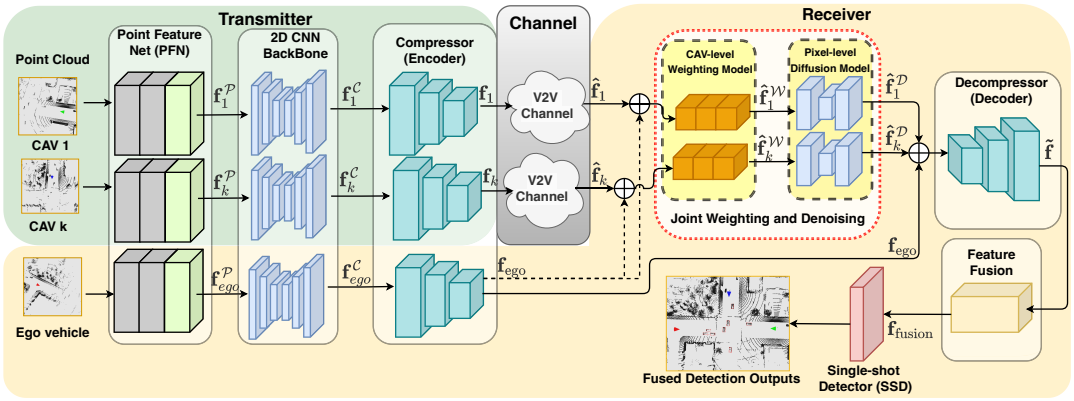


Figure 5.2: System architecture of cooperative inference with v2v communication.

where $\epsilon_{\text{CSI}}(t) \sim \mathcal{N}(0, \sigma^2_{\text{CSI}})$, and σ_{CSI} denotes the fluctuations of CSI error.

5.3 Cooperative inference with End-to-end Communicaiton

Figure 5.2 shows the system model for cooperative inference with the proposed joint adaptive weighting and denoising in networked vehicular systems. PointPillars (Lang et al., 2019) is adopted as the backbone algorithm with a pillar feature net (PFN), a 2D convolutional neural network (CNN) backbone, and a single shot detector (SSD) (W. Liu et al., 2016). Firstly, point clouds are divided into an x-y grid of pillars. To address the sparsity of pillars and points, each pillar is augmented with high-dimensional features, where limits are imposed on the number of non-empty pillars and points to formulate a dense tensor. Excess data is randomly sampled, and missing data is zero-padded. Subsequently, a simplified PointNet (Qi, Su, et al., 2017) extracts features with linear layers, batch normalization (BatchNorm), and rectified linear unit (ReLU). The resulting features are scattered back to their original positions, forming a pseudo-image denoted as $\mathbf{f}^{\mathscr{P}}$ in Figure 5.2. Then, pseudo-image features $(\mathbf{f}^{\mathscr{P}}_{ego}, \mathbf{f}^{\mathscr{P}}_{1}, \ldots, \mathbf{f}^{\mathscr{P}}_{k})$ are processed by the 2D CNN backbone, which adopts a residual structure with BatchNorm and ReLU layers to produce convoluted features $(\mathbf{f}^{\mathscr{C}}_{ego}, \mathbf{f}^{\mathscr{C}}_{1}, \ldots, \mathbf{f}^{\mathscr{C}}_{k})$. To prepare the sharing features, feature downsampling is conducted at the transmitter using a CNN-based encoder to compress

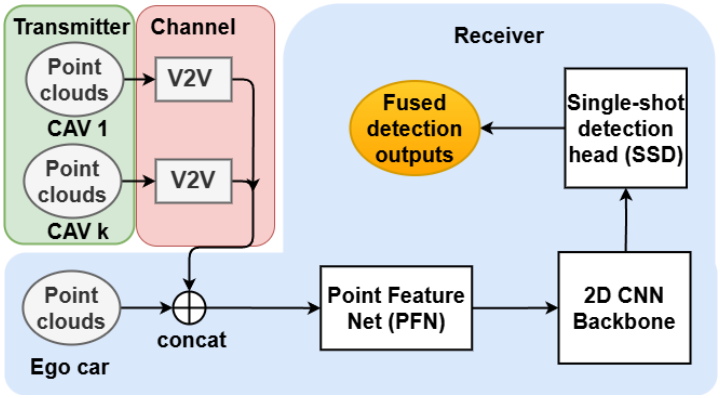
the feature and improve communication efficiency. At the receiver, the received and convoluted sharing features $(\hat{\mathbf{f}}_1,\ldots,\hat{\mathbf{f}}_k)$ are concatenated with $\mathbf{f}_{ego}$, respectively, and then deconvoluted to reconstruct the feature map $\tilde{\mathbf{f}}$. This downsampling and upsampling operation could take light channel impairments into account during model training. However, severe and dynamic channel impairments cannot be resolved by this encoder-decoder structure. Therefore, the proposed CAV-level weighting and pixel-level denoising method is implemented at the receiver before up-sampling to mitigate the effects of various channel impairments, which will be introduced in Section 5.4. Furthermore, feature fusion is conducted to aggregate the concatenated feature $\tilde{\mathbf{f}}$ into $\mathbf{f}_{\text{fusion}}$. Finally, an SSD is used to output the classification results of objects and the regression results of their box localization.

To enable effective collaboration among multiple CAVs, V2V communications are critical to enable them to share perceptual information. This learning-based detection framework can be seamlessly integrated with end-to-end communications systems to achieve global optimization. Specifically, the detection networks can be also trained to address wireless channel impairments using a distortion-in-the-loop training method. This framework serves as the benchmark and our work will focus on further improving cooperative inference under wireless channel impairments.

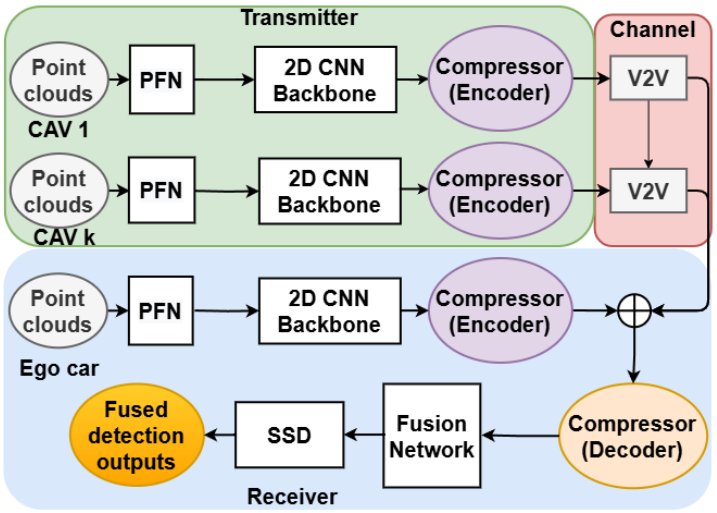
In this work, we utilize V2VNet (T.-H. Wang et al., 2020) as an intermediate-level fusion technique to combine the transmitted features by a graph neural network (GNN). V2VNet is particularly effective in balancing perception performance with the constraints of existing hardware transmission bandwidth by compressing the intermediate representations of the detector. The proposed joint weighting and denoising module will be introduced later.

5.3.1 Conventional Fusion Schemes with V2V Communication

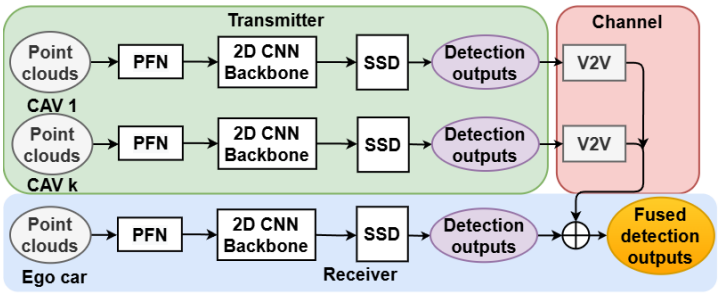
As mentioned in chapter 2, conventional cooperative inference frameworks can be categorized into three types: (i) raw-data-level fusion, (ii) intermediate-level fusion, and (iii) object-level fusion. The frameworks of the three conventional cooperative inference



(a) Raw-data-level fusion.



(b) Intermediate-level fusion.



(c) Object-level Fusion.

Figure 5.3: Cooperative inference with V2V communications: (a) Raw-data-level fusion, (b) Intermediate-level fusion, (c) Object-level fusion.

methods, with the V2V communication module highlighted in pink boxes, are shown in the three subfigures of Figure 5.3. In raw-data-level fusion (Figure 5.3(a)), the ego vehicle receives point clouds transmitted from other CAVs and concatenates them, as point cloud data are exhibiting unorderness and transformation invariance. Pre-processing, such as cropping and downsampling, is applied based on the ego vehicle's detection range to reduce computational cost. In intermediate-level fusion (Figure 5.3(b)), frameworks like V2VNet (T.-H. Wang et al., 2020) adopt an encoder to downsample shared features before transmission and then decode the features at the ego vehicle. These features are processed by a spatial-aware Graph Neural Network (GNN) and a convolutional gated recurrent unit (ConvGRU) to perform feature fusion. Unlike raw-data-level fusion and intermediate-level fusion, which require collaborative learning at the ego vehicle, object-level fusion (Figure 5.3(c)), relies only on detection outputs from all CAVs. This scheme facilitates information aggregation from CAVs, eliminating the need for offline collaborative learning. Object-level fusion refines detection outputs through non-maximum suppression (NMS) and range filtering to yield the final results.

5.3.2 Numerical Results and Discussion

Existing works on the three aforementioned fusion schemes often overlooks the impact of wireless channel impairments. However, sharing raw point cloud data, intermediate feature maps, and detection outputs via V2V communication exhibit varying robustness to such impairments. Therefore, we analyze their performance by utilizing the average precision (AP) as performance metric under realistic channel conditions.

5.3.2.1 Simulation Setup

We utilize the OPV2V dataset, proposed in (Xu, Xiang, Xia, et al., 2022) and constructed using the OpenCDA simulation tool, to conduct training and evaluation. The OPV2V dataset combines the default CARLA towns (6,765 training samples, 1,980 validation samples) and the Culver City dataset (550 samples) used as test set to evaluate the domain adaptability of the proposed model. We use V2VNet (T.-H. Wang et al., 2020)

for intermediate-level fusion. Training is conducted using the Adam optimizer with a learning rate of 0.002, over 60 epochs with a batch size of 2.

To account for wireless channel effects, we consider Rician fading with a Rician K-factor of 1 and AWGN with an SNR ranging from −10 to 30 dB. Given the varying coordinates of the CAVs, the raw point clouds and detection outputs are normalized along each axis to have a mean of 0 and a variance of 1, effectively following a standard normal distribution. This normalization is performed prior to transmission.

5.3.2.2 Effects of Channel Impairment

SNR	Raw-data-level		Intermediate-level		Object-level	
	AP@0.3	AP@0.7	AP@0.3	AP@0.7	AP@0.3	AP@0.7
Ideal	84.2%	73.2%	**90.5%**	**86.0%**	80.4%	74.7%
0	39.0%	31.3%	**63.7%**	**62.7%**	19.8%	11.8%
10	42.1%	31.7%	**85.1%**	**80.6%**	45.0%	8.0%
20	49.1%	33.3%	**88.8%**	**84.3%**	79.5%	33.9%
30	71.4%	47.6%	**90.0%**	**85.0%**	80.2%	67.4%

Table 5.1: Average precision under fading and noise conditions.

SNR	Raw-data-level		Intermediate-level		Object-level	
	AP@0.3	AP@0.7	AP@0.3	AP@0.7	AP@0.3	AP@0.7
Ideal	84.2%	73.2%	**90.5%**	**86.0%**	80.4%	74.7%
0	**32.1%**	**29.2%**	6.2%	4.4%	21.4%	15.9%
10	36.1%	30.6%	**36.2%**	**32.4%**	27.6%	8.2%
20	40.7%	31.6%	**78.1%**	**73.5%**	66.2%	15.4%
30	44.7%	32.3%	**90.1%**	**85.6%**	78.1%	54.2%

Table 5.2: Average precision under path loss, fading, and noise conditions.

Tables 5.1 and 5.2 show the performance of cooperative inference for different fusion schemes under fading, noise, and path loss conditions. Table 5.1 shows the cooperative inference under fading and noise scenarios. The accuracy of raw-data-level fusion steadily increases with SNR. For IoU = 0.7, accuracy improves from 31.3% to 47.6%, while for IoU = 0.3, it grows from 39% to over 71.4% with increasing SNR from −10 to 30 dB. Object-level fusion performs better than raw-data-level fusion for SNR values larger than 10 dB, with accuracy ranging from 19.8% to 80.2%for IoU = 0.3 and 8% to 67.4%

for IoU = 0.7. Intermediate-level fusion exhibits remarkable robustness, maintaining consistent level of accuracy over 62.7% for IoU = 0.7 and 63.7% for IoU = 0.3 across all SNR levels. Table 5.2 shows the cooperative inference in the presence of path loss, fading, and noise, intermediate-level fusion retains accuracy comparable to the noise-and-fading-only scenario when the SNR exceeds 10 dB. However, the AP experiences a substantial decline, decreasing from above 80% to around 10% as the SNR falls from 10 to -10 dB. Contrastingly, raw-data-level and object-level fusion show increasing accuracy as the SNR declines below 10 dB, attributed to the filtering of unreliable point cloud data and detection results caused by high distortion. This compels the ego vehicle to rely more on its measurements. These observations highlight intermediate-level fusion scheme is more resilience to noise and fading than raw-data-level fusion and object-level fusion.

5.4 Joint Adaptive Weighting and Denoising Approach

As analyzed in the previous section, cooperative inference contains communication impairments in V2V communication that may corrupt intermediate feature maps. These distortions compromise detection accuracy and affect system safety and reliability. To this end, we propose a joint CAV-level weighting and pixel-level denoising module to alleviate the performance degradation caused by the distorted shared feature maps.

In this section, a CAV-level weighting approach based on self-supervised learning is firstly introduced, followed by the pixel-level denoising approach that inspired by U-Net (Ronneberger et al., 2015) and CDiffuSE (Lu et al., 2022). Finally, the joint weighting and denoising framework will be presented. Figure 5.2 shows the system for cooperative inference with the joint adaptive weighting and denoising in networked vehicular systems. In which, PointPillar (Lang et al., 2019) is used as the 3D detection backbone model. It is important to note that, an autoencoder is exploited to minimize the volume of transmitted data, the feature maps undergo a 32-fold compression. This compression significantly reduces the data size and balances the requirements on detection accuracy

and bandwidth usage while preserving essential information for subsequent processing.

The motivation for proposing the synergy between CAV weighting and pixel-level denoising framework lies in mitigating the limitations associated with both CAV- and pixel-level processes. This hierarchical architecture integrates weighting and denoising, capitalizing on the simplicity of CAV weighting while addressing information loss through generative diffusion models. By incorporating contrastive self-supervision with generative learning, the framework achieves superior feature representation and synthesis capabilities to account for non-stationarity of V2V channel impairments. Moreover, the shared utilization of residual blocks and skip connections ensures structural compatibility between the two modules, facilitating seamless adaptation and effective integration.

5.4.1 CAV-level Weighting

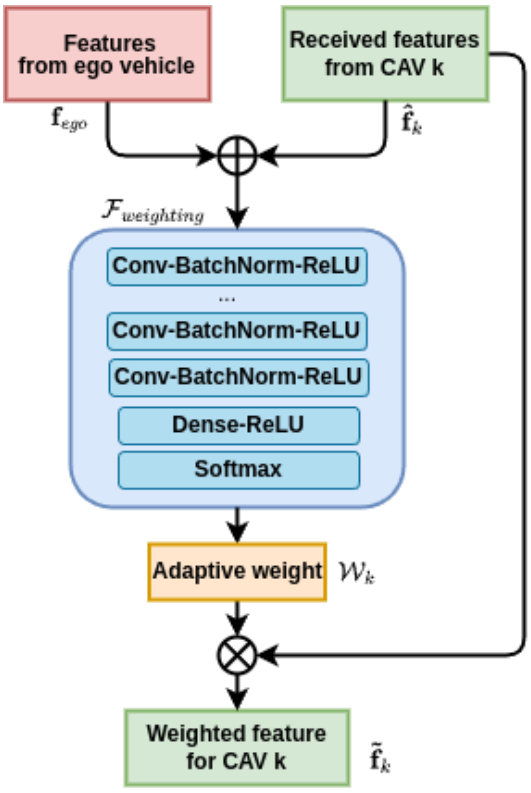


Figure 5.4: CAV-level weighting processes.

During the collaboration, after the ego vehicle receives shared features from other CAVs via the V2V communication channel, a CAV-level weighting network is firstly employed to address channel distortions. This network suppresses excessively distorted features and filters out outliers by assigning them very small weights, while features with minimal distortion are assigned larger weights. This process effectively reduces the impact of channel distortion, ensuring a more reliable and accurate feature fusion in subsequent stages. The weighted features of the k-th CAV can be obtained by

$$\mathscr{W}_k = \mathscr{F}_{weighting}(\mathbf{f}_{ego}, \hat{\mathbf{f}}_k) \tag{5.26}$$

$$\tilde{\mathbf{f}}_k = \mathscr{W}_k \hat{\mathbf{f}}_k \tag{5.27}$$

where $\mathbf{f}_{ego}$ and $\hat{\mathbf{f}}_k \in \mathbb{R}^{B\times C\times H\times W}$ denote the intermediate features from the ego vehicle and the k-th CAV, respectively, $\mathscr{F}_{weighting}$ denotes the adaptive weighting module, $\mathscr{W}_k \in \mathbb{R}^{B\times 1\times 1\times 1}$ denotes the output weight for received feature of each CAV, and $\tilde{\mathbf{f}}_k$ denotes the weighted feature. Figure 5.4 illustrates the workflow of the CAV-level weighting mechanism. We extract the weights using Conv2D-BatchNorm-ReLU blocks, the output of the former blocks is then processed by a linear dense layer and a Softmax operation. The linear dense layer is designed to accelerate faster convergence and address the vanishing gradient problem, while the Softmax function generates the regression-based weighting results. Consequently, the CAV-level weightings are normalized to values between 0 and 1 for each CAV.

The primary motivation is to leverage the complementary information embedded within the features of other CAVs. While the ego vehicle and other CAVs typically share similar data distributions and feature scales, especially under optimal communication conditions, significant discrepancies can arise when the communication channel experiences severe degradation. In such scenarios, the features from other CAVs diverge substantially from those of the ego vehicle. These contrasting characteristics serve as valuable inputs for the weighting module, enabling it to derive dynamic weights that effectively respond to varying channel conditions and ensure robust performance.

5.4.2 Pixel-level Denoise

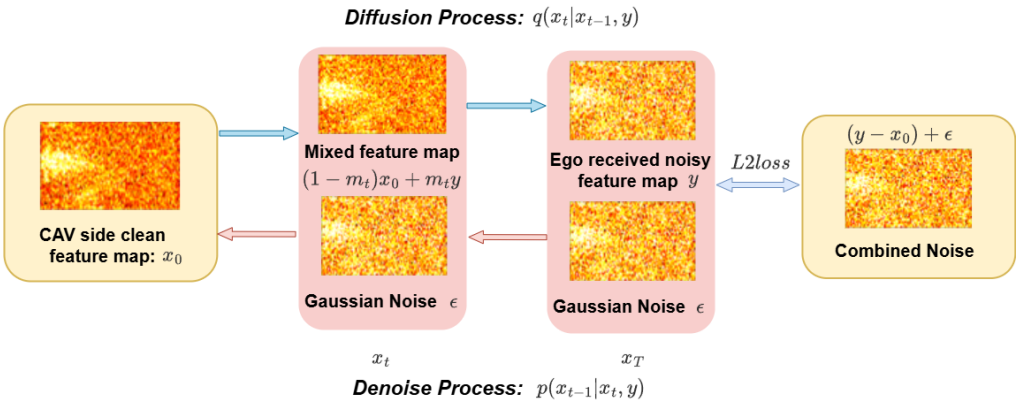


Figure 5.5: Pixel-level diffusion and denoising processes.

Figure 5.5 illustrates the pixel-level diffusion and reverse processes conditioned on the distorted features. This denoising module is designed by leverage recent advances in conditional diffusion probabilistic model (Lu et al., 2022) to recover and enhance distorted information from other CAVs in pixel-level.

The vanilla T-step denoising diffusion probabilistic model (DDPM) (Ho et al., 2020) comprises two main processes: (i) the **diffusion process**, in which Gaussian noise is progressively added to the data over T time steps ($t = 0$ to T); and and (ii) the **reverse process**, which reverses the diffusion process by removing the noise step-by-step($t = T$ to 0). Similarly, given the intermediate features at other CAVs x_0 as clean data, the ego vehicle side received features after V2V communication y as noisy data, the conditional diffusion model consists of conditional diffusion process and conditional reverse process. The diffusion process estimates the mixed noise between the transmitted clean feature x_0 and received distorted feature y, then the reverse process generates denoised feature maps.

5.4.2.1 Conditional Diffusion Process

In vanilla DDPM, the forward diffusion process q is conducted by gradually adding Gaussian noise to the clean data x_0 and transferred to a latent variable with a Gaussian

distribution of $p(x_T) = \mathcal{N}(0, I)$ over T steps. The step-dependent variable of diffusion at t-step is denoted as x_t. By defining a variance schedule $\{\beta_1, \beta_2, \cdots, \beta_T\}$, we can control the degree of noise injected at each step of the Markov chain that models the diffusion process from x_0 to the T-step variable x_T.

$$q(x_1, \cdots, x_T | x_0) = \prod_{t=1}^{T} q(x_t | x_{t-1}), \tag{5.28}$$

$$q(x_t | x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t} x_{t-1}, \beta_t I). \tag{5.29}$$

By incorporating the Gaussian model of $q(x_t|x_{t-1})$ into Eq. (5.28) and marginalizing over $\{x_1, x_2, \cdots, x_{t-1}\}$, the resulting sampling distribution for x_t is obtained as follows

$$q(x_t | x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1-\bar{\alpha}_t) I), \tag{5.30}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$.

The above vanilla diffusion process starts from the distribution of other CAV side clean data $q(x_0)$, while for the conditional diffusion process, we incorporate the received noisy feature y to act as the condition in each step. Specifically, the origin Gaussian model $q(x_t|x_0)$ is replaced by a conditional diffusion process $q_{cond}(x_t|x_0, y)$

$$q_{\text{cond}}(x_t | x_0, y) = \mathcal{N}(x_t; (1-m_t)\sqrt{\bar{\alpha}_t} x_0 + m_t \sqrt{\bar{\alpha}_t} y, \delta_t I). \tag{5.31}$$

Here, we model the mean of variable x_t in Eq. (5.31) as a linear combination of the clean features x_0 and the noisy features y. The contribution of y is controlled by a step-dependent ratio m_t, which is initialized at $m_0 = 0$ and gradually increases to approximately $m_T \approx 1$. In this diffusion process, the clean intermediate feature x_0 is gradually corrupted by adding the noise, conditioned on the noisy feature y, over T steps, which finally forms the complex noise patterns under dynamic channel conditions. By marginalizing y in Eq. (5.31), we obtain

$$q_{\text{cond}}(x_t | x_0) = \int q_{\text{cond}}(x_t | x_0, y) p_y(y | x_0) dy, \tag{5.32}$$

Then the $q_{\text{cond}}(x_t|x_0)$ is equvalent to the original diffusion process in Eq. (5.30) when

$$\delta_t = (1-\bar{\alpha}_t) - m_t^2 \bar{\alpha}_t. \tag{5.33}$$

5.4.2.2 Conditional Reverse Process

For the reverse process in the vanilla DDPM, the target features are generated by T refinement steps, starting from the prior distribution $p(x_T) = \mathcal{N}(0, I)$. The reverse process is modeled as a Markov chain with learned parameters θ, and can be obtained as follows

$$p_\theta(x_0, \cdots, x_{T-1}|x_T) = \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t). \tag{5.34}$$

where $p_\theta(x_{t-1}|x_t)$ represents the learned conditional distributions that refine the features step by step, gradually denoising them from Gaussian noise to recover the target features.

However, the following marginal likelihood, unlike the diffusion process, lacks a closed-form solution.

$$p_\theta(x_0) = \int p_\theta(x_0, \cdots, x_{T-1}|x_T) \cdot p_{\text{latent}}(x_T) dx_{1:T}. \tag{5.35}$$

Consequently, the Evidence Lower Bound (ELBO) is employed to formulate an approximate objective function. Previous studies (Ho et al., 2020) have demonstrated that enhanced generation quality can be achieved by minimizing the below equation.

$$c + \sum_{t=1}^{T} \kappa_t \mathbb{E}_{x_0,\epsilon} \parallel \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t) \parallel_2^2 . \tag{5.36}$$

where c and κ_t are constants, and ϵ_θ denotes the trained model employed to estimate the mixed noise within x_t. With Eq.(5.36) optimized, the corresponding reverse process is given by

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \tilde{\beta}_t I), \tag{5.37}$$

where the mean $\mu_\theta(x_t, t)$ is

$$\mu_\theta(x_t,t)=\frac{1}{\sqrt{\alpha_t}}(x_t-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t,t)), \tag{5.38}$$

and the variance is a constant value of $\tilde{\beta}_t$

$$\tilde{\beta}_t=\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \tag{5.39}$$

The vanilla reverse process described above starts with the prior distribution $p(x_T)=\mathcal{N}(0,I)$. The conditional reverse process predict x_T based on y as

$$p_{\text{cond}}(x_T|y)=\mathcal{N}(x_T,\sqrt{\bar{\alpha}_T}y,\delta_T I). \tag{5.40}$$

Unlike Eq. (5.37), the conditional reverse process $p_{\text{cond}}(x_{t-1}|x_t,y)$ aims to predict the variable x_{t-1} at $t-1$ step based on previous t step variable x_t and the noisy feature y:

$$p_{\text{cond}}(x_{t-1}|x_t,y)=\mathcal{N}(x_{t-1};\mu_\theta(x_t,y,t),\tilde{\delta}_t I), \tag{5.41}$$

where $\tilde{\delta}_t$ is the variance which will be introduced later; $\mu_\theta(x_t,y,t)$ is the estimated mean of x_{t-1}. Specifically, a weighted sum of x_t, y, and estimated noise ϵ with the coefficient c_{xt}, c_{yt}, $c_{\epsilon t}$ derives the mean:

$$\mu_\theta(x_t,y,t)=c_{xt}x_t+c_{yt}y-c_{\epsilon t}\epsilon_\theta(x_t,y,t), \tag{5.42}$$

$$c_{xt}=\frac{1-m_t}{1-m_{t-1}}\frac{\delta_{t-1}}{\delta_t}\sqrt{\alpha}_t+(1-m_{t-1})\frac{\delta_{t|t-1}}{\delta_t}\frac{1}{\sqrt{\alpha}_t}, \tag{5.43}$$

$$c_{yt}=(m_{t-1}\delta_t-\frac{m_t(1-m_t)}{1-m_{t-1}}\alpha_t\delta_{t-1})\frac{\sqrt{\bar{\alpha}_{t-1}}}{\delta_t}, \tag{5.44}$$

$$c_{\epsilon t}=(1-m_{t-1})\frac{\delta_{t|t-1}}{\delta_t}\frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\alpha_t}}, \tag{5.45}$$

$$\delta_{t|t-1}=\delta_t-\left(\frac{1-m_t}{1-m_{t-1}}\right)^2\alpha_t\delta_{t-1}, \tag{5.46}$$

$$\tilde{\delta}_t=\frac{\delta_{t|t-1}*\delta_t}{\delta_{t-1}}, \tag{5.47}$$

$\epsilon_\theta(x_t, y, t)$ is the estimated mixed noise by the diffusion model and $\tilde{\delta}_t$ denotes the variance in the reverse process. In this reverse process, the diffusion model iteratively predicts the mixed noise, starting from x_T and conditioning on the noisy feature y, to eventually get the denoised feature x_0 after T steps.

By modifying the derivations in (Ho et al., 2020), the ELBO for the conditional diffusion process can be expressed as

$$\begin{aligned} ELBO = -\mathbb{E}_q \Big[& D_{\mathrm{KL}}(q_{\mathrm{cond}}(x_T|x_0, y)||p_{\mathrm{latent}}(x_T|y)) \\ & + \sum_{t=2}^{T} D_{\mathrm{KL}}(q_{\mathrm{cond}}(x_{t-1}|x_t, x_0, y)||p_\theta(x_{t-1}|x_t, y)) \\ & - \log p_\theta(x_0|x_1, y) \Big]. \end{aligned} \tag{5.48}$$

Before optimizing Eq. (5.48), we need to determine the form of $q_{\mathrm{cond}}(x_t|x_{t-1}, y)$, which can be derived by comparing the predefined form of Eq. (5.31) and the coefficients of marginalized result to compute the exact coefficients of $q_{\mathrm{cond}}(x_t|x_{t-1}, y)$ as

$$\begin{aligned} q_{\mathrm{cond}}(x_t|x_{t-1}, y) = \mathcal{N}\Big(x_t; & \frac{1-m_t}{1-m_{t-1}}\sqrt{\alpha}_t x_{t-1} \\ & + \Big(m_t - \frac{1-m_t}{1-m_{t-1}} m_{t-1}\Big)\sqrt{\bar{\alpha}}_t y, \delta_{t|t-1} I\Big), \end{aligned} \tag{5.49}$$

where $\delta_{t|t-1}$ can be obtained via δ_t to satisfy Eq. (5.33):

$$\delta_{t|t-1} = \delta_t - \left(\frac{1-m_t}{1-m_{t-1}}\right)^2 \alpha_t \delta_{t-1}. \tag{5.50}$$

Then, by combining Eq. (5.31) and Eq. (5.49), $q_{\mathrm{cdiff}}(x_{t-1}|x_t, x_0, y)$ can be derived by combining Markov chain property with Bayes' theorem:

$$\begin{aligned} & q_{\mathrm{cond}}(x_{t-1}|x_t, x_0, y) = \\ & \mathcal{N}\Big(x_{t-1}; \frac{1-m_t}{1-m_{t-1}}\frac{\delta_{t-1}}{\delta_t}\sqrt{\alpha_t} x_t \quad + (1-m_{t-1})\frac{\delta_{t|t-1}}{\delta_t}\sqrt{\bar{\alpha}_{t-1}} x_0 \\ & + \Big(m_{t-1}\delta_t - \frac{m_t(1-m_t)}{1-m_{t-1}}\alpha_t \delta_{t-1}\Big)\frac{\sqrt{\bar{\alpha}_{t-1}}}{\delta_t} y, \tilde{\delta}_t I\Big), \end{aligned} \tag{5.51}$$

where $\tilde{\delta}_t$ is the variance of $q_{\text{cond}}(x_{t-1}|x_t, x_0, y)$ and can be calculated by the following equation.

$$\tilde{\delta}_t = \frac{\delta_{t|t-1} * \delta_t}{\delta_{t-1}}. \tag{5.52}$$

Finally, the ELBO in Eq. (5.48) can be optimized to

$$ELBO = c' + \sum_{t=1}^{T} \kappa'_t \mathbb{E}_{x_0,\epsilon,y} \left\| \frac{m_t \sqrt{\bar{\alpha}_t}}{\sqrt{1-\bar{\alpha}_t}}(y - x_0) + \frac{\sqrt{\delta_t}}{\sqrt{1-\bar{\alpha}_t}}\epsilon - \epsilon_\theta(x_t, y, t) \right\|_2^2. \tag{5.53}$$

where c' and κ'_t are constants, and ϵ represents the Gaussian noise in x_t. Compared to Eq. (5.36), the conditional diffusion model $\epsilon_\theta(x_t, y, t)$ extends its capbility by estimating not only Gaussian noise ϵ but also the non-Gaussian noisy $y - x_0$ within x_t.

5.4.3 Joint Training Scheme for Weighting and Denoising

As shown in Figure 5.6, the proposed joint CAV-level weighting and pixel-level denoising framework aims to offer an effective approach for fine-grained feature enhancements in cooperative inference. As previously stated, the weighting module can be expressed as

$$W_k = f_{\mathscr{W}}(\mathbf{f}_{\text{ego}}, \hat{\mathbf{f}}_k), \tag{5.54}$$

$$\mathbf{f}_k^{\mathscr{W}} = W_k \cdot \hat{\mathbf{f}}_k, \tag{5.55}$$

where W_k denotes the weighting output with a value from 0 to 1, $f_{\mathscr{W}}$ denotes the weighting module, $\mathbf{f}_{\text{ego}}$ and $\hat{\mathbf{f}}_k$ denote the feature from the ego CAV and the received feature from the k-th CAV, respectively, and $\mathbf{f}_k^{\mathscr{W}}$ is the k-th weighted feature. One key reason for adopting this module in the joint framework is the efficiency and adaptivity of using one value to filter out severe distorted features. However, this simplicity comes with the expense that some useful information in feature maps is also filtered out, reaching a bottleneck to improve performance when there is mild distortion.

To this end, the pixel-level denoising module is subsequently adopted to address information loss due to adaptive weighting and enhance features with moderate distor-

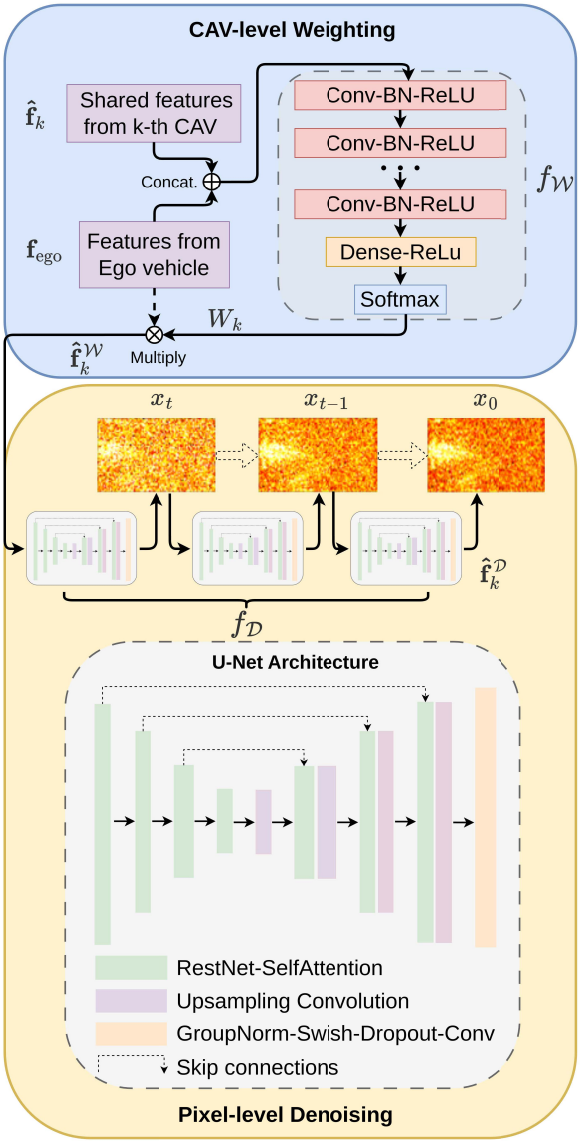


Figure 5.6: The proposed joint weighting and denoising algorithm pipeline.

tion. As shown in Figure 5.6, the features processed by the weighting module are then iteratively processed by the denoising module, which can be expressed as

$$\mathbf{f}_k^{\mathscr{D}} = f_{\mathscr{D}}(\mathbf{f}_k^{\mathscr{W}}) \tag{5.56}$$

where $\mathbf{f}_k^{\mathscr{D}}$ is the denoised features and $f_{\mathscr{D}}$ denotes the denoising function based on the U-Net architecture (Saharia et al., 2022).

The denoising module incorporates multiple network blocks, including ResNet-

SelfAttention, upsampling convolution, and a composite block consisting of GroupNorm (Group Normalization), Swish activation, dropout, and convolution, interconnected through direct and skip connections. Specifically, the ResNet-SelfAttention components are designed to effectively capture local features while simultaneously integrating global contextual information, leveraging long-range dependencies by the attention mechanism. The upsampling convolution layers progressively increase the spatial resolution of feature maps, facilitating the reconstruction of data at finer resolutions during the generative process. Skip connections link these upsampled features with features from earlier layers, aiding in the recovery of fine details. Finally, the integration of group normalization, Swish activation, dropout, and convolution provides robust handling of noisy inputs at varying levels while ensuring feature consistency across iterative denoising steps. The above learnable parameters are optimized through the generative learning process.

5.4.4 Numerical Results and Discussion

In this section, we evaluate the performance of our proposed joint weighting and denoising approach for cooperative inference in networked vehicular systems with various V2V communication channels.

5.4.4.1 Simulation Setup

The simulation settings, including the dataset, baseline, communication settings, and model training, are introduced in this section.

Dataset: To simulate real-world cooperative inference with V2V communication, we exploit the V2V4Real dataset (Xu, Xia, et al., 2023), a large-scale, multi-modal dataset collected from real-world vehicles and traffic scenarios for cooperative inference. This dataset encompasses a variety of driving environments, including intersections, highway entrance ramps, straight highway segments, and straight city roads, covering a total distance of 410 kilometers. It includes $20K$ LiDAR frames and $240K$ annotated 3D bounding boxes, providing a comprehensive foundation for both training and evaluation.

Baseline: For the cooperative inference, PointPillars (Lang et al., 2019) with V2VNet (T.-H. Wang et al., 2020) fusion module is used as the backbone for cooperative inference. The benchmark models are as below:

- Single AV detection is adopted by using only the ego vehicle, which does not suffer from distorted shared information, but also could not benefit from other CAVs.
- Cooperative inference without weighting and denoising (Coop) is trained using a supervised learning approach with distortion in the loop, which includes a simulated V2V communication channel.
- Cooperative inference with only CAV-level weighting (*Coop-W*) or only pixel-level denoising (*Coop-D*) is adopted for the ablation study of the proposed joint weighting and denoising approach.

Average precision (AP) is utilized as the performance metric, which calculates the AP at different levels of thresholds in terms of the Intersection-over-Union (IoU).

Communication Settings: To validate the effectiveness of our model in different communication conditions, we consider three types of communication models: (i) a simulated Rician fading channel model, (ii) a realistic WINNER II channel model (Bultitude & Rautiainen, 2007), and (iii) a non-stationary V2V channel model. A Rician K-factor of 1 with free-space path loss is adopted for the simulated Rician fading. The WINNER II channel model is implemented in an OFDM system with 64 subcarriers, where the carrier frequency is set to 2.6 GHz with 24 propagation paths and a maximum delay of 16. The non-stationary V2V channel model is simulated with time-varying distortion for the noise and the estimation error of the CSI.

Training: Firstly, the adaptive weighting module is trained by the self-supervised learning scheme in (C. Liu et al., 2024). Subsequently, the diffusion model is trained with the L2 loss function in Eq. (5.53). The U-Net architecture in (Saharia et al., 2022) is adopted as the diffusion model which contains three residual blocks. The training noise schedule is linearly spaced as $\beta_t \in [1 \times 10^{-4}, 0.035]$ with 50 diffusion steps. Besides, the fast sample scheme (Lu et al., 2022) is used in the reverse process with the inference schedule

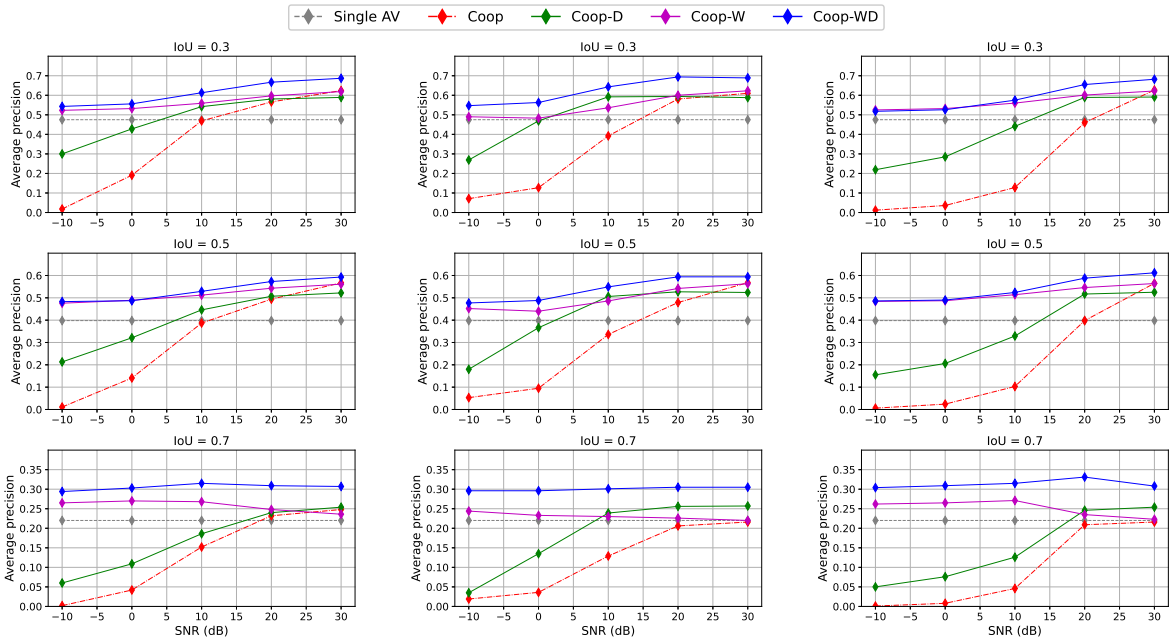
$[0.0001, 0.001, 0.01, 0.05, 0.2, 0.35]$. Furthermore, the non-stationary V2V channel is employed during the training phase. SNR of 15 dB is applied on the shared information from other CAVs.

5.4.4.2 Performance in Various Wireless Channels

In this section, we evaluate the cooperative inference under three different channels: (i) Rician fading, (ii) WINNER II, and (iii) non-stationary V2V channel. The model is trained with the non-stationary V2V channel and tested on the Rician fading, WINNER II channel and the non-stationary V2V channel to validate its generalizability on unseen channels.

As shown in Figure 5.7, the proposed *Coop-WD* model has the best robustness among all evaluated models to severe channel distortion. Additionally, it achieves the highest accuracy as channel conditions improve. Specifically, at an SNR of 30 dB, *Coop-WD* achieves AP scores of approximately 70% at IoU=0.3, 60% at IoU=0.5, and 30% at IoU=0.7. Even under challenging conditions with an SNR of -10 dB, it maintains AP scores of around 55%, 50%, and 30% for the same IoU thresholds, respectively.

In comparison, single CAV perception, which relies solely on the ego vehicle's own sensing data to avoid distortions from shared information among CAVs, achieves consistently low AP scores: 48% at IoU=0.3, 40% at IoU=0.5, and 22% at IoU=0.7, regardless of the SNR. *Coop-W*, which employs adaptive weighting to filter out distorted information, effectively mitigates the impact of severe channel impairments. However, it experiences slight performance degradation at IoU=0.3, as its simple weighting mechanism can result in information loss. On the other hand, *Coop-D*, leveraging a conditional diffusion model for denoising, performs better than *Coop* in general but struggles to address severe signal distortions as effectively as *Coop-W*. By integrating adaptive weighting with diffusion-based denoising, *Coop-WD* combines the strengths of both approaches. It mitigates severe channel distortions through weighting while leveraging the diffusion model to reconstruct features and compensate for light distortion. This synergy enables *Coop-WD* to consistently outperform both *Coop-W* and *Coop-D* across all SNR levels.



(a) (b) (c)

Figure 5.7: Average precision under various channels. (a) Rician fading. (b) WINNER II. (c) Non-stationary V2V channel.

Similar trends and results can also be observed in the realistic WINNER II channel and the more complex non-stationary V2V channel, considering various dynamic factors of CAVs.

5.4.4.3 Performance with Imperfect CSI and Different Path Loss Factors

In this section, the effects of imperfect CSI and different path loss factors are evaluated. The effect of imperfect CSI is simulated by adding Gaussian disturbance to the CSI.

Figure 5.8 shows the performance of cooperative inference under imperfect CSI. It is demonstrated that the proposed *Coop-WD* outperforms the single vehicle detection and *Coop* without weighting and denoising in the presence of CSI errors. Also, *Coop-WD* is more robust to the CSI errors than *Coop*.

Figure 5.9 shows the performance of the cooperative inference at 30 dB SNR for varying path loss factors in the non-stationary V2V channel. In the range of path loss factors from 1.0 to 2.25, the baseline and Coop-D approach experience a substantial decline in accuracy, decreasing from approximately 67%, 60%, 22% to 12%, 11%, 0.3%

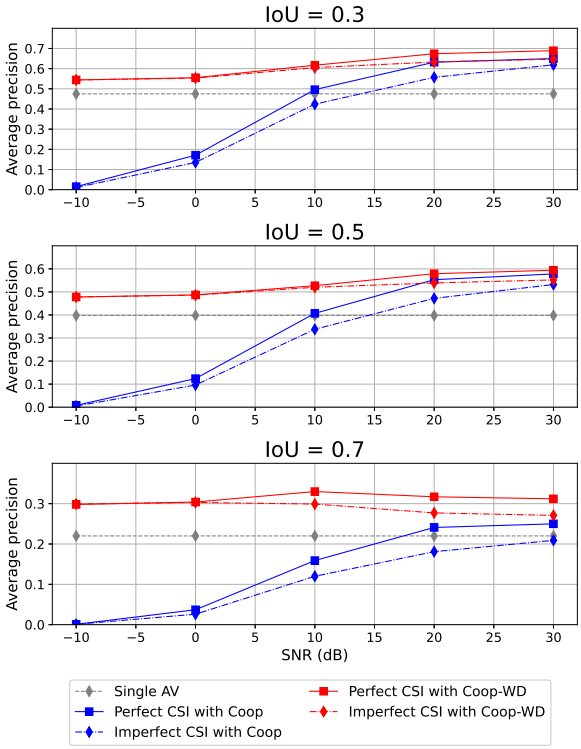


Figure 5.8: Average precision with imperfect CSI.

for IoU=0.3, 0.5, and 0.7, respectively. However, *Coop-W* and the *Coop-WD* demonstrate better robustness to changing path loss factors, where *Coop-WD* has a higher accuracy than *Coop-W* due to additional diffusion-based denoising module. It is validated that the proposed joint approach can enhance the system robustness to various communications environments.

5.4.4.4 Performance under Time-varying Distortion

In this section, we further evaluate the robustness and adaptivity of the proposed method under time-varying distortions. Specifically, we assume that the shared feature map is divided by multiple time slots for transmission. During each time slot, Gaussian disturbances are simulated separately for CSI errors and noise levels to model time-varying dynamic distortions.

Figure 5.10(a) shows the cooperative inference performance under the effects of noise variations over time. As σ_{SNR} increases to 10, the baseline, *Coop*, experiences a

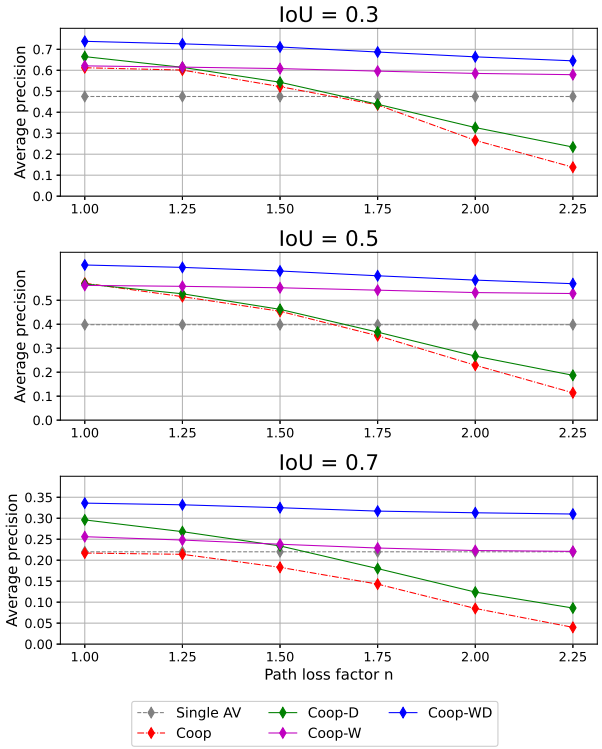


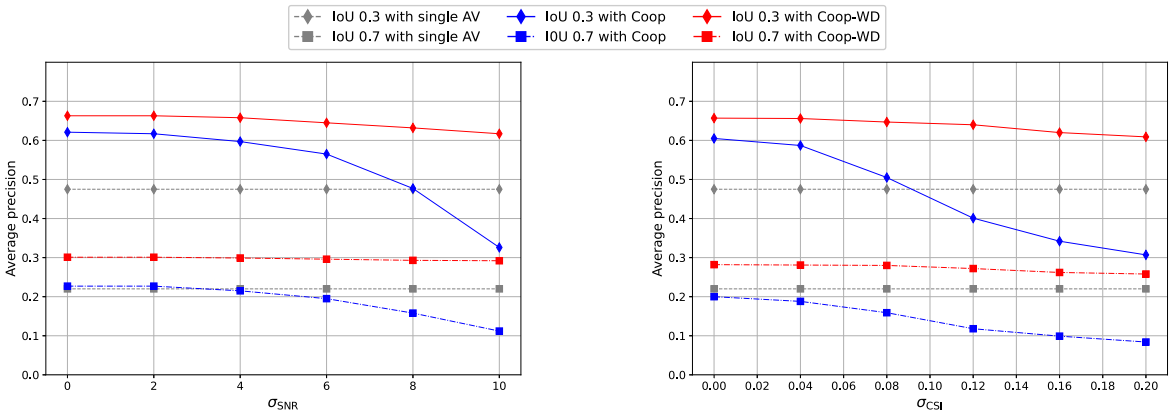
Figure 5.9: Average precision with different path loss factors.

substantial performance drop, retaining only 50% of the original AP scores due to the noise variation. In contrast, the proposed *Coop-WD* demonstrates a performance decline of less than 5% in AP, showing its significantly better robustness against time-varying noise variations. Similar trends and conclusions can be obtained for the variations of CSI errors in Figure 5.10(b).

5.4.4.5 Qualitative Analysis

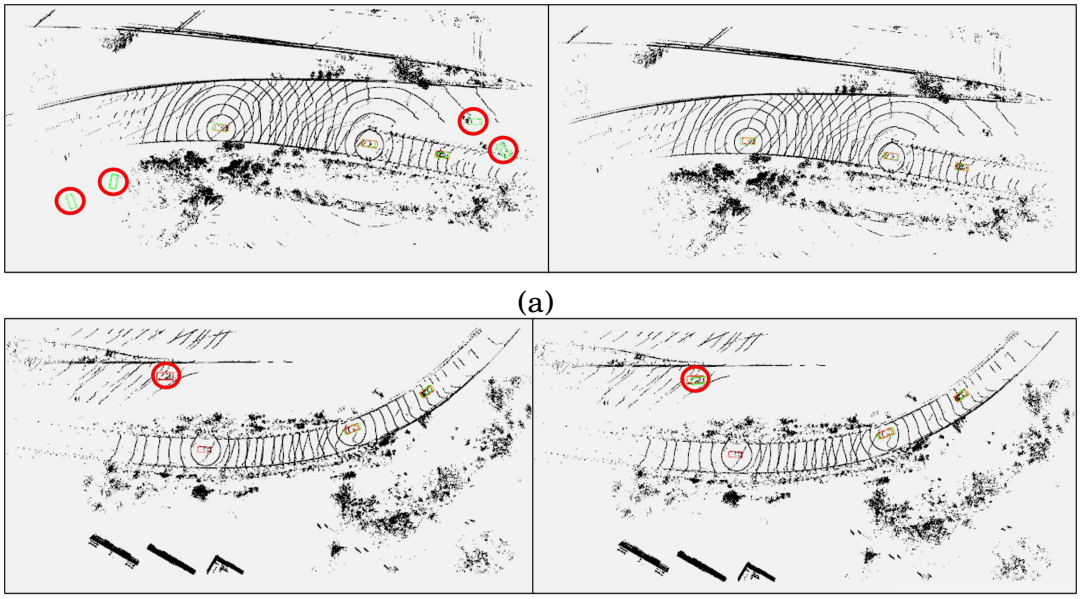
In this section, qualitative analysis with visual examples is provided for the ablation study to validate the effectiveness of the proposed method in Figures. 5.11 and 5.12, where the bounding boxes of ground truth and detection results are marked as red and green, respectively.

- **Comparison of the pixel-level *Coop-D* and the joint *Coop-WD*:** Figure 5.11 compares the performance of *Coop-D* and *Coop-WD* on the same data frame. In Figure 5.11 (a), *Coop-D* has false positive predictions, denoted by the red circles,



(a) (b)

Figure 5.10: Performance under time-varying disturbances, simulating disturbances with a fixed time duration following Gaussian distribution. (a) Time-varying noise levels (σ_{SNR}). (b) Time-varying CSI errors(σ_{CSI}).

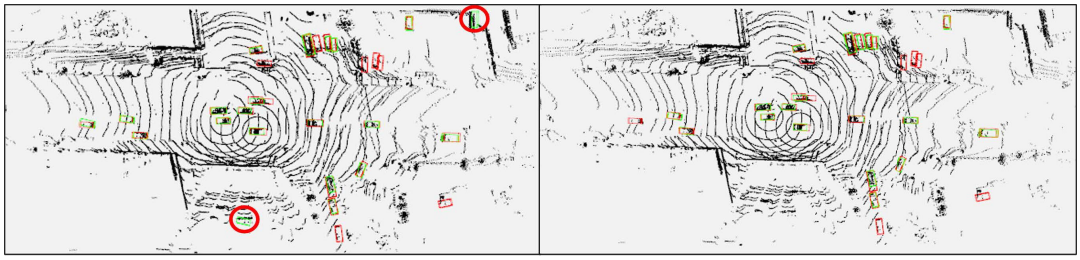


(a)

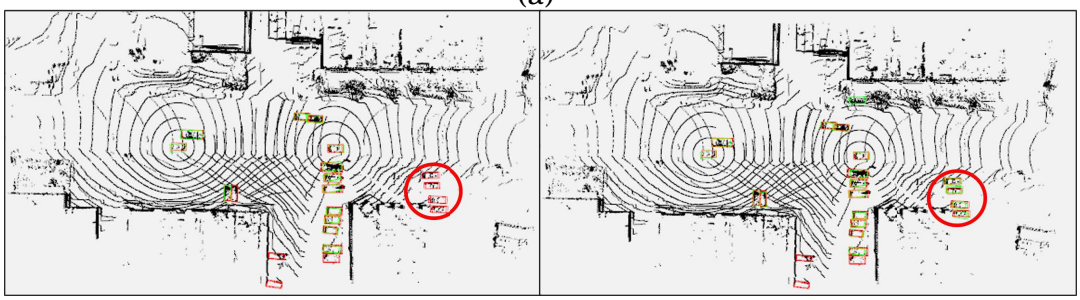
(b)

Figure 5.11: Visualization examples of Coop-D (top) and the Coop-WD (bottom). (a) False positive correction. (b) False negative correction.

due to channel distortions. However, the proposed *Coop-WD* can effectively avoid these false positive predictions. In Figure 5.11(b), *Coop-D* fails to detect a moving object denoted by the red circle, while the proposed *Coop-WD* successfully identifies it.



(a)



(b)

Figure 5.12: Visualization examples of Coop-W (top) and Coop-WD (bottom). (a) False positive correction. (b) False negative correction.

- **Comparison of the CAV-level *Coop-W* and the joint *Coop-WD*:** Figure 5.12 illustrates the visualization comparison of the same data frame between *Coop-W* and the proposed *Coop-WD*. *Coop-D* falsely identifies some areas without actual objects as targets in Figure 5.12 (a) and failed to detect some targets in Figure 5.12 (b). However, *Coop-WD* has effectively addressed these false detections or undetected results.

Furthermore, Table 5.3 compares each approach's average runtime and model size. Compared with the baseline without weighting and denoising modules, *Coop-W* increases runtime slightly to 57.88 ms with minimal impact on model size; *Coop-D* raises runtime to 131.25 ms and doubles the model size to 135.9 MB due to pixel-level feature processing. Although reducing the model complexity is not the primary focus of this work, it is noteworthy that *Coop-WD* achieves the best detection performance among the benchmarks at the expense of higher computational cost. These findings provide valuable insights for optimizing computational resource allocation in CAVs, particularly by prioritizing the ego vehicle, while also identifying this as a challenge for future exploration.

	Average Runtime (ms)	Model Size (Mb)
Coop	53.61	63.4
Coop-W	57.88	63.4
Coop-D	131.25	135.9
Coop-WD	137.45	135.9

Table 5.3: Comparison of runtime and size of the baseline without weighting and denoising, Coop-W, Coop-D, and Coop-WD.

5.5 Summary

In this work, we have proposed a novel joint weighting and denoising framework, *Coop-WD*, for enhancing cooperative inference in the presence of V2V channel impairments. Self-supervised training for feature weighting and generative diffusion models for denoising are integrated. A hierarchical feature processing mechanism was introduced to enhance features at both the CAV level and the pixel level, effectively mitigating the adverse effects of all-level channel-induced distortions. The performance of the proposed framework was evaluated across diverse channel models, including simulated Rician fading, realistic WINNER II, and non-stationary V2V channels. Numerical results have demonstrated that *Coop-WD* consistently outperforms standalone components (*Coop-W* and *Coop-D*) and conventional benchmarks without weighting and denoising under all conditions. It is also shown that Coop-WD not only mitigates severe signal distortions but also enhances performance under mild impairments, delivering fine-grained improvements to cooperative inference. Qualitative analysis with visualization examples have also demonstrated the better robustness and effectiveness of Coop-WD in comparison of Coop-W and Coop-D.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

This study addresses three urgent issues in cooperative model training and inference for the perception task within the AD system including IRSUs and CAVs, ranging from roadside sensor placement problems to communication-induced issues such as data transmission delay, and channel distortions during the collaborative training and inference process for the DNN-based detection model.

The first issue is how to leverage IRSUs in cooperative model training for the perception task within the AD system including IRSUs and CAVs. This issue was addressed by a novel cooperative model training framework among CAVs and IRSUs for AD systems, called RSFL framework. Two highlights along with this framework are a BSAP algorithm which can strategically reduce (increase) the number of sensors in low (high) complexity scenarios and a tailored multiple view federated distillation scheme, RSFL with BARL algorithm, which can be used to derive the information gained from road supervision and provide a fresh perspective on data annotation for the AD system. The results of high-fidelity experiments of using this framework in the CARLA simulator show the superiority of the proposed RSFL framework and BASP algorithm.

The second issue is how to integrate the domain specific perception task for FLCAV with V2X communication under practical communication topology among CAVs and IRSUs. This issue was addressed by a system-level road-assisted FLCAV framework and a multi-layer communication topology optimization algorithm which can reduce the communication delay among CAVs and IRSUs and improve the communication efficiency of the cooperative training process.

The third issue is how to enhance system robustness in the presence of competitive communication impairments, and channel distortions. Three cooperative fusion schemes regarding different levels of shared information have been estimated with V2V communication by considering communication noise, channel fadings, path loss, and imperfect CSI. Based on the evaluation result (i.e., the performance degradation), it urgent to explore a efficient approach to mitigate the adverse impact cause by V2V channel distortions. This issue was addressed by a joint CAV-level weighting and pixel-level denoising framework to mitigate the V2V channel distortion induced impact on cooperative inference, which can enhance the cooperative inference with V2V communication with high level of robustness. Specifically, this framework was proposed based on thorough evaluation of conventional fusion schemes for cooperative inference with V2V communication under various wireless channel conditions, including Rician fading, multi-path effects, and non-stationarity and consideration of factors such as varying SNRs levels, path loss imperfect CSI, and CAVs' time-vary factors. Numerical results showed that intermediate-level fusion by leveraging the features exhibits greater robustness to channel fading, varying noise, and path loss than raw-data-level fusion and object-level fusion. The outcomes of this study can significantly enhance the practicality of cooperative model training and inference for the perception task within the AD system.

6.2 Limitations and Future Work

The rapid advancement of IRSUs and the increasing utilization of CAVs have led to the development of various AV-related applications. Furthermore, with the commer-

cialization of 5G wireless communication technologies, the transition towards 6G has stimulated significant research efforts. AI has emerged as a critical technology for integration within CAVs and IRSUs, particularly in the context of intelligent transportation systems. However, despite the considerable potential of AI-driven autonomous driving technologies, their deployment remains limited due to ongoing technical and operational challenges.

The proposed road-assisted cooperative model training system presented in this study has been mainly validated in controlled simulation environments, and its adaptability to real-world transportation systems remains underexplored. In particular, the performance of the system in practical networked vehicular systems still requires comprehensive investigation due to the presence of dynamic traffic patterns, unpredictable human behaviors, and diverse environmental conditions in real-world scenarios. Furthermore, computational efficiency and resource optimization challenges persist in integrating CAVs and IRSUs, particularly in optimizing communication, processing power, resource allocation, and data sharing to improve road safety and overall traffic efficiency in intelligent transportation systems. In addition, concerns regarding data security and privacy have become increasingly critical as AI-enabled AV applications significantly rely on sensitive real-time data from multiple modalities. Addressing the above challenges will ensure the reliable deployment of AI-driven solutions in AV systems. The specific directions for future research are summarized as follows.

6.2.1 Real-world Roadside Infrastructures Deployment and Validation

In Chapter 3, the proposed roadside sensors placement algorithm and RSFL learning method were trained and validated over synthesized simulation data generated with the CARLA simulation environment. Refining the proposed algorithm for real-world data is equally important, as there is a domain gap between synthetic simulation data and real-world data. This gap arises primarily from limitations in accurately modeling traffic behavior and LiDAR rendering within simulators. Such discrepancies can lead to reduced

scalability and adaptability when transitioning to real-world deployments. Moreover, in practical scenarios, the deployment of CAVs and IRSUs often encounters computational constraints. This makes it crucial to design lightweight and efficient DNN-based detectors that can work effectively in resource-limited networked vehicular systems. In our current work (Chapters 3 and 4), we verified our novel framework and algorithms using a simplistic wireless channel. While effective for initial validation, this model does not fully account for the variability and interference that are commonly present in real-world communication networks. Future work will focus on addressing these gaps by incorporating realistic communication models into the cooperative model training process. Furthermore, real-world validation through field trials and experimentation with realistic networked vehicular systems will be essential to fully evaluate the performance, scalability, and adaptability of the proposed methods in dynamic transportation environments.

6.2.2 Task Offloading and Model Partition for Cooperative Inference

In Chapters 3 and 4, we have explored the cooperative model training/updating process with the collaboration of IRSUs and CAVs. The perception models are DNN-based models, consisting of a large number of parameters. For instance, the SECOND detector contains 5 million parameters. Given the limited computational resources available on CAVs and the constrained communication capacity of networked vehicular systems, achieving real-time performance for cooperative model training and inference presents a significant challenge. This challenge can be mitigated by leveraging the computational resources available at the IRSU side by offloading computation-intensive tasks from CAVs to roadside edge servers, thereby reducing the processing overhead on CAVs. However, fully offloading such tasks often necessitates the transmission of large volumes of data from CAVs to IRSUs. To address this, model partitioning for parallel processing offers a promising approach for enabling efficient collaboration between DNN models with large parameter sizes. As future work, we plan to develop efficient techniques for model

partitioning and task offloading approaches to minimize overall latency between CAVs and IRSUs, while also alleviating the computational burden on local CAVs. These strategies are essential for ensuring the scalability and real-time performance of cooperative inference and decision-making systems in AD environments.

6.2.3 Multi-modality Cooperative Perception

The work of Chapter 5 focused on LiDAR-based cooperative inference in the networked vehicular system and leveraging learning-based V2V communications to enhance the perception performance. While LiDAR offers reliable detection capabilities even in harsh weather conditions, it comes with high power consumption and sensor costs, making it a more resource-intensive option. Cameras, on the other hand, produce visual data rich in color and texture information, which can effectively complement LiDAR's point cloud data, providing a more well-rounded solution for perception tasks. This presents a promising avenue for exploring LiDAR-Camera fusion scheme in cooperative perception systems, where a tailored compression and fusion algorithm should be developed to optimize shared multi-modality information among CAVs, considering channel conditions and bandwidth constraints. Additionally, Radar technology can serve as a valuable component in this ecosystem, offering robust detection of objects, particularly moving ones, in adverse weather conditions where cameras and LiDAR might struggle. Radar's ability to operate effectively in conditions like fog, rain, or snow can significantly enhance perception systems, making it an essential element of cooperative perception frameworks. Furthermore, leveraging multi-modality data from IRSUs, such as Radar-equipped sensors, could provide critical real-time context, enhancing both the accuracy and efficiency of the cooperative perception system. In future work, we will delve deeper into road-assisted cooperative perception, incorporating multi-sensor fusion strategies involving LiDAR, Camera, and Radar to improve overall system robustness and performance

6.2.4 Active Perception

In Chapters 4 and 5, we focused on passive perception using DNNs-based detectors to extract features from collected sensor data. In which, the CAVs' trajectories were arbitrary, and the potential influence of the CAVs' routing path on perception quality was not considered yet. While effective in many scenarios, passive perception can be limited in dynamic and cluttered environments due to occlusions or suboptimal sensor viewpoints. In contrast, active perception aims to enhance the quality of sensor data by actively adjusting the CAVs' trajectories and sensor observation positions to optimize the perception process. In future work, we will explore active perception strategies integrating learning-based perception modules with optimization-based motion planning techniques. By doing so, the system can dynamically adjust CAV routes and sensor placements to avoid occlusions, thereby improving the accuracy and reliability of environmental perception. This approach would allow CAVs to proactively seek optimal viewpoints, enhancing situational awareness in complex traffic scenarios. Moreover, further research is needed to investigate how active perception can be implemented efficiently in real-time, particularly in highly dynamic environments. The combination of robust perception and motion planning is essential for achieving resilient and adaptive perception in cluttered and occlusion-prone urban settings.

6.2.5 Privacy Preserving for Cooperative Inference

In Chapter 5, we explored the impact of channel distortion on cooperation among CAVs in networked vehicular systems. In addition to channel impairment, it is also essential to explore the impact of various malicious attacks among CAVs. Such attacks pose significant risks to the safety of CAVs, potentially leading to accidents or disruptions in traffic flow. Large-scale disruptions caused by malicious attacks on AI-enabled AD systems could lead to significant economic ramifications, including expenses related to traffic accidents, IRSUs repairs, system restoration, and potential legal liabilities, placing a considerable financial strain on society. Understanding the implications of

these security breaches is critical in revealing the vulnerabilities of communication networks for CAVs and highlighting the urgent need for the development of secure and resilient AD systems. Furthermore, the nature of AI-enabled applications in intelligent transportation often entails the collection and analysis of vast amounts of personal data, raising serious concerns about user privacy. In the context of cooperative inference, this may involve the collection of sensitive data such as location information, and driving patterns. In future work, we will focus on developing privacy-preserving frameworks that strike a balance between utilizing this data to enhance system functionality and upholding individual privacy rights. Ensuring the security and privacy of AI-driven CAV networks will be pivotal to the widespread adoption and trustworthiness of intelligent transportation systems.

6.2.6 V2X Communication Channel Models for Cooperative Perception

In Chapters 4 and 5, we have examined how data transmission and signal distortion affect cooperative model training and inference. The communication models currently rely on free-space or long-distance path loss assumptions, which are insufficient for capturing the dynamic properties of V2X channels. The static positioning of IRSUs differentiates V2I communication from V2V, and the propagation characteristics of the V2I channel are still not thoroughly researched. Further studies should explore more advanced V2X channel models to facilitate more realistic communication frameworks for road-assisted cooperative model training and inference in ITS.

6.2.7 Physical Experiment in Real-world Devices

While in Chapters 4 and 5 the theoretical foundations and simulation of the proposed algorithm have shown promising outcomes, our next step will involve physical experiments using real-world devices, e.g., robots or vehicles, to validate the practical feasibility and robustness of the algorithm in the dynamic traffic environments. The insights gained

from these experiments will guide further optimization of the algorithms and inform future advancements in the field, bridging the gap between theoretical research and practical applications.

BIBLIOGRAPHY

Abdelkader, G., Elgazzar, K., & Khamis, A. (2021). Connected vehicles: Technology review, state of the art, challenges and opportunities. *Sensors*, *21*(22), 7712.

Agrawal, A., Verschueren, R., Diamond, S., & Boyd, S. (2018). A rewriting system for convex optimization problems. *J. Control and Decis.*, *5*(1), 42–60.

Alistarh, D., Hoefler, T., Johansson, M., Konstantinov, N., Khirirat, S., & Renggli, C. (2018). The convergence of sparsified gradient methods. *Advances in Neural Information Processing Systems*, *31*.

Ambrosin, M., Alvarez, I. J., Buerkle, C., Yang, L. L., Oboril, F., Sastry, M. R., & Sivanesan, K. (2019). Object-level perception sharing among connected vehicles. *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 1566–1573.

Arnold, E., Dianati, M., de Temple, R., & Fallah, S. (2020). Cooperative perception for 3D object detection in driving scenarios using infrastructure sensors. *IEEE Trans. Intell. Transp. Syst.*, *23*(3), 1852–1864.

Asghari, A., & Sohrabi, M. K. (2024). Server placement in mobile cloud computing: A comprehensive survey for edge computing, fog computing and cloudlet. *Computer Science Review*, *51*, 100616.

Beitollahi, M., Liu, M., & Lu, N. (2022). Dsfl: Dynamic sparsification for federated learning. *2022 5th International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, 1–6.

Bréhon–Grataloup, L., Kacimi, R., & Beylot, A.-L. (2022). Mobile edge computing for v2x architectures and applications: A survey. *Computer Networks*, *206*, 108797.

Bultitude, Y. d. J., & Rautiainen, T. (2007). Ist-4-027756 winner ii d1. 1.2 v1. 2 winner ii channel models. *EBITG, TUI, UOULU, CU/CRC, NOKIA, Tech. Rep*.

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., & Beijbom, O. (2020). Nuscenes: A multimodal dataset for autonomous driving. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11621–11631.

Cai, X., Jiang, W., Xu, R., Zhao, W., Ma, J., Liu, S., & Li, Y. (2023). Analyzing infrastructure lidar placement with realistic lidar simulation library. *IEEE Int. Conf. Robot. Automat.*, 5581–5587.

Chellapandi, V. P., Yuan, L., Brinton, C. G., Żak, S. H., & Wang, Z. (2023). Federated learning for connected and automated vehicles: A survey of existing approaches and challenges. *IEEE Transactions on Intelligent Vehicles*, *9*(1), 119–137.

Chen, D., & Krähenbühl, P. (2022). Learning from all vehicles. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17222–17231.

Chen, F., Luo, M., Dong, Z., Li, Z., & He, X. (2018). Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*.

Chen, J., You, D., Gündüz, D., & Dragotti, P. L. (2024). Commin: Semantic image communications as an inverse problem with inn-guided diffusion models. *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6675–6679.

Chen, M., Gündüz, D., Huang, K., Saad, W., Bennis, M., Feljan, A. V., & Poor, H. V. (2021). Distributed learning in wireless networks: Recent progress and future

challenges. *IEEE Journal on Selected Areas in Communications*, *39*(12), 3579–3605.

Chen, M., Yang, Z., Saad, W., Yin, C., Poor, H. V., & Cui, S. (2020). A joint learning and communications framework for federated learning over wireless networks. *IEEE Transactions on Wireless Communications*, *20*(1), 269–283.

Chen, Q., Ma, X., Tang, S., Guo, J., Yang, Q., & Fu, S. (2019). F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds. *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, 88–100.

Chen, Q., Tang, S., Yang, Q., & Fu, S. (2019). Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds. *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 514–524.

Chen, S., Liu, B., Feng, C., Vallespi-Gonzalez, C., & Wellington, C. (2020). 3d point cloud processing and learning for autonomous driving. *arXiv preprint arXiv:2003.00601*.

Chen, Z., Yi, W., & Nallanathan, A. (2023). Exploring representativity in device scheduling for wireless federated learning. *IEEE Transactions on Wireless Communications*, *23*(1), 720–735.

Cheng, H. K., Oh, S. W., Price, B., Schwing, A., & Lee, J.-Y. (2023). Tracking anything with decoupled video segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1316–1326.

Choukroun, Y., & Wolf, L. (2022). Denoising diffusion error correction codes. *arXiv preprint arXiv:2209.13533*.

Dahech, W., Pätzold, M., Gutiérrez, C. A., & Youssef, N. (2017). A non-stationary mobile-to-mobile channel model allowing for velocity and trajectory variations of

the mobile stations. *IEEE Transactions on Wireless Communications*, *16*(3), 1987–2000.

Deng, J., Shi, S., Li, P., Zhou, W., Zhang, Y., & Li, H. (2021). Voxel r-cnn: Towards high performance voxel-based 3d object detection. *Proceedings of the AAAI conference on artificial intelligence*, *35*(2), 1201–1209.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., & Koltun, V. (2017). Carla: An open urban driving simulator. *Conference on robot learning*, 1–16.

Du, J., Yu, F. R., Chu, X., Feng, J., & Lu, G. (2018). Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization. *IEEE Transactions on Vehicular Technology*, *68*(2), 1079–1092.

Du, J., Jiang, B., Jiang, C., Shi, Y., & Han, Z. (2023). Gradient and channel aware dynamic scheduling for over-the-air computation in federated edge learning systems. *IEEE Journal on Selected Areas in Communications*, *41*(4), 1035–1050.

Eskandarian, A., Wu, C., & Sun, C. (2019). Research advances and challenges of autonomous and connected ground vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *22*(2), 683–711.

Fan, L., Wang, F., Wang, N., & Zhang, Z.-X. (2022). Fully sparse 3d object detection. *Advances in Neural Information Processing Systems*, *35*, 351–363.

Fernandes, D., Silva, A., Névoa, R., Simões, C., Gonzalez, D., Guevara, M., Novais, P., Monteiro, J., & Melo-Pinto, P. (2021). Point-cloud based 3d object detection and classification methods for self-driving applications: A survey and taxonomy. *Information Fusion*, *68*, 161–191.

Gabb, M., Digel, H., Müller, T., & Henn, R.-W. (2019). Infrastructure-supported perception and track-level fusion using edge computing. *2019 IEEE Intelligent Vehicles Symposium (IV)*, 1739–1745.

Gao, H., Cheng, B., Wang, J., Li, K., Zhao, J., & Li, D. (2018). Object classification using cnn-based fusion of vision and lidar in autonomous vehicle environment. *IEEE Transactions on Industrial Informatics*, *14*(9), 4224–4231.

Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, *32*(11), 1231–1237.

Glaser, N. M., & Kira, Z. (2023). We need to talk: Identifying and overcoming communication-critical scenarios for self-driving. *arXiv preprint arXiv:2305.04352*.

Gong, X., Sharma, A., Karanam, S., Wu, Z., Chen, T., Doermann, D., & Innanje, A. (2021). Ensemble attention distillation for privacy-preserving federated learning. *IEEE/CVF Int. Conf. Comput. Vis*, 15076–15086.

Gonzalez-Barbosa, J.-J., Garcia-Ramirez, T., Salas, J., Hurtado-Ramos, J.-B., et al. (2009). Optimal camera placement for total coverage. *IEEE Int. Conf. Robot. Automat.*, 844–848.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., & He, K. (2017). Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*.

Grassucci, E., Barbarossa, S., & Comminiello, D. (2023). Generative semantic communication: Diffusion models beyond bit recovery. *arXiv preprint arXiv:2306.04321*.

Grassucci, E., Marinoni, C., Rodriguez, A., & Comminiello, D. (2024). Diffusion models for audio semantic communication. *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 13136–13140.

Guo, M., Liu, D., Simeone, O., & Wen, D. (2024). Efficient wireless federated learning via low-rank gradient factorization. *IEEE Transactions on Vehicular Technology*.

Hasan, M. K., Jahan, N., Nazri, M. Z. A., Islam, S., Khan, M. A., Alzahrani, A. I., Alalwan, N., & Nam, Y. (2024). Federated learning for computational offloading and resource management of vehicular edge computing in 6g-v2x network. *IEEE Transactions on Consumer Electronics*, *70*(1), 3827–3847.

He, Q., Wang, Z., Zeng, H., Zeng, Y., & Liu, Y. (2022). Svga-net: Sparse voxel-graph attention network for 3d object detection from point clouds. *Proceedings of the AAAI Conference on Artificial Intelligence*, *36*(1), 870–878.

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, *33*, 6840–6851.

Houston, J., Zuidhof, G., Bergamini, L., Ye, Y., Chen, L., Jain, A., Omari, S., Iglovikov, V., & Ondruska, P. (2021). One thousand and one hours: Self-driving motion prediction dataset. *Conference on Robot Learning*, 409–418.

Hu, Y., Fang, S., Lei, Z., Zhong, Y., & Chen, S. (2022). Where2comm: Communication-efficient collaborative perception via spatial confidence maps. *Advances in neural information processing systems*, *35*, 4874–4886.

Huang, X., Wang, P., Cheng, X., Zhou, D., Geng, Q., & Yang, R. (2019). The apolloscape open dataset for autonomous driving and its application. *IEEE transactions on pattern analysis and machine intelligence*, *42*(10), 2702–2719.

Itahara, S., Nishio, T., Koda, Y., Morikura, M., & Yamamoto, K. (2021). Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. *IEEE Transactions on Mobile Computing*, *22*(1), 191–205.

Jeong, E., Oh, S., Kim, H., Park, J., Bennis, M., & Kim, S.-L. (2018). Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*.

Jeong, W., & Hwang, S. J. (2022). Factorized-fl: Personalized federated learning with parameter factorization & similarity matching. *Advances in Neural Information Processing Systems*, *35*, 35684–35695.

Kang, M., Min, D., & Hwang, S. J. (2023). Grad-stylespeech: Any-speaker adaptive text-to-speech synthesis with diffusion models. *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5.

Kim, M., Fritschek, R., & Schaefer, R. F. (2023). Learning end-to-end channel coding with diffusion models. *WSA & SCC 2023; 26th International ITG Workshop on Smart Antennas and 13th Conference on Systems, Communications, and Coding*, 1–6.

Kim, T., Lin, E., Lee, J., Lau, C., & Mugunthan, V. (2024). Navigating data heterogeneity in federated learning: A semi-supervised approach for object detection. *Adv. Neural Inf. Process. Syst.*, *36*.

Koo, I., Lee, I., Kim, S.-H., Kim, H.-S., Jeon, W.-J., & Kim, C. (2023). Pg-rcnn: Semantic surface point generation for 3d object detection. *Proceedings of the IEEE/CVF international conference on computer vision*, 18142–18151.

Krajzewicz, D., Erdmann, J., Behrisch, M., & Bieker, L. (2012). Recent development and applications of sumo-simulation of urban mobility. *International journal on advances in systems and measurements*, *5*(3&4).

Krämmer, A., Schöller, C., Gulati, D., Lakshminarasimhan, V., Kurz, F., Rosenbaum, D., Lenz, C., & Knoll, A. (2019). Providentia–a large-scale sensor system for the

assistance of autonomous vehicles and its evaluation. *arXiv preprint arXiv:1906.06789*.

Kuo, H.-T., & Choi, T.-M. (2024). Metaverse in transportation and logistics operations: An ai-supported digital technological framework. *Transportation research part E: logistics and transportation review*, *185*, 103496.

Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., & Beijbom, O. (2019). Pointpillars: Fast encoders for object detection from point clouds. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12697–12705.

Lee, J., Solat, F., Kim, T. Y., & Poor, H. V. (2024). Federated learning-empowered mobile network management for 5g and beyond networks: From access to core. *IEEE Communications Surveys & Tutorials*.

Lei, Z., Ren, S., Hu, Y., Zhang, W., & Chen, S. (2022). Latency-aware collaborative perception. *European Conference on Computer Vision*, 316–332.

Li, B., Zhang, T., & Xia, T. (2016). Vehicle detection from 3d lidar using fully convolutional network. *arXiv preprint arXiv:1608.07916*.

Li, J., Dai, H., Han, H., & Ding, Y. (2023). Mseg3d: Multi-modal 3d semantic segmentation for autonomous driving. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 21694–21704.

Li, J., Xu, R., Liu, X., Ma, J., Chi, Z., Ma, J., & Yu, H. (2023). Learning for vehicle-to-vehicle cooperative perception under lossy communication. *IEEE Transactions on Intelligent Vehicles*, *8*(4), 2650–2660.

Li, J., Luo, C., & Yang, X. (2023). Pillarnext: Rethinking network designs for 3d object detection in lidar point clouds. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 17567–17576.

Li, W., Zhu, Q., Wang, C.-X., Bai, F., Chen, X., & Xu, D. (2020). A practical non-stationary channel model for vehicle-to-vehicle mimo communications. *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 1–6.

Li, Y., Fang, Q., Bai, J., Chen, S., Juefei-Xu, F., & Feng, C. (2023). Among us: Adversarially robust collaborative perception by consensus. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 186–195.

Li, Y., Ma, D., An, Z., Wang, Z., Zhong, Y., Chen, S., & Feng, C. (2022). V2x-sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving. *IEEE Robotics and Automation Letters*, *7*(4), 10914–10921.

Li, Y., Ren, S., Wu, P., Chen, S., Feng, C., & Zhang, W. (2021). Learning distilled collaboration graph for multi-agent perception. *Advances in Neural Information Processing Systems*, *34*, 29541–29552.

Li, Y., Bao, H., Ge, Z., Yang, J., Sun, J., & Li, Z. (2023). Bevstereo: Enhancing depth estimation in multi-view 3d object detection with temporal stereo. *Proceedings of the AAAI Conference on Artificial Intelligence*, *37*(2), 1486–1494.

Li, Z., Lan, S., Alvarez, J. M., & Wu, Z. (2024). Bevnext: Reviving dense bev frameworks for 3d object detection. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 20113–20123.

Lin, X., Liu, Y., Chen, F., Ge, X., & Huang, Y. (2023). Joint gradient sparsification and device scheduling for federated learning. *IEEE Transactions on Green Communications and Networking*, *7*(3), 1407–1419.

Lin, Y., Han, S., Mao, H., Wang, Y., & Dally, W. J. (2017). Deep gradient compression: Reducing the communication bandwidth for distributed training. *arXiv preprint arXiv:1712.01887*.

Liu, C., Chen, J., Chen, Y., Payton, R., Riley, M., & Yang, S.-H. (2024). Self-supervised adaptive weighting for cooperative perception in v2v communications. *IEEE Transactions on Intelligent Vehicles*, *9*(2), 3569–3580. https://doi.org/10.1109/TIV.2023.3345035

Liu, C., Chen, Y., Chen, J., Payton, R., Riley, M., & Yang, S.-H. (2023). Cooperative perception with learning-based v2v communications. *IEEE Wireless Communications Letters*, *12*(11), 1831–1835. https://doi.org/10.1109/LWC.2023.3295612

Liu, L., Chen, C., Pei, Q., Maharjan, S., & Zhang, Y. (2021). Vehicular edge computing and networking: A survey. *Mobile networks and applications*, *26*, 1145–1168.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, 21–37.

Liu, Y., Sun, Z., Li, G., & Hu, W. (2022). I know what you do not know: Knowledge graph embedding via co-distillation learning. *Proceedings of the 31st ACM international conference on information & knowledge management*, 1329–1338.

Liu, Y.-C., Tian, J., Glaser, N., & Kira, Z. (2020). When2com: Multi-agent perception via communication graph grouping. *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 4106–4115.

Liu, Y.-C., Tian, J., Ma, C.-Y., Glaser, N., Kuo, C.-W., & Kira, Z. (2020). Who2com: Collaborative perception via learnable handshake communication. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 6876–6883.

Liu, Y., Wang, S., Huang, J., & Yang, F. (2018). A computation offloading algorithm based on game theory for vehicular edge networks. *2018 IEEE International Conference on Communications (ICC)*, 1–6.

Liu, Z., Hou, J., Wang, X., Ye, X., Wang, J., Zhao, H., & Bai, X. (2025). Lion: Linear group rnn for 3d object detection in point clouds. *Advances in Neural Information Processing Systems*, *37*, 13601–13626.

Loutfi, S. I., Shayea, I., Tureli, U., El-Saleh, A. A., & Tashan, W. (2024). An overview of mobility awareness with mobile edge computing over 6g network: Challenges and future research directions. *Results in Engineering*, 102601.

Lu, Y.-J., Wang, Z.-Q., Watanabe, S., Richard, A., Yu, C., & Tsao, Y. (2022). Conditional diffusion probabilistic model for speech enhancement. *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7402–7406.

Mao, R., Guo, J., Jia, Y., Sun, Y., Zhou, S., & Niu, Z. (2022). Dolphins: Dataset for collaborative perception enabled harmonious and interconnected self-driving. *Proceedings of the Asian Conference on Computer Vision*, 4361–4377.

Mao, Y., Zhao, Z., Yan, G., Liu, Y., Lan, T., Song, L., & Ding, W. (2022). Communication-efficient federated learning with adaptive quantization. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *13*(4), 1–26.

McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. *Artificial intelligence and statistics*, 1273–1282.

Meyer, G. P., Laddha, A., Kee, E., Vallespi-Gonzalez, C., & Wellington, C. K. (2019). Lasernet: An efficient probabilistic 3d object detector for autonomous driving.

Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 12677–12686.

Mirzadeh, S. I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., & Ghasemzadeh, H. (2020). Improved knowledge distillation via teacher assistant. *Proceedings of the AAAI conference on artificial intelligence*, *34*(04), 5191–5198.

Nishio, T., & Yonetani, R. (2019). Client selection for federated learning with heterogeneous resources in mobile edge. *ICC 2019-2019 IEEE international conference on communications (ICC)*, 1–7.

Papernot, N., Abadi, M., Erlingsson, Ú., Goodfellow, I., & Talwar, K. (2022). Semi-supervised knowledge transfer for deep learning from private training data. *International Conference on Learning Representations*.

Pedersen, K. I., Mogensen, P. E., & Fleury, B. H. (2000). A stochastic model of the temporal and azimuthal dispersion seen at the base station in outdoor propagation environments. *IEEE Transactions on Vehicular Technology*, *49*(2), 437–447.

Posner, J., Tseng, L., Aloqaily, M., & Jararweh, Y. (2021). Federated learning in vehicular networks: Opportunities and solutions. *IEEE Network*, *35*(2), 152–159.

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, *30*.

Qureshi, K. N., Bashir, F., & Iqbal, S. (2018). Cloud computing model for vehicular ad hoc networks. *2018 IEEE 7th International Conference on Cloud Networking (CloudNet)*, 1–3.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, 234–241.

Saad, A., Senouci, M. R., & Benyattou, O. (2020). Toward a realistic approach for the deployment of 3d wireless sensor networks. *IEEE Transactions on Mobile Computing*, *21*(4), 1508–1519.

Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., & Norouzi, M. (2022). Image super-resolution via iterative refinement. *IEEE transactions on pattern analysis and machine intelligence*, *45*(4), 4713–4726.

Saleem, M. A., Li, X., Mahmood, K., Tariq, T., Alenazi, M. J., & Das, A. K. (2024). Secure rfid-assisted authentication protocol for vehicular cloud computing environment. *IEEE Transactions on Intelligent Transportation Systems*.

Schwartz, M., & Stern, T. (1980). Routing techniques used in computer communication networks. *IEEE Transactions on Communications*, *28*(4), 539–552.

Senouci, M. R., & Lehtihet, H. (2018). Sampling-based selection-decimation deployment approach for large-scale wireless sensor networks. *Ad Hoc Networks*, *75*, 135–146.

Seo, H., Park, J., Oh, S., Bennis, M., & Kim, S.-L. (2020). Federated knowledge distillation. *arXiv preprint arXiv:2011.02367*.

Shan, M., Narula, K., Wong, Y. F., Worrall, S., Khan, M., Alexander, P., & Nebot, E. (2020). Demonstrations of cooperative perception: Safety and robustness in connected and automated vehicle operations. *Sensors*, *21*(1), 200.

Shi, S., Wang, X., & Li, H. (2019). Pointrcnn: 3d object proposal generation and detection from point cloud. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 770–779.

Shi, S., Wang, Z., Shi, J., Wang, X., & Li, H. (2020). From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE transactions on pattern analysis and machine intelligence*, *43*(8), 2647–2664.

Shi, S., Cui, J., Jiang, Z., Yan, Z., Xing, G., Niu, J., & Ouyang, Z. (2022). Vips: Real-time perception fusion for infrastructure-assisted autonomous driving. *Proceedings of the 28th annual international conference on mobile computing and networking*, 133–146.

Shlezinger, N., Chen, M., Eldar, Y. C., Poor, H. V., & Cui, S. (2020). Uveqfed: Universal vector quantization for federated learning. *IEEE Transactions on Signal Processing*, *69*, 500–514.

Smith, V., Chiang, C.-K., Sanjabi, M., & Talwalkar, A. S. (2017). Federated multi-task learning. *Advances in neural information processing systems*, *30*.

Song, Z., Yang, L., Xu, S., Liu, L., Xu, D., Jia, C., Jia, F., & Wang, L. (2024). Graphbev: Towards robust bev feature alignment for multi-modal 3d object detection. *European Conference on Computer Vision*, 347–366.

Su, L., Zhou, R., Wang, N., Chen, J., & Li, Z. (2023). Low-latency hierarchical federated learning in wireless edge networks. *IEEE Internet of Things Journal*.

Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al. (2020). Scalability in perception for autonomous driving: Waymo open dataset. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2446–2454.

Tareq, M. M. K., Semiari, O., Salehi, M. A., & Saad, W. (2018). Ultra reliable, low latency vehicle-to-infrastructure wireless communications with edge computing. *2018 IEEE Global Communications Conference (GLOBECOM)*, 1–7.

Tihanyi, V., Rovid, A., Remeli, V., Vincze, Z., Csontho, M., Petho, Z., Szalai, M., Varga, B., Khalil, A., & Szalay, Z. (2021). Towards cooperative perception services for its: Digital twin in the automotive edge cloud. *Energies*, *14*, 5930.

Tsukada, M., Oi, T., Kitazawa, M., & Esaki, H. (2020). Networked roadside perception units for autonomous driving. *Sensors*, *20*(18), 5320.

Vijay, R., Cherian, J., Riah, R., De Boer, N., & Choudhury, A. (2021). Optimal placement of roadside infrastructure sensors towards safer autonomous vehicle deployments. *IEEE Int. Intell. Transp. Syst. Conf.*, 2589–2595.

Vijayakumar, A., & Vairavasundaram, S. (2024). Yolo-based object detection models: A review and its applications. *Multimedia Tools and Applications*, *83*(35), 83535–83574.

Vu, D.-Q., Le, N., & Wang, J.-C. (2021). Teaching yourself: A self-knowledge distillation approach to action recognition. *IEEE Access*, *9*, 105711–105723.

Wang, L., Fan, X., Chen, J., Cheng, J., Tan, J., & Ma, X. (2020). 3d object detection based on sparse convolution neural network and feature fusion for autonomous driving in smart cities. *Sustainable Cities and Society*, *54*, 102002.

Wang, S., Hong, Y., Wang, R., Hao, Q., Wu, Y.-C., & Ng, D. W. K. (2022). Edge federated learning via unit-modulus over-the-air computation. *IEEE Transactions on Communications*, *70*(5), 3141–3156.

Wang, S., Li, C., Ng, D. W. K., Eldar, Y. C., Poor, H. V., Hao, Q., & Xu, C. (2022). Federated deep learning meets autonomous vehicle perception: Design and verification. *IEEE network*, *37*(3), 16–25.

Wang, T.-H., Manivasagam, S., Liang, M., Yang, B., Zeng, W., & Urtasun, R. (2020). V2vnet: Vehicle-to-vehicle communication for joint perception and prediction. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, 605–621.

Wang, X., Li, J., Ning, Z., Song, Q., Guo, L., Guo, S., & Obaidat, M. S. (2023). Wireless powered mobile edge computing networks: A survey. *ACM Computing Surveys*, *55*(13s), 1–37.

Wen, J., Zhang, Z., Lan, Y., Cui, Z., Cai, J., & Zhang, W. (2023). A survey on federated learning: Challenges and applications. *International Journal of Machine Learning and Cybernetics*, *14*(2), 513–535.

Wen, W., Chen, Z., Yang, H. H., Xia, W., & Quek, T. Q. (2022). Joint scheduling and resource allocation for hierarchical federated edge learning. *IEEE Transactions on Wireless Communications*, *21*(8), 5857–5872.

Wu, C., Wu, F., Lyu, L., Huang, Y., & Xie, X. (2022). Communication-efficient federated learning via knowledge distillation. *Nature communications*, *13*(1), 2032.

Wu, Q., Chen, X., Zhou, Z., & Zhang, J. (2020). Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring. *IEEE Trans. Mob. Comput.*, *21*(8), 2818–2832.

Wu, T., Chen, Z., He, D., Qian, L., Xu, Y., Tao, M., & Zhang, W. (2024). Cddm: Channel denoising diffusion models for wireless semantic communications. *IEEE Transactions on Wireless Communications*.

Xia, B., Zhang, Y., Wang, S., Wang, Y., Wu, X., Tian, Y., Yang, W., & Van Gool, L. (2023). Diffir: Efficient diffusion model for image restoration. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13095–13105.

Xu, R., Guo, Y., Han, X., Xia, X., Xiang, H., & Ma, J. (2021). Opencda: An open cooperative driving automation framework integrated with co-simulation. *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 1155–1162.

Xu, R., Tu, Z., Xiang, H., Shao, W., Zhou, B., & Ma, J. (2023). Cobevt: Cooperative bird,äôs eye view semantic segmentation with sparse transformers. *Conference on Robot Learning*, 989–1000.

Xu, R., Xia, X., Li, J., Li, H., Zhang, S., Tu, Z., Meng, Z., Xiang, H., Dong, X., Song, R., et al. (2023). V2v4real: A real-world large-scale dataset for vehicle-to-vehicle cooperative perception. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13712–13722.

Xu, R., Xiang, H., Tu, Z., Xia, X., Yang, M.-H., & Ma, J. (2022). V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. *European conference on computer vision*, 107–124.

Xu, R., Xiang, H., Xia, X., Han, X., Li, J., & Ma, J. (2022). Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. *2022 International Conference on Robotics and Automation (ICRA)*, 2583–2589.

Yan, Y., Mao, Y., & Li, B. (2018). Second: Sparsely embedded convolutional detection. *Sensors*, *18*(10), 3337.

Yang, B., Luo, W., & Urtasun, R. (2018). Pixor: Real-time 3d object detection from point clouds. *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 7652–7660.

Yang, J., Shi, S., Wang, Z., Li, H., & Qi, X. (2022). St3d++: Denoised self-training for unsupervised domain adaptation on 3d object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, *45*(5), 6354–6371.

Yoo, J. H., Kim, Y., Kim, J., & Choi, J. W. (2020). 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, 720–736.

Yu, H., Luo, Y., Shu, M., Huo, Y., Yang, Z., Shi, Y., Guo, Z., Li, H., Hu, X., Yuan, J., et al. (2022). Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21361–21370.

Yuan, Y., Cheng, H., & Sester, M. (2022). Keypoints-based deep feature fusion for cooperative vehicle detection of autonomous driving. *IEEE Robotics and Automation Letters*, *7*(2), 3054–3061.

Zajic, A. G., & Stuber, G. L. (2008). Space-time correlated mobile-to-mobile channels: Modelling and simulation. *IEEE Transactions on Vehicular Technology*, *57*(2), 715–726.

Zarzar, J., Giancola, S., & Ghanem, B. (2019). Pointrgcn: Graph convolution networks for 3d vehicles detection refinement. *arXiv preprint arXiv:1911.12236*.

Zhang, W., Wang, Y., You, Z., Cao, M., Huang, p., Shan, J., Yang, Z., & Cui, B. (2022). Information gain propagation: A new way to graph active learning with soft labels. *Int. Conf. Learn. Represent. (ICLR)*.

Zhang, Z., Wang, S., Hong, Y., Zhou, L., & Hao, Q. (2021). Distributed dynamic map fusion via federated learning for intelligent networked vehicles. *2021 IEEE International conference on Robotics and Automation (ICRA)*, 953–959.

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., & Chandra, V. (2018). Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*.

Zhao, Y., & Haggman, S.-G. (2001). Intercarrier interference self-cancellation scheme for ofdm mobile communication systems. *IEEE transactions on Communications*, *49*(7), 1185–1191.

Zheng, T., Li, A., Chen, Z., Wang, H., & Luo, J. (2023). Autofed: Heterogeneity-aware federated multimodal learning for robust autonomous driving. *Int. Conf. Mobile Comput. Netw.*, 1–15.

Zhou, X., Liu, C., & Zhao, J. (2023). Resource allocation of federated learning for the metaverse with mobile augmented reality. *IEEE Transactions on Wireless Communications*.

Zhou, Y., & Tuzel, O. (2018). Voxelnet: End-to-end learning for point cloud based 3d object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4490–4499.

Zhu, G., Du, Y., Gündüz, D., & Huang, K. (2020). One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis. *IEEE Transactions on Wireless Communications*, *20*(3), 2120–2135.

Zhu, Q., Li, H., Fu, Y., Wang, C.-X., Tan, Y., Chen, X., & Wu, Q. (2018). A novel 3d non-stationary wireless mimo channel simulator and hardware emulator. *IEEE Transactions on Communications*, *66*(9), 3865–3878.

Zhu, Q., Yang, Y., Chen, X., Tan, Y., Fu, Y., Wang, C.-X., & Li, W. (2018). A novel 3d non-stationary vehicle-to-vehicle channel model and its spatial-temporal correlation properties. *IEEE access*, *6*, 43633–43643.

Zhuang, W., Wen, Y., Zhang, X., Gan, X., Yin, D., Zhou, D., Zhang, S., & Yi, S. (2020). Performance optimization of federated person re-identification via benchmark analysis. *Proceedings of the 28th ACM International Conference on Multimedia*, 955–963.