Title
Cooperative Driving Automation: Simulation and Perception

Author
Xu, Runsheng

Publication Date
2023

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Cooperative Driving Automation: Simulation and Perception

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Civil Engineering

by

Runsheng Xu

2023

ABSTRACT OF THE DISSERTATION

Cooperative Driving Automation: Simulation and Perception

by

Runsheng Xu
Doctor of Philosophy in Civil Engineering
University of California, Los Angeles, 2023
Professor Jiaqi Ma, Chair

Automated driving technology has emerged in recent years due to its potential to revolutionize transportation, bringing enhanced safety and efficiency. However, large-scale deployment is restricted by challenges inherent to single-vehicle systems, including occlusions, interactions with diverse traffic elements, and complicated decision-making. This dissertation advances the realm of Cooperative Driving Automation (CDA) as a solution, focusing on simulation frameworks and cooperative perception algorithms design.

The research starts with introducing OpenCDA, a comprehensive simulation framework for CDA system prototyping, and OPV2V, the first large-scale simulated cooperative perception dataset. These tools address the need for a simulated environment to prototype and validate CDA algorithms, bridging existing gaps in cooperative perception advancement.

Built upon OpenCDA and OPV2V, I present two state-of-the-art cooperative perception algorithms. The first, a cooperative 3D LiDAR detection framework, employs a Vision Transformer architecture to tackle challenges like sensor heterogeneity, localization error, and bandwidth constraints. The second, CoBEVT, is a pioneering multi-agent, multi-camera perception framework that uses economical RGB cameras to generate Bird-eye-view map predictions, offering a cost-effective solution.

The final segment of the research emphasizes real-world deployment. I present V2V4Real, the first real-world dataset for V2V perception, detailing its comprehensive benchmarks and introducing novel tasks. Further, I delve into strategies to optimally train cooperative perception models using simulated data, introducing a novel module, the Homogeneous Training Augmenter, which demonstrates the efficacy of simulation in real-world applications.

In essence, this thesis provides significant contributions to the domain of CDA, offering tools, datasets, and algorithms that pave the way for the broader, real-world implementation of cooperative automated driving.

The dissertation of Runsheng Xu is approved.

Sriram Narasimhan

Hongkai Yu

Bolei Zhou

Jiaqi Ma, Committee Chair

University of California, Los Angeles

2023

To my wife Xiaoyu Dong who always supports me

Table of Contents

List of Figures

Acknowledgment

Three years ago, when I quit my well-paid job at Mercedes-Benz and packed all my stuff in the apartment, I looked around the empty house and started to question myself: Is this a good decision? My wife really understood my feelings and encouraged me, saying, "Your dream is to change the world, and the PhD journey is a very good start. I will be your back no matter what challenges we will face."

Three years later, as I stand at the center of the stage to defend my PhD thesis and summarize how researchers from at least 40 different countries all over the world are using my work to build their own projects, I feel that I am already changing the world, even though it's just to a small extent. Therefore, I would like to thank my wife first for her companionship and support during the PhD journey. I couldn't have done it without her.

Next, I want to express my gratefulness to my supervisor, Dr. Jiaqi Ma. I used to regard the research problem and engineering problem as the same, but Dr. Ma pushed me to think deeper and only do meaningful work. He reshaped the way I think about research, which is the most precious gift for a researcher.

I would also like to thank my committee members: Sriram Narasimhan, Hongkai Yu, and Bolei Zhou. They have offered me many insightful suggestions during my PhD journey.

I want to extend my gratitude to all my collaborators on this journey: Zhengzhong Tu, Hao Xiang, Qinhua Jiang, and Xin Xia. You gave me the courage to overcome the challenges in the research.

In the end, I would like to say to my parents: "You made me very proud when I was a kid, and it is time to let me make you proud. I am very lucky to have the best parents in the world, and I know that I can chase my dream because you are always behind me to support me."

Curriculum Vitae

2012 – 2014	B.S. in Electrocal Engineering, North China Electrical Power University, Beijing, China.
2014 – 2016	B.S. in Electrocal Engineering, Illinois Institute of Technology, Illinois, USA.
2016 – 2017	MSc in Electrocal Engineering, Northwestern University, Illinois, USA.
2018 – 2019	Computer Vision Engineer, Oppo Mobile Research Center, California, USA.
2019 – 2020	Senior Sensor Fusion Engineer, Mercedes-Benz R&D North America, California, USA.
2021 – Present	Ph.D. student in Civil Engineering, University of California, Los Angeles (UCLA).
2023 – Present	Research Intern, Perception and Simulation, Waymo LLC

Publications

[1] Runsheng Xu, Yi Guo, Xu Han, Xin Xia, Hao Xiang, and Jiaqi Ma, "OpenCDA: an open cooperative driving automation framework integrated with co-simulation," (2021), in 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), IEEE, pp. 1155–1162 .

[2] Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and Jiaqi Ma, "Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication," (2022), in 2022 International Conference on Robotics and Automation (ICRA), IEEE, pp. 2583–2589 .

[3] Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma, "V2x-vit: Vehicle-to-everything cooperative perception with vision transformer," (2022), in European Conference on Computer Vision (ECCV), Springer, pp. 107–124 .

[4] Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, and Jiaqi Ma, "CoBEVT: Cooperative Bird's Eye View Semantic Segmentation with Sparse Transformers," (2022), in 6th Annual Conference on Robot Learning (CoRL) .

[5] Runsheng Xu, Weizhe Chen, Hao Xiang, Xin Xia, Lantao Liu, and Jiaqi Ma, "Model-agnostic multi-agent perception framework," (2023), in 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 1471–1478 .

[6] Runsheng Xu, Jinlong Li, Xiaoyu Dong, Hongkai Yu, and Jiaqi Ma, "Bridging the domain gap for multi-agent perception," (2023), in 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 6035–6042 .

[7] Runsheng Xu, Xin Xia, Jinlong Li, Hanzhao Li, Shuo Zhang, Zhengzhong Tu, Zonglin Meng, Hao Xiang, Xiaoyu Dong, Rui Song, and others, "V2v4real: A real-world large-scale dataset for vehicle-to-vehicle cooperative perception," (2023), in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. .

[8] Runsheng Xu, Zhengzhong Tu, Yuanqi Du, Xiaoyu Dong, Jinlong Li, Zibo Meng, Jiaqi Ma, Alan Bovik, and Hongkai Yu, "Pik-fix: Restoring and colorizing old photos," (2023), in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 1724–1734 .

[9] Runsheng Xu, Hao Xiang, Xu Han, Xin Xia, Zonglin Meng, Chia-Ju Chen, Camila Correa-Jullian, and Jiaqi Ma, "The opencda open-source ecosystem for cooperative driving automation research," (2023), IEEE Transactions on Intelligent Vehicles, IEEE .

.

CHAPTER 1

Introduction

In recent years, automated driving technology has witnessed significant advancements, attracting considerable interest and substantial investments. as it has a great potential to fundamentally reshape transportation [3, 4, 5, 6, 7, 8]. Automated vehicles offer many benefits, including enhanced road safety [9], more efficient traffic flow [10], reduced fuel consumption [11], as well as environmental and economic advantages [12]. Furthermore, these vehicles can expand mobility options for those unable to drive, such as the elderly or disabled, thereby contributing to greater social equity [13].

Despite the impressive progress made in the field of automated driving, large-scale deployment in the real world is still far. A primary challenge stems from the limitations of single-vehicle perception systems. They usually suffer from occlusions and long-distance objects [6, 14], and such deficiencies can lead to catastrophic accidents [15]. Furthermore, automated vehicles are tasked with navigating interactions with the complicated transportation system, including human-operated vehicles, vulnerable road users, diverse roadside infrastructures, and intricate traffic regulations. These variables complicate the decision-making processes, making individual safety decisions particularly challenging [16, 17].

Cooperative Driving Automation (CDA) is emerging as a promising solution to these challenges. As defined by SAEJ3216 [18], CDA employs machine-to-machine communication to foster cooperation among various entities such as vehicles, pedestrians, and infrastructure equipped with advanced communication technologies. A critical application within CDA is cooperative perception, which uses Vehicle-to-Everything (V2X) communication to facilitate

information exchange between vehicles and other traffic participants. This ensures a comprehensive understanding of their surroundings. Combined with collaborative decision-making, CDA aids vehicles in navigating occlusions, broadening their sensing capabilities, and making sound safety decisions In this dissertation, I will undertake comprehensive research to facilitate the development of CDA, with a focus on simulation frameworks, and cooperative perception datasets and algorithms. My thesis is organized as follows:

1.1 Part I: Tools and Dataset for Cooperative Driving Automation

Motivation: While CDA holds the promise to significantly advance automated driving technology, it remains in its early stages of development. A primary challenge to its progression is the expensive cost and safety concerns associated with conducting field experiments. Unlike testing single-vehicle automation, which by itself is costly, CDA requires multiple automated vehicles to operate concurrently. This not only amplifies the expenses but also introduces heightened safety challenges. The most direct approach to mitigate these obstacles is to prototype and validate CDA algorithms in a simulated environment. Yet, a noticeable gap exists: there is no suitable simulation framework that is designed specifically for CDA, offering both vehicle cooperation and communication capabilities as well as the support to develop full-stack software modules of automated driving, including perception, localization, planning, and control. Consequently, progress in cooperative perception has also been slow, as there exist neither real-world datasets nor simulation platforms to generate synthetic data. To address these gaps, the initial part of my dissertation is dedicated to the development of a simulation framework designed to support the prototyping of CDA systems and the collection of benchmark datasets to facilitate the advancement of cooperative perception.

Chapter2 [7]: I introduce OpenCDA, a comprehensive simulation framework tailored for Cooperative Driving Automation. OpenCDA supports full-stack automated driving development, including the common self-driving modules composed of sensing, computation, and

actuation capabilities, and cooperative features as defined in SAE J3216 [18] (e.g., vehicular communication, information sharing, agreements seeking). Built upon these basic modules, OpenCDA supports a range of common cooperative driving applications, such as platooning, cooperative perception, and cooperative merge. Such capabilities not only streamline the CDA system development process but also lead to significant reductions in associated costs and development duration.

Chapter3 [6]: By leveraging OpenCDA, I further contribute OPV2V, the world's first large-scale, open-source simulated cooperative perception dataset. It contains over 70 interesting scenes, in which multiple automated vehicles equipped with multiple sensors will show up simultaneously to capture different views of the same scenes. This dataset is designed to serve as a standard benchmark, facilitating advancements in the realm of cooperative perception algorithms.

1.2 Part II: Algorithms for Cooperative Perception

Motivation: Leveraging the capabilities of the OPV2V dataset, I've oriented my research towards specialized algorithm design for cooperative perception. The realm of cooperative perception, in contrast to single-vehicle perception, introduces a set of distinct challenges:

- *Collaboration Efficiency:* Collaborative perception aims to enhance detection accuracy by sharing complementary information amongst various agents, whether they be vehicles or infrastructure. Yet, practical scenarios often offer limited communication bandwidth. Thus, a key challenge emerges: optimizing perception accuracy while being constrained by data transfer limits.

- *Sensor Heterogeneity:* Not all participating agents in the network will have identical sensors; some may even have the same sensors but with varied placements or orientations, for instance, the LiDAR installed in infrastructure usually has a much higher

viewpoint. Such heterogeneity can complicate the fusion of perception information from diverse agents.

- *Localization and Synchronization Issues:* Automated vehicles primarily rely on GPS for localization, which usually has unavoidable errors. Ensuring that data from different vehicles aligns correctly becomes a challenge, especially with the potential for GPS inaccuracies. Moreover, transmission delays inherent in V2X communication can introduce data misalignment due to asynchrony.

- *Economic Solution for Scalability:* LiDAR is naturally the most suitable sensor for cooperative perception, given its capability to provide direct and accurate 3D geometric information. However, LiDAR systems tend to be costly. Exploring the use of cameras, which are more economical and widely available, becomes crucial for a scalable and cost-effective implementation of cooperative perception in real-world scenarios.

The second part of my thesis is focused on solving the domain challenges in cooperative perception by designing two state-of-the-art algorithms.

Chapter4 [19]: In this chapter, I propose a robust cooperative 3D LiDAR detection framework that leverages vision transformers to solve the sensor heterogeneity, localization error, and asynchronization issues while keeping low bandwidth requirements. This framework employs a novel Vision Transformer architecture composed of alternating layers of delay-aware positional embedding, heterogeneous multi-agent self-attention, and multi-scale window self-attention. These layers are designed to capture both inter-agent interactions and individual spatial relationships between agents to solve the abovementioned issues.

Chapter5 [20]: Subsequently, I introduce CoBEVT, the world's first generic multi-agent, multi-camera perception framework capable of cooperatively generating Bird-eye-view (BEV) map predictions. Leveraging only economical RGB cameras, this novel approach is both cost-effective and innovative. To optimize the fusion of multi-view and multi-agent camera

features within a Transformer architecture, I propose the Fused Axial Attention Module (FAX). This module adeptly captures both local and global spatial interactions across varying views and agents. The efficacy of both the V2X-ViT and CoBEVT frameworks was verified using the simulated dataset established in the first part of my research.

1.3 Part III: Towards Real-world Deployment for Cooperative Perception

Motivation: Building upon the foundational models and benefits demonstrated in the earlier parts of this dissertation, the final section shifts its focus toward the practicalities of real-world deployment. This includes tackling realistic challenges that may arise in real-world scenarios, generating a real-world cooperative perception dataset, and examining training strategies to minimize the costs associated with collecting real-world data.

Chapter6, Chapter7 [21, 22]: All of the previous literature assume that every collaborator utilizes the same model with identical parameters and architecture. However, this assumption is hard to satisfy in practice, particularly in automated driving. Different companies will have distinct models, and even for the same company, the model version may differ for different users. Without adequately handling the inconsistency, the shared sensory information can have a large domain gap, and the advantage brought by cooperative perception will be diminished rapidly. Both Chapter 6 and 7 are target to solve this challenge, but they focus on distinct fusion strategies. In chapter 6, I propose a confidence calibrator that can eliminate the prediction confidence score bias caused by the model distinctions and integrate it with late fusion strategy. In chapter 7, I first conduct detailed analysis on the feature domain gap from different models in the intermediate fusion strategy and then present the Multi-agent Perception Domain Adaption framework (MPDA) to bridge such domain gap.

Chapter8 [23]: We present V2V4Real, the first large-scale real-world multimodal dataset for V2V perception. The data is collected by two vehicles equipped with multi-modal sensors driving together through diverse scenarios. Our V2V4Real dataset covers a driving area of 410 km, comprising 20K LiDAR frames, 40K RGB frames, 240K annotated 3D bounding boxes for 5 classes, and HDMaps that cover all the driving routes. V2V4Real introduces three perception tasks, including cooperative 3D object detection, cooperative 3D object tracking, and Sim2Real domain adaptation for cooperative perception. We provide comprehensive benchmarks of recent cooperative perception algorithms on three tasks. This dataset will bridge gap of lack of existence of real-world cooperative perception dataset.

Chapter9 As Chapter8 indicates, collecting and labeling real-world cooperative perception dataset is both costing and time-consuming. In this chapter, I explore how to utilize large-scale simulated data and mini-scale labeled real-world data to achieve comparable performance with models trained on large-scale labeled real-world data. I conducted extensive experiments to explore several categories of training strategies using the simulated OPV2V dataset and the real-world V2V4Real dataset, employing state-of-the-art models. More importantly, I designed a domain-tailored plug-in training module named Homogeneous Training Augmenter (HTA), comprising a homogeneous aligner coupled with a data augmenter specially tailored for cooperative perception. The experiments demonstrate that the best training strategy integrated with HTA can even beat some models trained on the large-scale real-world data.

CHAPTER 2

OpenCDA: An Open Cooperative Driving Automation Framework Integrated with Co-Simulation

Although Cooperative Driving Automation (CDA) has attracted considerable attention in recent years, there remain numerous open challenges in this field. The gap between existing simulation platforms that mainly concentrate on single-vehicle intelligence and CDA development is one of the critical barriers, as it inhibits researchers from validating and comparing different CDA algorithms conveniently. To this end, we propose OpenCDA, a generalized framework and tool for developing and testing CDA systems in simulation. Specifically, OpenCDA is composed of three major components: a co-simulation platform with simulators of different purposes and resolutions, a full-stack prototype cooperative driving system, and a scenario manager. Through the interactions of these three components, our framework offers a straightforward way for researchers to test different CDA algorithms at both levels of traffic and individual autonomy. More importantly, OpenCDA is highly modularized and installed with benchmark algorithms and test cases. Users can conveniently replace any default module with customized algorithms and use other default modules of the CDA platform to perform evaluations of the effectiveness of new functionalities in enhancing the overall CDA performance. An example of platooning implementation is used to illustrate the framework's capability for CDA research. The codes of OpenCDA are available at the UCLA Mobility Lab GitHub page.

2.1 INTRODUCTION

By leveraging cutting-edge technologies to circumvent traditional infrastructure enhancement constraints, Intelligent Transportation Systems (ITS) are reshaping transportation and have demonstrated a tremendous potential to boost the transportation system management, operations, safety, and efficiency. One of the essential sub-fields in ITS is Cooperative Driving Automation (CDA), which is defined in SAE J3216 [18] and refers to vehicle-highway automation that uses Machine-to-Machine communication to enable cooperation among two or more entities (e.g., vehicles, pedestrians, infrastructure components) with capable communication technologies. By enabling the status-sharing, intent-sharing, and maneuver cooperation between entities, the traffic efficiency, energy consumption, and safety of the driving can be significantly improved [24]. Developed by the Federal Highway Administration (FHWA), the CARMA Program [25] is a leading research program on CDA, leveraging emerging capabilities in automation and cooperation to advance transportation systems management and operations (TSMO) strategies.

Although CDA has been an active field in recent years, it is still in its infancy. One of the major barriers to the development of CDA is the high cost and potential safety issues to conduct field experiments as they usually require multiple expensive connected automated vehicles (CAVs) and extra-large testing space [26]. One approach to facilitating experimental research with minimum cost is to prototype and validate the CDA algorithms in a simulated environment. However, existing simulation platforms featured with full-stack software development of autonomous driving provided limited supports for CDA capabilities. As far as we know, there is no existing open-source (or commercial) tool dedicated to CDA by featuring both traffic and vehicles with full CDA vehicle software pipeline. As a result, it becomes very challenging to find an easy and flexible way for researchers to deploy, validate and compare the impact of different CDA algorithms on the dynamic driving tasks of CAVs in simulation. Most of the research uses ad-hoc simulation capabilities with different qualities,

making the algorithmic and functional performance not comparable between studies.

To overcome such challenges, we introduce OpenCDA, a generalized open-source framework integrated with co-simulation for CDA research. OpenCDA provides a full-stack CDA software in simulation that contains the common self-driving modules composed of sensing, computation, and actuation capabilities, and cooperative features as defined in SAE J3216 (e.g., vehicular communication, information sharing, agreements seeking). OpenCDA is developed purely in Python [27] for fast prototyping. Built upon these basic modules, OpenCDA supports a range of common cooperative driving applications, such as platooning, cooperative perception, cooperative merge, and speed harmonization. More importantly, OpenCDA offers a scenario database that includes various standard scenarios for testing of different cooperative driving applications as benchmarks. Users can easily replace any default modules in OpenCDA with their designs and test them in the supplied scenarios. If users desire to produce their scenarios, our framework also provides simple APIs to support such customization. We select CARLA [8] and SUMO [28] to render the environment, simulate the vehicle dynamics, and generate the traffic flow. Since our framework is designed with high flexibility, it can also be extended to integrate additional simulators, such as communication simulators (ns-3 [29]) and vehicle dynamics simulators (e.g., CARSim [30]).

The key features of OpenCDA can be summarized as **IFMBC**:

- **I**ntegration: OpenCDA integrates CARLA and SUMO together for realistic scene rendering, vehicle modeling and traffic simulation.
- **F**ull-stack prototype CDA Platform in Simulation: OpenCDA provides a simple prototype automated driving and cooperative driving platform, all in Python, that contains perception, localization, planning, control, and V2X communication modules.
- **M**odularity: OpenCDA is highly modularized, enabling users to conveniently replace any default algorithms or protocols with their own customzied design.

- **B**enchmark: OpenCDA offers benchmark testing scenarios, state-of-the-art benchmark algorithms for all modules, benchmark testing road maps, and benchmark evaluation metrics.

- **C**onnectivity and **C**ooperation: OpenCDA supports various levels and categories of cooperation between CAVs in simulation. This differentiates OpenCDA from other single vehicle simulation tools.

The paper is organized as follows. Section 2 will review existing frameworks aiming to support cooperative driving automation and different cooperative driving applications. In section 3, we will describe the overall architecture of OpenCDA and reveal the details of each major component. In section 4, we will showcase a concrete example of how platooning, one of the most important applications in CDA, is implemented under our framework. Afterward, a case study for platooning scenario testings will be introduced. Section 5 will present the experiment results using our default guidance algorithm and compare it with customized algorithms to prove the effectiveness of our framework.

2.2 Related Work

In the past decades, various CDA applications have emerged and imposed significant impacts on ITS. One of the representative applications is the Cooperative Adaptive Cruise Control(CACC), which has been studied extensively, such as [31, 32]. CAVs can form stable strings with short following gaps by utilizing CACC, improving the stability, safety, comfort, and traffic performance in terms of throughput and delay [33]. For freeway traffic, the cooperative merge has been a popular topic as it allows the speed coordination between mainline vehicles and merging vehicles to create qualified gaps for safe merging [34]. Speed harmonization also attracts much attention due to its capability of gradually decreasing upstream traffic speed in a heavily congested area to reduce the stop-and-go traffic and prevent

congestion formation [35, 36, 37, 38]. Moreover, there are also some CDA applications for intersection control, including vehicular trajectory control [39], traffic signal control [40], and joint control of traffic signal and vehicular trajectories [41].

Although extensive research has been carried out in this field, there are few open-sourced simulation platforms for CDA. Segata *et al.* [42] proposes an extension for Veins [43] to provide the basic platooning capability. Recently, SUMO [28] integrates the Simpla package for basic platoon formations. Wu *et al.* [44] also present a platform for integrating the traffic simulation and reinforcement learning controllers. However, these platforms only stay at the level of traffic analysis and fail to support full-stack software development and testing for CDA, including perception, planning, decision-making, control, and communication.

The FHWA CARMA Program [25] has developed software platforms for vehicle and infrastructure and tools for full-scale vehicle software simulation and testing. In collaboration with the CARMA Program, OpenCDA, as an open-source project, makes a unique contribution from the perspective of early-stage development and testing using simulation, enabling users can conveniently conduct both task-specific evaluation (e.g. object detection accuracy) and pipeline-level assessment (e.g. traffic safety) on their customized algorithms.

2.3 Overview of OpenCDA

OpenCDA is a generalized framework integrated with co-simulation for intelligent and dynamic cooperative driving. It supports various cooperation between automated vehicles and provides benchmarking scenario database and CDA algorithms. As Fig. 2.1 depicts, OpenCDA is composed of three major components – the simulation tools, cooperative driving automation system built in Python, and scenario manager.

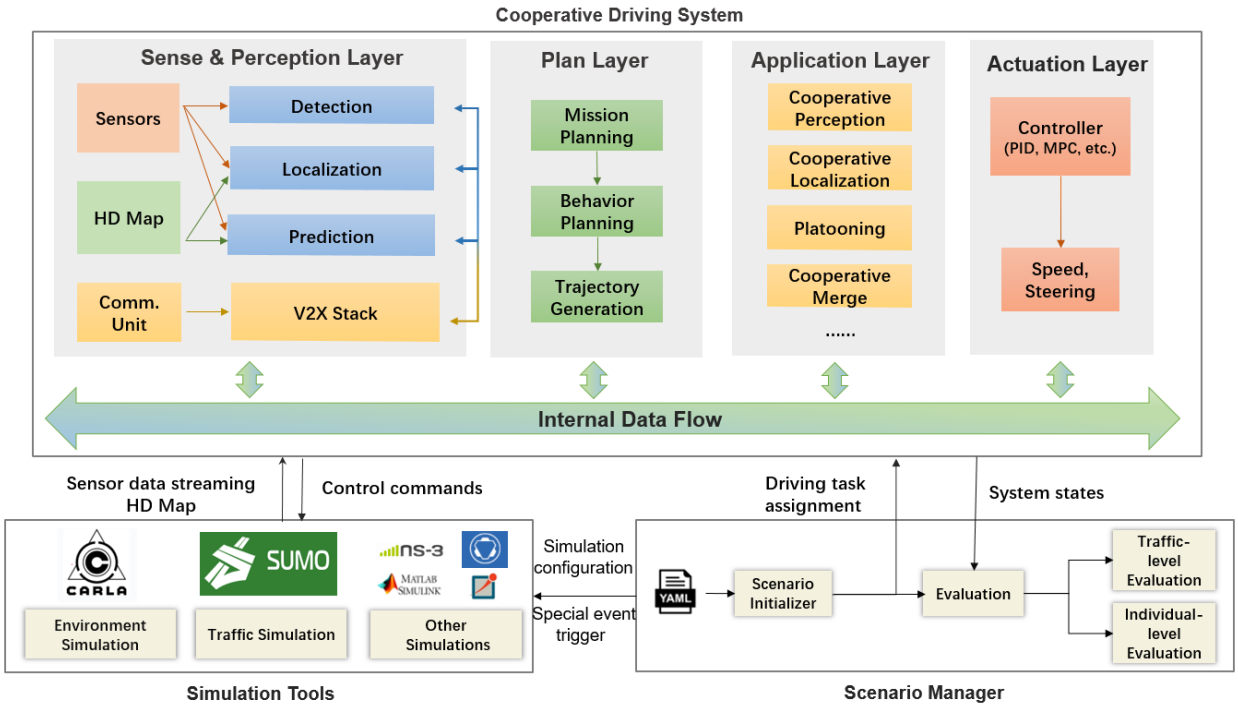


Figure 2.1: The overall architecture design of OpenCDA.The full-stack software of the designed cooperative driving system interacts with simulation tools to test the system performance in provided scenarios

2.3.1 Simulation Tools

CARLA [8] is selected as one of the simulation tools in OpenCDA for automated driving simulation. CARLA is a free, open-source automated driving simulator that aims to accelerate the development of new automated driving technologies. It utilizes Unreal Engine [45] to produce high-quality scene rendering, realistic physics, and basic sensor modeling. The CARLA platform defines a versatile simulation API that users and developers can control over all the elements of the simulation from sensor placement to prototyping and testing the perception, planning, and control algorithms. A key feature of CARLA is its scalable architecture, following a server-multi-client approach to allow for the distribution of computation into multiple nodes. The server will keep updating the physics of the environment, and the client-side will be controlled by users through the CARLA API. Our cooperative driving system is embedded with CARLA API to perform cooperative dynamic driving tasks and evaluate the vehicle performance under the individual autonomy level.

However, CARLA lacks the manageability of large volumes of traffic and fails to represent realistic traffic behavior, thus not ideal for creating a complex traffic environment for CDA testing [46]. Additionally, CDA's potential in improving overall traffic system performance is also of interest. Therefore, SUMO [28], an open-source traffic/driver behavior simulator, is involved in the framework because of its capability of handling large-scale and realistic traffic flows. SUMO has dynamic modeling for each vehicle and allows users to quickly construct customized traffic scenarios through the TraCI (Traffic Control Interface) API. Note that even though CARLA provides a traffic manager module for generating background traffic, they are based on simplistic behavior rules, which cannot represent real driver behavior.

Further, SUMO can generate traffic using different well-accepted driver models (e.g., Intelligent Driver Model [47]), and it is more convenient to use SUMO to take in naturalistic trajectory data (e.g., NGSIM [48]) and use them directly as the surrounding environment for CDA testing. Since CARLA has developed a co-simulation feature with SUMO, we provide

the option for researchers to test their algorithms and protocols solely using CARLA, SUMO, or employing them together. When the co-simulation is activated, SUMO will control the traffic and transform the background human-driven vehicles into the CARLA server, and the CAVs controlled by CARLA will react with the traffic to finish their driving tasks. By distributing the tasks to both CARLA and SUMO, the evaluation of designed algorithms or protocols can be processed at both individual level and traffic level.

Note that we do not recommend using the full co-simulation in OpenCDA in all testing tasks. The users need to understand the evaluation needs (e.g., vehicular or traffic behavior) and then select corresponding tools. For example, if only traffic performance is to be understood, there is no point to conduct a fully automated driving simulation and investigate the detailed level of questions such as how sensor outputs and fusion impact traffic performance, because they are not at the same level of analysis. Traffic performance is mostly derived directly from driver/vehicle behavior (as a result of internal mechanisms or algorithms). To this end, in OpenCDA, our benchmark algorithms, to be discussed in the next section, are implemented in both SUMO and CARLA (in a consistent manner, but with some differences due to the fundamental differences between the two simulators in controlling vehicles), so analysis at any level is possible and consistent.

The OpenCDA framework can also be flexibly enhanced with additional tools, such as ns-3 or other customized models for wireless communication. However, we do not consider extremely complex integration of different tools as necessary for the simple reason that no models can fully replicate the real world and models should be built only to meet the testing needs of specific purposes. For example, when evaluating if a certain level of communication packet drops impact traffic stability, using a full ns-3 tool in the simulation loop will not only significantly slow down the simulation but also does not present many benefits as compared to using only simple Monte Carlo simulation [49].

2.3.2 Cooperative Driving System

OpenCDA encapsulates the cooperative driving system with CARLA and SUMO via simple API to operate cooperative driving tasks. Sensors mounted on CAVs in CARLA will collect raw sensing information from the simulation environment and proceed to the sensing layer of the system. The received information is then processed by the perception module to perceive the operational environment, utilizing the plan layer to deliver a series of actions, and finally, spawn the control commands through the actuation layer. The actuation will be sent back to CARLA actors to execute the movement at each simulation time step. It is interesting to note that such architecture is also suitable for single-vehicle intelligence development when there is no cooperation needed. This means that the OpenCDA tool can simulate mixed traffic of human-driven, connectivity, and automation.

The cooperation between automated vehicles is activated at the application layer. In this layer, each CAV will exchange status information (e.g., vehicle position, signal phasing, and timing), intent information (e.g., perceived sensing context, planned vehicle trajectory) through the V2X stack, and seek agreement on a plan (e.g., forming a platoon). Based on different agreements, there will be corresponding protocols that potentially modify the default settings of the layers. For instance, when the cooperative perception application is launched, each CAV doesn't solely utilize its own raw sensing information to locate dynamic objects but also retrieves and fuses others' sensing information to achieve multi-modal, cooperative object detection.

One key feature of the OpenCDA framework is modularity. All layers mentioned above come with default algorithms or protocols, and users can replace the default ones with their customization without influencing others parts by just applying one line of code. We consider this as a desirable feature because researchers can utilize the default modules and algorithms to evaluate the ultimate performance of the entire CDA system and it is also possible for different groups of researchers to compare the algorithms under the same framework. Ad-

ditionally, the default algorithms in OpenCDA applications, such as cooperative platooning and merge, are also state-of-the-art algorithms that are qualified to serve as benchmark algorithms. Researchers can compare their algorithms with the OpenCDA benchmarks to demonstrate the capability and enhancement of the new algorithms.

2.3.3 Scenario Manager

The scenario manager in OpenCDA contains four parts: the scenario configuration file, scenario initializer, special event trigger, and evaluation functions.

A scenario is a description of how the view of the world alters to time. In the context of cooperative driving, it encompasses the information of the static elements of the world (e.g., road topology, surrounding buildings, static objects on the road surface), and dynamic elements such as the traffic flow, traffic signal state, and weather. In OpenCDA, the static elements of a scenario are defined by the default maps in CARLA map library or customized maps built by xdor [50] and fbx [51] file. The dynamic elements are controlled by a yaml [52] file. In the yaml file, users can define the traffic flow for each lane generated by SUMO, including traffic volume and desire speed. If background traffic produced by CARLA is also introduced, then the number and spawn positions of these vehicles and the CARLA traffic manager's settings also need to be recorded. As mentioned before, our framework comes with an existing scenario database that stores predefined scenario testings, but users are welcome to contribute their customized testings to the database.

After the yaml file is created, a configuration loader will load the file into a Python dictionary. This dictionary will guide the simulation environment construction and determine the major driving tasks for the target CAVs. A driving task composed of the starting locations and destinations of the CAVs, and the intermediate locations to reach.

When the CAVs are executing driving tasks, special events may be triggered. A good example of such events is that a human-driven vehicle in front of a platoon suddenly de-

celerates or stops. These special events are normally triggered by certain time-step or the positions of CAVs to test the performance of the cooperative driving system in the corner cases.

A driving task is regarded as finished when the CAV arrives at the destination. Then the evaluation is carried out to measure the performance of the whole driving period at an individual level with CARLA and at traffic level with SUMO.

2.3.4 Software Class Design and Logic Flow

To better demonstrate how the interaction of the three major components of OpenCDA are realized, in this section, we will describe the major software class components and the procedure of simulation information transferring between these components by utilizing an example CDA application – vehicle platooning.

As Fig.2.2 depicts, we apply hierarchical class management to control the simulation neatly. The most fundamental class is called `VehicleManager`, which contains the full-stack CDA and Automated Driving System(ADS) benchmark software for a single CAV. The class member `PerceptionManager` and `LocalizationManager` are responsible for perceiving the surrounding environment and localize the ego vehicle. The `BehaviorAgent` plans the driving behavior (e.g. car following, overtaking, lane changing behavior) for the single CAV, and the attribute `LocalPlanner` in `BehaviorAgent` will generate the trajectory using cubic spline interpolation and basic vehicle kinematics:

$$y_t = \alpha_0 + \alpha_1 x_t + \alpha_2 x_t^2 + \alpha_3 x_t^3 \tag{2.1}$$

$$a_t = \begin{cases} \min(\frac{v_{target} - v_t}{\Delta t}, a^1), & \text{if } v_{target} \geq v_t \\ \max(\frac{v_{target} - v_t}{\Delta t}, a^2), & \text{otherwise} \end{cases} \tag{2.2}$$

$$x_t = v_{t-1}\Delta t + \frac{a_{t-1}\Delta t^2}{2} \tag{2.3}$$

$$v_t = v_{t-1} + a_{t-1}\Delta t \tag{2.4}$$

where x_t, y_t are the planned x and y coordinates of the vehicle at time step t, $\alpha_0, \alpha_1, \alpha_2$ are the coefficients of cubic polynomial, a_t is the desired acceleration at the time step t, a^1, a^2 are the comfort-related acceleration and deceleration, Δt is the time resolution, i.e., simulation step , v_{target}, v_t are the final target speed and desired speed at time step t. This produced trajectory will be delivered to `ControlManager` to generate the throttle, brake, and steering control commands. The `V2XManager` will send and receive the packets (currently regarded as lossless transfer) generated by the components mentioned above to other CAVs for cooperative driving applications.

Fig. 2.3 shows the logic flow of the simulation during run time. To run a scenario test, the users are required to create a yaml file based on the template that OpenCDA provides to configure the settings of CARLA server (e.g., synchronous mode versus asynchronous mode), the specifications of the traffic flow (e.g., the number of human drive vehicles, spawn positions), and the parameters of each Connected Automated Vehicle (e.g., sensor parameters, detection model selection, target speed). Subsequently, the `ScenarionManager` will load the configuration file, retrieve the necessary parameters, and deliver them to the CARLA server to settle the simulation environment, generate traffic flow, and create the `VehicleManager` for each CAV.

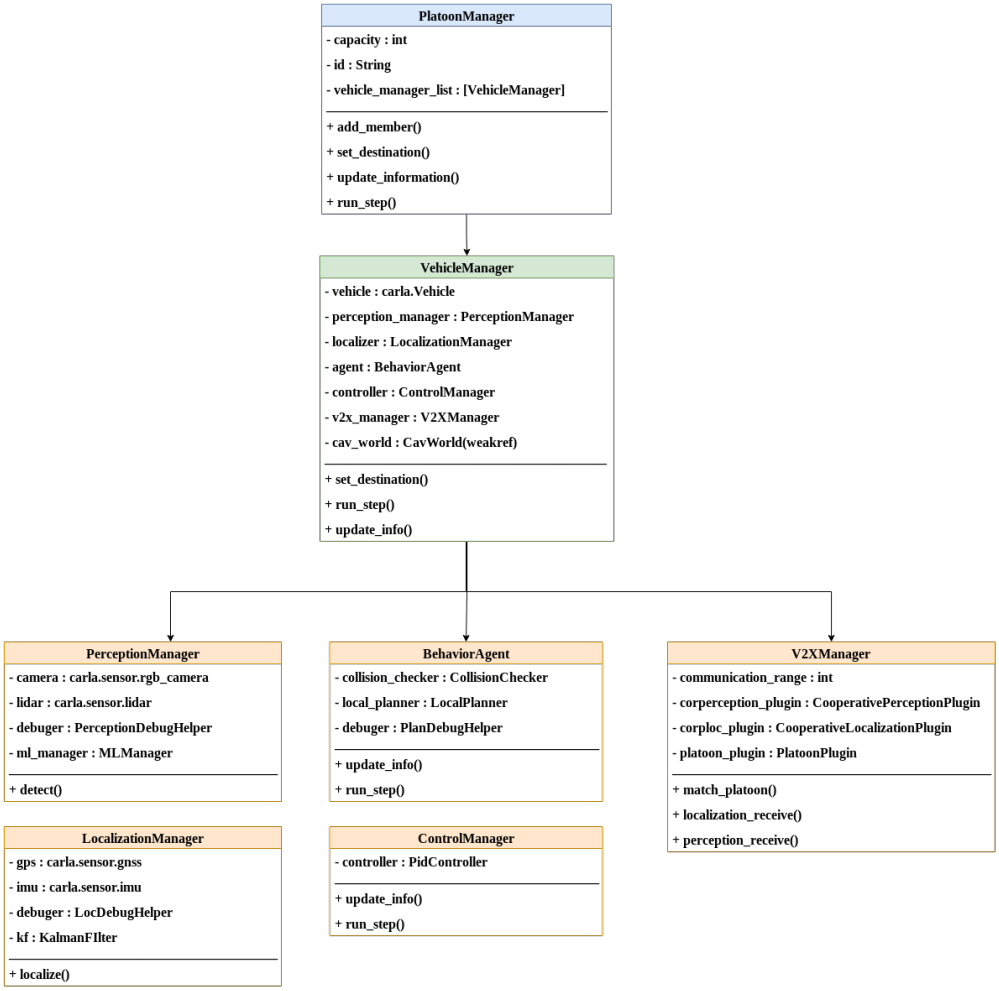


Figure 2.2: Simplified class diagram of platooning. Note we only exhibits partial design here.

After the server updates the information given by `ScenarionManager`, the sensors mounted at each CAV will collect the surrounding environment as well as the ego vehicle information (e.g., 3D LiDAR points, GNSS data) and share those through `V2XManager`. If the upstream cooperative application is activated, `CoopPerceptionManager` and `CoopLocalizationManager` will be utilized to fuse all contexts obtained from other CAVs for object detection and localization. Otherwise, the vehicle will switch to the default `PerceptionManager` and `LocalizationManager`, which do not employ shared data. The processed sensing information (i.e., object 3D pose, ego position) is delivered to the downstream modules for planning. Similarly, the CAV will select cooperative strategies to make decisions if corresponding applications are activated; otherwise, the origin `BehaviorAgent` and `TrajecotryPlanner` will plan the behavior and generate a smooth trajectory, which is passed to the `ControlManager` to output the final control commands. The CARLA server will apply these commands on the corresponding vehicles, execute a single simulation step, and return the updated information to the `VehicleManager` for the next round of simulation.

It is obvious that the design of the logic flow enhances the flexibility and modularity of OpenCDA as users are capable of choosing the level of cooperation by just modifying the activation indicator. When the simulation is terminated, the embedded evaluation toolboxes will assess the driving performance. We provide default performance measurement for various modules, including perception (e.g., mean average precision of the 3D bounding box detection), localization(e.g. error between estimated and true ego-position), planning (e.g., the smoothness of the planned trajectory), control(e.g, tracking error) and safety (e.g., hazard frequency). If users demand any evaluation measurements that are out of the default scope, they can build customized metrics following the predefined template, which is another key advantage of OpenCDA.

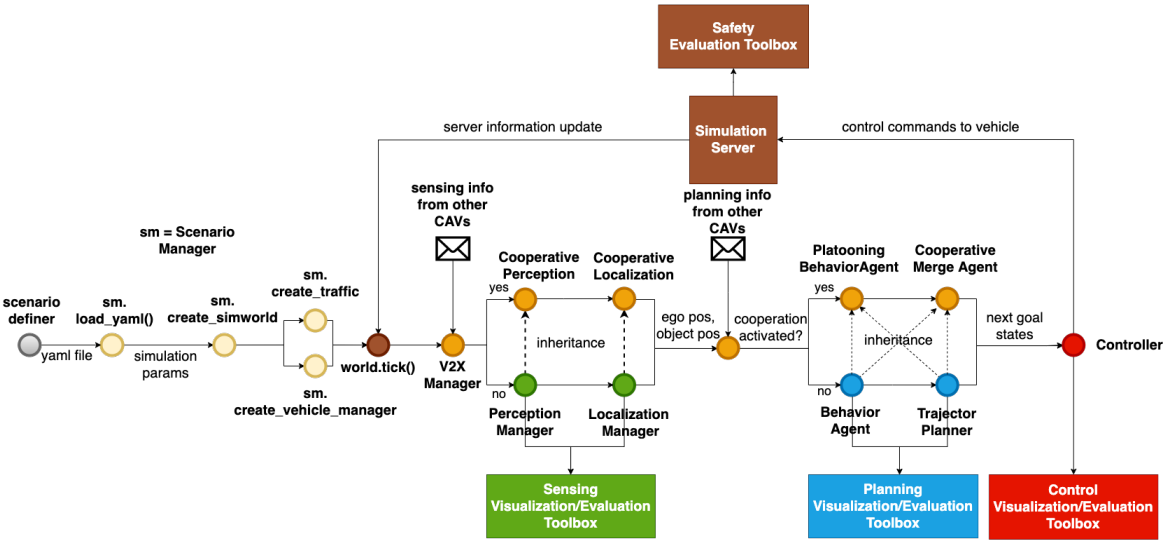


Figure 2.3: Logic flow of the simulation process in OpenCDA.

2.4 Experiment Setup and Evaluation Measurement

To prove the effectiveness of OpenCDA, in this section we continue with an example of vehicle platooning. Our platooning benchmark includes four parts – the rule-based platooning protocol and algorithms, a customized map with a long freeway basic and merge segment as Fig. 2.5 presents, several designed testing scenarios, and evaluation metrics. Note that we use such customized map because it leaves enough distance for vehicles to reach high target speed and perform various cooperative maneuvers. For all the experiments conducted, we set the simulation time step as 0.05 second, which means the update frequency of the CARLA server and SUMO is 20Hz.

2.4.1 Platooning Protocol Design

As Fig. 2.2 shows, in the platooning application, all CAVs in the same platoon will be managed by the `PlatoonManager` through a pre-defined protocol. Fig. 2.4 displays the default platooning protocol in OpenCDA. Overall, the driving task in a platoon can be divided into

different sub-tasks, and platoon members have various driving modes based on the current platooning status.

When the platooning application is activated, the leading vehicle of the existing platoon will keep listening to the joining requests from CAVs through `V2XManager`. If no such requests are received, the whole platoon will keep moving forward steadily while the leading vehicle will stay in the leader drive mode, in which the vehicle shares a similar behavior pattern with CAVs outside a platoon except overtaking is forbidden. Meanwhile, if the cooperative perception application is also activated, each platoon member will also share its raw sensing information (e.g., camera RGB images, 3d lidar points) and processed sensing information (e.g., detected objects, calibrated vehicle position) retrieved from `PerceptionManager` with the leading vehicle for a better perception.

When there is no joining request, the following vehicles in the platoon will enter the maintaining mode, in which the driving task is defined as adjusting the velocity smoothly to keep a constant inter-vehicular time gap to the preceding members. To accomplish such tasks, the members need to receive some of the preceding vehicle's trajectory (e.g., leader, immediate predecessor) from `V2XManager` to assist the `LocalPlanner` creating the trajectories, as shown below.

$$pos_j^t = \frac{pos_{j-1}^t - L_{j-1} + pos_j^{t-\Delta t} \times gap/\Delta t}{1 + gap/\Delta t} \tag{2.5}$$

$$v_j^t = \frac{||pos_j^t - pos_j^{t-\Delta t}||}{\Delta t} \tag{2.6}$$

where pos_j^t, pos_{j-1}^t are the position of vehicle j and its proceeding vehicle $j-1$ at time step t, L_{j-1} is the length of vehicle $j-1$, Δt is the time resolution i.e, simulation step, gap is the desired inter-vehicular time gap, and v_j^t is the desired speed of the jth vehicle at time step t. In this example, the platooning algorithm only considers the planned trajectory of

the immediate preceding vehicle.

If the platoon receives a joining request, the leading vehicle will exchange destination, current position, and planned routes with the requesting CAV to decide whether a feasible joining can be operated. If the request is rejected, the single CAV will keep searching and stay in single-vehicle driver mode. Otherwise, the `PlatoonManager` will choose the best meeting position for the merging vehicle to join depending on the internal and surrounding information, and if needed, certain platoon members will adjust their speed to open a gap for joining. Then the merging vehicle can move to the meeting point and finish the joining maneuver.

2.4.2 Platooning Scenario Testing Design

Fig. 2.5 shows a snippet of the platooning co-simulation testing using the customized benchmark map of a basic freeway merge segment included in OpenCDA. This map is formed by a 2800 meters two-lane freeway for the mainstream traffic and a single-lane on-ramp to allow the merging vehicles to enter the freeway. In this section, we will exhibit two different platooning testing scenarios from our database in this benchmark map. Note that all tests are operated within perception and localization algorithms. We apply yolov5 [53] for object detection and utilize GNSS/IMU fusion algorithm similar with [54, 55] for localization.

2.4.2.1 Single Lane Platooning

As Fig. 2.6(a) describes, there is a five-vehicle platoon that keeps driving in the same lane in this scenario. The objective is to test the platoon's stability, which is indicated by the degree of amplified oscillations when the leading vehicle changes speed dramatically. To meet such a purpose, the platoon leader will follow a given speed profile to accelerate and decelerate frequently to identify whether the following members are capable of maintaining desired inter-vehicular time gap and dampen the speed oscillation. The OpenCDA provides

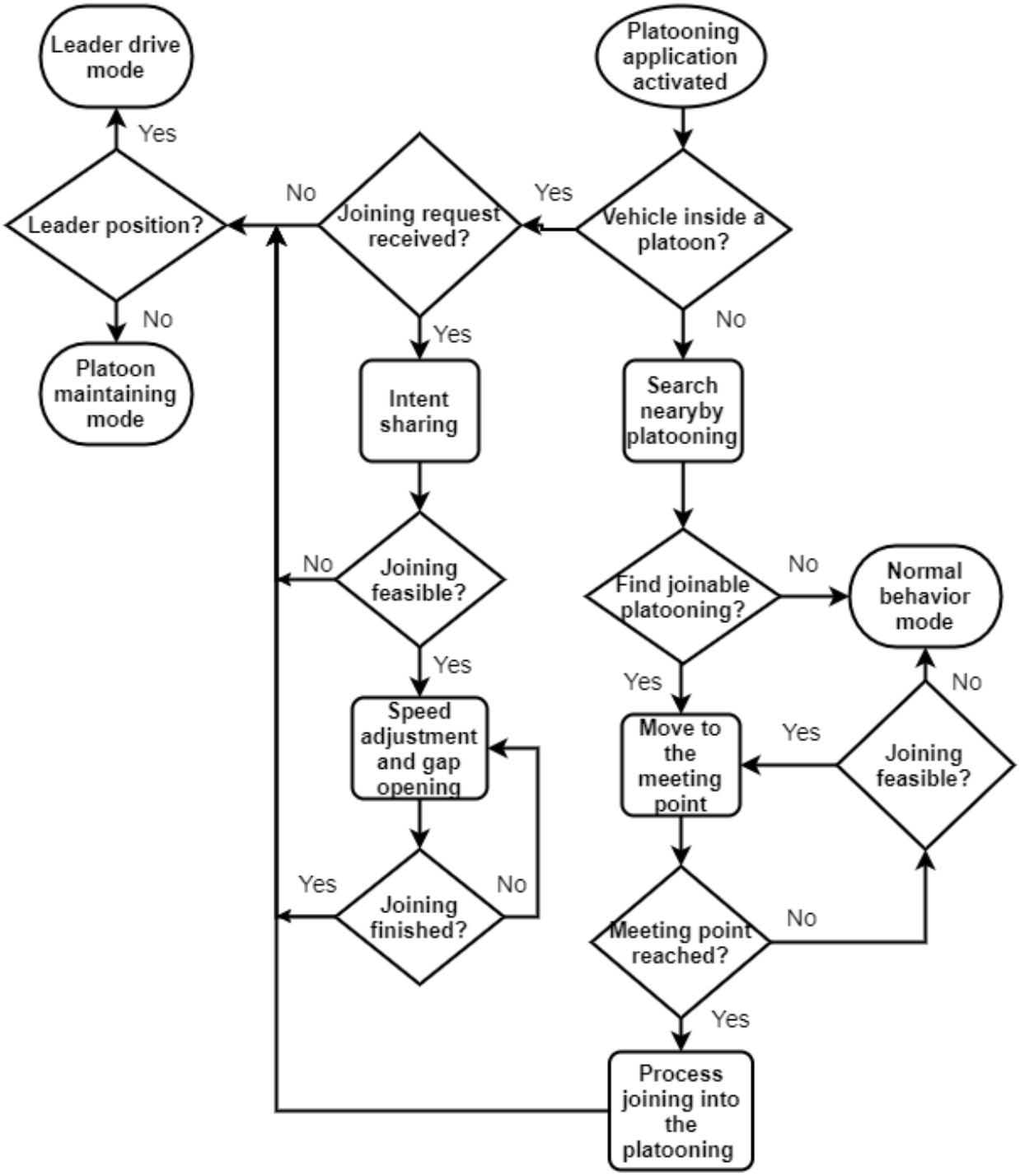


Figure 2.4: Logic Flow of Platooning Protocol.

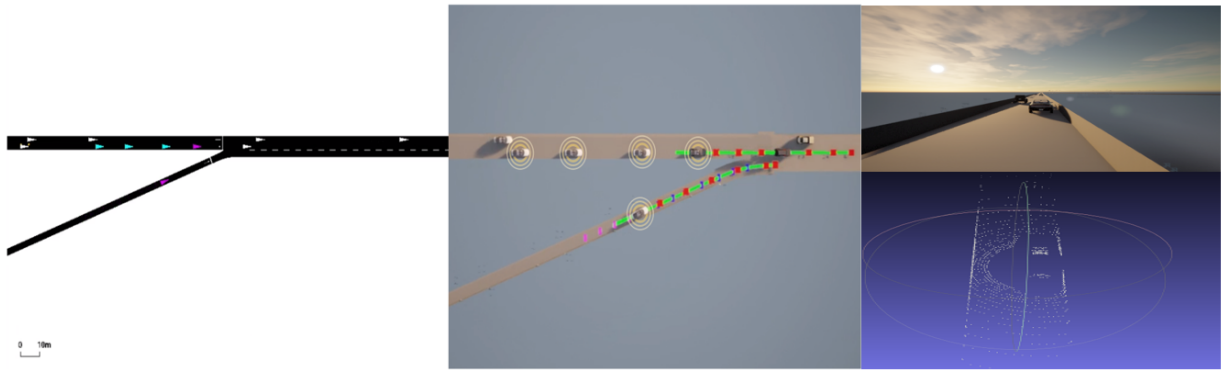


Figure 2.5: A snippet of platooning scenario testing under co-simulation setting. From left to right: Sample simulation snippet in SUMO, the corresponding view in CARLA where the green lines and red dots represent planned trajectory path and points respectively, and the RGB image with 3D lidar points together collected from the sensors mounted at the CAV.

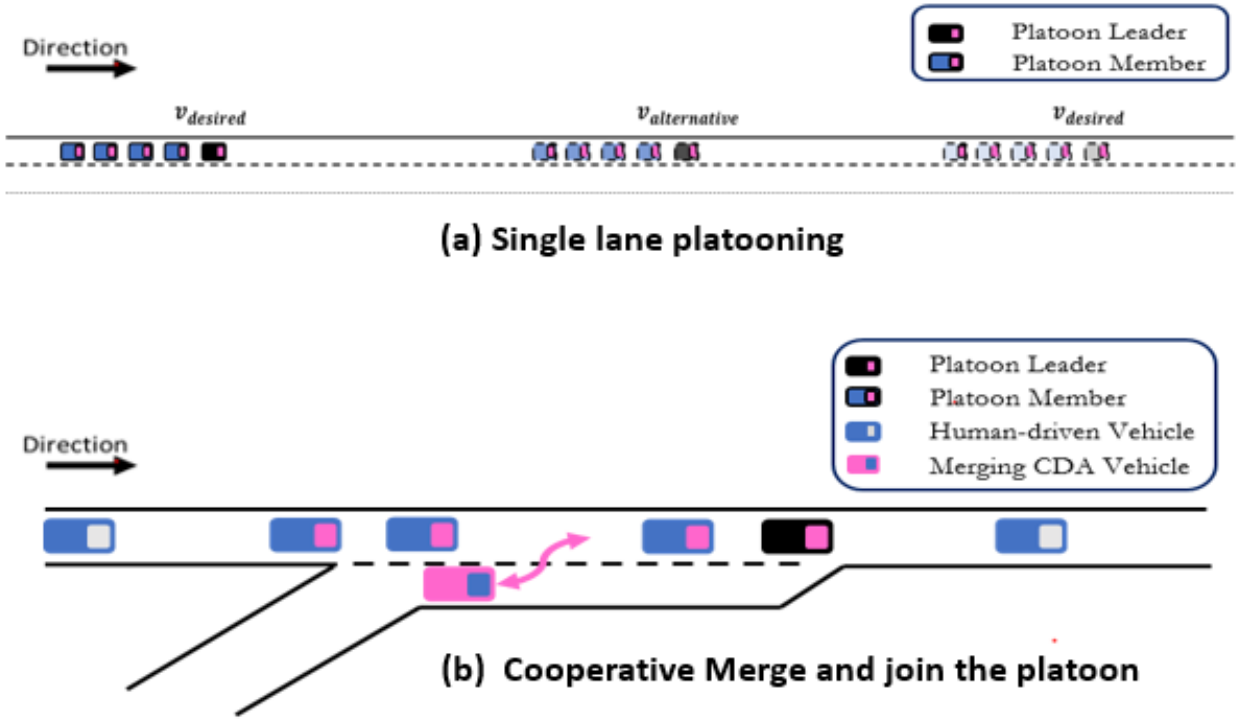


Figure 2.6: Two different platooning scenario testings.

benchmarking testing scenarios of front vehicle trajectories, e.g., two types of cycles of testing with distinct speed profiles. Users can also use their own scenarios for specific purposes by using the example format.

- In the first cycle, the platoon leader will follow a synthetic speed trajectory. It drives at 25 m/s for 20 seconds, then accelerates until reaching the target speed of 30 m/s. The platoon leader will keep this speed for 20 seconds, then decelerate to reach the initial speed of 25 m/s, and keep this speed for 20 seconds. There is no traffic flow generated as we aim to sorely evaluate the platooning protocol in this cycle. Furthermore, the aggressiveness of the acceleration or deceleration and speed maintaining duration can be easily modified to divergent levels, and here we just showcase a single instance.
- In the second cycle, we placed a human-driven vehicle in front of the platoon. This human-driven vehicle will follow representative speed profiles extracted from NGSIM data, which is collected from real-world experiments as Fig. 2.7 displays. The leader demands to have a decent car following behavior, and the platoon needs to remain stable while the human-driven vehicle radically adjusts the speed. In such a way, both the car following behaviors of the platoon leader and the platoon followers will be validated.

2.4.2.2 Cooperative Platoon Joining from Other Lanes

As shown in Fig. 2.6(b), the mainline has a high-speed traffic flow mixed with human-driven vehicles managed by SUMO and CAVs controlled by CARLA. When the single CAV is near the merging area, it will communicate with the mainline platoon and make a request to join. Once they achieve an agreement, the single CAV has to finish the merge and join the platoon simultaneously before the acceleration lane ends. The leader will decide the best merging position and command certain platoon members to create a gap for the new member.

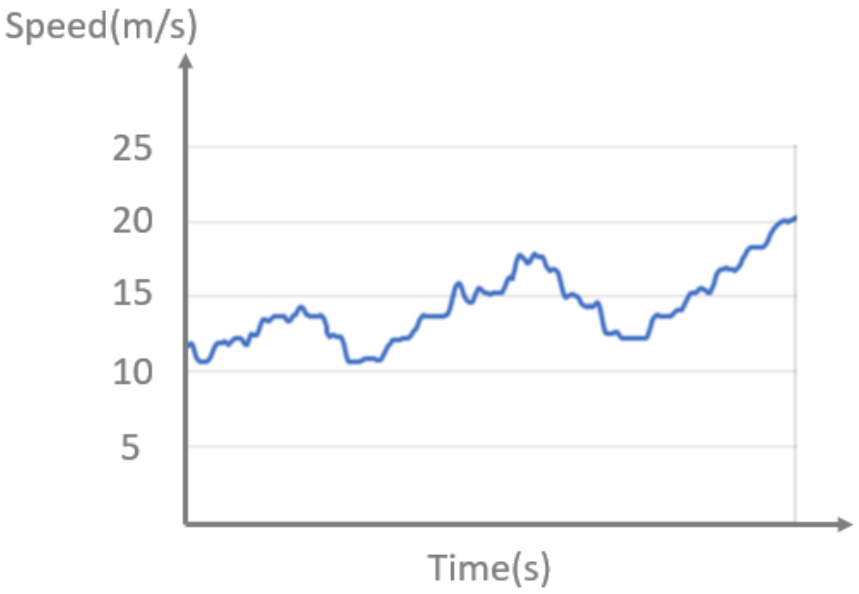


Figure 2.7: Real-world human-driven vehicle speed profile.

To demonstrate OpenCDA's high modularity and extensibility, we further compare two different algorithms of choosing the best merging position. The first approach is heuristic-based. The single CAV will choose the vehicle in the platoon that has the shortest Euclidean distance as the frontal vehicle for merging. The second method is Genetic Fuzzy System [56], which utilizes fuzzy logic to decide the best merging position. Different from a heuristic-based method, it also takes platoon members' speed and surrounding human-driven vehicles' information into consideration.

2.4.3 Evaluation Measurements

As adequate performance measurements are essential in the testing, we also provide default evaluation metrics in the scenario benchmark. For platooning application, we assess the performance from **safety**, **stability**, and **efficiency**.

2.4.3.1 Safety

Safety is always the most critical factor for any automated driving system. In platooning, not only the leading vehicle needs to avoid collisions with surrounding human-driven vehicles, but also the following members are required to keep a safe distance from each other. The safety element can be measured from two perspectives:

- **Time-to-Collision**: Time-to-Collision(TTC) refers to the time required for two vehicles to collide if they continue at their present speed and on the same path. Here we can extract the TTC performance series of each vehicle throughout the simulation. It is also possible to estimate the average TTC for each platoon member across all simulation time steps to represent overall safety by the following equation:

$$ATTC = \frac{\sum_{t=1}^{N} \frac{x_i^t - x_{i-1}^t - l}{v_i^t - v_{i-1}^t}}{N} \tag{2.7}$$

  where x_i^t is the position of vehicle i at time-step t, x_{i-1}^t is the position of the preceding vehicle i at time-step t, l is the length of vehicle i, v_i^t, v_{i-1}^t are the speed of vehicle i and $i-1$ at time step t and N is the number of simulation time-steps at which meets the condition $v_i^t < v_{i-1}^t$.

- **Hazard frequency**: The number of events that $TTC < TTC_t$, where TTC_t is the warning threshold of Time-to-Collision to distinguish between safe and unsafe events. In this experiment, we set it as 2.5 second, which is suggested by [57].

2.4.3.2 Stability

The stability of a platoon indicates whether oscillations are amplified from downstream to upstream vehicles [58]. As it is directly correlated with safety and energy consumption, proposing corresponding appropriate evaluation measurements are crucial. In OpenCDA,

the following three metrics are used and users can easily define advanced metrics using the data provided by OpenCDA.

- **Inter-vehicular time gap**: The time gap between a platoon member and its proceeding vehicle. The time gap at each simulation step is collected and plotted, and its mean value and standard deviation across the whole episode are calculated as well.
- **Acceleration** The time-series data of acceleration and statistics of the data (e.g., mean, standard deviation) are calculated to reflect the driving smoothness of the platoon members.

Table 2.1: **Quantitative results of two different scenario tests**. The desired platoon time gap is set to 0.6 second. attc:average time-to-collision(second), hf:hazard frequency(number of times), atg:average platoon time gap(second), tg_std:platoon time gap standard deviation(second), tcm: time to complete maneuver(second)

Vehicle id	Safety		Stability			Efficiency	
	attc	hf	atg	tg_std	acc_std	tcm	acc_std
0	NA	0	NA	NA	0.98	NA	NA
1	30.55	0	0.603	0.007	0.73	NA	NA
2	30.50	0	0.602	0.003	0.65	NA	NA
3	30.43	0	0.602	0.004	0.62	NA	NA
4	30.40	0	0.602	0.005	0.60	NA	NA
a) Single platooning cycle 1							
0	32.68	0	NA	NA	1.42	NA	NA
1	17.1	0	0.614	0.03	1.08	NA	NA
2	17.3	0	0.609	0.012	0.75	NA	NA
3	18.16	0	0.608	0.007	0.55	NA	NA

Continued on next page

Table 2.1 – continued from previous page

Vehicle id	Safety		Stability			Efficiency	
	attc	hf	atg	tg_std	acc_std	tcm	acc_std
4	18.92	0	0.605	0.003	0.49	NA	NA
b) Single platooning cycle 2							
0	49.8	0	NA	NA	0.92	NA	0.03
1	25.5	0	0.607	0.005	0.63	NA	0.01
2	31.03	0	0.607	0.003	0.56	NA	0.01
3	31.53	0	0.612	0.013	1.33	13.5	2.83
4	32.59	0	0.707	0.23	0.82	NA	1.37
c) Cooperative merge and platoon join using heuristic method							
0	49.8	0	NA	NA	0.95	NA	0.02
1	31.40	0	0.608	0.007	1.27	9.9	2.51
2	31.24	0	0.674	0.16	0.79	NA	1.18
3	30.14	0	0.609	0.008	0.65	NA	0.88
4	29.9	0	0.607	0.006	0.61	NA	0.59
d) Cooperative merge and platoon join using GFS							

2.4.3.3 Efficiency

The efficiency refers to the time duration required for platoon joining and the smoothness of the joining process. It can be evaluated as follows:

- **Time to complete the maneuver** The time duration starting from the joining request approved to joining concluded.
- **Acceleration** The standard deviation of acceleration during the joining procedure.

- **Traffic delay and throughput** If SUMO are used and traffic performance is of interest, overall delay and throughout of traffic are calculated during the specified simulation period.

2.5 Results Analysis

In this section, the results for our benchmark platooning algorithm are presented and discussed.

2.5.1 Single Lane Platooning

Table 2.1 (a) presents the average performance of platooning in cycle 1. As we can see, in spite of the dramatic velocity fluctuation, the platoon members can still maintain the desired 0.6 second time gap safely with minor deviations. Similarly, as Table 2.1 (b) demonstrates, the leading vehicle is able to follow the human-driven vehicle safely and smoothly while the whole platoon can achieve good safety and stability.

Fig. 2.8 further describes the driving performance at each simulation time step. For the first cycle, as Fig. 2.8(a) demonstrates, the platoon followers are able to keep the designed time gap 0.6s during the whole process, even with the leading vehicle dramatically increasing and decreasing speeds. When the platoon leader starts to accelerate suddenly, the platoon members are able to follow it tightly without any speed-overshooting. When the platoon leader rapidly steps on the brake, the followers can smoothly decelerate at a comfortable rate and stay constant time gaps between each other, which indicates the stability of the platooning. In the real trajectory testing, as Fig. 2.4(b) depicts, despite the frequent speed changes of the human-driven vehicle, the platoon leader is able to follow it safely, and the time gap between them is around 1.5s. Meanwhile, the platoon member can still keep a constant time gap of 0.6s even when the front human-driven vehicle rapidly accelerates or

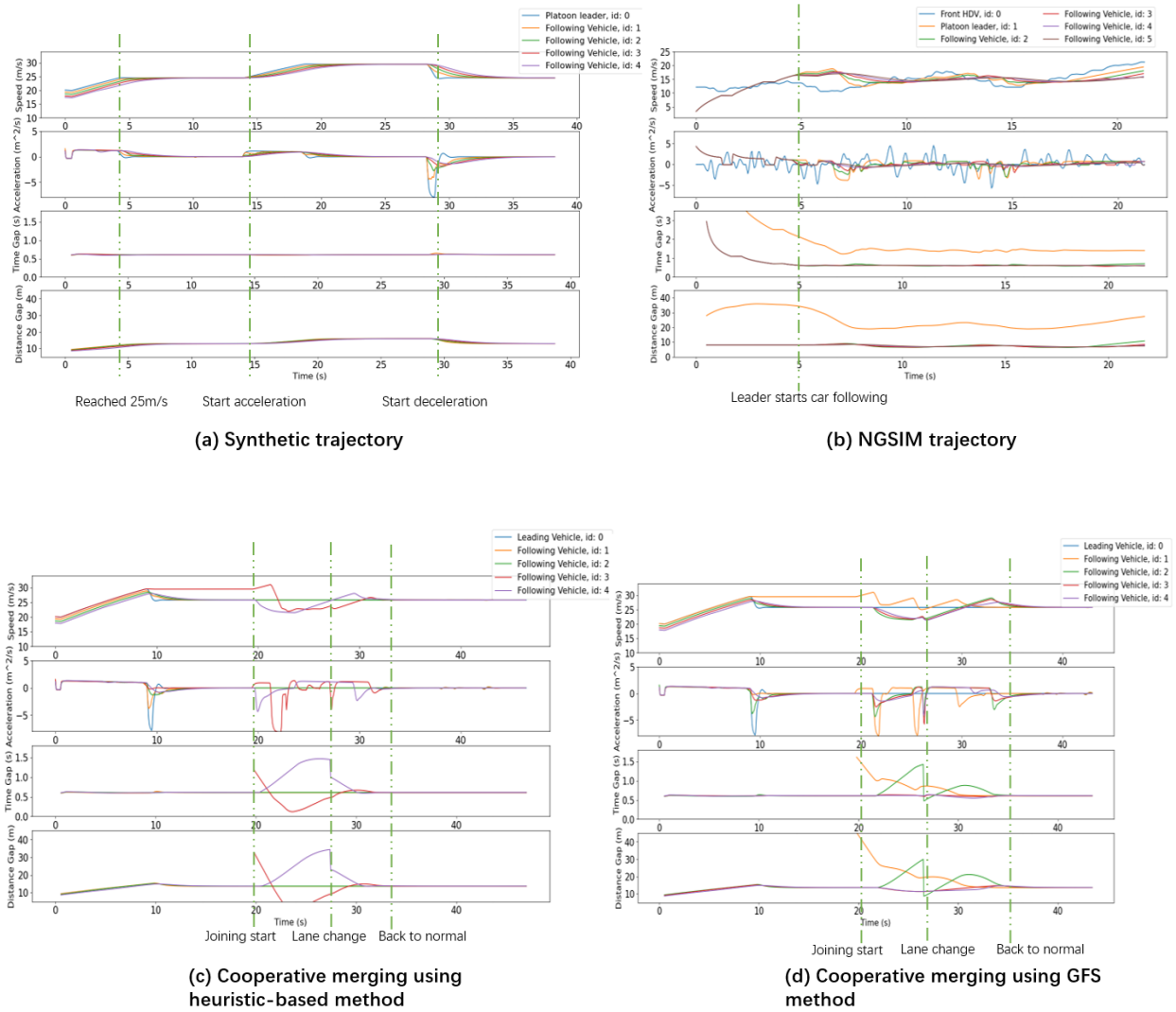


Figure 2.8: The speed, acceleration, time gap and distance gap plotting for each CAV in the four testing scenarios

decelerates.

These results of the benchmark algorithms illustrate the whole module pipeline of the cooperative driving system in our framework is complete and can work properly for cooperative driving tasks in the simulation environment under various settings.

2.5.2 Cooperative Merge and Join Platoon

Fig. 2.8 (c) and 2.8 (d) display the profiles of velocity, acceleration, inter-vehicular time gap, and distance gap of each of the platoon members during the scenario testing utilizing two different merging position decision algorithms. First, the results of these two algorithms are noticeably distinct. The heuristic-based method chooses the third platoon member as the immediate proceeding vehicle for joining, while the GFS chooses the leading vehicle. Second, when the merging CAV operates the cut-in joining using a heuristic-based method, the time gap between it and the rear member drops under 0.2 seconds, which is potentially dangerous. In contrast, the GFS allows the merging vehicle to keep the time gap above 0.6 seconds during the whole joining process, which makes the merging process much safer. Last, the GFS is more efficient as Table2.1(d) depicts, it takes 9.9 seconds to end the joining maneuver while the heuristic-based method needs 13.1 seconds.

In conclusion, the evaluation shows that the GFS is superior to the heuristic-based method. More importantly, our framework allows efficient and straightforward method replacement in as simple as one line of code while maintaining the functionality of the system and the accuracy of other existing modules. This example perfectly proves the effectiveness of OpenCDA in terms of validating any customized CDA algorithms.

2.6 Conclusion

In this article, we introduce OpenCDA, a generalized framework and tool for research and development of Cooperative Driving Automation (CDA). OpenCDA addresses the gap in the community and is one of the first of its kind – an easy-to-use fast-prototyping tool that has a full-stack CDA software platform that covers perception, communication, planning, and control, to enable researchers to evaluate and compare new CDA algorithms and functions with benchmarks. The six key features of OpenCDA – **Connectivity**, **Integration**, **Full-stack System**, **Modularity**, and **Benchmark** – have been discussed in detail through the introduction of the OpenCDA architecture, simulation flow, testing scenarios and processes, and software design. By exploiting a practical example of the platooning application, we demonstrate that the modular pipeline in OpenCDA can function properly for CDA applications and the whole framework is flexible enough for any customization. Last, but not least, OpenCDA is an evolving project, and we expect that our team at the UCLA Mobility Lab and interested parties in the community to continuously contribute to the project with additional CDA applications, testing scenarios, enhancements to the existing CDA platform, and integration with other tools for necessary testing purposes.

CHAPTER 3

OPV2V: An Open Benchmark Dataset and Fusion Pipeline for Perception with Vehicle-to-Vehicle Communication

Employing Vehicle-to-Vehicle communication to enhance perception performance in self-driving technology has attracted considerable attention recently; however, the absence of a suitable open dataset for benchmarking algorithms has made it difficult to develop and assess cooperative perception technologies. To this end, we present the first large-scale open simulated dataset for Vehicle-to-Vehicle perception. It contains over 70 interesting scenes, 11,464 frames, and 232,913 annotated 3D vehicle bounding boxes, collected from 8 towns in CARLA and a digital town of Culver City, Los Angeles. We then construct a comprehensive benchmark with a total of 16 implemented models to evaluate several information fusion strategies (i.e. early, late, and intermediate fusion) with state-of-the-art LiDAR detection algorithms. Moreover, we propose a new Attentive Intermediate Fusion pipeline to aggregate information from multiple connected vehicles. Our experiments show that the proposed pipeline can be easily integrated with existing 3D LiDAR detectors and achieve outstanding performance even with large compression rates. To encourage more researchers to investigate Vehicle-to-Vehicle perception, we will release the dataset, benchmark methods, and all related codes in https://mobility-lab.seas.ucla.edu/opv2v/

3.1 INTRODUCTION

Perceiving the dynamic environment accurately is critical for robust intelligent driving. With recent advancements in robotic sensing and machine learning, the reliability of perception has been significantly improved [59, 60, 61], and 3D object detection algorithms have achieved outstanding performance either with LiDAR point clouds [2, 1, 62, 63] or multi-sensor data [64, 65].

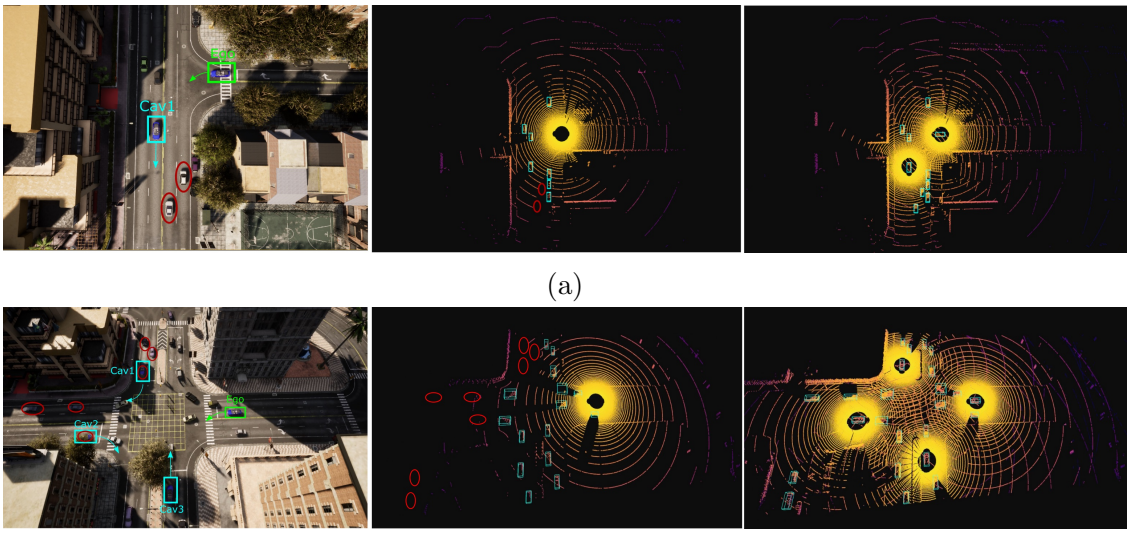
Despite the recent breakthroughs in the perception field, challenges remain. When the objects are heavily occluded or have small scales, the detection performance will dramatically drop. Such problems can lead to catastrophic accidents and are difficult to solve by any algorithms since the sensor observations are too sparse. An example is revealed in Fig. 3.1a. Such circumstances are very common but dangerous in real-world scenarios, and these blind spot issues are extremely tough to handle by a single self-driving car.

To this end, researchers started recently investigating dynamic agent detection in a cooperative fashion, such as USDOT CARMA [25] and Cooper [66]. By leveraging the Vehicle-to-Vehicle (V2V) communication technology, different Connected Automated Vehicles (CAVs) can share their sensing information and thus provide multiple viewpoints for the same obstacle to compensate each other. The shared information could be raw data, intermediate features, single CAV's detection output, and metadata e.g., timestamps and poses. Despite the big potential in this field, it is still in its infancy. One of the major barriers is the lack of a large open-source dataset. Unlike the single vehicle's perception area where multiple large-scale public datasets exist [67, 4, 68], most of the current V2V perception algorithms conduct experiments based on their customized data [69, 70, 71]. These datasets are either too small in scale and variance or they are not publicly available. Consequently, there is no large-scale dataset suitable for benchmarking distinct V2V perception algorithms, and such deficiency will preclude further progress in this research field.

To address this gap, we present OPV2V, the first large-scale **O**pen Dataset for **P**erception with **V2V** communication. By utilizing a cooperative driving co-simulation framework named OpenCDA [7] and CARLA simulator [8], we collect 73 divergent scenes with a various number of connected vehicles to cover challenging driving situations like severe occlusions. To narrow down the gap between the simulation and real-world traffic, we further build a digital town of Culver City, Los Angeles with the same road topology and spawn dynamic agents that mimic the realistic traffic flow on it. Data samples are shown in Fig. 3.1 and Fig. 3.4. We benchmark several state-of-the-art 3D object detection algorithms combined with different multi-vehicle fusion strategies. On top of that, we propose an Attentive Intermediate Fusion pipeline to better capture interactions between connected agents within the network. Our experiments show that the proposed pipeline can efficiently reduce the bandwidth requirements while achieving state-of-the-art performance.

3.2 Related Work

Vehicle-to-Vehicle Perception: V2V perception methods can be divided into three categories: early fusion, late fusion, and intermediate fusion. Early fusion methods [66] share raw data with CAVs within the communication range, and the ego vehicle will predict the objects based on the aggregated data. These methods preserve the complete sensor measurements but require large bandwidth and are hard to operate in real time [69]. In contrast, late fusion methods transmit the detection outputs and fuse received proposals into a consistent prediction. Following this idea, Rauch. [72] propose a Car2X-based perception module to jointly align the shared bounding box proposals spatially and temporally via an EKF. In [73], a machine learning-based method is utilized to fuse proposals generated by different connected agents. This stream of work requires less bandwidth, but the performance of the model is highly dependent on each agent's performance within the vehicular network. To meet requirements of both bandwidth and detection accuracy, intermediate fusion [74, 69]



(a)

(b)

Figure 3.1: Two examples from our dataset. *Left*: Screenshot of the constructed scenarios in CARLA. *Middle*: The LiDAR point cloud collected by the ego vehicle. *Right*: The aggregated point clouds from all surrounding CAVs. The red circles represent the cars that are invisible to the ego vehicle due to the occlusion but can be seen by other connected vehicles. (a): The ego vehicle plans to turn left in a T-intersection and the roadside vehicles block its sight to the incoming traffic. (b): Ego-vehicle's LiDAR has no measurements on several cars because of the occlusion caused by the dense traffic.

has been investigated, where intermediate features are shared among connected vehicles and fused to infer the surrounding objects. F-Cooper [74] utilizes max pooling to aggregate shared Voxel features, and V2VNet [69] jointly reason the bounding boxes and trajectories based on shared messages.

Vehicle-to-Vehicle Dataset: To the best of our knowledge, there is no large-scale open-source dataset for V2V perception in the literature. Some work [66, 74] adapts KITTI [68] to emulate V2V settings by regarding the ego vehicle at different timestamps as multiple CAVs. Such synthetic procedure is unrealistic and not appropriate for V2V tasks since the dynamic agents will appear at different locations, leading to spatial and temporal inconsistency. [69] utilizes a high-fidelity LiDAR simulator [75] to generate a large-scale V2V dataset. However, neither the LiDAR simulator nor the dataset is publicly available. Recently, several works [71,

Sensors	Details
4x Camera	RGB, 800×600 resolution, 110° FOV
1x LiDAR	64 channels, 1.3 M points per second, 120 m capturing range, -25° to 5° vertical FOV, ± 2 cm error
GPS & IMU	20 mm positional error, 2° heading error

Table 3.1: Sensor specifications.

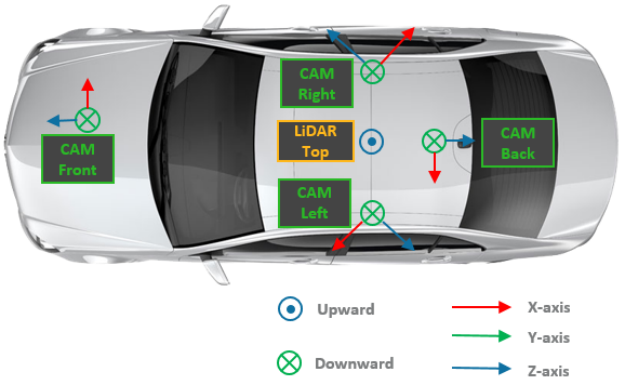


Figure 3.2: Sensor setup for each CAV in OPV2V.

76] manage to evaluate their V2V perception algorithms on the CARLA simulator, but the collected data has a limited size and is restricted to a small area with a fixed number of connected vehicles. More importantly, their dataset is not released and difficult to reproduce the identical data based on their generation approach. T&J dataset [66, 74] utilizes two golf carts equipped with 16-channel LiDAR for data collection. Nevertheless, the released version only has 100 frames without ground truth labels and only covers a restricted number of road types. A comparison to existing dataset is provided in Table 3.2.

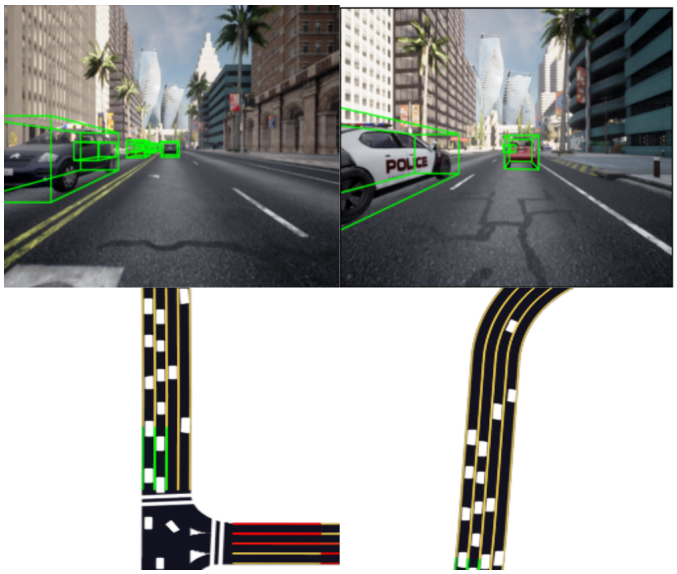
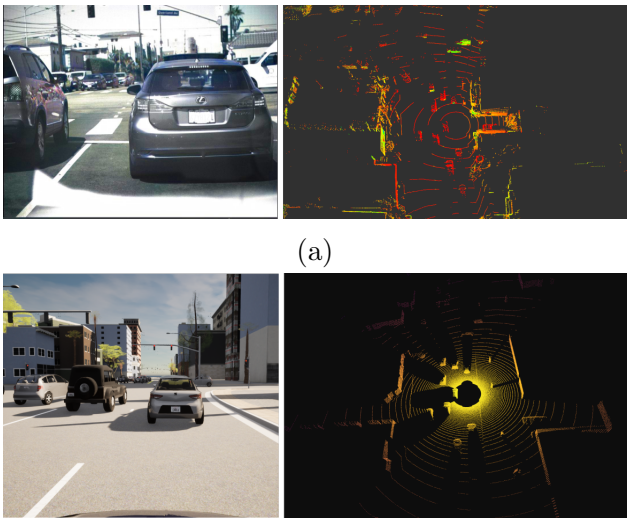


Figure 3.3: Examples of the front camera data and BEV map of two CAVs in OPV2V. The yellow, green, red, and white lanes in the BEV map represent the lanes without traffic light control, under green light control, under red light control, and crosswalks.

Table 3.2: Dataset comparison. (†) The number is reported based on data used during their experiment. (††) Single LiDAR resolution's data is counted. (‡) Ground truth data is not released in the T&J dataset and it only has 100 frames and LiDAR data. (-) means that the number is not reported in the paper and can't be found in open dataset. (*) means the data has the format mean±std.

Dataset	frames	GT 3D boxes	Dataset Size	CAV range	cities	Code	Open Dataset	Reproducibility& Extensibility
V2V-Sim [69]	51,200	-	-	$10 \pm 7^*$	> 1			
[71]	1,310†	-	-	3, 5	1			
[76]	6,000††	-	-	2	1	✓		
T&J [66, 74]	100‡	0‡	183.7MB	2	1	✓	✓	
OPV2V	11,464	232,913	249.4GB	$2.89 \pm 1.06^*$	9	✓	✓	✓



(a)

(b)

Figure 3.4: A caparison between the real Culver City and its digital town. (a) The RGB image and LiDAR point cloud captured by our vehicle in Culver City. (b) The corresponding frame in the digital town. The road topology, building layout, and traffic distribution are similar to reality.

3.3 OPV2V Dataset

3.3.1 Data Collection

Simulator Selection. CARLA is selected as our simulator to collect the dataset, but CARLA itself doesn't have V2V communication and cooperative driving functionalities by default. Hence, we employ OpenCDA [7], a co-simulation tool integrated with CARLA and SUMO [46], to generate our dataset[1]. It is featured with easy control of multiple CAVs, embedded vehicle network communication protocols, and more convenient and realistic traffic management.

Sensor Configuration. The majority of our data comes from eight default towns provided by CARLA. Our dataset has on average approximately 3 connected vehicles with a minimum

[1]Codes for generating our dataset have been released in https://github.com/ucla-mobility/OpenCDA/tree/feature/data_collection

of 2 and a maximum of 7 in each frame. As Fig. 3.2 shows, each CAV is equipped with 4 cameras that can cover 360° view together, a 64-channel LiDAR, and GPS/IMU sensors. The sensor data is streamed at 20 Hz and recorded at 10 Hz. A more detailed description of the sensor configurations is depicted in Table 3.1.

Culver City Digital Town. To incorporate scenarios that can better imitate real-world challenging driving environments and evaluate models' domain adaptation capability, we further gather several scenes imitating realistic configurations. An automated vehicle equipped with a 32-channel LiDAR and two cameras is sent out to Culver City during rush hour to collect sensing data. Then, we populate the road topology of digital town via RoadRunner [77], select buildings based on agreement with collected data, and then spawn cars mimicking the real-world traffic flow with the support of OpenCDA. We collect 4 scenes in Culver City with around 600 frames in total (See Fig. 3.4). These scenes will be used for validation of models trained with simulated datasets purely generated in CARLA. Future addition of data from real environments is planned and can be added to the model training set.

Data Size. Overall, 11,464 frames (i.e. time steps) of LiDAR point clouds (see Fig. 3.1) and RGB images (see Fig. 3.3) are collected with a total file size of 249.4 GB. Moreover, we also generate Bird Eye View (BEV) maps for each CAV in each frame to facilitate the fundamental BEV semantic segmentation task.

Downstream Tasks. By default, OPV2V supports cooperative 3D object detection, BEV semantic segmentation, tracking, and prediction either employing camera rigs or LiDAR sensors. To enable users to extend the initial data, we also provide a driving log replay tool[2]. along with the dataset. By utilizing this tool, users can define their own tasks (e.g., depth estimation, sensor fusion) and set up additional sensors (e.g., depth camera) without changing any original driving events. Note that in this paper, we only report the benchmark results on 3D Lidar-based object detection.

[2]The tool can be found here.

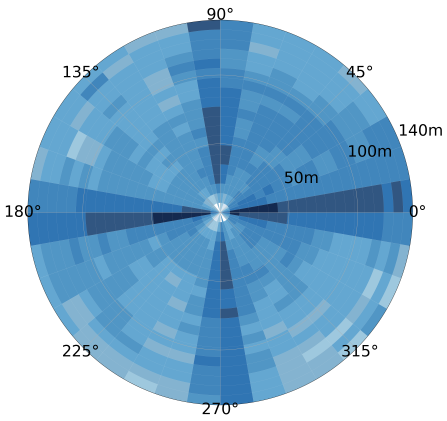


Figure 3.5: Polar density map in log scale for ground truth bounding boxes. The polar and radial axes indicate the angle and distance (in meters) of the bounding boxes with respect to the ego vehicle. The color indicates the number of bounding boxes (log scale) in the bin. The darker color means a larger number of boxes in the bin.

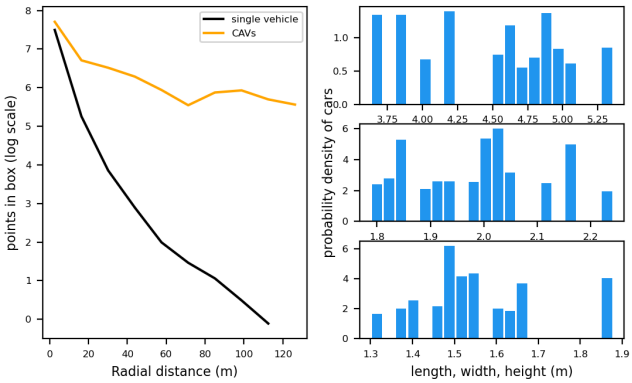


Figure 3.6: *Left*: Number of points in log scale within the ground truth bounding boxes with respect to radial distance from ego vehicles. *Right*: Bounding box size distributions.

Table 3.3: Summary of OPV2V dataset statistics. Traffic density means the number of vehicles spawned around the ego vehicle within a 140m radius and aggressiveness represents the probability of a vehicle operating aggressive overtakes. The speed is in km/h.

Road Type	Ratio	Length(s) mean/std	CAV number mean/std	Traffic density mean/std	Traffic Speed mean/std	CAV speed mean/std	Aggressiveness mean/std
4-way Intersection	24.5	12.5/4.2	2.69/0.67	29.6/26.1	19.3/8.8	21.3/10.2	0.09/0.30
T Intersection	24.1	14.3/12.8	2.55/1.3	27.9/18.65	26.3/7.5	26.2/10.0	0.11/0.32
Straight Segment	20.7	20.2/12.7	3.54/1.21	38.0/36.3	45.7/14.8	54.3/20.1	0.82/0.40
Curvy Segment	23.3	17.8/6.8	2.86/0.95	19.1/9.2	45.8/15.1	51.6/19.2	0.50/0.51
Midblock	4.7	10.0/1.3	3.00/1.22	21.8/8.2	45.1/8.3	50.7/11.5	0.20/0.44
Entrance Ramp	2.7	9.3/0.9	2.67/0.57	20.3/2.8	54.8/1.7	66.7/4.8	0.67/0.57
Overall	100	16.4/9.1	2.89/1.06	26.5/17.2	33.1/15.8	37.5/21.0	0.34/0.47

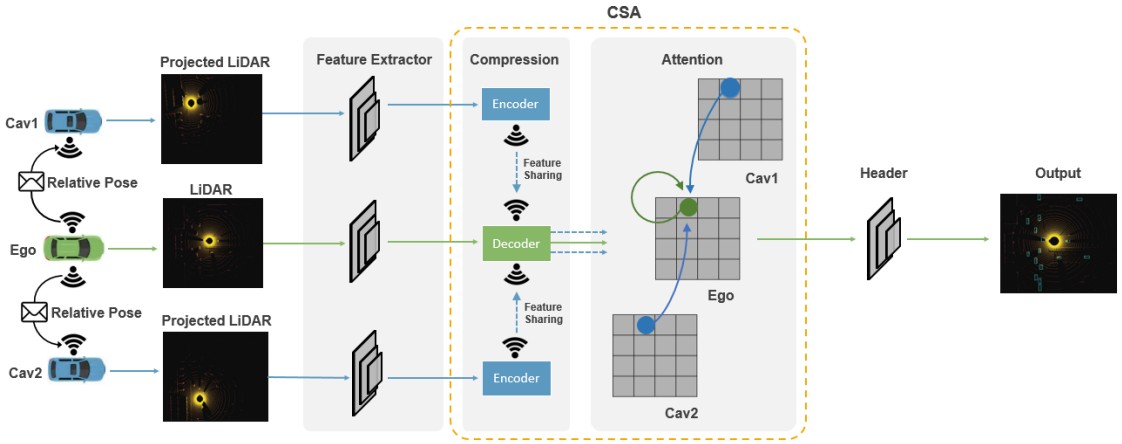


Figure 3.7: The architecture of Attentive Intermediate Fusion pipeline. Our model consists of 6 parts: 1) Metadata Sharing: build connection graph and broadcast locations among neighboring CAVs. 2) Feature Extraction: extract features based on each detector's backbone. 3) Compression (optional): use Encoder-Decoder to compress/decompress features. 4) Feature sharing: share (compressed) features with connected vehicles. 5) Attentive Fusion: leverage self-attention to learn interactions among features in the same spatial location. 6) Prediction Header: generate final object predictions.

3.3.2 Data Analysis

As Table 3.3 depicts, six distinct categories of road types are included in our dataset for simulating the most common driving scenarios in real life. To minimize data redundancy, we attempt to avoid overlong clips and assign the ego vehicles short travels with an average length of 16.4 seconds, dissimilar locations, and divergent maneuvers for each scenario. We

also allocate the gathered 73 scenes with diverse traffic and CAV configurations to enlarge dataset variance.

Fig. 3.5 and Fig. 3.6 reveal the statistics of the 3D bounding box annotations in our dataset. Generally, the cars around the ego vehicle are well-distributed with divergent orientations and bounding box sizes. This distribution is in agreement with the data collection process where the object positions are randomly selected around CAVs and vehicle models are also arbitrarily chosen. As shown in Fig. 3.5, unlike the dataset for the single self-driving car, our dataset still has a large portion of objects in view with distance $\geq$ 100m, given that the ground truth boxes are defined with respect to the aggregated lidar points from all CAVs. As displayed in Fig. 3.6, although a single vehicle's LiDAR points for distant objects are especially sparse, other CAVs are able to provide compensations to remarkably boost the LiDAR points density. This demonstrates the capability of V2V technology to drastically increase perception range and provide compensation for occlusions.

3.4 Attentive Intermediate Fusion Pipeline

As sensor observations from different connected vehicles potentially carry various noise levels (e.g., due to distance between vehicles), a method that can pay attention to important observations while ignoring disrupted ones is crucial for robust detection. Therefore, we propose an Attentive Intermediate Fusion pipeline to capture the interactions between features of neighboring connected vehicles, helping the network attend to key observations. The proposed Attentive Intermediate Fusion pipeline consists of 6 modules: Metadata sharing, Feature Extraction, Compression, Feature sharing, Attentive Fusion, and Prediction. The overall architecture is shown in Fig. 3.7. The proposed pipeline is flexible and can be easily integrated with existing Deep Learning-based LiDAR detectors (see Table 3.4).

Metadata Sharing and Feature Extraction: We first broadcast each CAVs' relative pose

and extrinsics to build a spatial graph where each node is a CAV within the communication range and each edge represents a communication channel between a pair of nodes. After constructing the graph, an ego vehicle will be selected within the group.[3] And all the neighboring CAVs will project their own point clouds to the ego vehicle's LiDAR frame and extract features based on the projected point clouds. The feature extractor here can be the backbones of existing 3D object detectors.

Compression and Feature sharing: An essential factor in V2V communication is the hardware restriction on transmission bandwidth. The transmission of the original high-dimensional feature maps usually requires large bandwidth and hence compression is necessary. One key advantage of intermediate fusion over sharing raw point clouds is the marginal accuracy loss after compression [69]. Here we deploy an Encoder-Decoder architecture to compress the shared message. The Encoder is composed of a series of 2D convolutions and max pooling, and the feature maps in the bottleneck will broadcast to the ego vehicle. The Decoder that contains several deconvolution layers [78] on the ego-vehicles' side will recover the compressed information and send it to the Attentive Fusion module.

Attentive Fusion: Self-attention models [79] are adopted to fuse those decompressed features. Each feature vector (green/blue circles shown in Fig.3.7) within the same feature map corresponds to certain spatial areas in the original point clouds. Thus, simply flattening the feature maps and calculating the weighted sum of features will break spatial correlations. Instead, we construct a local graph for each feature vector in the feature map, where edges are built for feature vectors in the same spatial locations from disparate connected vehicles. One such local graph is shown in Fig.3.7 and self-attention will operate on the graph to reason the interactions for better capturing the representative features.

Prediction Header: The fused features will be fed to the prediction header to generate

[3]During training, a random CAV within the group is selected as ego vehicle while in the inference, the ego vehicle is fixed for a fair comparison.

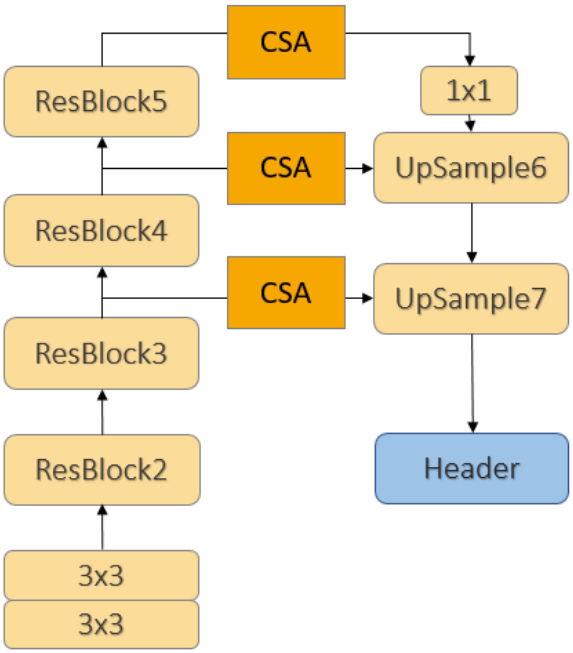


Figure 3.8: The architecture of PIXOR with Attentive Fusion.

bounding box proposals and associated confidence scores.

3.5 Experiments

3.5.1 Benchmark models

We implement four state-of-the-art LiDAR-based 3D object detectors on our dataset and integrate these detectors with three different fusion strategies i.e., early fusion, late fusion, and intermediate fusion. We also investigate the model performance under a single-vehicle setting, named no fusion, which neglects V2V communication. Therefore, in total 16 models will be evaluated in the benchmark. All the models are implemented in a unified code framework, and our code and develop tutorial can be found in the project website.

Selected 3D Object Detectors: We pick SECOND [80], VoxelNet [2], PIXOR [81], and

PointPillar [1] as our 3D LiDAR detectors for benchmarking analysis.

Early fusion baseline: All the LiDAR point clouds will be projected into ego-vehicles' coordinate frame, based on the pose information shared among CAVs, and then the ego vehicle will aggregate all received point clouds and feed them to the detector.

Late fusion baseline: Each CAV will predict the bounding boxes with confidence scores independently and broadcast these outputs to the ego vehicle. Non-maximum suppression (NMS) will be applied to these proposals afterwards to generate the final object predictions.

Intermediate fusion: The Attentive Fusion pipeline is flexible and can be easily generalized to other object detection networks. To evaluate the proposed pipeline, we only need to add the Compression, Sharing, and Attention (CSA) module to the existing network architecture. Since 4 different detectors add CSA modules in a similar way, here we only show the architecture of intermediate fusion with the PIXOR model as Fig. 3.8 displays. Three CSA modules are added at the 2D backbone of PIXOR to aggregate multi-scale features while all other parts of the network remain the same.

3.5.2 Metrics

We select a fixed vehicle as the ego vehicle among all spawned CAVs for each scenario in the test and validation set. Detection performance is evaluated near the ego vehicle in a range of $x \in [-140, 140]m, y \in [-40, 40]m$. Following [69], we set the broadcast range among CAVs to be 70 meters. Sensing messages outside of this communication range will be ignored by the ego vehicle. Average Precisions (AP) at Intersection-over-Union (IoU) threshold of both 0.5 and 0.7 are adopted to assess different models. Since PIXOR ignores the z coordinates of the bounding box, we compute IoU only on x-y plane to make the comparison fair. For the evaluation targets, we include vehicles that are hit by at least one LiDAR point from any connected vehicle.

Table 3.4: Object detection results on Default CARLA Towns and digital Culver City.

Method		Default AP@IoU 0.5	Default AP@IoU 0.7	Culver AP@IoU 0.5	Culver AP@IoU 0.7
PIXOR	No Fusion	0.635	0.406	0.505	0.290
	Late Fusion	0.769	0.578	0.622	0.360
	Early Fusion	0.810	0.678	**0.734**	**0.558**
	Intermediate Fusion	**0.815**	**0.687**	0.716	0.549
PointPillar	No Fusion	0.679	0.602	0.557	0.471
	Late Fusion	0.858	0.781	0.799	0.668
	Early Fusion	0.891	0.800	0.829	0.696
	Intermediate Fusion	**0.908**	**0.815**	**0.854**	**0.735**
SECOND	No Fusion	0.713	0.604	0.646	0.517
	Late Fusion	0.846	0.775	0.808	0.682
	Early Fusion	0.877	0.813	0.821	0.738
	Intermediate Fusion	**0.893**	**0.826**	**0.875**	**0.760**
VoxelNet	No Fusion	0.688	0.526	0.605	0.431
	Late Fusion	0.801	0.738	0.722	0.588
	Early Fusion	0.852	0.758	0.815	0.677
	Intermediate Fusion	**0.906**	**0.864**	**0.854**	**0.775**

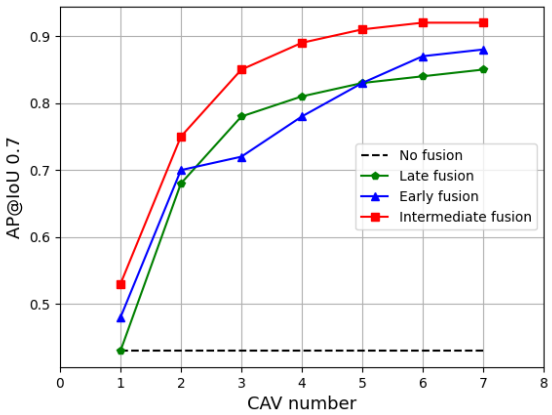


Figure 3.9: Average Precision at IoU=0.7 with respect to CAV number.

3.5.3 Experiment Details

The train/validation/test splits are 6764/1981/2719 frames. The testing frames contain all road types and are further split into two parts–CARLA default maps and Culver City digital town. For each frame, we assure that the minimum and maximum numbers of CAVs are

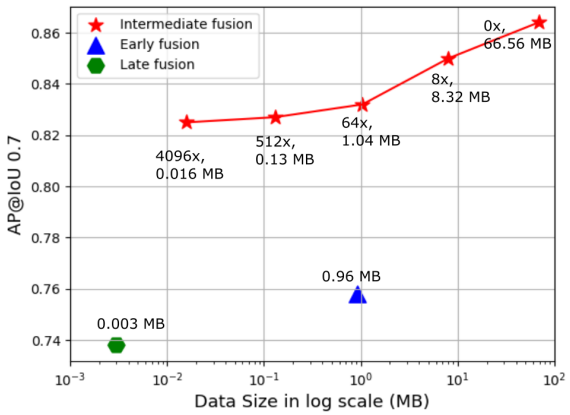


Figure 3.10: Average Precision at IoU=0.7 with respect to data size in log scale based on VoxelNet detector. The number× refers to the compression rate.

2 and 7 respectively. We use Adam Optizer [82] and early stop to train all models, and it takes us 14 days to finish all training on 4 RTX 3090 GPUs.

3.5.4 Benchmark Analysis

Table 3.4 depicts the performance of the selected four LiDAR detectors combined with different fusion strategies. All fusion methods achieve ≥10% AP gains at IoU 0.7 over no fusion counterparts for both default CARLA towns and Culver City, showing the advantage of aggregating information from all CAVs for V2V perception. Generally, because of the capability of preserving more sensing measurements and visual cues, early fusion methods outperform late fusion methods. Except for PIXOR at Culver City, intermediate fusion achieves the best performance on both testing sets compared with all other methods. We argue that the AP gains over early fusion originate from the mechanism of the self-attention module, which can effectively capture the inherent correlation between each CAV's perception information. It is also worth noting that the prediction results for Culver City are generally inferior to CARLA towns. Such a phenomenon is expected as the traffic pattern in Culver City is more similar to real life, which causes a domain gap with the training

data. Furthermore, we collect the Culver City data in a busy hour under a very congested driving environment, which leads to vastly severe occlusions and makes the detection task very challenging.

3.5.5 Effect of CAV Quantity

We explore the detection performance as affected by the number of CAVs in a complex intersection scenario where 150 vehicles are spawned in the surrounding area. A portion of them will be transformed into CAVs that can share information. We gradually increase the number of the CAVs up to 7 and apply VoxelNet with different fusion methods for object detection. As shown in Fig. 3.9, the AP has a positive correlation with the number of CAVs. However, when the quantity reaches 4, the increasing rate becomes lower. This can be due to the fact that the CAVs are distributed on different sides of the intersection and four of them can already provide enough viewpoints to cover most of the blind spots. Additional enhancements with 5 or more vehicles come from denser measurements on the same object.

3.5.6 Effect of Compression Rates

Fig. 3.10 exhibits the data size needed for a single transmission between a pair of vehicles and corresponding AP for all fusion methods on the testing set in CARLA towns. We pick VoxelNet for all fusion methods here and simulate distinct compression rates by modifying the number of layers in Encoder-Decoder. By applying a straightforward Encoder-Decoder architecture to squeeze the data, the Attentive Intermediate Fusion obtains an outstanding trade-off between the accuracy and bandwidth. Even with a 4096x compression rate, the performance still just drop marginally (around 3%) and surpass the early fusion and late fusion. Based on the V2V communication protocol [83], data broadcasting can achieve 27 Mbps at the range of 300 m. This represents that the time delay to deliver the message with a 4096x compression rate is only about 5 ms.

3.6 CONCLUSIONS

In this paper, we present the first open dataset and benchmark fusion strategies for V2V perception. We further come up with an Attentive Intermediate Fusion pipeline, and the experiments show that the proposed approach can outperform all other fusion methods and achieve state-of-the-art performance even under large compression rates.

In the future, we plan to extend the dataset with more tasks as well as sensors suites and investigate more multi-modal sensor fusion methods in the V2V and Vehicle-to-infrastructure (V2I) setting. We hope our open-source efforts can make a step forward for the standardizing process of the V2V perception and encourage more researchers to investigate this new direction.

CHAPTER 4

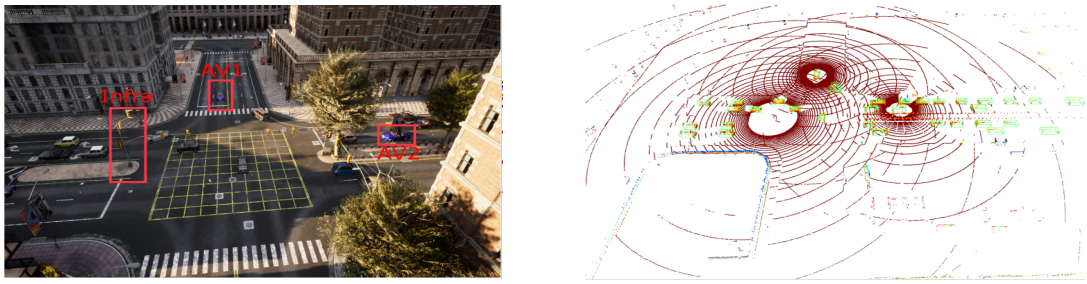
V2X-ViT: Vehicle-to-Everything Cooperative Perception with Vision Transformer

In this paper, we investigate the application of Vehicle-to-Everything (V2X) communication to improve the perception performance of autonomous vehicles. We present a robust cooperative perception framework with V2X communication using a novel vision Transformer. Specifically, we build a holistic attention model, namely V2X-ViT, to effectively fuse information across on-road agents (i.e., vehicles and infrastructure). V2X-ViT consists of alternating layers of heterogeneous multi-agent self-attention and multi-scale window self-attention, which captures inter-agent interaction and per-agent spatial relationships. These key modules are designed in a unified Transformer architecture to handle common V2X challenges, including asynchronous information sharing, pose errors, and heterogeneity of V2X components. To validate our approach, we create a large-scale V2X perception dataset using CARLA and OpenCDA. Extensive experimental results demonstrate that V2X-ViT sets new state-of-the-art performance for 3D object detection and achieves robust performance even under harsh, noisy environments. The code is available at https://github.com/DerrickXuNu/v2x-vit.

4.1 Introduction

Perceiving the complex driving environment precisely is crucial to the safety of autonomous vehicles (AVs). With recent advancements of deep learning, the robustness of single-vehicle perception systems has demonstrated significant improvement in several tasks such as semantic segmentation [84, 85], depth estimation [86, 87], and object detection and tracking [1, 88, 89, 90]. Despite recent advancements, challenges remain. Single-agent perception system tends to suffer from occlusion and sparse sensor observation at a far distance, which can potentially cause catastrophic consequences [91]. The cause of such a problem is that an individual vehicle can only perceive the environment from a single perspective with limited sight-of-view. To address these issues, recent studies [15, 92, 6, 93] leverage the advantages of multiple viewpoints of the same scene by investigating Vehicle-to-Vehicle (V2V) collaboration, where visual information (*e.g.*, detection outputs, raw sensory information, intermediate deep learning features, details see Sec. 4.2) from multiple nearby AVs are shared for a complete and accurate understanding of the environment.

Although V2V technologies have the prospect to revolutionize the mobility industry, it ignores a critical collaborator – roadside infrastructure. The presence of AVs is usually unpredictable, whereas the infrastructure can always provide supports once installed in key scenes such as intersections and crosswalks. Moreover, infrastructure equipped with sensors on an elevated position has a broader sight-of-view and potentially less occlusion. Despite these advantages, including infrastructure to deploy a robust V2X perception system is non-trivial. Unlike V2V collaboration, where all agents are homogeneous, V2X systems often involve a heterogeneous graph formed by infrastructure and AVs. The configuration discrepancies between infrastructure and vehicle sensors, such as types, noise levels, installation height, and even sensor attributes and modality, make the design of a V2X perception system challenging. Moreover, the GPS localization noises and the asynchronous sensor measurements of AVs and infrastructure can introduce inaccurate coordinate transformation and lagged sensing



(a) Snapshot of Simulation (b) Aggregated LiDAR point cloud

Figure 4.1: **A data sample from the proposed V2XSet**. (a) A simulated scenario in CARLA where two AVs and infrastructure are located at different sides of a busy intersection. (b) The aggregated LiDAR point clouds of these three agents.

information. Failing to properly handle these challenges will make the system vulnerable.

In this paper, we introduce a unified fusion framework, namely V2X Vision Transformer or **V2X-ViT**, for V2X perception, that can jointly handle these challenges. Fig. 4.2 illustrates the entire system. AVs and infrastructure capture, encode, compress, and send intermediate visual features with each other, while the ego vehicle (*i.e.*, receiver) employs V2X-Transformer to perform information fusion for object detection. We propose two novel attention modules to accommodate V2X challenges: 1) a customized heterogeneous multi-agent self-attention module that explicitly considers agent types (vehicles and infrastructure) and their connections when performing attentive fusion; 2) a multi-scale window attention module that can handle localization errors by using multi-resolution windows in parallel. These two modules will adaptively fuse visual features in an iterative fashion to capture inter-agent interaction and per-agent spatial relationship, correcting the feature misalignment caused by localization error and time delay. Moreover, we also integrate a delay-aware positional encoding to further handle the time delay uncertainty. Notably, all these modules are incorporated in a single transformer that learns to address these challenges end-to-end.

To evaluate our approach, we collect a new large-scale open dataset, namely V2XSet, that explicitly considers real-world noises during V2X communication using the high-fidelity simulator CARLA [8], and a cooperative driving automation simulation tool OpenCDA.

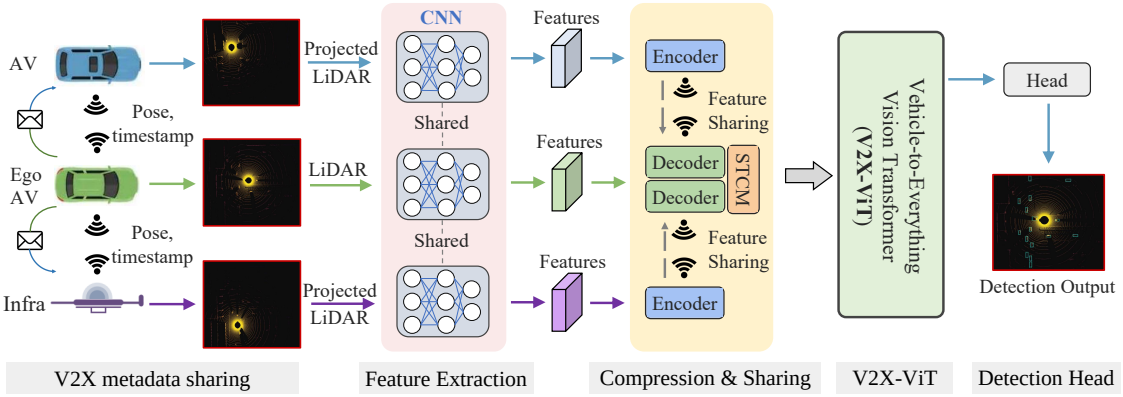


Figure 4.2: **Overview of our proposed V2X perception system.** It consists of five sequential steps: V2X metadata sharing, feature extraction, compression & sharing, V2X-ViT, and the detection head.

Fig. 4.1 shows a data sample in the collected dataset. Experiments show that our proposed V2X-ViT significantly advances the performance on V2X LiDAR-based 3D object detection, achieving a 21.2% gain of AP compared to single-agent baseline and performing favorably against leading intermediate fusion methods by at least 7.3%. Our contributions are:

- We present the first unified transformer architecture (V2X-ViT) for V2X perception, which can capture the heterogeneity nature of V2X systems with strong robustness against various noises. Moreover, the proposed model achieves state-of-the-art performance on the challenging cooperative detection task.
- We propose a novel heterogeneous multi-agent attention module (HMSA) tailored for adaptive information fusion between heterogeneous agents.
- We present a new multi-scale window attention module (MSWin) that simultaneously captures local and global spatial feature interactions in parallel.
- We construct V2XSet, a new large-scale open simulation dataset for V2X perception, which explicitly accounts for imperfect real-world conditions.

4.2 Related work

V2X perception. Cooperative perception studies how to efficiently fuse visual cues from neighboring agents. Based on its message sharing strategy, it can be divided into 3 categories: 1) early fusion [92] where raw data is shared and gathered to form a holistic view, 2) intermediate fusion [15, 6, 94, 93] where intermediate neural features are extracted based on each agent's observation and then transmitted, and 3) late fusion [95, 96] where detection outputs (*e.g.*, 3D bounding box position, confidence score) are circulated. As early fusion usually requires large transmission bandwidth and late fusion fails to provide valuable scenario context [15], intermediate fusion has attracted increasing attention because of its good balance between accuracy and transmission bandwidth. Several intermediate fusion methods have been proposed for V2V perception recently. OPV2V [6] implements a single-head self-attention module to fuse features, while F-Cooper employs *maxout* [97] fusion operation. V2VNet [15] proposes a spatial-aware message passing mechanism to jointly reason detection and prediction. To attenuate outlier messages, [94] regresses vehicles' localization errors with consistent pose constraints. DiscoNet [98] leverages knowledge distillation to enhance training by constraining the corresponding features to the ones from the network for early fusion. However, intermediate fusion for V2X is still in its infancy. Most V2X methods explored late fusion strategies to aggregate information from infrastructure and vehicles. For example, a late fusion two-level Kalman filter is proposed by [99] for roadside infrastructure failure conditions. Xiangmo *et al.* [100] propose fusing the lane mark detection from infrastructure and vehicle sensors, leveraging Dempster-Shafer theory to model the uncertainty.

LiDAR-based 3D object detection. Numerous methods have been explored to extract features from raw points, voxels, bird-eye-view (BEV) images, and their mixtures. PointRCNN [101] proposes a two-stage strategy based on raw point clouds, which learns rough estimation in the first stage and then refines it with semantic attributes. The authors of [2, 80] propose to split the space into voxels and produce features per voxel. Despite having high

accuracy, their inference speed and memory consumption are difficult to optimize due to reliance on 3D convolutions. To avoid computationally expensive 3D convolutions, [1, 81] propose an efficient BEV representation. To satisfy both computational and flexible receptive field requirements, [102, 103, 104] combine voxel-based and point-based approaches to detect 3D objects.

Transformers in vision. The Transformer [79] is first proposed for machine translation [79], where multi-head self-attention and feed-forward layers are stacked to capture long-range interactions between words. Dosovitskiy *et al*. [105] present a Vision Transformer (ViT) for image recognition by regarding image patches as visual words and directly applying self-attention. The full self-attention in ViT [79, 105, 106], despite having global interaction, suffers from heavy computational complexity and does not scale to long-range sequences or high-resolution images. To ameliorate this issue, numerous methods have introduced locality into self-attention, such as Swin [107], CSwin [108], Twins [109], window [110, 111], and sparse attention [112, 113, 20]. A hierarchical architecture is usually adopted to progressively increase the receptive fields for capturing longer dependencies.

While these vision transformers have proven efficient in modeling homogeneous structured data, their efficacy to represent heterogeneous graphs has been less studied. One of the developments related to our work is the heterogeneous graph transformer (HGT) [114]. HGT was originally designed for web-scale Open Academic Graph where the nodes are text and attributes. Inspired by HGT, we build a customized heterogeneous multi-head self-attention module tailored for graph attribute-aware multi-agent 3D visual feature fusion, which is able to capture the heterogeneity of V2X systems.

4.3 Methodology

In this paper, we consider V2X perception as a heterogeneous multi-agent perception system, where different types of agents (*i.e.*, smart infrastructure and AVs) perceive the surrounding environment and communicate with each other. To simulate real-world scenarios, we assume that all the agents have imperfect localization and time delay exists during feature transmission. Given this, our goal is to develop a robust fusion system to enhance the vehicle's perception capability and handle these aforementioned challenges in a unified end-to-end fashion. The overall architecture of our framework is illustrated in Fig. 4.2, which includes five major components: 1) metadata sharing, 2) feature extraction, 3) compression and sharing, 4) V2X vision Transformer, and 5) a detection head.

4.3.1 Main architecture design

V2X metadata sharing. During the early stage of collaboration, every agent $i \in \{1 \dots N\}$ within the communication networks shares metadata such as poses, extrinsics, and agent type $c_i \in \{I, V\}$ (meaning infrastructure or vehicle) with each other. We select one of the connected AVs as the ego vehicle (e) to construct a V2X graph around it where the nodes are either AVs or infrastructure and the edges represent directional V2X communication channels. To be more specific, we assume the transmission of metadata is well-synchronized, which means each agent i can receive ego pose $x_e^{t_i}$ at the time t_i. Upon receiving the pose of the ego vehicle, all the other connected agents nearby will project their own LiDAR point clouds to the ego-vehicle's coordinate frame before feature extraction.

Feature extraction. We leverage the anchor-based PointPillar method [1] to extract visual features from point clouds because of its low inference latency and optimized memory usage [6]. The raw point clouds will be converted to a stacked pillar tensor, then scattered to a 2D pseudo-image and fed to the PointPillar backbone. The backbone extracts informative

feature maps $\mathbf{F}_i^{t_i} \in \mathbb{R}^{H \times W \times C}$, denoting agent i's feature at time t_i with height H, width W, and channels C.

Compression and sharing. To reduce the required transmission bandwidth, we utilize a series of 1×1 convolutions to progressively compress the feature maps along the channel dimension. The compressed features with the size (H, W, C') (where $C' \ll C$) are then transmitted to the ego vehicle (e), on which the features are projected back to (H, W, C) using 1×1 convolutions.

There exists an inevitable time gap between the time when the LiDAR data is captured by connected agents and when the extracted features are received by the ego vehicle (details see appendix). Thus, features collected from surrounding agents are often temporally misaligned with the features captured on the ego vehicle. To correct this delay-induced global spatial misalignment, we need to transform (*i.e.*, rotate and translate) the received features to the current ego-vehicle's pose. Thus, we leverage a spatial-temporal correction module (STCM), which employs a differential transformation and sampling operator Γ_ξ to spatially warp the feature maps [115]. An ROI mask is also calculated to prevent the network from paying attention to the padded zeros caused by the spatial warp.

V2X-ViT. The intermediate features $\mathbf{H}_i = \Gamma_\xi \left(\mathbf{F}_i^{t_i}\right) \in \mathbb{R}^{H \times W \times C}$ aggregated from connected agents are fed into the major component of our framework *i.e.*, V2X-ViT to conduct an iterative inter-agent and intra-agent feature fusion using self-attention mechanisms. We maintain the feature maps in the same level of high resolution throughout the entire Transformer as we have observed that the absence of high-definition features greatly harms the objection detection performance. The details of our proposed V2X-ViT will be unfolded in Sec. 4.3.2.

Detection head. After receiving the final fused feature maps, we apply two 1×1 convolution layers for box regression and classification. The regression output is $(x, y, z, w, l, h, \theta)$, denoting the position, size, and yaw angle of the predefined anchor boxes, respectively. The classification output is the confidence score of being an object or background for each anchor

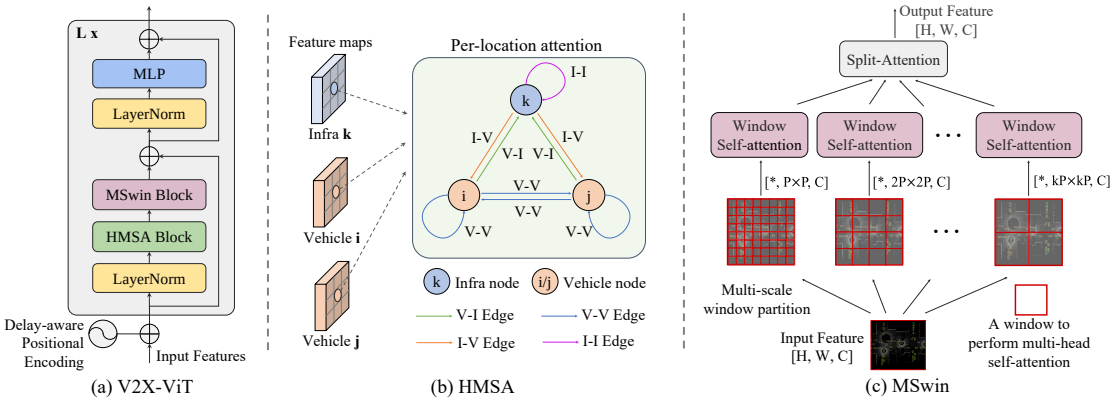


Figure 4.3: **V2X-ViT architecture.** (a) The architecture of our proposed V2X-ViT model. (b) Heterogeneous multi-agent self-attention (HMSA) presented in Sec. 4.3.2.1. (c) Multi-scale window attention module (MSwin) illustrated in Sec. 4.3.2.2.

box. We use the smooth ℓ_1 loss for regression and a focal loss [116] for classification.

4.3.2 V2X-Vision Transformer

Our goal is to design a customized vision Transformer that can jointly handle the common V2X challenges. Firstly, to effectively capture the heterogeneous graph representation between infrastructure and AVs, we build a heterogeneous multi-agent self-attention module that learns different relationships based on node and edge types. Moreover, we propose a novel spatial attention module, namely multi-scale window attention (MSwin), that captures long-range interactions at various scales. MSwin uses multiple window sizes to aggregate spatial information, which greatly improves the detection robustness against localization errors. Lastly, these two attention modules are integrated into a single V2X-ViT block in a factorized manner (illustrated in Fig. 4.3a), enabling us to maintain high-resolution features throughout the entire process. We stack a series of V2X-ViT blocks to iteratively learn inter-agent interaction and per-agent spatial attention, leading to a robust aggregated feature representation for detection.

4.3.2.1 Heterogeneous multi-agent self-attention

The sensor measurements captured by infrastructure and AVs possibly have distinct characteristics. The infrastructure's LiDAR is often installed at a higher position with less occlusion and different view angles. In addition, the sensors may have different levels of sensor noise due to maintenance frequency, hardware quality *etc.*. To encode this heterogeneity, we build a novel heterogeneous multi-agent self-attention (HMSA) where we attach types to both nodes and edges in the directed graph. To simplify the graph structure, we assume the sensor setups among the same category of agents are identical. As shown in Fig. 4.3b, we have two types of nodes and four types of edges, *i.e.*, node type $c_i \in \{I, V\}$ and edge type $\phi(e_{ij}) \in \{V-V, V-I, I-V, I-I\}$. Note that unlike traditional attention where the node features are treated as a vector, we only reason the interaction of features *in the same spatial position* from different agents to preserve spatial cues. Formally, HSMA is expressed as:

$$\mathbf{H}_i = \underset{\forall j \in N(i)}{\mathsf{Dense}_{c_i}} \left(\mathbf{ATT}(i,j) \cdot \mathbf{MSG}(i,j) \right) \tag{4.1}$$

which contains 3 operators: a linear aggregator Dense_{c_i}, attention weights estimator **ATT**, and message aggregator **MSG**. The Dense is a set of linear projectors indexed by the node type c_i, aggregating multi-head information. **ATT** calculates the importance weights between pairs of nodes conditioned on the associated node and edge types:

$$\mathbf{ATT}(i,j) = \underset{\forall j \in N(i)}{\mathsf{softmax}} \left(\Big\Vert_{m \in [1,h]} \text{head}^m_{\text{ATT}}(i,j) \right) \tag{4.2}$$

$$\text{head}^m_{\text{ATT}}(i,j) = \left(\mathbf{K}^m(j) \, \mathbf{W}^{m,\text{ATT}}_{\phi(e_{ij})} \mathbf{Q}^m(i)^T \right) \frac{1}{\sqrt{C}} \tag{4.3}$$

$$\mathbf{K}^m(j) = \mathsf{Dense}^m_{c_j}(\mathbf{H}_j) \tag{4.4}$$

$$\mathbf{Q}^m(i) = \mathsf{Dense}^m_{c_i}(\mathbf{H}_i) \tag{4.5}$$

where $\|$ denotes concatenation, m is the current head number and h is the total number of heads. Notice that Dense here is indexed by both node type $c_{i/j}$, and head number m. The linear layers in $\mathbf{K}$ and $\mathbf{Q}$ have distinct parameters. To incorporate the semantic meaning of edges, we calculate the dot product between Query and Key vectors weighted by a matrix $\mathbf{W}_{\phi(e_{ij})}^{m,\text{ATT}} \in \mathbb{R}^{C\times C}$. Similarly, when parsing messages from the neighboring agent, we embed infrastructure and vehicle's features separately via $\mathsf{Dense}_{c_j}^m$. A matrix $\mathbf{W}_{\phi(e_{ij})}^{m,\text{MSG}}$ is used to project the features based on the edge type between source node and target node:

$$\mathbf{MSG}\left(i,j\right) = \underset{m\in[1,h]}{\|} \text{head}_{\text{MSG}}^{m}\left(i,j\right) \tag{4.6}$$

$$\text{head}_{\text{MSG}}^{m}\left(i,j\right) = \mathsf{Dense}_{c_j}^{m}\left(\mathbf{H}_j\right)\mathbf{W}_{\phi(e_{ij})}^{m,\text{MSG}}. \tag{4.7}$$

4.3.2.2 Multi-scale window attention

We present a new type of attention mechanism tailored for efficient long-range spatial interaction on high-resolution detection, called multi-scale window attention (MSwin). It uses a pyramid of windows, each of which caps a different attention range, as illustrated in Fig. 4.3c. The usage of variable window sizes can greatly improve the detection robustness of V2X-ViT against localization errors (see ablation study in Fig. 4.5b). Attention performed within larger windows can capture long-range visual cues to compensate for large localization errors, whereas smaller window branches perform attention at finer scales to preserve local context. Afterward, the split-attention module [117] is used to adaptively fuse information coming from multiple branches, empowering MSwin to handle a range of pose errors. Note that MSwin is applied on each agent independently without considering any inter-agent fusion; therefore we omit the agent subscript in this subsection for simplicity.

Formally, let $\mathbf{H} \in \mathbb{R}^{H\times W\times C}$ be an input feature map of a single agent. In branch j out of k parallel branches, $\mathbf{H}$ is partitioned using window size $P_j \times P_j$, into a tensor of shape $(\frac{H}{P_j}\times\frac{W}{P_j}, P_j\times P_j, C)$, which represents a $\frac{H}{P_j}\times\frac{W}{P_j}$ grid of non-overlapping patches each with

size $P_j \times P_j$. We use h_j number of heads to improve the attention power at j-th branch. More detailed formulation can be found in Appendix. Following [107, 118], we also consider an additional relative positional encoding $\mathbf{B}$ that acts as a bias term added to the attention map. As the relative position along each axis lies in the range $[-P_j + 1, P_j - 1]$, we take $\mathbf{B}$ from a parameterized matrix $\hat{\mathbf{B}} \in \mathbb{R}^{(2P_j-1)\times(2P_j-1)}$.

To attain per-agent multi-range spatial relationship, each branch partitions input tensor $\mathbf{H}$ with different window sizes $i.e.\{P_j\}_{j=1}^{k} = \{P, 2P, ..., kP\}$. We progressively decrease the number of heads when using a larger window size to save memory usage. Finally, we fuse the features from all the branches by a Split-Attention module [117], yielding the output feature $\mathbf{Y}$. The complexity of the proposed MSwin is *linear* to image size HW, while enjoying long-range multi-scale receptive fields and adaptively fuses both local and (sub)-global visual hints in parallel. Notably, unlike Swin Transformer [107], our multi-scale window approach requires no masking, padding, or cyclic-shifting, making it more efficient in implementations while having larger-scale spatial interactions.

4.3.2.3 Delay-aware positional encoding

Although the global misalignment is captured by the spatial warping matrix Γ_ξ, another type of local misalignment, arising from object motions during the delay-induced time lag, also needs to be considered. To encode this temporal information, we leverage an adaptive delay-aware positional encoding (DPE), composed of a linear projection and a learnable embedding. We initialize it with sinusoid functions conditioned on time delay Δt_i and channel $c \in [1, C]$:

$$\mathbf{p}_c\left(\Delta t_i\right) = \begin{cases} \sin\left(\Delta t_i / 10000^{\frac{2c}{C}}\right), & c = 2k \\ \cos\left(\Delta t_i / 10000^{\frac{2c}{C}}\right), & c = 2k+1 \end{cases} \tag{4.8}$$

A linear projection $f : \mathbb{R}^C \to \mathbb{R}^C$ will further warp the learnable embedding so it can

generalize better for unseen time delay [114]. We add this projected embedding to each agents' feature $\mathbf{H}_i$ before feeding into the Transformer so that the features are temporally aligned beforehand.

$$\text{DPE}\left(\Delta t_i\right) = f\left(\mathbf{p}\left(\Delta t_i\right)\right) \tag{4.9}$$

$$\mathbf{H}_i = \mathbf{H}_i + \text{DPE}\left(\Delta t_i\right) \tag{4.10}$$

4.4 Experiments

4.4.1 Experimental setup

Th evaluation range in x and y direction are $[-140, 140]$ m and $[-40, 40]$ m respectively. We assess models under two settings: 1) *Perfect Setting*, where the pose is accurate, and everything is synchronized across agents; 2) *Noisy Setting*, where pose error and time delay are both considered. In the *Noisy Setting*, the positional and heading noises of the transmitter are drawn from a Gaussian distribution with a default standard deviation of 0.2 m and 0.2° respectively, following the real-world noise levels [119, 120, 121]. The time delay is set to 100 ms for all the evaluated models to have a fair comparison of their robustness against asynchronous message propagation.

Evaluation metrics. The detection performance is measured with Average Precisions (AP) at Intersection-over-Union (IoU) thresholds of 0.5 and 0.7. In this work, we focus on LiDAR-based vehicle detection. Vehicles hit by at least one LiDAR point from any connected agent will be included as evaluation targets.

Implementation details. During training, a random AV is selected as the ego vehicle, while during testing, we evaluate on a fixed ego vehicle for all the compared models. The communication range of each agent is set as 70 m based on [122], whereas all the agents out of this broadcasting radius of ego vehicle is ignored. For the PointPillar backbone, we set

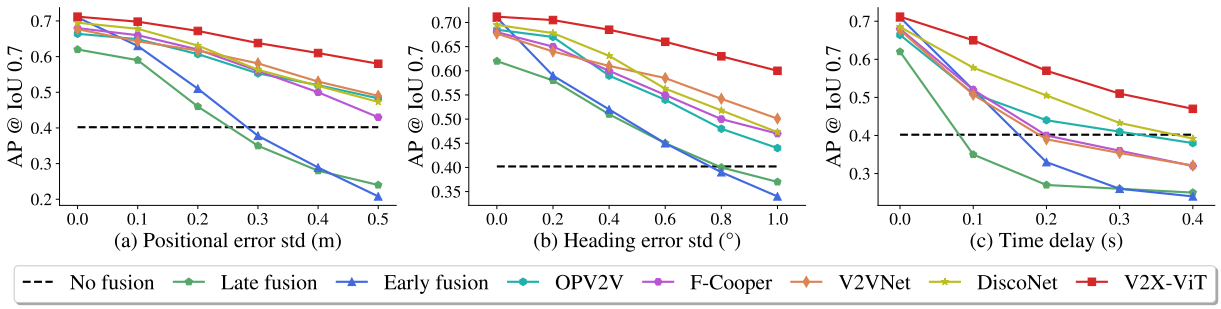


Figure 4.4: **Robustness assessment** on positional and heading errors.

the voxel resolution to 0.4 m for both height and width. The default compression rate is 32 for all intermediate fusion methods. Our V2X-ViT has 3 encoder layers with 3 window sizes in MSwin: 4, 8, and 16. We first train the model under the *Perfect Setting*, then fine-tune it while fixing the backbone for *Noisy Setting*. We adopt Adam optimizer [82] with an initial learning rate of 10^{-3} and steadily decay it every 10 epochs using a factor of 0.1. All models are trained on Tesla V100 with 10^5 iterations.

Compared methods. We consider *No Fusion* as our baseline, which only uses ego-vehicle's LiDAR point clouds. We also compare with *Late Fusion*, which gathers all detected outputs from agents and applies Non-maximum suppression to produce the final results, and *Early Fusion*, which directly aggregates raw LiDAR point clouds from nearby agents. For *intermediate fusion* strategy, we evaluate four state-of-the-art approaches: OPV2V [6], F-Cooper [93] V2VNet [15], and DiscoNet [98]. For a fair comparison, all the models use PointPillar as the backbone, and every compared V2V methods also receive infrastructure data, but they do not distinguish between infrastructure and vehicles.

4.4.2 Quantitative evaluation

Main performance comparison. Tab. 4.1 shows the performance comparisons on both *Perfect* and *Noisy Setting*. Under the *Perfect Setting*, all the cooperative methods significantly outperform *No Fusion* baseline. Our proposed V2X-ViT outperforms SOTA interme-

Table 4.1: **3D detection performance comparison on V2XSet.** We show Average Precision (AP) at IoU=0.5, 0.7 on *Perfect* and *Noisy* settings, respectively.

	Perfect		Noisy	
Models	AP0.5	AP0.7	AP0.5	AP0.7
No Fusion	0.606	0.402	0.606	0.402
Late Fusion	0.727	0.620	0.549	0.307
Early Fusion	0.819	0.710	0.720	0.384
F-Cooper [93]	0.840	0.680	0.715	0.469
OPV2V [6]	0.807	0.664	0.709	0.487
V2VNet [15]	0.845	0.677	0.791	0.493
DiscoNet [98]	0.844	0.695	0.798	0.541
V2X-ViT (Ours)	**0.882**	**0.712**	**0.836**	**0.614**

diate fusion methods by 3.8%/1.7% for AP@0.5/0.7. It is even higher than the ideal *Early fusion* by 0.2% AP@0.7, which receives complete raw information. Under noisy setting, when localization error and time delay are considered, the performance of *Early Fusion* and *Late Fusion* drastically drop to 38.4% and 30.7% in AP@0.7, even worse than single-agent baseline *No Fusion*. Although OPV2V [6], F-Cooper [93] V2VNet [15], and DiscoNet [98] are still higher than *No fusion*, their performance decrease by 17.7%, 21.1%, 18.4% and 15.4% in AP@0.7, respectively. In contrast, V2X-ViT performs favorably against the *No fusion* method by a large margin, *i.e.*23% and 21.2% higher in AP@0.5 and AP@0.7. Moreover, when compared to the *Perfect Setting*, V2X-ViT only drops by less than 5% and 10% in AP@0.5 and AP@0.7 under *Noisy Setting*, demonstrating its robustness against normal V2X noises. The real-time performance of V2X-ViT is also shown in Tab. 4.4. The inference time of V2X-ViT is 57 ms, and by using only 1 encoder layer, V2X-ViT$_S$ can still beat DiscoNet while reaching only 28 ms inference time, which achieves real-time performance.

Sensitivity to localization error. To assess the models' sensitivity to pose error, we sample noises from Gaussian distribution with standard deviation $\sigma_{xyz} \in [0, 0.5]$ m, $\sigma_{heading} \in [0°, 1°]$. As Fig. 4.4 depicts, when the positional and heading errors stay within a normal range (*i.e.*, $\sigma_{xyz} \leq 0.2m, \sigma_{heading} \leq 0.4°$ [121, 120, 119]), the performance of V2X-ViT only drops by less than 3%, whereas other intermediate fusion methods decrease at least 6%. Moreover, the accuracy of *Early Fusion* and *Late Fusion* degrade by nearly 20% in AP@0.7.

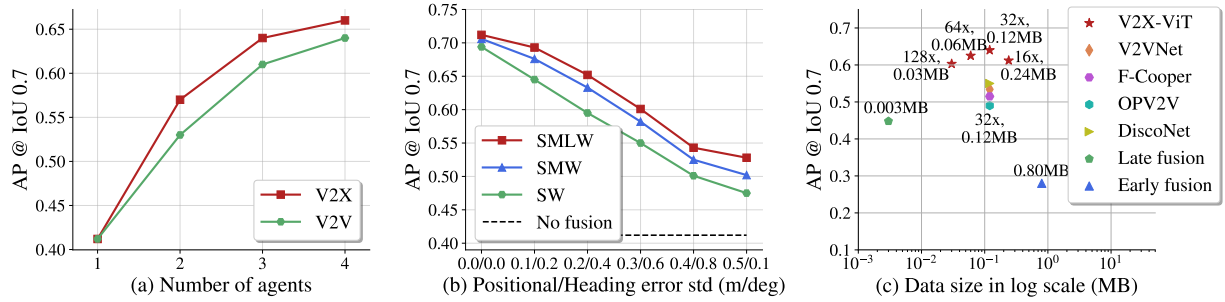


Figure 4.5: **Ablation studies.** (a) AP *vs.*number of agents. (b) MSwin for localization error with window sizes: 4^2 (S), 8^2 (M), 16^2 (L). (c) AP *vs.*data size.

Table 4.2: **Component ablation study**. MSwin, SpAttn, HMSA, DPE represent adding i) multi-scale window attention, ii) split attention, iii) heterogeneous multi-agent self-attention, and iv) delay-aware positional encoding, respectively.

MSwin	SpAttn	HMSA	DPE	AP0.5 / AP0.7
				0.719 / 0.478
✓				0.748 / 0.519
✓	✓			0.786 / 0.548
✓	✓	✓		0.823 / 0.601
✓	✓	✓	✓	**0.836** / **0.614**

Table 4.3: **Effect of DPE** w.r.t. time delay on AP@0.7.

Delay/Model	w/o DPE	w/ DPE
100 ms	0.639	0.650
200 ms	0.558	0.572
300 ms	0.496	0.514
400 ms	0.458	0.478

Table 4.4: **Inference time** measured on GPU Tesla V100.

Model	Time	AP0.7(prf/nsy)
V2X-ViT$_S$	28ms	0.696 / 0.591
V2X-ViT	57ms	0.712 / 0.614

When the noise is massive (*e.g.*, 0.5 m and 1°std), V2X-ViT can still stay around 60% detection accuracy while the performance of other methods significantly degrades, showing the robustness of V2X-ViT against pose errors.

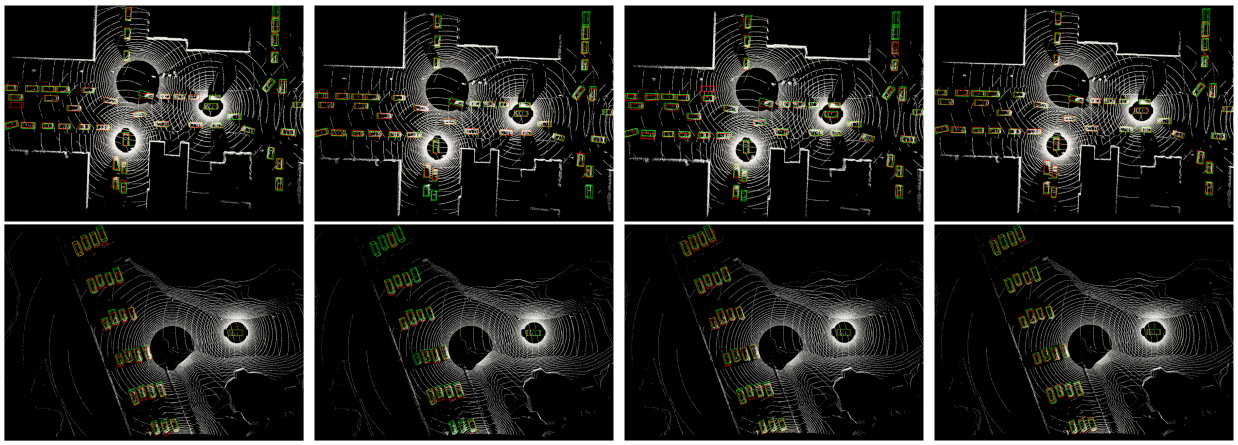
Time delay analysis. We further investigate the impact of time delay with range $[0, 400]$ ms. As Fig. 4.4c shows, the AP of *Late Fusion* drops dramatically below *No Fusion* with only 100 ms delay. *Early Fusion* and other intermediate fusion methods are relatively less sensitive, but they still drop rapidly when delay keeps increasing and are all below the baseline after 400 ms. Our V2X-ViT, in contrast, exceeds *No Fusion* by 6.8% in AP@0.7 even under 400 ms delay, which is much larger than usual transmission delay in real-world system[123]. This clearly demonstrates its great robustness against time delay.

Infrastructure *vs*.vehicles. To analyze the effect of infrastructure in the V2X system, we evaluate the performance between V2V, where only vehicles can share information, and V2X, where infrastructure can also transmit messages. We denote the number of agents as the total number of infrastructure and vehicles that can share information. As shown in Fig. 4.5a, both V2V and V2X have better performance when the number of agents increases. The V2X system has better APs compared with V2V in our collected scenes. We argue this is due to the better sight-of-view and less occlusion of infrastructure sensors, leading to more informative features for reasoning the environmental context.

Effects of transmission size. The size of the transmitted message can significantly affect the transmission delay, thereby affecting the detection performance. Here we study the model's detection performance with respect to transmitted data size. The data transmission time is calculated by $t_c = f_s/v$, where f_s denotes the feature size and transmission rate v is set to 27 Mbps [83]. Following [124], we also include another system-wise asynchronous delay that follows a uniform distribution between 0 and 200 ms. See supplementary materials for more details. From Fig. 4.5c, we can observe: 1) Large bandwidth requirement can eliminate the advantages of cooperative perception quickly, *e.g.*, *Early Fusion* drops to 28%, indicating the necessity of compression; 2) With the default compression rate (32x), our V2X-ViT outperforms other intermediate fusion methods substantially; 3) V2X-ViT is insensitive to large compression rate. Even under a 128x compression rate, our model can still maintain high performance.

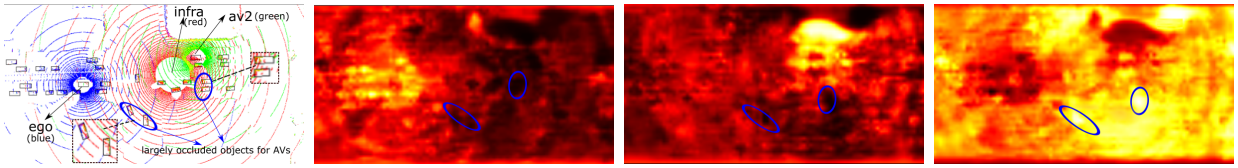
4.4.3 Qualitative evaluation

Detection visualization. Fig. 4.6 shows the detection visualization of OPV2V, V2VNet, DiscoNet, and V2X-ViT in two challenging scenarios under *Noisy setting*. Our model predicts highly accurate bounding boxes which are well-aligned with ground truths, while other approaches exhibit larger displacements. More importantly, V2X-ViT can identify more



(a) OPV2V [6] (b) V2VNet [15] (c) DiscoNet [98] (d) V2X-ViT (ours)

Figure 4.6: **Qualitative comparison in a congested intersection and a highway entrance ramp.** Green and red 3D bounding boxes represent the ground truth and prediction respectively. Our method yields more accurate detection results. More visual examples are provided in the supplementary materials.



(a) LiDAR points (better zoom-in) (b) attention weights ego paid to ego (c) attention weights ego paid to av2 (d) attention weights ego paid to infra

Figure 4.7: **Aggregated LiDAR points and attention maps for ego.** Several objects are occluded (blue circle) from both AV's perspectives, whereas infra can still capture rich point clouds. V2X-ViT learned to pay more attention to infra on occluded areas, shown in (d). We provide more visualizations in Appendix.

dynamic objects (more ground-truth bounding boxes have matches), which proves its capability of effectively fusing all sensing information from nearby agents. Please see Appendix for more results.

Attention map visualization. To understand the importance of infra, we also visualize the learned attention maps in Fig. 4.7, where brighter color means more attention ego pays. As shown in Fig. 4.7a, several objects are largely occluded (circled in blue) from both ego and AV2's perspectives, whereas infrastructure can still capture rich point clouds. There-

fore, V2X-ViT pays much more attention to infra on occluded areas (Fig. 4.7d) than other agents (Figs. 4.7b and 4.7c), demonstrating the critical role of infra on occlusions. Moreover, the attention map for infra is generally brighter than the vehicles, indicating more importance on infra seen by the trained V2X-ViT model.

4.4.4 Ablation studies

Contribution of major components in V2X-ViT. Now we investigate the effectiveness of individual components in V2X-ViT. Our base model is PointPillars with naive multi-agent self-attention fusion, which treats vehicles and infrastructure equally. We evaluate the impact of each component by progressively adding i) MSwin, ii) split attention, iii) HMSA, and iv) DPE on the *Noisy Setting*. As Tab. 4.2 demonstrates, all the modules are beneficial to the performance gains, while our proposed MSwin and HMSA have the most significant contributions by increasing the AP@0.7 4.1% and 6.6%, respectively.

MSwin for localization error. To validate the effectiveness of the multi-scale design in MSwin on localization error, we compare three different window configurations: i) using a single small window branch (SW), ii) using a small and a middle window (SMW), and iii) using all three window branches (SMLW). We simulate the localization error by combining different levels of positional and heading noises. From Fig. 4.5b, we can clearly observe that using a large and small window in parallel remarkably increased its robustness against localization error, which validates the design benefits of MSwin.

DPE Performance under delay. Tab. 4.3 shows that DPE can improve the performance under various time delays. The AP gain increases as delay increases.

4.5 Conclusion

In this paper, we propose a new vision transformer (V2X-ViT) for V2X perception. Its key components are two novel attention modules *i.e.*HMSA and MSwin, which can capture heterogeneous inter-agent interactions and multi-scale intra-agent spatial relationship. To evaluate our approach, we construct V2XSet, a new large-scale V2X perception dataset. Extensive experiments show that V2X-ViT can significantly boost cooperative 3D object detection under both perfect and noisy settings.

CHAPTER 5

CoBEVT: Cooperative Bird's Eye View Semantic Segmentation with Sparse Transformers

Bird's eye view (BEV) semantic segmentation plays a crucial role in spatial sensing for autonomous driving. Although recent literature has made significant progress on BEV map understanding, they are all based on single-agent camera-based systems. These solutions sometimes have difficulty handling occlusions or detecting distant objects in complex traffic scenes. Vehicle-to-Vehicle (V2V) communication technologies have enabled autonomous vehicles to share sensing information, dramatically improving the perception performance and range compared to single-agent systems. In this paper, we propose CoBEVT, the first generic multi-agent multi-camera perception framework that can cooperatively generate BEV map predictions. To efficiently fuse camera features from multi-view and multi-agent data in an underlying Transformer architecture, we design a fused axial attention module (FAX), which captures sparsely local and global spatial interactions across views and agents. The extensive experiments on the V2V perception dataset, OPV2V, demonstrate that CoBEVT achieves state-of-the-art performance for cooperative BEV semantic segmentation. Moreover, CoBEVT is shown to be generalizable to other tasks, including 1) BEV segmentation with single-agent multi-camera and 2) 3D object detection with multi-agent LiDAR systems, achieving state-of-the-art performance with real-time inference speed. The code is available at https://github.com/DerrickXuNu/CoBEVT.

5.1 Introduction

Autonomous vehicles (AVs) need the accurate surrounding perception and robust online mapping capabilities for robust and safe autonomy. AVs are normally located on the ground plane, so it is natural to represent semantic and geometric information of surroundings in the bird's eye view (BEV) maps. Projecting multi-camera views onto the holistic BEV space brings clear strengths in preserving the location and scale of road elements both spatially and temporally, which is critical for various autonomous driving tasks, including scene understanding and planning [125, 126]. It also presents a scalable vision-based solution for real-world deployment without relying on costly LiDAR sensors.

Map-view (or BEV) semantic segmentation is a fundamental task that aims to predict road segments from single- or multi-calibrated camera inputs. Significant efforts have been made toward precise camera-based BEV semantic segmentation. One of the most popular techniques is to leverage depth information to infer the correspondences between camera views and the canonical maps [127, 128, 129, 130]. Another family of works directly learns the camera-to-BEV space transformation, either implicitly or explicitly, using attention-based models [126, 131, 132, 133]. Despite the promising results, vision-based perception systems have intrinsic limitations – camera sensors are known to be sensitive to object occlusions and limited depth-of-field, which can lead to inferior performance in areas that are heavily occluded or far from the camera lens [126].

Recent advancements in Vehicle-to-Vehicle (V2V) communication technologies have made it possible to overcome the limitations of single-agent line-of-sight sensing. That is, multiple connected AVs can share their sensory information with each other through broadcasting, thereby providing multiple viewpoints of the same scene. Several prior works have demonstrated the efficacy of cooperative perception utilizing LiDAR sensors [6, 19, 98, 134]. Nevertheless, whether, when, and how this V2V cooperation can benefit camera-based perception systems has not been explored yet.

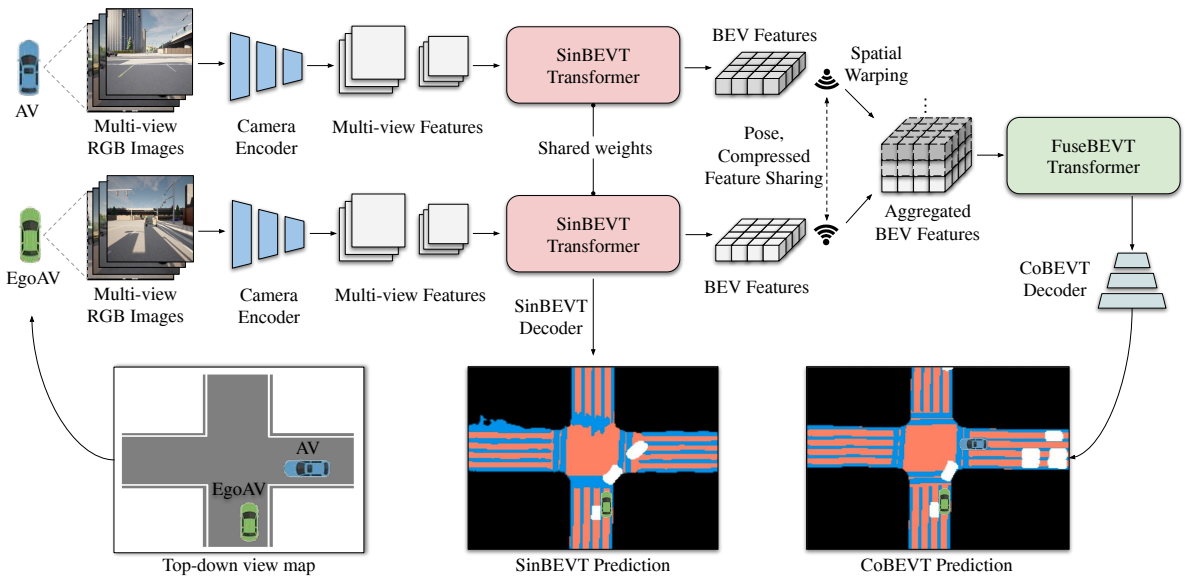


Figure 5.1: The overall framework of CoBEVT. White boxes in prediction maps indicate car segmentation results.

In this paper, we present CoBEVT, the first-of-its-kind framework that employs multi-agent multi-camera sensors to generate BEV segmented maps via sparse vision transformers cooperatively. Fig. 5.1 illustrates the proposed framework. Each AV computes its own BEV representation from its camera rigs with the SinBEVT Transformer and then transmits it to others after compression. The receiver (i.e. other AVs) transforms the received BEV features onto its coordinate system, and employs the proposed FuseBEVT for BEV-level aggregation. The core ingredient of these two transformers is a novel fused axial attention (FAX) module, which can search over the whole BEV or camera image space across all agents or camera views via local and global spatial sparsity. FAX contains global attention to model long-distance dependencies, and local attention to aggregate regional detailed features, with low computational complexity. Our extensive experiments on the V2V perception dataset [6] show that CoBEVT achieves performance gains of 22.7% and 6.9% over single-agent baseline and leading multi-agent fusion models, respectively.

Furthermore, we demonstrate the generalizability of the proposed framework in two additional tasks. First, we evaluate SinBEVT alone for single-agent multi-view BEV segmen-

tation. Second, we validate the attention fusion on a different sensor modality – multi-agent LiDAR fusion. Our experiments on the nuScenes dataset [3] and the LiDAR-track of OPV2V [6] show that CoBEVT exhibits outstanding performance and capably generalize to many other tasks. Our contributions are:

- We present the generic Transformer framework (CoBEVT) for cooperative camera-based BEV semantic segmentation. CoBEVT delivers superior performance and flexibility, achieving state-of-the-art results on multi-agent camera-based, single-vehicle multi-view BEV semantic segmentation, and multi-agent LiDAR-based 3D detection.
- We propose a novel sparse attention module called fused axial (FAX) attention, which can efficiently capture both local and global relationships between different agents or cameras. We build two instantiations – self-attention (FAX-SA) and cross-attention (FAX-CA) to accommodate different application scenarios.
- We construct a large-scale benchmark study on the cooperative BEV map segmentation task with a total of eight strong baseline models. Extensive experimental results and ablation studies show the strong performance and efficiency of the proposed model. All code, baselines, and pre-trained models will be released.

5.2 Related Work

5.2.1 V2V Perception

V2V perception leverages communication technologies to enable AVs to share their sensing information to enhance their perception. Previous works mainly focus on cooperative 3D object detection with LiDAR. A straightforward sharing strategy is to transmit raw point cloud (i.e. early fusion) [92] or detection outputs (i.e. late fusion) [96]. However, they either require a large bandwidth or ignore the context information. Recently, V2VNet [15] pro-

poses to circulate the intermediate features extracted from 3D backbones (i.e., intermediate fusion), then utilize a spatial-aware graph neural network for multi-agent feature aggregation. Following a similar transmission paradigm, OPV2V [6] employs a simple agent-wise single-head attention to fuse all features. F-Cooper [93] uses a simple `maxout` operation to fuse features. DiscoNet [98] explores knowledge distillation by constraining intermediate feature maps to match the correspondences in the early-fusion teacher model.

Compared to the previous multi-agent algorithms, our CoBEVT is the first to employ sparse transformers to explore the correlations between vehicles efficiently and exhaustively. Furthermore, previous approaches mainly focus on cooperative perception with LiDAR, while we aim to propose a low-cost camera-based cooperative perception solution free of LiDAR devices.

5.2.2 BEV Semantic Segmentation

BEV semantic segmentation aims to take camera views as input and predict a rasterized map with surrounding semantics under the BEV view. A common approach for this task is to use inverse perspective mapping (IPM) [135] to learn the homography matrix for view transformation [136, 137, 138]. As camera images lack explicit 3D information, another family of models includes depth estimation to inject auxiliary 3D information [127, 128, 125]. Recently, researchers start to directly model the image-to-map correspondence using transformers or MLPs. VPN [139] learns map-view transformation in a spatial MLP module on flattened camera-view image features. CVT [126] develops positional embedding for each individual camera depending on their intrinsic and extrinsic calibrations. BEVFormer [131] exploits the camera intrinsic and extrinsic explicitly to compute the spatial features in the regions of interest of the BEV grid across camera views using deformable transformer [140]. Our CoBEVT builds upon CVT but further improves on CVT with our proposed 3D FAX attention, which is more efficient and thus supports a larger BEV embedding size to retrieve

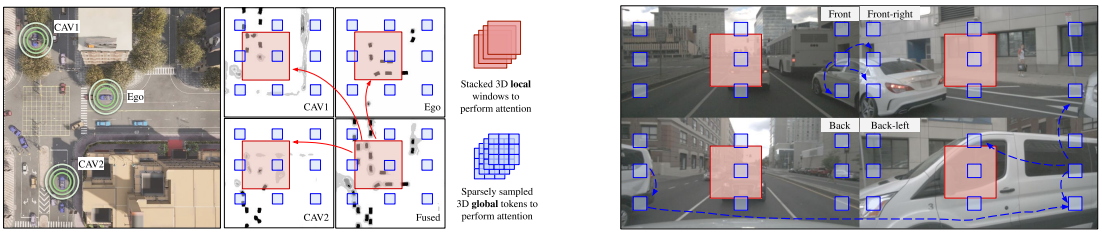
better accuracy. Furthermore, we developed a hierarchical architecture that can aggregate multi-scale camera features to preserve finer image details with only a low computational cost.

5.2.3 Transformers in Vision

Transformers are originally proposed for natural language processing [79]. ViT [105] has demonstrated for the first time that, a pure Transformer that simply regards image patches as visual words, is sufficient for vision tasks by large-scale pre-training. Swin Transformer [107] further improves the generality and flexibility of pure Transformers via restricting attention fields in local (shifted) windows. For high dimensional data, video Swin Transformer [141] extends the Swin approach onto shifted 3D space-time windows, achieving high performance with low complexity. Recent works have been focused on improving the architectures of attention models, including sparse attention [108, 142, 143, 144, 145, 111, 146], enlarged receptive fields [147, 148], pyramidal designs [149, 150, 151], efficient alternatives [152, 153, 154], etc. Our work belongs to efficient model designs of 3D Transformers for high dimensional data. While we have only validated the efficacy of the proposed FAX attention for multi-view and multi-agent autonomous perception, we expect its broad applications to other vision tasks such as video and multi-modality.

5.3 Methodology

We consider a V2V communication system where all AVs can exchange sensing information with others. Assuming the poses of all the agents are accurate and transmitted messages are synchronized, we propose a robust cooperative framework that can exploit the shared information across multiple agents to obtain a holistic BEV segmentation map. The overall architecture of CoBEVT is illustrated in Fig. 5.1, which consists of: SinBEVT for BEV fea-



(a) 3D FAX attention for multi-agent BEV fusion. (b) 3D FAX attention for multi-view fusion.

Figure 5.2: **Illustrated examples of fused axial attention (FAX) in two use cases – (a) multi-agent BEV fusion and (b) multi-view camera fusion**. FAX attends to 3D local windows (red) and sparse global tokens (blue) to attain location-wise and contextual-aware aggregation. In (b), for example, the white van is torn apart in three views (front-right, back, and back-left), our sparse global attention can capture long-distance relationships across parts in different views to attain global contextual understanding.

ture computation (Sec. 5.3.2), feature compression and sharing (Sec. 5.3.3), and FuseBEVT for multi-agent BEV fusion (Sec. 5.3.3). We propose a novel 3D attention mechanism called fused axial attention (FAX, Sec. 5.3.1) as the core component of SinBEVT and FuseBEVT that can efficiently aggregate features across agents or camera views both locally and globally. We will later show that this FAX attention has great generality, showing efficacy on different modalities for multiple perception tasks, including cooperative/single-agent BEV segmentation based on multi-view cameras and cooperative 3D LiDAR object detection.

5.3.1 Fused Axial Attention (FAX)

Fusing BEV features from multiple agents requires both local and global interactions across all agents' spatial positions. On the one hand, neighboring AVs often have different occlusion levels on the same object; hence, local attention, which cares more about details, can help construct pixel-to-pixel correspondence on that object. Take the scene in Fig. 5.2(a) as an example. The ego vehicle should aggregate all the BEV features per location from nearby AVs to obtain reliable estimates. On the other hand, long-term global contextual awareness can also assist in understanding the road topological semantics or traffic states – the road topology and traffic density ahead of the vehicle are often highly correlated with the one

behind. This global reasoning is also beneficial for multi-camera views understanding. In Fig. 5.2(b), for instance, the same vehicle is torn apart into multi-views, and global attention is highly capable of connecting them for semantic reasoning.

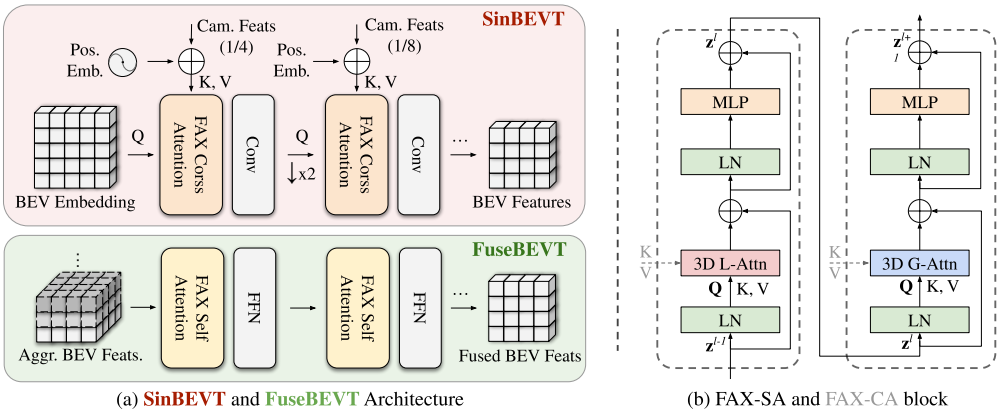
To attain such local-global properties efficiently, we propose a sparse 3D attention model called fused axial attention (FAX), which performs both local window-based attention and sparse global interactions, inspired by [141, 155, 113]. Formally, let $X \in \mathbb{R}^{N \times H \times W \times C}$ be the stacked BEV features with spatial dimension $H \times W$ from N agents. In the local branch, we partition the feature map into 3D non-overlapping windows, each of size $N \times P \times P$. The partitioned tensor of shape $(\frac{H}{P} \times \frac{W}{P}, \underline{N \times P^2}, C)$ is then fed into the self-attention model, representing mixing information along the second axis i.e., within local 3D windows [141]. Likewise, in the global branch, feature X is divided using a uniform 3D grid $N \times G \times G$ into the shape $(\underline{N \times G^2}, \frac{H}{G} \times \frac{W}{G}, C)$. Employing attention on the first axis of this tensor representing attending to sparsely sampled tokens [155, 113]. Fig. 5.2 illustrates the attended regions using red and blue colored boxes for local and global branches, respectively.

Combining this 3D local and global attention with typical designs of Transformers [105, 107, 141], including Layer Normalization (LN) [156], MLPs [105], and skip-connections, forms our proposed FAX attention block, as shown in Fig. 5.3b. Our 3D FAX attention only requires $\mathcal{O}(2(NP)^2HWC)$ complexity assuming $P \sim G$ (typically $N <= 5$, $P, G \in \{8, 16\}$), significantly cheaper than the full attention $\mathcal{O}((NHW)^2C)$. Still, it enjoys non-local 3D interactions by seeing through all the agents, which is more expressive than local attention approaches [107, 141]. The 3D FAX self-attention (FAX-SA) block can be expressed as:

$$\hat{\mathbf{z}}^{\ell} = \text{3DL-Attn}(\text{LN}(\mathbf{z}^{\ell-1})) + \mathbf{z}^{\ell-1}, \qquad \mathbf{z}^{\ell} = \text{MLP}(\text{LN}(\hat{\mathbf{z}}^{\ell})) + \hat{\mathbf{z}}^{\ell}, \tag{5.1}$$

$$\hat{\mathbf{z}}^{\ell+1} = \text{3DG-Attn}(\text{LN}(\mathbf{z}^{\ell})) + \mathbf{z}^{\ell}, \qquad \mathbf{z}^{\ell+1} = \text{MLP}(\text{LN}(\hat{\mathbf{z}}^{\ell})) + \hat{\mathbf{z}}^{\ell}, \tag{5.2}$$

where $\hat{\mathbf{z}}^{\ell}$ and $\mathbf{z}^{\ell}$ denote the output features of the 3DL(G)-Attn module and MLP module for block ℓ. The 3DL-Attn and 3DG-Attn represent the above-defined 3D local and global



(a) SinBEVT and FuseBEVT Architecture (b) FAX-SA and FAX-CA block

Figure 5.3: Architectures of (a) SinBEVT and FuseBEVT, and (b) the FAX-SA and FAX-CA block.

attention, respectively.

5.3.2 SinBEVT for Single-agent BEV Feature Computation

Given monocular views from m cameras on the i-th agent $(I_k^i, K_k^i, R_k^i, t_k^i)_{k=1}^m$ denoting input images $I_k \in \mathbb{R}^{h \times w \times 3}$, camera intrinsic $K_k \in \mathbb{R}^{3 \times 3}$, rotation extrinsic $R_k \in \mathbb{R}^{3 \times 3}$, and translation $t_k \in \mathbb{R}^3$, every agent needs to compute a BEV feature representation $\mathbf{F}_i \in \mathbb{R}^{H \times W \times C}$ (height H, width W, and channels C) before any cross-agent collaboration. $\mathbf{F}_i$ can be either fed into a decoder to perform single-agent predictions or shared to the ego vehicle for multi-agent feature fusion.

We take a BEV processing architecture similar to CVT [126], wherein a learnable BEV embedding is initialized as the query to interact with encoded multi-view camera features, as shown in Fig. 5.3a. We have observed that CVT uses a low-resolution BEV query that fully cross-attends to image features, which leads to degraded performance on small objects, despite being efficient. Thus, CoBEVT learns a high-resolution BEV embedding instead, then uses a hierarchical structure to refine the BEV features with reduced resolution. To efficiently query features from camera encoders at high resolution, the FAX-SA module is further extended to build a FAX cross-attention (FAX-CA) module (Fig. 5.3b), in which

the query vector is obtained using the BEV embedding, whereas the key/value vectors are projected by multi-view camera features. Before applying cross-attention, we add a camera-aware positional encoding derived from camera intrinsics and extrinsic, to learn implicit geometric reasoning from individual camera views to a canonical map-view representation, following CVT. This rather simple, implicit approach demonstrates a good balance of performance and efficiency, and our FAX attention allows for global interactions in a hierarchical network, showing better accuracy against low-resolution isotropic approaches such as CVT.

5.3.3 FuseBEVT for Multi-agent BEV Feature Fusion

Feature Compression and Sharing. Transmission data size is critical to V2V applications, as large bandwidth requirements will likely cause severe communication delays. Therefore, it is necessary to compress the BEV features before broadcasting. Similar to [19, 98], we apply a simple 1x1 auto-encoder [157] to compress and decompress the BEV features. Once receiving the broadcasted messages that contain intermediate BEV representations, and the pose of the sender, the ego vehicle applies a differentiable spatial transformation operator $\mathbf{\Gamma}_\xi$, to geometrically warp the received features [115] onto the ego's coordinate system: $\mathbf{H}_i = \mathbf{\Gamma}_\xi\left(\mathbf{F}_i\right) \in \mathbb{R}^{H\times W\times C}$.

Feature Fusion. We design a customized 3D vision Transformer called FuseBEVT that can attentively fuse information of the received BEV features from multiple agents. The ego vehicle first stacks the received and projected BEV features $\mathbf{H}_i,\ i = 1, ..., N$ into a high dimensional tensor $\mathbf{h} \in \mathbb{R}^{N\times H\times W\times C}$, then feeds them into the FuseBEVT encoder which consists of multiple layers of FAX-SA blocks (Fig. 5.3a). Benefiting from the linear complexity of FAX attention (Sec. 5.3.1), this agent-wise fusion Transformer is also efficient. Each FAX-SA block conducts a 3D global and local BEV feature transformation via Eqs. 5.1-5.2. As exemplified in Fig. 5.2(a), the 3D FAX-SA can attend to the same region of estimations (red boxes) drawn from multiple agents to derive the final aggregated representations. More-

over, the sparsely sampled tokens (blue boxes) can interact globally to attain a contextual understanding of the map semantics such as road, traffic, etc.

Decoder. We apply a series of lightweight convolutional layers and bi-linear upsampling operations on the aggregated BEV representation and generate the final segmentation output.

5.4 Experiments

We evaluate the effectiveness of the proposed CoBEVT on the camera track of the V2V perception dataset OPV2V [6]. To show the flexibility and generality of our CoBEVT, we also conduct experiments on the LiDAR track of OPV2V and the autonomous driving dataset nuScenes [3].

5.4.1 Datasets and Evaluations

OPV2V is a large-scale V2V perception dataset that is collected in CARLA [8] and the cooperative driving automation tool OpenCDA [7]. It contains 73 diverse scenarios, which have an average of 25 seconds duration. In each scenario, various numbers (2 to 7) of AVs show up simultaneously, and each one is equipped with one LiDAR sensor and 4 cameras in different directions to cover 360° horizontal field-of-view. Our main experiment only utilizes the camera rigs of the dataset, and we use Intersection over Union (IoU) between map prediction and ground truth map-view labels as the performance metric. Since OPV2V has multiple AVs in the same scene, we select a fixed one as the ego vehicle during testing and evaluate the 100m×100m area around it with a 39cm map resolution.

To demonstrate its generality, we also evaluated our proposed CoBEVT on the OPV2V LiDAR-track 3D detection task. We use the same evaluation range in [6, 21], and the detection performance is measured by Average Precisions (AP) at an IoU threshold of 0.7.

For both camera and LiDAR track, there are 6764/1981/2719 frames for train/validation/test set, respectively.

The nuScenes dataset contains 1000 diverse scenes, each of around 20 seconds long. In total, there are 40K sampled frames in this dataset, and the dumped data captures a 360° view of surroundings using 6 cameras. We use the groundtruth in [126]. The evaluation ranges are [-50m, 50m] for the X and Y axis, and the resolution of the BEV grid is 0.5m.

5.4.2 Experiments Setup

Implementation details. We assume all the AVs have a 70m communication range following [15], and all the vehicles out of this broadcasting radius of ego vehicle will not have any collaboration. For the OPV2V camera-track,we choose ResNet34 [60] as the image feature extractor in SinBEVT. The transmitted BEV intermediate representation has a resolution of $32 \times 32 \times 128$. For the multi-agent fusion, our FuseBEVT component has 3 encoded layers and a window size of 8 for both local and global attention. We train the whole model end-to-end with Adam [158] optimizer and cosine annealing learning rate scheduler [159]. We use weighted cross entropy loss and train all models with 90 epochs, with a batch size of 1 per GPU. Please refer to the supplementary materials for more details, as well as the configurations on nuScenes and OPV2V LiDAR-track.

Compared methods. For multi-agent perception task, we consider single-agent perception system *No Fusion* as the baseline. We compare with the state-of-the-art multi-agent perception algorithms: F-Cooper [93], AttFuse [6], V2VNet [15], and DiscoNet [98]. We also implement a straightforward fusion strategy *Map Fusion*, which transmits the segmentation map instead of BEV features and fuses all maps by selecting the closest agent's prediction for each pixel.

For the nuScenes dataset, we compare against state-of-the-art models including CVT [126], FIERY [127], View Parsing Network (VPN) [139], Orthographic Feature Transform (OFT) [160],

Table 5.1: **Map-view segmentation on OPV2V camera-track.** We report IoU for all classes. All fusion methods employs CVT [126] backbone, except for CoBEVT which uses SinBEVT backbone.

Method	Veh.	Dr.Area	Lane
No Fusion	37.7	57.8	43.7
Map Fusion	45.1	60.0	44.1
F-Cooper [93]	52.5	60.4	46.5
AttFuse [6]	51.9	60.5	46.2
V2VNet [15]	53.5	60.2	47.5
DiscoNet [98]	52.9	60.7	45.8
FuseBEVT	**59.0**	**62.1**	**49.2**
CoBEVT	**60.4**	**63.0**	**53.0**

Table 5.2: **3D detection results on the OPV2V LiDAR-track**. All methods employ the PointPillars [1] backbone. (C) denotes using 64× feature compression.

Method	AP0.7	AP0.7(C)
No Fusion	60.2	60.2
Late Fusion	78.1	78.1
Early Fusion	80.0	-
F-Cooper	79.0	78.8
AttFuse	81.5	81.0
V2VNet	82.2	81.4
DiscoNet	83.6	83.1
FuseBEVT	**85.2**	**84.9**

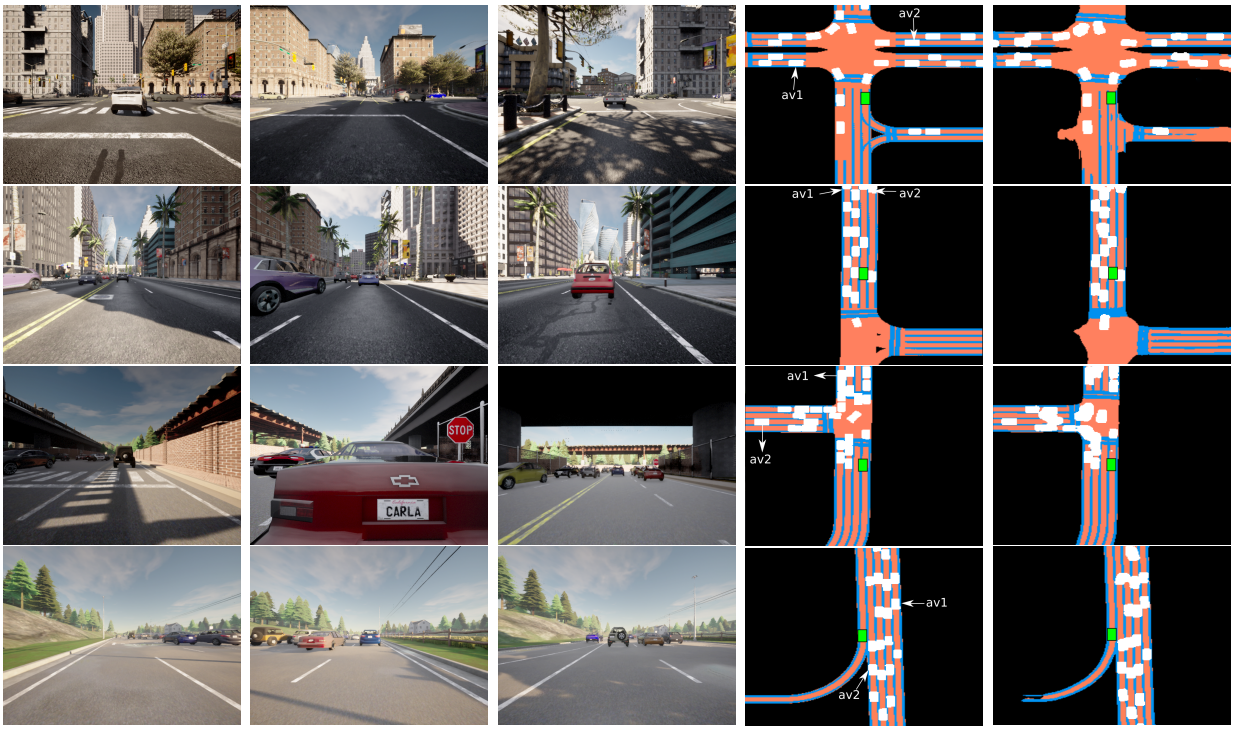
Table 5.3: **Vehicle map-view segmentation on nuScenes.** All models use only a single timestamp. ∗ denotes our reproduced result with the EfficientNet-b4 backbone.

Method	Veh.	Par(M)	FPS
VPN* [139]	29.3	4.	31
OFT [160]	30.1	-	-
Lift-Splat	32.1	14	25
FIERY [127]	35.8	7	8
CVT [126]	36.0	1.2	35
SinBEVT	**37.1**	1.6	35

and Lift-Splat-Shoot [128]. All models only utilize single-step timestamp data for fair comparisons. We intentionally use the same image feature extractor Efficient-B4 [161] and decoder as CVT and FIERY.

5.4.3 Quantitative Evaluation

OPV2V camera-track results. To make a fair comparison, we first employ CVT [126] to extract the BEV feature from camera rigs for all methods and only use the fusion component (i.e. FuseBEVT) of CoBEVT to compare with other fusion models. Then we compare it with our complete CoBEVT to show the effectiveness of SinBEVT as well. As shown in Tab. 5.1, all cooperative methods perform better than *No Fusion*, which proves the benefits from multi-agent perception system. Among all fusion models, our FuseBEVT achieves the best IoU for all classes, outperforming the second-best method by 5.5%, 1.4%, and 3.4% on vehicle, drivable area, and lane, respectively. More importantly, by replacing the CVT with our SinBEVT for feature extraction, our CoBEVT can further increase the accuracy



(a) ego (b) av1 (c) av2 (d) groundtruth (e) prediction

Figure 5.4: **Qualitative results of CoBEVT.** From left to right: the front camera image of (a) ego, (b) av1, (c) av2, (d) groundtruth and (e) prediction. The green bounding boxes represent ego vehicles, while the white boxes denote the segmented vehicles. CoBEVT demonstrates robust performance under various traffic situations and road types. It is also capable of detecting occluded or distant vehicles (white circled) benefiting from the collaboration.

by 1.4%, 0.9%, and 3.8% on the three classes compared to using FuseBEVT only.

OPV2V LiDAR-track results. As Tab. 5.2 reveals, our FuseBEVT also has the best performance on the LiDAR-track task, which improves the single-agent system by 25.0% and outperforms the leading algorithm DiscoNet by 1.7%. Furthermore, our method exhibits great robustness against LiDAR feature compression, with only a 0.3% drop with the 64× compression rate.

nuScenes vehicle map-view segmentation. Our SinBEVT can run 35 FPS on RTX2080 with 37.1 IoU score and 1.6 M parameters, achieving the best accuracy with real-time performance. Compared to the state-of-the-art method CVT, we are 1.1% higher with similar

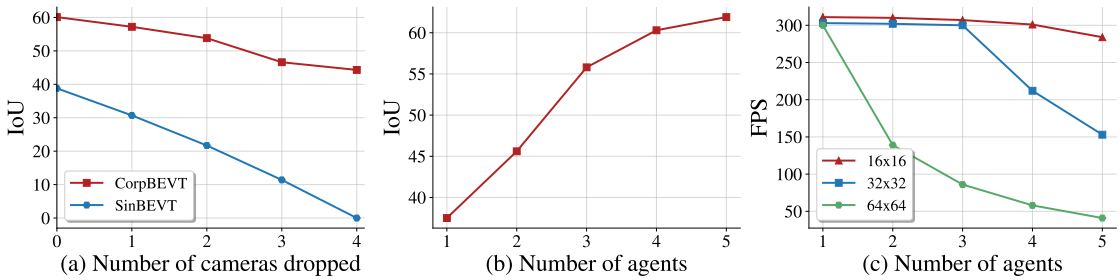


Figure 5.5: **Ablation studies.** (a) IoU *vs.* number of dropped cameras (b) IoU *vs.* number of agents. (c) FPS *vs.* number of agents. The channel dimension of BEV feature map is fixed as 128 for (c).

parameters and latency.

Effect of compression rate. Data transmission size is a critical factor in V2V applications. Here we study the effect of different compression rates on our CoBEVT by adjusting the 1×1 convolution. Tab. 5.4 shows that CoBEVT is insensitive to compression, and it can still beat other fusion methods even with a large compression rate of 64.

Table 5.4: **Compression effect on OPV2V Camera.**

CPR-rate	Size (KB)	IoU
0x	524	60.4
8x	66	60.1
16x	33	58.9
32x	16	56.2
64x	8	54.8

5.4.4 Qualitative Analysis

Fig. 5.4 shows the qualitative results of CoBEVT on scenes containing 3 AVs. In each row, we draw the front camera image of each AV along with the ground truth and prediction pairs. Our framework can overcome most of the occlusions and perceive distant objects accurately, benefiting from our Transformer design that learns from all agents and views. However, one limitation we have observed is the "merging" predictions of multiple nearby vehicles, which may be attributed to the combined effects of low-resolution BEV embedding and the complicated ground truth in dense traffic.

5.4.5 Ablation Study

Robustness to camera dropout. Sensor failure during driving can lead to fatal accidents. Therefore, here we investigate how well our CoBEVT handles it. We random drop $n \in [1, 4]$ cameras of the ego vehicle, and demonstrate the performance decrease for both SinBEVT (no collaboration) and CoBEVT in Fig. 4.5a. It can be seen that by introducing sensing cooperation, driving safety can be significantly improved, as even if all ego cameras break down, CoBEVT can still reach an IoU score of 44.3.

Number of agents. Here we study the influence brought by the number of collaborators on CoBEVT. As Fig. 5.5b describes, increasing the collaborators can generally bring performance improvement, whereas such gain will be marginal when the agent number is greater than 4.

Inference speed of FuseBEVT. Real-time multi-agent feature fusion is critical for real-world deployment. Here we examine the inference speed of FuseBEVT with different BEV feature map spatial resolution (from 16 to 64) and the number of agents on RTX3090. Fig. 5.5c shows that our fusion algorithm can achieve real-time performance under distinct collaboration scenarios.

5.5 Conclusion and Limitations

In this paper, we propose a holistic vision Transformer dubbed CoBEVT for multi-view cooperative semantic segmentation. We propose a fused axial attention (FAX) mechanism that allows for local and global interactions across all views and agents. Extensive experiments on both simulated and real-world datasets show that CoBEVT achieves superior performance on multi-camera cooperative BEV segmentation. It can also be adapted to other tasks and substantially improve multi-agent LiDAR detection and single-agent map-view segmentation.

Limitations. Despite the proposed single-agent model outperforming the real-world nuScenes dataset, the entire cooperative framework has been trained and validated on simulated datasets only, and thus its real-world generalization capability remains unknown. The proposed approach does not explicitly model realistic V2V challenges such as asynchronization and position errors, which may impair its robustness under these noises. The perception robustness against different domains such as severe weather or lighting conditions needs further examination. Addressing these limitations needs future research on real-world, realistic, and diverse cooperative datasets and benchmarks.

CHAPTER 6

Bridging the Domain Gap for Multi-Agent Perception

Existing multi-agent perception algorithms usually select to share deep neural features extracted from raw sensing data between agents, achieving a trade-off between accuracy and communication bandwidth limit. However, these methods assume all agents have identical neural networks, which might not be practical in the real world. The transmitted features can have a large domain gap when the models differ, leading to a dramatic performance drop in multi-agent perception. In this paper, we propose the first lightweight framework to bridge such domain gaps for multi-agent perception, which can be a plug-in module for most of the existing systems while maintaining confidentiality. Our framework consists of a learnable feature resizer to align features in multiple dimensions and a sparse cross-domain transformer for domain adaption. Extensive experiments on the public multi-agent perception dataset V2XSet have demonstrated that our method can effectively bridge the gap for features from different domains and outperform other baseline methods significantly by at least 8% for point-cloud-based 3D object detection.

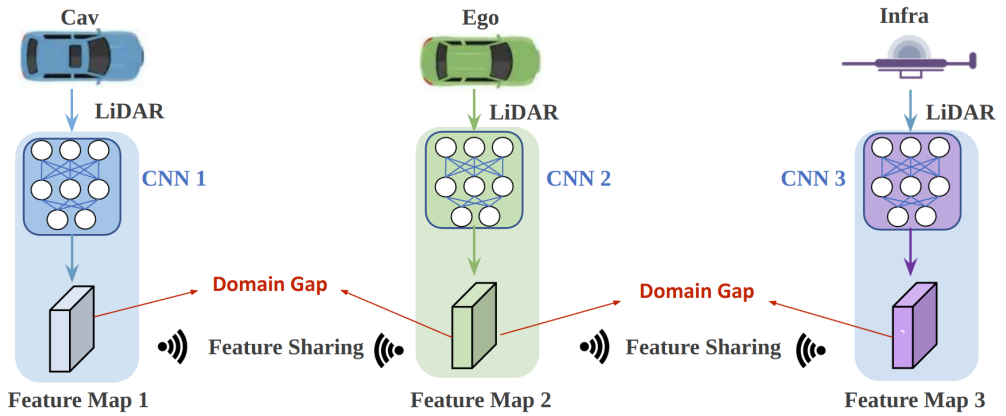
6.1 INTRODUCTION

Recent studies have demonstrated that by leveraging Vehicle-to-Everything (V2X) communication technology to share visual information, the multi-agent perception system can significantly improve the performance of the single-agent system by seeing through occlusions and perceiving longer range [6, 20, 19, 162, 98, 163, 134, 164]. Instead of sharing raw

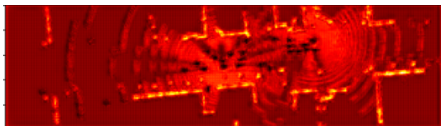
sensing data or detected outputs, state-of-the-art methods usually share the intermediate neural features computed from the sensor data, as they can achieve the best trade-off between accuracy and bandwidth requirements [15, 6]. Furthermore, transmitted intermediate features are more robust to the GPS noise and communication delay [19, 165, 164]. Despite the advancements in intermediate fusion strategy, previous methods conduct experiments under a strong assumption that all agents are equipped with identical neural networks to extract neural features. This overlooks a critical fact: deploying the same model for all agents is unrealistic, especially for connected autonomous driving [166, 167]. For example, as shown in (a) from Fig. 6.1, the detection models on connected automated vehicles (CAV) and infrastructure products of distinct companies are usually dissimilar. Even for the same company, diverse detection models may exist due to the different on-vehicle software versions. When the shared features come from different backbones, a noticeable domain gap exists, which can easily diminish the benefits of collaborations.

In this paper, we dive into this unsolved and practical problem in multi-agent perception, especially for autonomous driving. We first carefully investigate the domain gap of different feature maps and then propose our framework based on the analysis. Fig. 6.1 shows intermediate feature representations obtained from two distinct point cloud based 3D object detection backbones, PointPillar [1] and VoxelNet [2], in the same scenario. We apply the same techniques as [168] to make the visualization informative by summing up all channels' absolute value together. In general, we can observe the features are dissimilar in three aspects:

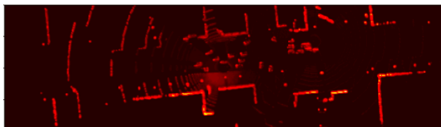
- **Spatial resolution.** Because of the different voxelization parameters, LiDAR cropping range, and downsampling layers, the spatial resolutions are different.
- **Channel number.** The channel dimensions are distinct due to the difference in convolution layers' settings.



(a) Multi-agent perception pipeline in the context of V2X perception



(b) PointPillar feature map



(c) VoxelNet feature map

Figure 6.1: **Illustration of domain gap of different feature maps for multi-agent perception.**(a) Ego vehicle receives the shared feature maps from other CAV and infrastructure with different CNN models, which causes domain gaps. (b) Visualization of feature map from ego, which is extracted from PointPillar [1]. (c) Feature map from CAV, which is extracted from VoxelNet [2]. Brighter pixels represent higher feature values.

- **Patterns.** As Fig. 6.1 shows, PointPillar and VoxelNet have the opposite patterns: The object positions have relatively low values on the feature map for PointPillar but high values for VoxelNet.

To address the three dominant distinctions, we present the first **M**ulti-agent **P**erception **D**omain **A**daption framework, dubbed as MPDA, to bridge the domain gap. Fig. 6.2 depicts the overall architecture. Specifically, two components, namely Learnable Resizer and Sparse Cross-Domain Transformer, are proposed. As multiple factors could cause different spatial resolutions, we argue that using rudimentary resizing algorithms such as bilinear and nearest interpolation may cause severe misalignment. Therefore, we propose to resize the received intermediate features in a learnable way and optimize with the multi-agent fusion algorithms jointly to improve the detection performance. Moreover, aligning the channel dimension by

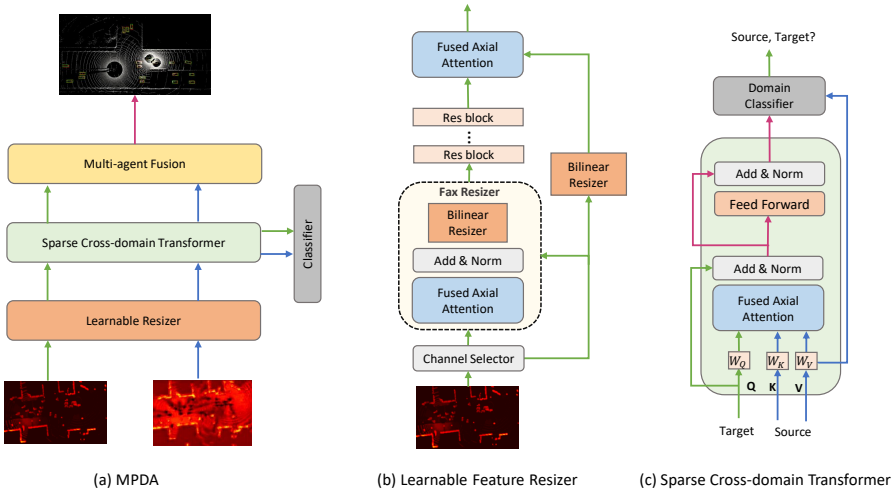


Figure 6.2: **The overview and core components of our framework.** Our MPDA first aligns feature dimensions through a learnable feature resizer and then unifies the pattern through the sparse cross-domain transformer.

simply dropping channels can potentially lead to losing important information; thus, our resizer also includes a learnable channel selector to alleviate such loss. To diminish the pattern disparity, the sparse cross-domain transformer will efficiently reason the received and ego features locally and globally and generates domain-invariant representations by adversarially fooling a domain classifier. Finally, the state-of-the-art multi-agent fusion algorithm V2X-ViT [19] is utilized to fuse information across multiple agents. Since the framework does not require any key information from other models (e.g., model type, parameters), it can maintain confidentiality. We conduct extensive experiments on the public dataset V2XSet [19], and the result demonstrates that our framework can increase the accuracy of V2X-ViT by at least 8% under various realistic settings. Overall, our contributions are summarized as follows:

- We pioneer the domain gap identification (spatial resolution, channel number, pattern) in multi-agent perception and propose a new Multi-agent Perception Domain Adaption (MPDA) framework, which is the **first work to bridge the domain gap for multi-agent perception**.

- We present a novel Learnable Resizer to better align spatial and channel features from other agents in an adaptive way.
- We propose a sparse cross-domain transformer that can efficiently unify the feature patterns from various agents.
- The proposed MPDA framework can be easily combined with other multi-agent fusion algorithms and does not require confidential model information from other agents. Extensive experiments on the public dataset V2XSet demonstrate that our method achieves the best performance with real-time performance.

6.2 Related Work

Multi-Agent Perception: Despite the great progress in autonomous driving in the past years, there still exist many challenges that single-vehicle systems can not get over [169, 170, 171, 172, 173, 174, 175, 176, 177, 178]. Especially, the single-agent perception systems suffer severely from occlusions and limitations in sensor range [179]. Multi-agent perception was born to alleviate such pains. As a pioneering work, V2VNet [15] first proposes the intermediate fusion approach, in which all agents should broadcast the features extracted from the raw point cloud to achieve the trade-off between bandwidth requirement and accuracy. Following this design ethos, OVP2V [6] uses a single-head self-attention module to fuse the received intermediate features. DiscoNet [98] utilizes a graph neural network and knowledge distillation to aggregate the shared representations. Recently, V2X-ViT [19] first proposed to use a vision transformer for multi-agent perception and achieves robust performance under GPS error and communication delay. [180] captures both aleatoric and epistemic uncertainties with one inference pass and tailors a moving block bootstrap algorithm with direct modeling of the multivariant Gaussian distribution of each corner of the bounding box. The method can be used with different collaborative object detectors and helps to improve safety-critical

systems such as CAVs. Despite the prominent performance achieved by these methods, none of them consider the realistic domain gap issue caused by the model discrepancy. We aim to fill such a gap in this paper.

Transformers in Vision: Since Dosovitskiy *et al.* [105] successfully adapt the Transformer architecture [79] into the computer vision area by regarding image patches as visual words, Vision Transformers (ViT) have gained increasing attention [181, 182, 183]. For example, [184] shows that by applying a 3D attention mechanism, the object detection performance can outperform traditional convolutional neural networks. Despite obtaining great benefits from global interactions, the full attention in ViT [79, 105] usually requires large computation resources. To avoid such costs, recent methods [107, 106, 113] have explored different sparse attention mechanisms such as local and sparsely global schemes. In this work, we adapt an efficient 3D attention called Fused Axial Attention (FAX) in CoBEVT [20] to our domain adaption framework as it already shows excellent efficiency on multi-agent fusion.

Domain Adaptation: Due to the time consumption of data annotation and the domain gap between different domains, domain adaptation is utilized to solve these problems by adapting the model trained on a labeled source domain to address an unlabeled target domain. Recent works on domain adaptation mainly address different computer vision tasks [185, 186, 187, 188, 189, 190, 191, 192, 90, 193].

In domain adaptation, to minimize the domain shift between different domains, feature distribution can be aligned in common levels: domain level [194, 195, 196] and category level [190, 187, 197, 198, 199]. Domain level alignment generally involves minimizing some measure of distance between the source and target feature distributions like maximum mean discrepancy [194]. [200] proposes a novel prototype-based shared-dummy classifier (PSDC) model to address the challenges of open-set domain adaptation, including distinguishing between unknown target instances and shared classes and aligning shared class prototypes, which outperforms existing methods on several datasets. [201] proposes a novel

class-Balanced Multicentric Dynamic prototype (BMD) strategy for the Source-free Domain Adaptation (SFDA) task to adapt pre-trained source models to the target domain without accessing the well-labeled source data. The proposed BMD strategy avoids the gradual dominance of easy-transfer classes on prototype generation, introduces a novel inter-class balanced sampling strategy, and incorporates dynamic network update information during model adaptation. While category level alignment aligned each category distribution between source and target domain using an adversarial manner between the feature extractor and domain classifiers. A fine-grained alignment leads to more accurate distribution alignment in the same label space. Xu et al. [190] adopted Transformers for category-level domain adaptation to show great potential in image classification. In this paper, similar to [190], a sparse cross-domain Transformer is proposed to unify the feature patterns from different agents.

Learnable Resizer: [202] first comes up the concept of learnable resizer for image classifications. Instead of using rudimentary interpolation, they employ a convolution neural network to resize the RGB images for classification and jointly train with vision models. Our learnable feature resizer is inspired by this work but differs in three major aspects: 1) We investigate an unexplored practical application scenario for a learnable resizer – domain adaption for multi-agent perception. 2) Our resizing target is the LiDAR feature, which is more sparse than images. Therefore, instead of using a pure convolution neural network, we integrate our resizer with a sparse transformer. 3) Besides resizing the spatial dimension, we also embed a simple but effective algorithm to resize the channel dimension to the required number.

6.3 Methodology

In this paper, we consider a realistic scenario for multi-agent perception, where each agent in the collaboration may be equipped with a separate model and transmit visual features with

domain discrepancy. We mainly focus on the cooperative perception task of LiDAR-based 3D object detection for autonomous driving, where the agents are connected to autonomous vehicles and intelligent roadside infrastructure, but our framework is generally-applicable to other multi-agent perception applications as long as they broadcast neural features for collaborations. Since we focus on the problem of domain gaps in this work, we assume the relative poses between agents are accurate and no communication delay exists.

Fig. 6.2(a) shows the overall architecture of our MPDA, which consists of 1) a learnable feature resizer, 2) a sparse cross-domain transformer, 3) a domain classifier, and 4) multi-agent feature fusion. In this section, we will describe the details of each module.

6.3.1 Learnable Feature Resizer

We regard the feature maps computed locally on ego vehicle as source domain features $F_S \in \mathbb{R}^{1\times H_S\times W_S\times C_S}$ and received features from other agents as target domain features $F_T \in \mathbb{R}^{N\times H_T\times W_T\times C_T}$, where N is the number of other collaborators/agents, H is the height, W is the width, C is the channel number, and $H_S \neq H_T, W_S \neq W_T, C_S \neq C_T$. The goal of our feature resizer Φ is to align the dimensions of the source domain feature with the target domain in a learnable way:

$$F_T^{'} = \Phi(F_T),\ \text{s.t. } F_T^{'} \in \mathbb{R}^{N\times H_S\times W_S\times C_S}. \tag{6.1}$$

We jointly train Φ with multi-agent detection models so it can intelligently learn the optimal approach to resize the features, which is fundamentally different from the naive resizing method such as bilinear interpolation. The architecture of our learnable feature resizer is designed as Fig 6.2(b) shows, which includes four major components: channel aligner, FAX resizer, skip connection, and res-block.

Channel Aligner: We use a simple 1×1 convolution layer to align the channel dimension,

whose input channel number is $C_{in} = 2C_S$ and outputs C_S channels. When $C_T > C_{in}$, we randomly drop $C_{in} - C_T$ channels and apply the 1×1 convolution layer to obtain a new feature. We repeat this process on F_T for n times to get features with $n \times H_T \times W_T \times C_S$ dimensions and average them along the first dimension. In this way, we ameliorate the loss of information due to channel dropping. When $C_T < C_{in}$, we perform padding with randomly selected channels from F_T to meet the required input channel number for the 1×1 convolution.

FAX Resizer: To search for the optimal resizing solution, the neural network is supposed to have a large receptive field to gain the global information and pay attention to details to capture the critical object information. Since LiDAR features are usually sparse due to empty voxels, applying large-kernel convolution to get global information may diffuse the meaningless information to the important area. Therefore, we apply the fused axial (FAX) attention block [20] before bilinear resizing to fetch better feature representations. FAX sparsely employs local window and grid attention to efficiently capture global and local interactions. More importantly, it can discard empty voxels through a dynamic attention mechanism to eliminate their potential negative effects. After FAX, a bilinear resizer is implemented to reshape the feature map to the same spatial dimension as the source feature map. Compared to simple bilinear interpolation, our FAX resizer can adjust the input features first to avoid misalignment and distortion issues during resizing.

Skip connection: We also employ the bilinear feature resizing method in the skip connection to make learning easier.

Res-Block: We implement standard residual blocks [60] r times after resizing the feature maps to further refine them.

6.3.2 Sparse Cross-Domain Transformer

After retrieving the resized feature $F_T^{'}$, we need to convert its pattern to be indistinguishable from the domain classifier to obtain the domain-invariant features. To reach this goal, we need to effectively reason the correlations between $F_T^{'}$ and F_S both locally and globally. Therefore, we propose the sparse cross-domain transformer, which enjoys the benefits of dynamic and global attention brought by the transformer architecture while avoiding expensive computation. Fig. 6.2(c) shows the details of our proposed architecture. We first apply different convolution layers W_Q, W_K, W_V on $F_T^{'}$ and F_S to obtain query, key, and value, respectively. Then the query from the target domain and key/value from the source domain will be fed into the FAX block, capturing sparsely local and global spatial interactions across target and source domain features. Finally, a standard feed-forward neural network (FFN) is implemented to refine the interacted feature further. The whole process can be formulated as below:

$$Q = W_Q(F_T^{'}), \quad K = W_K(F_S), \quad V = W_V(F_S), \tag{6.2}$$

$$\hat{F_T^{'}} = Q + LN(FAX(Q, K, V)), \tag{6.3}$$

$$F_T^{''} = \hat{F_T^{'}} + LN(FFN(\hat{F_T^{'}})), \tag{6.4}$$

where LN is layer normalization, Q is the query, K is the key, and V is the value. Afterward, we pair $F_T^{''}$ and F_S together and send them the domain classifier and multi-agent fusion module.

6.3.3 Domain Classifier

We use the H-divergence [203] to measure the divergence between $F_T^{''}$ and F_S. Let us denote X as a feature map that may come from the source or target domain and $h : X \to \{0, 1\}$ a domain classifier, which tries to predict source domain sample X_S as 0 and target domain

sample X_T as 1. In our paper, the domain classifier comprises two convolution layers. Suppose H is the hypothesis space for the domain classifier and G is the combination of our learnable resizer and sparse cross-domain transformer, then G needs to be optimized towards the following objective:

$$\max_{G} \min_{h \in H} (\mathbf{E}_S(h(X)) + \mathbf{E}_T(h(X)) \tag{6.5}$$

where $\mathbf{E}_S(h(X))$ and $\mathbf{E}_T(h(X))$ are the domain classification error over the source domain and target domain respectively and X is produced by G. This optimization can be achieved in an adversarial training manner by a gradient reverse layer (GRL) [204].

6.3.4 Multi-Agent Fusion

Our MPDA framework is very flexible and can integrate most of the multi-agent fusion algorithms. In this work, we select a state-of-the-art model, V2X-ViT [19], as our multi-agent fusion algorithm. V2X-ViT employs a heterogeneous multi-agent self-attention block and a multi-scale windowed attention block sequentially to intelligently fuse the different agents' features. To achieve the best performance, besides learning to fool the domain classifier, G also targets to directly optimize the detection performance. Let us denote M as the multi-agent fusion algorithm, then the second training objective for G is:

$$\min_{G,M} (\mathbf{E}_D(V)), \quad V = M(F_S, F_T''), \tag{6.6}$$

where $\mathbf{E}_D(V)$ is the 3D detection error and V is the fused feature with shape of $1 \times H_S \times W_S \times C_S$.

6.3.5 Loss

For 3D object detection, we use the smooth L1 loss for bounding box regression and focal loss [116] for classification. For the domain classifier, we utilize cross-entropy loss to

learn domain-invariant features. The final loss is the combination of detection and domain adaptation loss:

$$L = \alpha L_{det} + \beta L_{domain}, \tag{6.7}$$

where α and β are the balance coefficients within range $[0, 1]$.

6.4 Experiments

6.4.1 Dataset

We conduct experiments on the public large-scale V2X perception dataset V2XSet [19]. V2XSet is collected together by the high-fidelity simulator CARLA [8] and cooperative driving automation frame [7]. It provides LiDAR data from different autonomous vehicles and roadside intelligent infrastructure at the same timestamp and scenario. In total, V2XSet has 11,447 frames and can be split into 6,694/1,920/2,833 frames for training/validation/testing respectively.

6.4.2 Experiments Setup

Evaluation metrics. We evaluate the performance of our proposed framework by the final 3D detection accuracy. Similar to previous works in this area [6, 19], we set the evaluation range as $x \in [-140, 140]$ meters, $y \in [-40, 40]$ meters and measure the accuracy with Average Precisions (AP) at Intersection-over-Union (IoU) threshold of 0.7.

Evaluation protocols During training, we randomly select one agent as the ego agent. During testing, we choose a fixed one as the ego for each scenario. We estimate our model under three distinct settings:

1. *Normal scenario:* In this scenario, all ego agents and other agents use PointPillar [1] with identical parameter as the detection backbone, named p_0. **Among all experi-**

ments, the ego vehicle will always have p_0 as the backbone.

2. *Hetero scenario 1:* During training, ego agents use PointPillar p_0, whereas other agents employ p_1, which also belongs to the PointPillar family but with heterogeneous configurations including voxelization resolutions and the number of convolution layers. To assess the generalization probability of the proposed MPDA, another trained PointPillar model p_2 will be used for testing, which has different parameters from any training model.

3. *Hetero scenario 2:* We assume even the model types are heterogeneous in this scenario. All ego vehicles are still trained based on p_0, and other agents are based on the different detection model SECOND [80] s_0. During the testing stage, we use another trained SECOND model s_1 with distinct parameters with s_0.

To ensure all the backbones are trained properly, we first assume that all agents are equipped with the same backbone and combine it with the V2X-ViT model [19] to perform 3D detection. As shown in Table 6.1, all backbones have achieved reasonable accuracy. We also demonstrate partial parameters of various backbones in Tabel 6.2, and there are noticeable differences between them in terms of voxel resolution, LiDAR cropping range, and the number of convolutional layers.

Table 6.1: **Detection backbone models' performance on the testing set without domain gap.** We assume all agents have the same detection model in this experiment.

	p_0	p_1	p_2	s_0	s_1
AP@0.7	71.2	68.3	70.1	74.5	77.0

Compared methods: We consider *No Fusion* as the baseline, which does not involve any collaboration in the system. To demonstrate the significant effect of the domain gap, we first directly use the pre-trained model provided by [19] and simply apply bilinear interpolation with the channel dropping technique to align the dimensions. We then let the pre-trained

Table 6.2: **Parameters of different detection backbone.**

Backbone	Voxel Resolution	Half Lidar Cropping Range (x&y)	# of 2D&3D CNN Layers
p_0	0.4, 0.4, 4	140.8 & 38.4	19 & 0
p_1	0.8, 0.6, 4	140.8 & 38.4	16 & 0
p_2	0.6, 0.6, 4	153.6 & 38.4	17 & 0
s_0	0.2, 0.2, 0.2	140.8 & 41.6	12 & 12
s_1	0.1, 0.1, 0.1	140.8 & 41.6	13 & 13

model finetune on *Hetero1* and *Hetero2* scenarios to make the comparison fair since our framework will see features from different domains in training. To show the effectiveness of the two critical components in our framework, we first only add the learnable resizer. Then we add the sparse cross-domain transformer as well to be our complete framework, MPDA. We will also compare with *Late Fusion* method, which directly transmits the detected 3d bounding box along with the confidence score and merges all the overlapped predictions according to the sorted confidence scores. Though *Late Fusion* does not have the domain gap issue like intermediate fusion, it still suffers from the confidence score discrepancy issue, e.g., different models can have diverse confidence estimation biases.

Implementation details: For the multi-agent fusion method, we follow the same hyper-parameters for V2X-ViT as its original implementation in [19]. For all backbones training, we use Adam [158] as the optimizer, decay the learning rate by 0.1 for every 10 epochs with an initial learning rate of 0.001. The coefficient of detection loss L_{det} is set to 1.0 and that of domain classification loss L_{domain} is set to 0.1.

6.4.3 Quantitaive Evaluation

Major performance analysis: Table 6.3 depicts the performance comparison of various methods on *Normal*, *Hetro1*, and *Hetero2* settings, respectively. Under the *Normal* scenario, all methods exceed the baseline *No Fusion* by a large margin. Nevertheless, the results

Table 6.3: **3D detection performance in Normal scenario (w/o domain gap) and Hetero scenarios (w/ domain gap).** We show the Average Precision (AP) at IoU=0.7. DC stands for domain classifier. * notes that we do not use the domain classifier when training on the normal scenario.

Method	Normal	Hetero 1	Hetero 2
No Fusion	40.2	40.2	40.2
Late Fusion	60.2	51.7	52.8
V2X-ViT	71.2	26.7	34.5
V2X-ViT (finetuned)	71.2	48.6	64.8
V2X-ViT + Resizer	72.3	54.8	72.1
V2X-ViT + MPDA (w/o DC)	73.4	56.3	72.5
V2X-ViT + MPDA	**73.4***	**57.6**	**73.3**

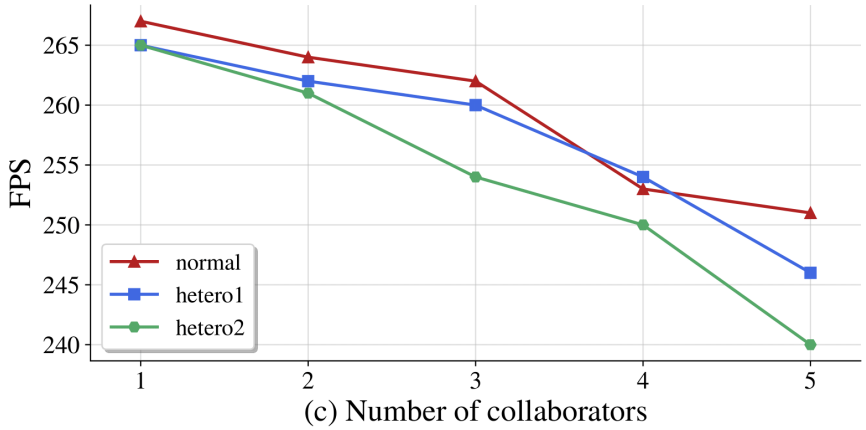
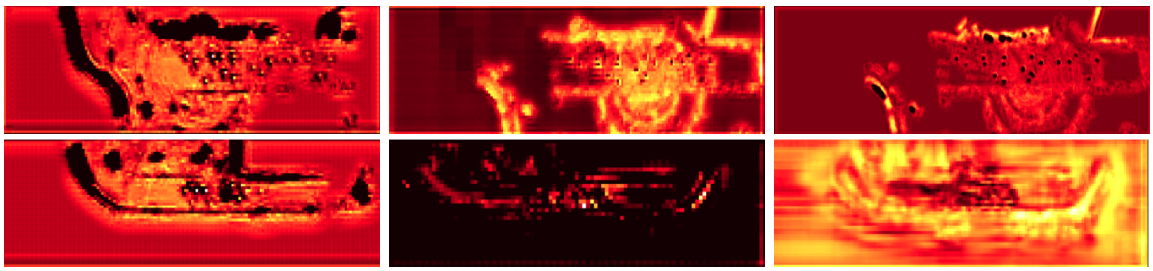


Figure 6.3: **Inference speed of MPDA under different settings.**

are different when the models deployed on the agents are heterogeneous. The pre-trained V2X-ViT drops to 26.7% and 34.5% on *Hetero1* and *Hetero2* respectively, which is even much lower than the single agent perception system. **This dramatic performance drop indicates highly negative impacts by the domain gap.** After directly finetune on *Hetero1* and *Hetero2*, V2X-ViT's performance increases though still not satisfying and lower than *Late Fusion* in *Hetero1*. On the contrary, our MPDA has achieved 57.6%, and 73.3% on the two heterogeneous settings, which performs favorably against other methods and significantly outperforms *No Fusion*'s baseline. Note that the performance on *Hetero1* is



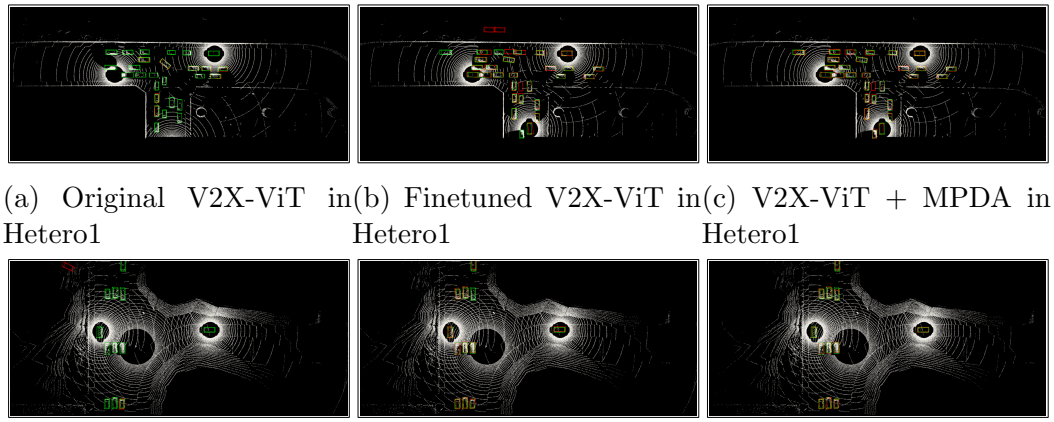
(a) ego's feature (b) feature before MPDA (c) feature after MPDA

Figure 6.4: **Visalzation of intermediate features before and after domain adaption.** From left to right: (a) ego's feature, (b) collaborator's feature before domain adaption, (c) collaborator's feature after domain adaption. Row 1 is the *Hetero1* scenario where ego and others both use PointPillar, but the parameters differ. Row 2 is the *Hetero2* scenario where ego uses PointPillar, and others use SECOND. It is obvious that after domain adaption, others' intermediate features have more similar patterns as ego's.

relatively lower for all methods. A potential reason for it is p_2's voxel resolution, and the LiDAR cropping range is quite distinct from p_0 and p_1, which makes the adaption challenging. With the deployment of our framework, the accuracy of V2X-ViT increases by 9% and 8.5% on *Hetero1* and *Hetero2* respectively. We also found that our MPDA can enhance the performance on *Normal* setting as well by 2.2%, which attributes the capability of our resizer and sparse cross-domain transformer to help generate more robust feature representations.

Main component analysis: As Table 6.3 describes, all of our designed components in MPDA have contributed to more accurate detections. Adding the learnable resizer improves the detection performance by 6.2% and 7.3% under two heterogeneous settings. The sparse cross-domain transformer combined with the domain classifier can further increase the AP by 2.8% and 1.2%.

Inference time: Real-time performance is critical for real-world deployment. Thus, here we calculate the inference speed of our proposed MPDA framework under different scenarios concerning various collaborator numbers. As Fig. 6.3 shows, our MPDA can always achieve more than 200FPS under different settings, indicating our design's efficiency.



(a) Original V2X-ViT in Hetero1 (b) Finetuned V2X-ViT in Hetero1 (c) V2X-ViT + MPDA in Hetero1

(d) Original V2X-ViT in Hetero2 (e) Finetuned V2X-ViT in Hetero2 (f) V2X-ViT + MPDA in Hetero2

Figure 6.5: **3D detection visualization.** Green and red 3D bounding boxes represent the ground truth and prediction respectively. With our MPDA, the detection results are clearly more accurate.

6.4.4 Qualitative Evaluation

Domain adaption visualization: Similar to [168], we sum up all the absolute values of all channels to visualize the feature maps to investigate their patterns. As Fig. 6.4 shows, without any domain adaption, there are noticeable gaps between the ego agent's and other collaborators' features. After applying our MPDA, the converted features become more similar to the ego's, which visually proves the effectiveness of MPDA.

3D detection visualization: We visually compare different methods in the same scenario under two heterogeneous settings and show the result in Fig. 6.5. Obviously, without seeing any features from different backbones, the pre-trained V2X-ViT model provided by the authors from [19] has many missing detections. After directly finetuning under *Hetero1* and *Hetero2* settings, the results get improved, but there still exist noticeable missing detections, false positives, and large displacement. On the contrary, our MPDA has a more robust performance, detecting most of the objects and predicting accurate bounding box positions.

6.5 CONCLUSIONS

This paper is the first work that investigates the domain gap issue in multi-agent perception. Based on the analysis, we propose the first multi-agent perception domain adaption framework, which mainly contains a learnable feature resizer and a sparse cross-domain transformer. Extensive experiments on the V2XSet dataset prove that our framework can effectively bridge the domain gap. In the future, we will combine robust generative representation learning techniques such as Diffusion [205] and conduct real-world field experiments on this practical issue.

CHAPTER 7

Model-Agnostic Multi-Agent Perception Framework

Existing multi-agent perception systems assume that every agent utilizes the same model with identical parameters and architecture. The performance can be degraded with different perception models due to the mismatch in their confidence scores. In this work, we propose a model-agnostic multi-agent perception framework to reduce the negative effect caused by the model discrepancies without sharing the model information. Specifically, we propose a confidence calibrator that can eliminate the prediction confidence score bias. Each agent performs such calibration independently on a standard public database to protect intellectual property. We also propose a corresponding bounding box aggregation algorithm that considers the confidence scores and the spatial agreement of neighboring boxes. Our experiments shed light on the necessity of model calibration across different agents, and the results show that the proposed framework improves the baseline 3D object detection performance of heterogeneous agents. The code can be found at this url.

7.1 Introduction

Recent advancements in deep learning have improved the performance of modern perception systems on many tasks, such as object detection [2, 206, 90], semantic segmentation [139, 207], and visual navigation [208, 209, 210]. Despite the remarkable progress, single-agent perception systems still have many limitations due to single-view constraints. For instance, autonomous vehicles (AVs) usually suffer from occlusion [211, 162, 163], and such situations

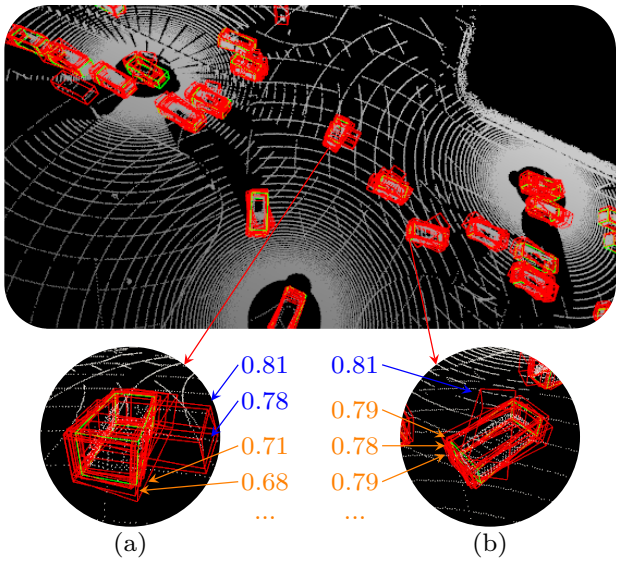


Figure 7.1: **Ground truth (green) and bounding box candidates (red) produced by three connected autonomous vehicles**. (a) Some agents have confidence scores that are systematically larger than others, e.g., the blue scores versus the orange scores. However, they might be confidently wrong, which mislead the fusion process. (b) Candidates with slightly lower confidence scores (orange) but higher spatial agreement with neighboring boxes can be better than a singleton with a higher confidence score (blue).

are difficult to handle because of the lack of sensory observations of the occluded area. To address this issue, recent studies [15, 98, 6, 19, 212, 22] have explored wireless communication technology to enable nearby agents to share the sensory information and collaboratively perceive the surrounding environment.

Although existing fusion frameworks have obtained a significant 3D object detection performance boost, they assume that all the collaborating agents share an identical model with the same parameters. This assumption is hard to satisfy in practice, particularly in autonomous driving. Distributing the model parameters among AVs might raise privacy and confidentiality concerns, especially for vehicles from different automotive companies. Even for AVs from the same company, the detection models can have various versions, depending on the vehicle type and model updating frequency. Without adequately handling the inconsistency, the shared sensory information can have a large domain gap, and the advantage brought by multi-agent perception will be diminished rapidly.

To this end, we propose a model-agnostic multi-agent perception framework to handle the model heterogeneity while maintaining confidentiality. The perception outputs (i.e., detected bounding boxes and confidence scores) are shared to bypass the dependency on the underlying model's detailed information. Due to the distinct models used by the agents, the confidence scores provided by different agents can be systematically misaligned. Some agents may be over-confident, whereas others tend to be under-confident. Directly fusing bounding box proposals from neighboring agents using, for example, Non-Maximum Suppression (NMS) [213] can result in poor detection accuracy due to the presence of over-confident and low-quality candidates.

We propose a simple yet flexible confidence calibrator, called Doubly Bounded Scaling (DBS), to mitigate the misalignment. We also propose a corresponding bounding box aggregation algorithm, named Promote-Suppress Aggregation (PSA), that considers the confidence scores and the spatial agreement of neighboring boxes. Fig. 7.1 illustrates the importance of these two components. This framework does not reveal model design and parameters, ensuring confidentiality. We evaluate our approach on an open-source large-scale multi-agent perception dataset – OPV2V [6]. Experiments show that in the presence of model discrepancies among agents, our framework significantly improves multi-agent LiDAR-based 3D object detection performance, outperforming the baselines by at least 6% in terms of Average Precision (AP).

7.2 Related Work

Multi-Agent Perception. Multi-agent systems have been extensively studied recently because of their potential to revolutionize robotics industry [169, 172, 170, 171, 210, 177, 178, 175, 176]. Multi-agent perception, as an important branch in multi-agent systems, investigates how to leverage visual cues from neighboring agents through the communication system to enhance the perception capability. There are three categories of existing work

according to the information sharing schema: 1) early fusion, where raw point clouds are transmitted directly and projected into the same coordinate frame, 2) late fusion [96], where detected bounding boxes and confidence scores are shared, and 3) intermediate fusion [98, 6, 15, 19, 180], where compressed latent neural features extracted from point clouds are propagated. Though early fusion has no information loss, it usually requires large bandwidth. Intermediate fusion can achieve a good balance between accuracy and transmission data size, but it requires complete knowledge of each agent's model, which is non-trivial to satisfy in reality due to intellectual property concerns. On the contrary, late fusion only needs the outputs of the detector without demanding access to the underlying neural networks, which are typically confidential for automotive companies. Therefore, our approach adopts the late fusion strategy but further designs customized new components to address the model discrepancy issue in vanilla late fusion.

Confidence Calibration. For a probabilistic classifier, the probability associated with the predicted class label should reflect its correctness likelihood. However, many modern neural networks do not have such property [214]. Confidence calibration aims to endow a classifier with such property. Calibration methods can be tightly coupled with the neural networks, such as Bayesian neural networks and regularization techniques [215, 216, 217], or serve as a post-processing step. Post-processing methods include histogram binning methods [218], scaling methods [219, 220], and mixtures [221] that combine the first two branches. Due to the popularity of the Temperature Scaling method [214] which is a single-parameter version of Platt Scaling [219], scaling methods are widely adopted for calibrating neural networks. Our proposed method follows the same fashion.

Bounding Box Aggregation. Object detection models typically require bounding box aggregation to lump the proposals corresponding to the same object. Recent study [184] demonstrates that bounding box aggregation can effectively improve small object detection accuracy. The *de facto* standard post-processing method is Non-Maximum Suppres-

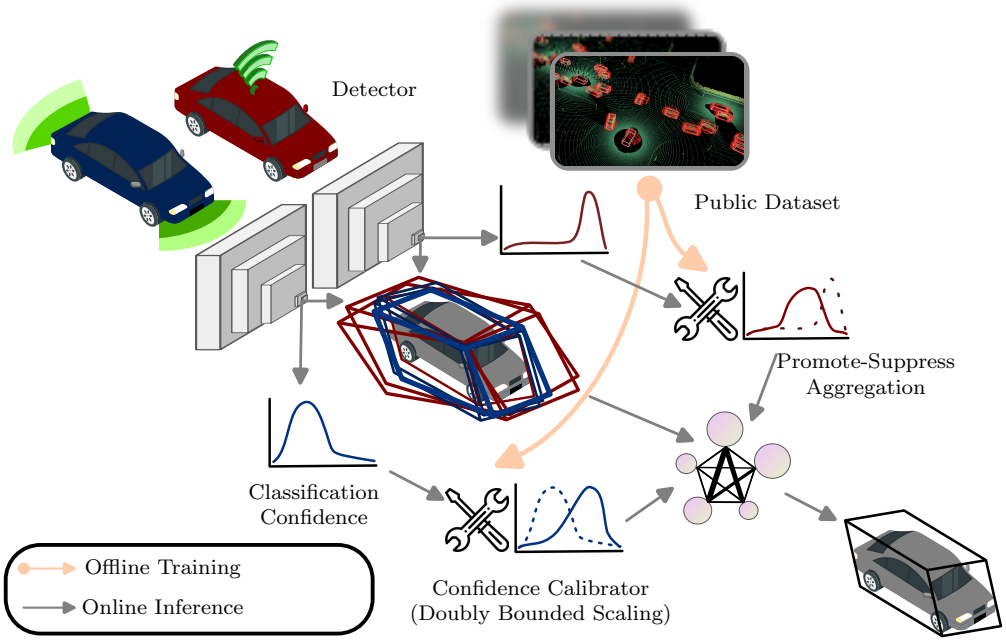


Figure 7.2: **Overview of the proposed framework**. Each agent trains its confidence calibrator (i.e., Doubly Bounded Scaling) on the same public dataset offline (orange arrows). Promote-Suppress Aggregation yields the final detection result, considering the spatial information and calibrated confidence of bounding boxes given by connected autonomous vehicles.

sion (NMS) [213], which sequentially selects the proposals with the highest confidence score and then suppresses other overlapped boxes. NMS does not fully exploit information in the proposals because it only uses the relative order of confidence, ignoring the absolute confidence scores and the spatial information hidden in the bounding box coordinates. Several works have been proposed to refine the box aggregation strategies. Soft-NMS [222] softly decays the confidence scores of the proposals proportional to the degree of overlap. In [223] NMS can be learned by a neural network to achieve better occlusion handling and bounding box localization. Adaptive NMS [224] applies a dynamic suppression threshold to an instance according to the target object density. [225] formulate NMS as a clustering problem and use Affinity Propagation Clustering to solve the problem. The idea of message passing between proposals is related to the aggregation algorithm introduced in Sec. 7.3.3, but our update rules are simpler and more efficient.

7.3 Methodology

In this paper, we consider the cooperative perception of a heterogeneous multi-agent system, where agents communicate to share sensing information from different perception models without revealing model information, i.e., model-agnostic collaboration. We focus on a 3D LiDAR detection task in autonomous driving, but the methodology can also be customized and used in other cooperative perception applications. Our goal is to develop a robust framework to handle the heterogeneity among agents while preserving confidentiality. The proposed model-agnostic collaborative perception framework is shown in Fig. 7.2, which can be divided into two stages. In the offline stage, we train a model-specific calibrator. During the online phase, real-time on-road sensing information is calibrated and aggregated.

7.3.1 Model-Agnostic Fusion Pipeline

Agents with distinct perception models usually generate systematically different confidence. The mismatch in confidence distributions can affect the fusion performance. For instance, an inferior model may be over-confident and dominate the aggregation process, decreasing the accuracy of the final results.

To address the issue, we train a calibrator offline for each model, aligning its confidence score with its empirical accuracy on a calibration dataset. First, each model runs its well-trained detector on a public dataset to produce a model-specific dataset with labels and confidence scores. The public dataset, like nuScenes [67] or Waymo open dataset [4], should be independent of the manufacturer and sensor setup, serving only to test the model's performance. The calibration dataset is then used to train the calibrator (see Sec. 7.3.2 for more details). After training, the calibrator is saved locally for each agent.

When the vehicle is driving on-road and making predictions from the sensor measurements, the calibrator will align the predicted confidence score towards the same standard,

thus alleviating the aforementioned mismatch. Then the bounding box coordinates and calibrated confidence scores are packed together and transmitted to neighboring agents. The receiving agent (i.e., ego vehicle) will fuse the shared information via the Promote-Suppress Aggregation algorithm (see Sec. 7.3.3 for details) to output the final results. Since each agent learns its calibrator independently in the offline stage and only shares the detection outputs during the online phase, the detector architecture and parameters are invisible to other agents, protecting the intellectual property.

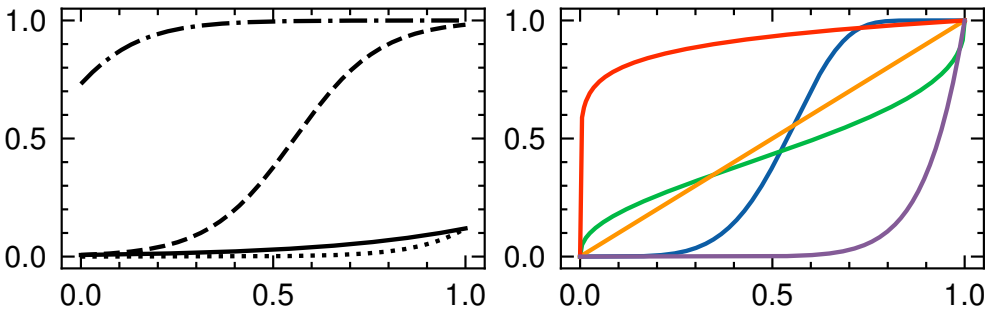
7.3.2 Classification Confidence Calibration

To eliminate the bias brought by the system heterogeneity, the models need to be *well-calibrated*. If the confidence scores can imply the likelihood of correct prediction, for example, 80% confidence leads to 80% accurate predictions, this model is *well-calibrated*. Formally, let $\tilde{s}$ be the confidence score produced by the model and $y \in \{0, 1\}$ be the label indicating vehicle or background[1]. A model is *well-calibrated* if its confidence score $\tilde{s}$ matches the expectation of correctly predicting the label:

$$\mathbb{E}[y = 1 \mid \tilde{s}] = \tilde{s}. \tag{7.1}$$

Scaling-Based Confidence Calibration. The goal of scaling-based confidence calibration is to learn a parametric scaling function (i.e., calibrator) $\mathsf{c}_{\boldsymbol{\theta}}(\tilde{s}) : [0, 1] \mapsto [0, 1]$ on a calibration dataset to transform the uncalibrated confidence scores $\tilde{s}$ into well-calibrated ones s. Given a calibration set $\mathcal{D} \triangleq \{(\tilde{s}_n, y_n)\}_{n=1}^{N}$ containing the model-dependent confidence scores $\tilde{s}$ and ground-truth labels y, we optimize the parameters $\boldsymbol{\theta}$ of the calibrator $\mathsf{c}_{\boldsymbol{\theta}}(\tilde{s})$ by gradient

[1]We discuss binary classification here for simplicity but the proposed framework can be generalized to the multi-class case.



(a) Logistic (b) Kumaraswamy

Figure 7.3: **Scaling functions with various parameters** that follow (a) the logistic form and (b) the Kumaraswamy CDF. Note that, in (b), the "inverse-sigmoid" shape (green curve, $a = 0.4, b = 0.4$) and the identity map (orange curve, $a = 1, b = 1$) are not in the logistic family.

descent on the binary cross entropy loss

$$\ell_{CE} = -y_n \log(s_n) - (1 - y_n) \log(1 - s_n), \tag{7.2}$$

where $s_n = \mathsf{c}_{\boldsymbol{\theta}}(\tilde{s}_n)$. Training a parametric function by optimizing Eq. (7.2) is similar to standard binary classification, however, extra constraints are required on the scaling function for confidence calibration . Designing a suitable calibrator for our application requires satisfying three conditions: (a) The scaling function needs to be *monotonically non-decreasing* as a higher confidence score is supposed to indicate a higher expected accuracy; (b) The scaling function should be relatively smooth to avoid over-fitting to the calibration set; (c) The scaling function is supposed to be *doubly bounded*, meaning that it maps a confidence interval $[0, 1]$ to the same $[0, 1]$ range. In the following sub-sections, we will explain why the commonly used calibration methods do not meet all these conditions, which motivates the development of our proposed calibrator.

Platt Scaling and Temperature Scaling. The most popular scaling methods are arguably Platt Scaling [219] and Temperature Scaling [214]. Platt Scaling uses the logistic

family as the calibrator:

$$\mathsf{c}_{\text{Platt}}(\tilde{s}; a, b) = \frac{1}{1 + \exp\left(-(a \times \tilde{s} + b)\right)}, \tag{7.3}$$

where a, b are parameters with $a \geq 0$ to ensure that the calibration map is *monotonically non-decreasing*. Temperature Scaling is a special case of Eq. (7.3) where b is fixed to 0. Fig. 7.3 shows several scaling functions from this family. Platt Scaling can fail if its parametric assumptions are not met [226]. For example, we cannot learn an "inverse-sigmoid" (see the green curve in Fig. 7.3b) scaling function within this family. Furthermore, the identity function is also not a member of the logistic family. In addition to the aforementioned limitations, the logistic family is also not a function family that can naturally map $[0, 1]$ to $[0, 1]$ as its input domain is $\mathbb{R}$, therfore, these popular choices are not our ideal candidates.

Doubly Bounded Scaling (DBS). We propose to use the Kumaraswamy Cumulative Density Function (CDF) [227] that meets all the three constraints while being sufficiently flexible. To the best of our knowledge, this is the first time that this function family has been adopted in confidence calibration. Specifically, we learn a scaling function with the following form

$$\mathsf{c}(\tilde{s}; a, b) = 1 - \left(1 - \tilde{s}^{a}\right)^{b}, \tag{7.4}$$

where $a > 0$ and $b > 0$ are the parameters. Scaling functions that follow Eq. (7.4) are *monotonically non-decreasing*, *smooth*, and *doubly bounded*, hence the name. we can see that DBS is more flexible than the logistic form by comparing Fig. 7.3a and Fig. 7.3b. For each detector, we optimize the a and b on a calibration dataset by minimizing Eq. (7.2).

7.3.3 Promote-Suppress Aggregation (PSA)

Detection models typically output a bunch of overlapped bounding box candidates for the same detected object, thus we need a post-processing step to select from these candidates.

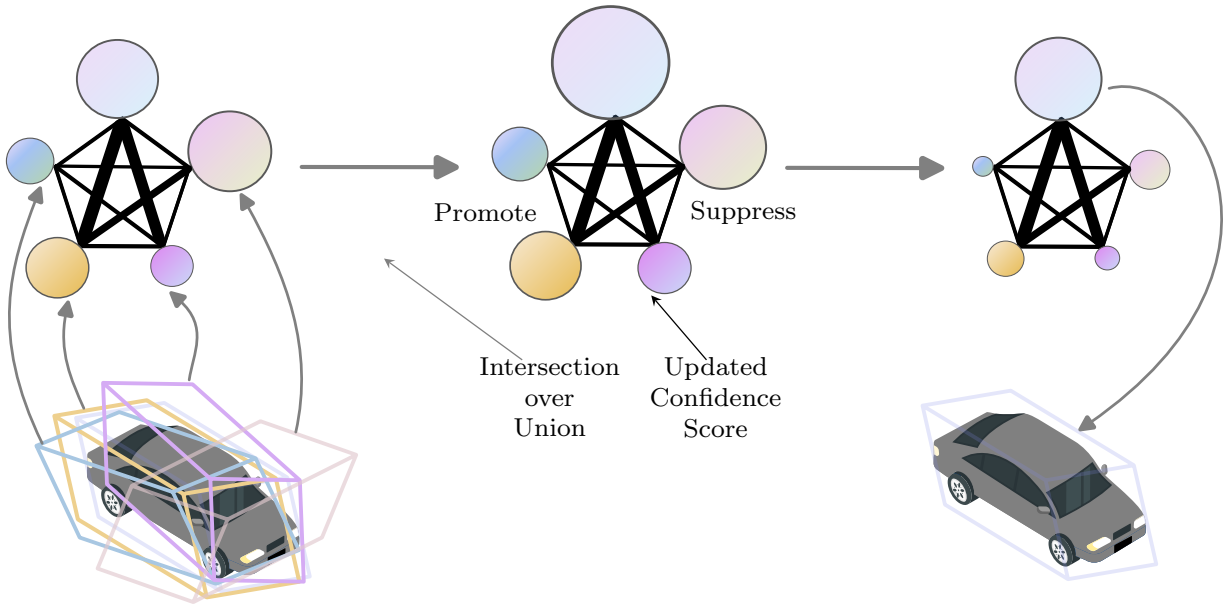


Figure 7.4: **Illustration of Promote-Suppress Aggregation**. The size of a node indicates the confidence score of the bounding box and the edge width represents the Intersection-over-Union of two boxes.

In most of the detection algorithms, the optimization objective function is a summation of a bounding box regression loss and a classification loss. The detector can express its "confidence" by assigning high classification scores to the promising bounding boxes or allocating more bounding boxes to the region that it finds relevant features. To select the high-score bounding boxes with many confident neighbors, we propose Promote-Suppress Aggregation (PSA), which takes into account both the regression and classification confidences.

Fig. 7.4 illustrates the idea of PSA. We first construct a spatial graph of bounding box candidates based on Intersection-over-Union (IoU) values and the confidence scores. In the promotion step, the IoU weighted confidence scores are propagated to the neighboring nodes. We design the propagation rule to meet the following desiderata:

- A candidate should be promoted if many other candidate boxes have *large intersections* with it;

- A candidate with many *high-score neighbors* should be promoted;
- If possible, the update rules should be *parallelizable* and *permutation-invariant*. Namely, the propagation order does not change the result.

In the suppression step, the candidate with the highest updated score will softly suppress the scores of other candidates. Finally, we select one or more bounding boxes that rank in the first (few) places. The idea of soft suppression and selecting more than one candidate is akin to soft-NMS [222], which is beneficial when the bounding box of a small object is within the box of a large object. Below we formally describe the PSA algorithm.

Lemma 1 *Let $\mathcal{G}$ be a weighted graph with a set of edges $\mathcal{E}$ and a set of nodes/vertices $\mathcal{V}$, where each vertex $v \in \mathcal{V}$ represents a bounding box candidate b with an associated confidence score s after calibration. The edge weigh w_{ij} between vertex v_i and v_j is defined as the Intersection-over-Union value $\texttt{IoU}(b_i, b_j) \triangleq \frac{\cap(b_i,b_j)}{\cup(b_i,b_j)}$. An edge connects vertex v_i and v_j if the edge weight is non-zero.*

Lemma 2 *The graph consists of a number of connected components in which every pair of nodes is connected via a sequence of edges.*

Problem 1 (Bounding Box Aggregation) *Given the Intersection-over-Union matrix $\mathbf{U} \in [0,1]^{N\times N}$ among N bounding box candidates $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_N]^\intercal$ and their confidence scores $\mathbf{s} = [s_1, \ldots, s_N]^\intercal$, our goal is to compute an index set $\mathcal{I}$ to select/filter candidates that best match the ground-truth bounding boxes.*

Algorithm 1 shows how PSA computes the index set. Given the IoU adjacency matrix, we can find out the indices of each component and put them into a component set $\mathcal{C} = \{\mathbf{c}_m\}_{m=1}^{M}$, where M is the number of components and $\mathbf{c}_m$ contains the indices of N_m vertices (line 3). For each component, we extract the IoU matrix $\mathbf{U}_m \in [0,1]^{N_m\times N_m}$ and confidence score vector

Algorithm 1: Promote-Suppress Aggregation

Arguments: bounding boxes $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_N]^\intercal$,
confidence score vector $\mathbf{s} = [s_1, \ldots, s_N]^\intercal$,
soft selection parameters ε, and threshold ϕ

1: Initialize selected box indices to an empty set $\mathcal{I} = \emptyset$
2: Compute IoU matrix $\mathbf{U} \in [0,1]^{N \times N}$ using $\mathbf{B}$
3: Find vertex indices of connected components $\mathcal{C} \triangleq \{\mathbf{c}_m\}_{m=1}^{M}$
4: **for each** $\mathbf{c}_m \in \mathcal{C}$ **do**
5: Extract IoU sub-matrix $\mathbf{U}_m \in [0,1]^{N_m \times N_m}$ via $\mathbf{c}_m$
6: Extract score sub-vector $\mathbf{s}_m \in [0,1]^{N_m}$ via $\mathbf{c}_m$
7: $\hat{\mathbf{s}}_m = \mathbf{U}_m \mathbf{s}_m$ ▷ Promote
8: $\bar{\mathbf{s}}_m = \texttt{softmax}(\hat{\mathbf{s}}_m / \varepsilon)$ ▷ Suppress
9: $\mathcal{I} = \mathcal{I} \cup \{c_m^{(n)} \mid \bar{s}_m^{(n)} > \phi, n = 1, \ldots, N_m\}$ ▷ Select
10: **end for**
11: **return** selected candidate indices $\mathcal{I}$

$\mathbf{s}_m \in [0,1]^{N_m}$ corresponding to this component (line 5-6). Then, we perform the promotion step $\hat{\mathbf{s}}_m = \mathbf{U}_m \mathbf{s}_m$ where each vertex updates its score to be the IoU-weighted sum of scores from other vertices in the component (line 7). In the suppression step, we normalize the updated scores back to $[0,1]$ and distill the winning candidate via $\bar{\mathbf{s}}_m = \texttt{softmax}(\hat{\mathbf{s}}_m / \varepsilon)$ (line 8). In the end, indices with updated scores larger than a threshold are added to the set $\mathcal{I}$ (line 9). We can select multiple candidates if $\varepsilon \in (0,1]$ is large and ϕ is small. In our application, however, one component typically contains only one object/vehicle, so we use a small ε and $\phi = 0.5$. Overall, PSA is highly parallelizable as each component operates independently and each step only requires simple linear search or small matrix-vector multiplication.

7.4 Experiments

7.4.1 Dataset

We evaluate the proposed framework on a large-scale open-source multi-agent perception dataset OPV2V [6], which is simulated using the high-fidelity simulator CARLA [8] and a cooperative driving automation simulation framework OpenCDA [7]. It includes 73 sce-

Table 7.1: Object detection performance. Average Precision (AP) at IoU=0.7 on *Homo*, *Hetero1*, and *Hetero2* setting.

Methods	**Homo** ↑AP@0.7	**Hetero1** ↑AP@0.7	**Hetero2** ↑AP@0.7
No fusion	0.602	0.602	0.602
Intermediate w/o calibration	**0.815**	0.677	0.571
Late fusion w/o calibration	0.781	0.691	0.723
Our method	0.813	**0.750**	**0.784**

narios with an average of 25 seconds duration. In each scene, various numbers (2 to 7) of Autonomous Vehicles (AVs) provide LiDAR point clouds from their viewpoints. The train/validation/test splits are 6764/1981/2169 frames, respectively. For details of the dataset, please refer to [6].

7.4.2 Experiment Setup

Evaluation metric. Following [7], we evaluate the detection accuracy in the range of $x \in [-140, 140]$m and $y \in [-40, 40]$m, centered at the ego-vehicle coordinate frame. The detection performance is measured with Average Precision (AP) at $IoU = 0.7$.

Evaluation setting. We evaluate our method under three different settings: 1) *Homo Setting*, where the detectors of agents are homogeneous with the same architecture and trained parameters. This setting has no confidence distribution gap and is used to demonstrate the performance drop when taking heterogeneity into account; 2) *Hetero Setting 1*, where the agents have the same model architecture but different parameters; 3) *Hetero Setting 2*, where the detector architectures are disparate. For *Homo Setting*, we select pre-trained Pointpillar [1] as the backbone for all the AVs. For *Hetero Setting 1*, the ego vehicle employs the same pre-trained Pointpillar model as in *Homo Setting*, whereas other AVs pick the parameters of Pointpillar from a different epoch during training. Likewise, in the *Hetero Setting 2*, the ego vehicle utilizes Pointpillar while other AVs use SECOND [80] for detection. As intermediate fusion requires equal feature map resolution, we apply simple bi-linear interpolation

Table 7.2: Component ablation study.

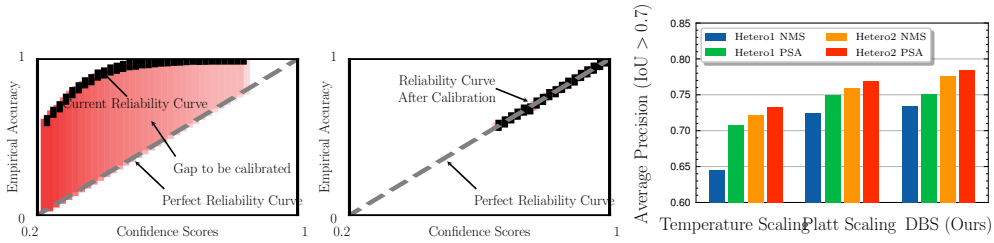
Components		Hetero1	Hetero2
DBS	PSA	↑AP@0.7	↑AP@0.7
		0.691	0.723
✓		0.734	0.776
✓	✓	**0.750**	**0.784**

under this setting. The ego vehicle uses the identical model with the same parameters across all settings for the *No Fusion* and *Late Fusion*. To compare with existing calibrators, we use the same calibration method for all agents, but the parameters are agent-specific. The proposed framework should also work even when the calibration methods across agents are heterogeneous, as long as the prediction bias is effectively reduced.

Compared methods. We regard *No Fusion* as the baseline, which only takes the ego vehicle's LiDAR data as input and omits any collaboration. Ideally, the multi-agent system should at least outperform this baseline. To validate the necessity of the calibration, we compare our method with naive late fusion and intermediate fusion that ignore calibrations. The naive late fusion gathers all detected bounding box positions and confidence scores together and simply applies NMS to produce the final results. The intermediate fusion method is the same as the one in [6]. We exclude the early fusion in the comparison as it requires large bandwidth, which leads to high communication delay thus is impractical to be deployed in the real world. Moreover, we also compare the proposed Doubly Bounded Scaling (DBS) with two other commonly used scaling-based calibrators: Temperature Scaling (TS) [214] and Platt Scaling (PS) [219].

7.4.3 Quantitative Evaluation

Main performance analysis. Tab. 7.1 describes the performance comparisons of different methods under *Homo*, *Hetero1*, and *Hetero2 Setting*. In the unrealistic *Homo* setting, all methods exceed the baseline remarkably while intermediate fusion and our method have

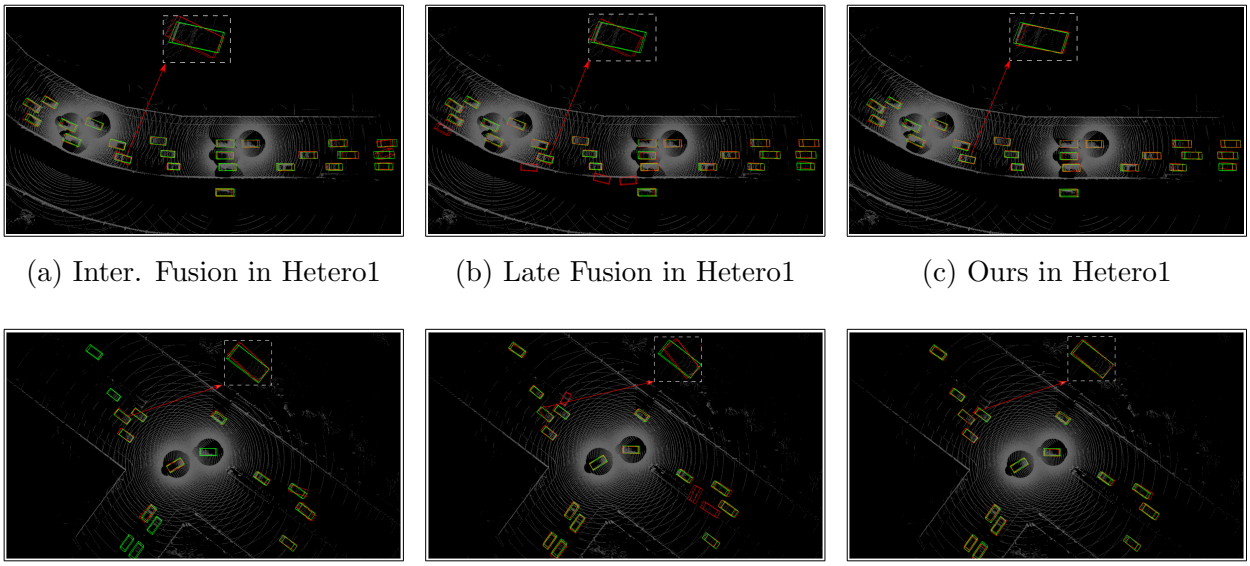


(a) Reliability Diagram before Calibration (b) Calibrated by Doubly Bounded Scaling (c) Performance of Different Calibrators

Figure 7.5: The reliability diagrams in (a) and (b) reveal that Doubly Bounded Scaling method can effectively calibrate the classification confidence scores. In (c), the proposed Doubly Bounded Scaling outperforms Temperature Scaling and Platt Scaling under various experiment setups and aggregation algorithms.

very close performance (0.2% difference). However, when we consider the realistic model discrepancy factor, our method outperforms the classic late fusion and intermediate fusion significantly by 5.9%, 7.3% under *Hetero1 Setting*, and by 6.1%, 21.3% under *Hetero2 Setting*, respectively. The classic late fusion and intermediate fusion suffer from the model discrepancy, leading to clear accuracy decreases. In the *Hetero2 Setting*, the intermediate fusion even becomes lower than the baseline. On the contrary, our method only drops around 6% and 3% under the two realistic settings, indicating the effectiveness of the proposed calibration for the heterogeneity of the multi-agent perception system. Note that although the design essence of our framework aims to handle the heterogeneous situations, we also obtain performance boost under the *Homo Setting* compared with the standard late fusion that shares detection proposals. We attribute this gain to PSA and the filtering operation of low-confidence proposals after confidence calibration that removes some potential false positives.

Major component analysis. Here we investigate the contribution from each component by incrementally adding DBS and PSA. Tab. 7.2 reveals that both modules are beneficial for the performance boost, while the calibration exhibits more contributions – increasing the AP by 4.3% and 5.3% .



(a) Inter. Fusion in Hetero1 (b) Late Fusion in Hetero1 (c) Ours in Hetero1

(d) Inter. Fusion in Hetero2 (e) Late Fusion in Hetero2 (f) Ours in Hetero2

Figure 7.6: **Qualitative comparison in a busy freeway and a congested intersection.** Green and red 3D bounding boxes represent the ground truth and prediction, respectively. Our method yields more accurate detection results.

Confidence calibration evaluation. Fig. 7.5a show the reliability diagram of Pointpillar used by the ego vehicle, in which a perfect calibration will produce a diagonal reliability curve, indicating the real accuracy matches the predictive confidence score. Reliability curves under or above the diagonal line represent over-confident or under-confident models, respectively. Pointpillar has much higher empirical accuracy than its reported confidence score. When using NMS to fuse the predictions of Pointpillar with that of another inaccurate but over-confident detector, the under-estimated confidence will result in the removal of Pointpillar's good predictions. After being calibrated by DBS, in Fig. 7.5b, the reliability curve of Pointpillar lies on the diagonal line.

Comparison with other calibration methods. Fig. 7.5c describes the comparison between our DBS calibration and other calibration methods, including TS and PS. Our DBS achieves better performance than others under both heterogeneous settings. Moreover, PSA can also improve the accuracy of different calibrators and experimental settings, showing the

generalized capability to refine the prediction results.

7.4.4 Qualitative Results

Fig. 7.6 shows the detection results of intermediate fusion, classic late fusion, and our method under *Hetero1* and *Hetero2 Setting*. Our method can identify more objects while keeping very few false positives. The zoom-in examples show that our method can regress the bounding box positions more accurately, indicating the robustness against the model discrepancy in multi-agent perception systems.

7.5 Conclusions

In the context of cooperative perception, agents from different stakeholders have heterogeneous models. For the sake of confidentially, information related to the models and parameters should not be revealed to other agents.

In this work, we present a model-agnostic collaboration framework that addresses two critical challenges of the vanilla late fusion strategy. First, we propose a confidence calibrator to align the classification confidence distributions of different agents. Second, we present a bounding box aggregation algorithm that takes into account both the calibrated classification confidence and the spatial congruence information given by bounding box regression. Experiments on a large-scale cooperative perception dataset shed light on the necessity of model calibration across heterogeneous agents. The results show that combining the two proposed techniques can improve the state-of-the-art for cooperative 3D object detection when different agents use distinct perception models.

7.6 Acknowledgment

This work is part of the OpenCDA Ecosystem [228] and supported in part by the Federal Highway Administration Exploratory Advanced Research (EAR) Program,

CHAPTER 8

V2V4Real: A Real-world Large-scale Dataset for Vehicle-to-Vehicle Cooperative Perception

Modern perception systems of autonomous vehicles are known to be sensitive to occlusions and lack the capability of long perceiving range. It has been one of the key bottlenecks that prevents Level 5 autonomy. Recent research has demonstrated that the Vehicle-to-Vehicle (V2V) cooperative perception system has great potential to revolutionize the autonomous driving industry. However, the lack of a real-world dataset hinders the progress of this field. To facilitate the development of cooperative perception, we present V2V4Real, the first large-scale real-world multi-modal dataset for V2V perception. The data is collected by two vehicles equipped with multi-modal sensors driving together through diverse scenarios. Our V2V4Real dataset covers a driving area of 410 km, comprising 20K LiDAR frames, 40K RGB frames, 240K annotated 3D bounding boxes for 5 classes, and HDMaps that cover all the driving routes. V2V4Real introduces three perception tasks, including cooperative 3D object detection, cooperative 3D object tracking, and Sim2Real domain adaptation for cooperative perception. We provide comprehensive benchmarks of recent cooperative perception algorithms on three tasks. The V2V4Real dataset and codebase can be found at https://github.com/ucla-mobility/V2V4Real.

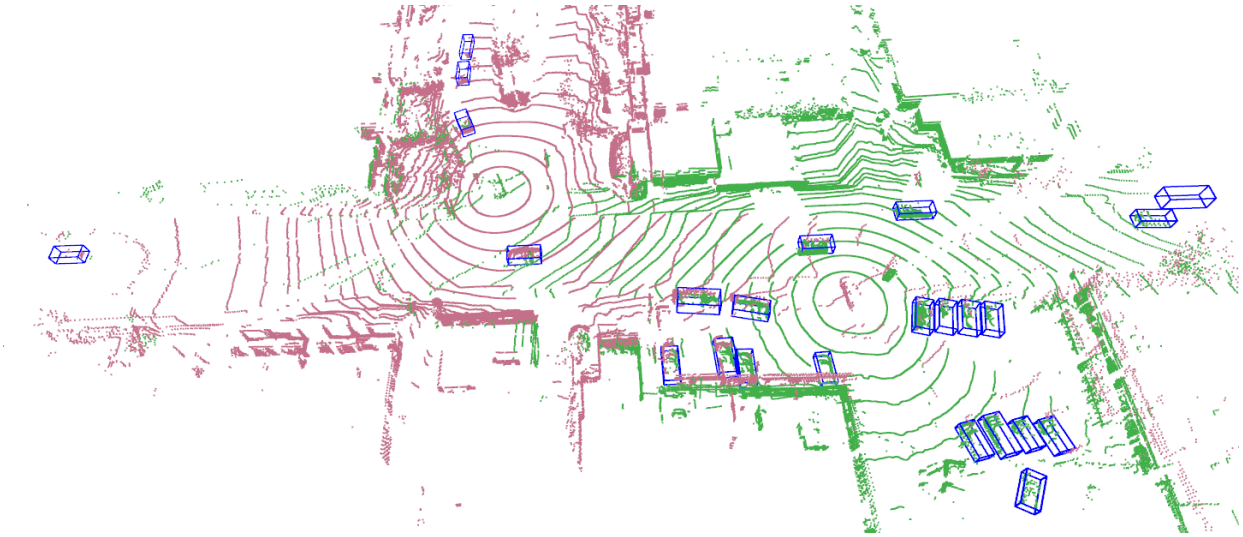
Dataset	Year	Real/ Sim	V2X	Size (km)	RGB images	LiDAR	Maps	3D boxes	Classes
Kitti [233]	2012	Real	No	-	15k	15k	No	200k	8
nuScenes [3]	2019	Real	No	33	1.4M	400k	Yes	1.4M	23
Argo [5]	2019	Real	No	290	107k	22k	Yes	993k	15
Waymo Open [4]	2019	Real	No	-	1M	200k	Yes	12M	4
OPV2V [6]	2022	Sim	V2V	-	44k	11k	Yes	230k	1
V2X-Sim [14]	2022	Sim	V2V&I	-	60K	10k	Yes	26.6k	1
V2XSet [19]	2022	Sim	V2V&I	-	44K	11k	Yes	230k	1
DAIR-V2X [234]	2022	Real	V2I	20	39K	39K	No	464K	10
V2V4Real (ours)	2022	Real	V2V	410	40K	20K	Yes	240K	5

Table 8.1: **Comparison of the proposed dataset and existing representative autonomous driving datasets.**

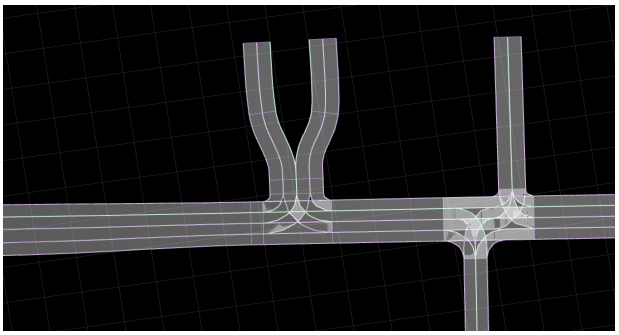
8.1 INTRODUCTION

Perception is critical in autonomous driving (AV) for accurate navigation and safe planning. The recent development of deep learning brings significant breakthroughs in various perception tasks such as 3D object detection [229, 230], object tracking [231, 232], and semantic segmentation [126, 20]. However, single-vehicle vision systems still suffer from many real-world challenges, such as occlusions and short-range perceiving capability [15, 19], which can cause catastrophic accidents. The shortcomings stem mainly from the limited field-of-view of the individual vehicle, leading to an incomplete understanding of the surrounding traffic.

A growing interest and recent advancement in cooperative perception systems have enabled a new paradigm that can potentially overcome the limitation of single-vehicle perception. By leveraging vehicle-to-vehicle (V2V) technologies, multiple connected and automated vehicles (CAVs) can communicate and share captured sensor information simultaneously. As shown in a complex intersection in Fig. 8.1, for example, the ego vehicle (red liDAR) struggles to perceive the upcoming objects located across the way due to occlusions. Incorporating the LiDAR features from the nearby CAV (green scans) can largely broaden the sensing range of the vehicle and make it even see across the occluded corner.



(a) Aggregated LiDAR data



(b) HD map

(c) Satallite Map

Figure 8.1: **A data frame sampled from V2V4Real**: (a) aggregated LiDAR data, (b) HD map, and (c) satellite map to indicate the collective position. More qualitative examples of V2V4Real can be found in the supplementary materials.

Despite the great promise, however, it remains challenging to validate V2V perception in real-world scenarios due to the lack of public benchmarks. Most of the existing V2V datasets, including OPV2V [6], V2X-Sim [14], and V2XSet [19], rely on open-source simulators like CARLA [8] to generate synthetic road scenes and traffic dynamics with simulated connected vehicles. However, it is well known that there exists a clear domain gap between synthetic data and real-world data, as the traffic behavior and sensor rendering in simulators are often not realistic enough [235, 236]. Hence, models trained on these benchmarks may not generalize well to realistic driving situations.

To further advance innovative research on V2V cooperative perception, we present a large-scale multimodal and multitask V2V autonomous driving dataset, which covers 410 km road and contains $20K$ LiDAR frames with more than $240K$ 3D bounding box annotations. Compared to the only existing real-world cooperative dataset DAIR-V2X [234], our proposed V2V4Real dataset shows several strengths: (1) DAIR-V2X focuses on Vehicle-to-Infrasctrure (V2I) applications without supporting V2V perception. Compared to V2I, V2V does not require the pre-installed sensors restricted in a certain area, which is more flexible and scalable. Our dataset fills the gap by focusing on the important V2V cooperation. (2) V2V4Real includes four diverse road types, including intersection, highway entrance ramp, highway straight road, and city straight road, covering broader driving areas and greater mileage. (3) We also provide high-definition (HD) maps that can be used for road topology prediction and semantic bird's-eye-view (BEV) map understanding. (4) We construct several benchmarks that can train and evaluate recent autonomous perception algorithms, including 3D object detection, object tracking, and Sim2Real domain adaption, while DAIR-V2X only has a single track. 5) We have provided 8 state-of-the-art cooperative perception algorithms for benchmarking, whereas DAIR-V2X only implements 3 baseline methods. Unlike DAIR-V2X, which can be only accessed within China[1], we will make all the data, benchmarks, and models publically available across the globe. Our contributions can be summarized as follows:

- We build the V2V4Real, a large real-world dataset dedicated to V2V cooperative autonomous perception. All the frames are captured by multi-modal sensor readings from real-world diverse scenarios in Columbus, Ohio, in the USA.
- We provide more than $240K$ annotated 3D bounding boxes for 5 vehicle classes, as well as corresponding HDMaps along the driving routes, which enables us to train and test cooperative perception models in real-world scenarios.

[1]https://thudair.baai.ac.cn/index

- We introduce three cooperative perception tasks, including 3D object detection, object tracking, and Sim2Real, providing comprehensive benchmarks with several SOTA models. The results show the effectiveness of V2V cooperation in multiple tasks.

8.2 Relaed Work

8.2.1 Autonomous Driving Datasets.

Public datasets have contributed to the rapid progress of autonomous driving technologies in recent years. Tab. 8.1 summarizes the recent autonomous driving datasets. The earlier datasets mainly focus on 2D annotations (boxes, masks) for RGB camera images, such as Cityscapes [237], Synthia [238], BDD100K [239], to name a few. However, achieving human-level autonomous driving requires accurate perception and localization in the 3D real world, whereas learning the range or depth information from pure 2D images is an ill-posed problem.

To enable robust perception in 3D or map-view, multimodal datasets that typically involve not only camera images but also range data such as Radar or LiDAR sensors have been developed [233, 3, 4]. KITTI [233] was a pioneering dataset that provides multimodal sensor readings, including front-facing stereo camera and LiDAR for 22 sequences, annotated with 200k 3D boxes and tasks of 3D object detection, tracking, stereo, and optical flow. Subsequently, NuScenes [3] and Waymo Open dataset [4] is the most recent multimodal datasets providing an orders-of-magnitude larger number of scenes (over 1K), with 1.4M and 993K annotated 3D boxes, respectively. Despite remarkable progress, those datasets only aim at developing single-vehicle driving capability, which has been demonstrated to have limited ability to handle severe occlusions as well as long-range perception [6, 15].

The recent development of V2V technologies has made it possible for vehicles to communicate and fuse multimodal features collaboratively, thus yielding a much broader perception range beyond the limit of single-view methods. OPV2V [6] builds the first-of-a-kind 3D co-

operative detection dataset using CARLA and OpenCDA co-simulation. V2XSet [19] and V2X-Sim [14] further explore the viability of vehicle-to-everything (V2X) perception using synthesized data generated from CARLA simulator [240]. Unlike the above-simulated datasets, DAIR-V2X is the first real-world dataset for cooperative detection. However, DAIR-V2X only concentrates on V2I cooperation, neglecting the important V2V application, which can be more flexible and more likely to be scalable. As V2V and V2I perception has major differences, i.e., V2V perception needs to deal with more diverse traffic scenarios and occlusions [19], a real-world dataset for V2V perception is needed. Furthermore, DAIR-V2X only spans limited road types (i.e., only intersections) and constrained driving route length (only 20km).

8.2.2 3D Detection

3D object detection plays a critical role in the success of autonomous driving. Based on available sensor modality, 3D detection has roughly three categories. (1) **Camera-based** detection denotes approaches that detect 3D objects from a single or multiple RGB images [241, 160, 230, 242, 229]. For instance, ImVoxelNet [230] builds a 3D volume in 3D world space and samples multi-view features to obtain the voxel representation. DETR3D [229] models 3D objects using queries to index into extracted 2D multi-camera features, which directly estimate 3D bounding boxes in 3D spaces. (2) **LiDAR-based** detection typically converts LiDAR points into voxels or pillars, resulting in 3D voxel-based [2, 80] or 2D pillar-based methods [1, 81]. Since 3D voxels are usually expensive to process, PointPillars [1] propose to compress all the voxels along the z-axis into a single pillar, then predicting 3D boxes in the bird's-eye-view space. Benefiting from its fast processing and real-time performance, many recent 3D object detection models follow this pillar-based approach [243, 244]. (3) **Camera-LiDAR fusion** presents a recent trend in 3D detection that fuses information from both image and LiDAR points. One of the key challenges in multimodal fusion is how

to align the image features with point clouds. Some methods [245, 246] use a two-step framework, e.g., first detect the object in 2D images, then use the obtained information to further process point clouds; more recent works [247, 248] develop end-to-end fusion pipelines and leverage cross-attention [79] to perform feature alignment.

8.2.3 V2V/V2X Cooperative Perception

Due to the intrinsic limitation of camera/LiDAR devices, occlusions and long-distance perception are extremely challenging for single-vehicle systems, which can potentially cause catastrophic consequences in complex traffic environments [6]. Cooperative systems, on the other hand, can unlock the possibility of multi-vehicle detection that tackles the limitation of single-vehicle perception. Among these, V2V (Vehicle-to-Vehicle) approaches center on collaborations between vehicles, while V2X (Vehicle-to-Everything) involves correspondence between vehicles and infrastructure. V2V/V2X cooperative perception can be roughly divided into three categories: (1) Early Fusion [92] where raw data is shared among CAVs, and the ego vehicle makes predictions based on the aggregated raw data, (2) Late Fusion [96] where detection outputs (*e.g.*, 3D bounding boxes, confidence scores) are shared, then fused to a 'consensus' prediction, and (3) Intermediate Fusion [15, 93, 6] where intermediate representations are extracted based on each agent's observation and then shared with CAVs.

Recent methods typically choose the intermediate neural features computed from each agent's sensor data as the transmitted features, which achieves the best trade-off between accuracy and bandwidth requirements. For instance, V2VNet [15] adopted graph neural networks to fuse intermediate features. F-Cooper [93] employed max-pooling fusion to aggregate shared Voxel features. Coopernaut [249] used Point Transformer [250] to deliver point features and conduct experiments under AustoCastSim [251]. CoBEVT [20] proposed local-global sparse attention that captures complex spatial interactions across views and agents to improve the performance of cooperative BEV map segmentation. AttFuse [6] proposed an

agent-wise self-attention module to fuse the received intermediate features. V2X-ViT [19] presented a unified vision transformer for multi-agent multi-scale perception and achieves robust performance under GPS error and communication delay.

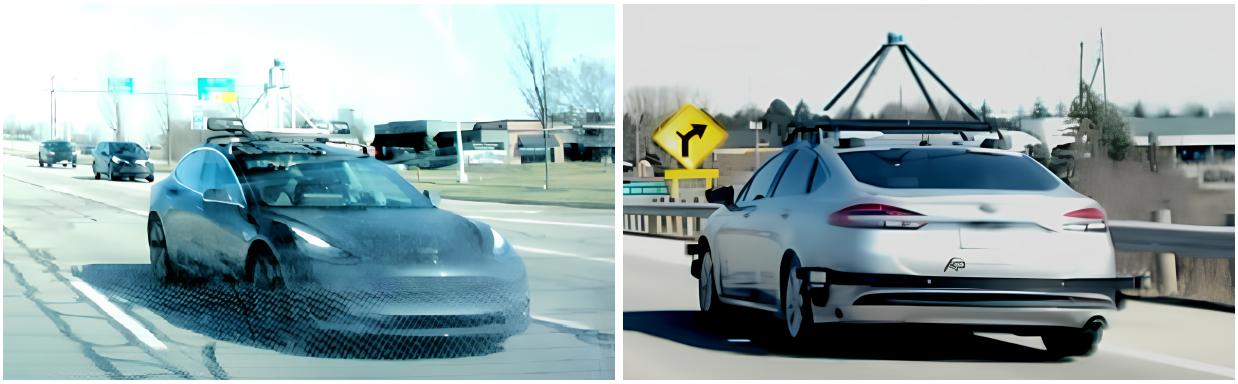
8.3 V2V4Real Dataset

To expedite the development of V2V Cooperative Perception for autonomous driving, we propose **V2V4Real**, the real-world, large-scale, multi-modal dataset with diverse driving scenarios. This dataset is annotated with both 3D bounding boxes and HDMaps for the research of multi-vehicle cooperative perception. In this section, we first detail the setup of data collection (Sec. 8.3.1), and then describe the data annotation approach (Sec. 8.3.2), and finally analyze the data statistics (Sec. 8.3.3).

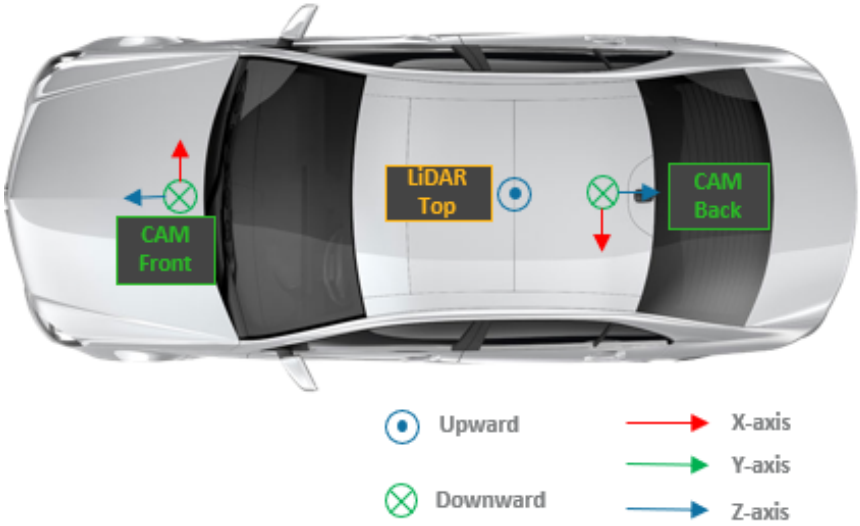
8.3.1 Data Acquisition

Sensor Setup. We collect the V2V4Real via two experimental connected automated vehicles including a Tesla vehicle (Fig. 8.2a) and a Ford Fusion vehicle (Fig. 8.2b) retrofitted by Transportation Research Center(TRC) company and AutonomouStuff (AStuff) Company respectively. Both vehicles are equipped with a Velodyne VLP-32 LiDAR sensor, two mono cameras (front and rear), and GPS/IMU integration systems. The sensor layout configuration can be found in Fig. 8.2c, and the detailed parameters are listed in Table. 8.2.

Driving Route. The two vehicles drive simultaneously in Columbus, Ohio, and their distance is maintained within 150 meters to ensure overlap between their views. To enrich the diversity of sensor-view combinations, we vary the relative poses of the two vehicles across different scenarios (see Sec. 8.3.3 for details). We collect driving logs for three days that cover 347 km of highway road and 63 km of city road. The driving routes are visualized in Fig. 8.3, wherein the red route is on day 1 (freeway with one to five lanes), the yellow



(a) Tesla Vehicle (b) Ford Fusion Vehicle



(c) Sensor setup for our data collection platform

Figure 8.2: **The information of the collection vehicles**.a) The Tesla vehicle. b) The Ford Fusion vehicle. c) The sensor setup for both vehicles.Note that the photo of Tesla is taken from the rear camera of Ford, and that of Ford is taken from the front camera of Tesla.

route is on day 2 (city road, one to two lanes), and the green route is on day 3 (highway, two to four lanes).

Data Collection. We collect 19 hours of driving data of 310K frames. We manually select the most representative 67 scenarios, each 10-20 seconds long. We sample the frames at 10Hz, resulting in a total of 20K frames of LiDAR point cloud and 40K frames of RGB images. For each scene, we ensure that the asynchronizations between two vehicles' sensor

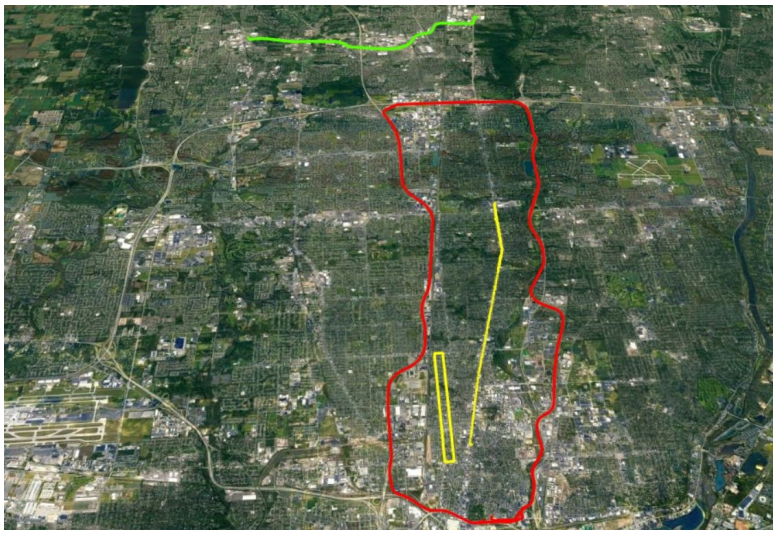


Figure 8.3: Driving routes of our two collection vehicles. Different colors represent the routes collected on different days.

systems are less than 50 ms. All the scenarios are aligned with maps containing drivable regions, road boundaries, as well as dash lines.

8.3.2 Data Annotation

Coordinate System. Our dataset includes four different coordinate systems: the LiDAR coordinate system for Tesla and Ford Fusion, the HDmap coordinate, and the earth-earth, fixed-coordinate(ECEF). We annotate the 3D bounding boxes separately based on each vehicle's LiDAR coordinate system such that each vehicle's sensor data alone can also be treated as single-agent detection tasks. We utilize the positional information provided by GPS on the two vehicles to initialize the relative pose of the two vehicles for each frame. The origin of the HDMap aligns with the initial frame of Tesla for each driving route.

3D Bounding boxes annotation. We employ SusTechPoint [252], a powerful opensource labeling tool, to annotate 3D bounding boxes for the collected LiDAR data. We hire two groups of professional annotators. One group is responsible for the initial labeling, and the other further refines the annotations. There are five object classes in total, including cars,

Sensors	Details
2x Camera	RGB, 1920×1080 resolution, 110° FOV
1x LiDAR	32 channels, 1.2 M points per second, 200 m capturing range, -25° to 15° vertical FOV, ± 3 cm error, 10Hz
GPS & IMU	Tesla: RT3000, Ford: Novatel SPAN E1

Table 8.2: Sensor specifications for each vehicle.

vans, pickup trucks, semi-truck, and buses. For each object, we annotate its 7-degree-of-freedom 3D bounding box containing x, y, z for the centroid position and l, w, h, yaw for the bounding box extent and yaw angles. We also record each object's driving state ($i.e.$dynamic or parking). To facilitate downstream applications such as tracking and behavior prediction, we assign consistent id and size for the same object in different timestamps.

Since the bounding boxes are annotated separately for the two collection vehicles, an object in the Tesla's frame could have the same id as a different object in Ford Fusion's frame. To avoid such issues, all the object ids in Tesla are labeled between $0 - 1000$, while ids in Ford Fusion range from $1001 - 2000$. Moreover, identical objects could have different ids in the annotation files of the two collection vehicles. To solve this issue, we transform the objects from different coordinates to a unified coordinate system and calculate the BEV IoU between all objects. For the objects that have IoU larger than a certain threshold, we assign them the same object id and unify their bounding box sizes.

Map Annotation. The HD map generation pipeline refers to generating a global point cloud map and vector map. To generate the point cloud map, we fuse a sequence of point cloud frames together. More specifically, we first pre-process each LiDAR frame by removing the dynamic objects while keeping the static elements. Then, a Normal Transformation Distribution scan matching algorithm is applied to compute the relative transformation between two consecutive LiDAR frames. The LiDAR odometry can then be constructed by taking the transformation. However, the noise imbued in the LiDAR data can lead to accumulated errors in the estimated transformation matrix as the frame index increases. Therefore, we

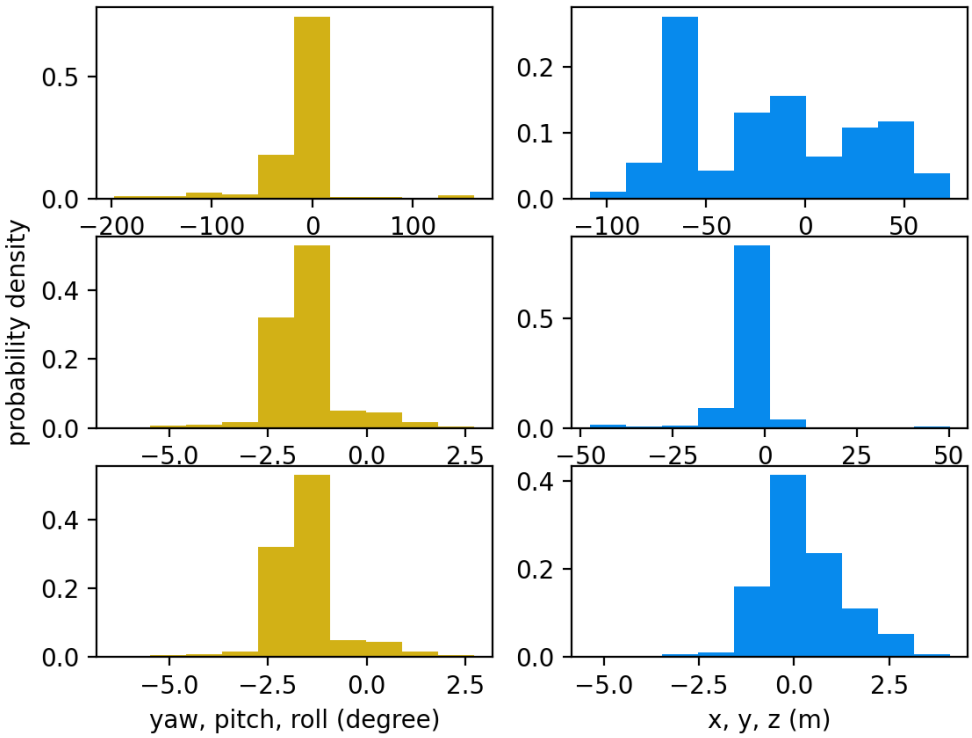


Figure 8.4: The distribution of the relative poses between the two collection vehicles.

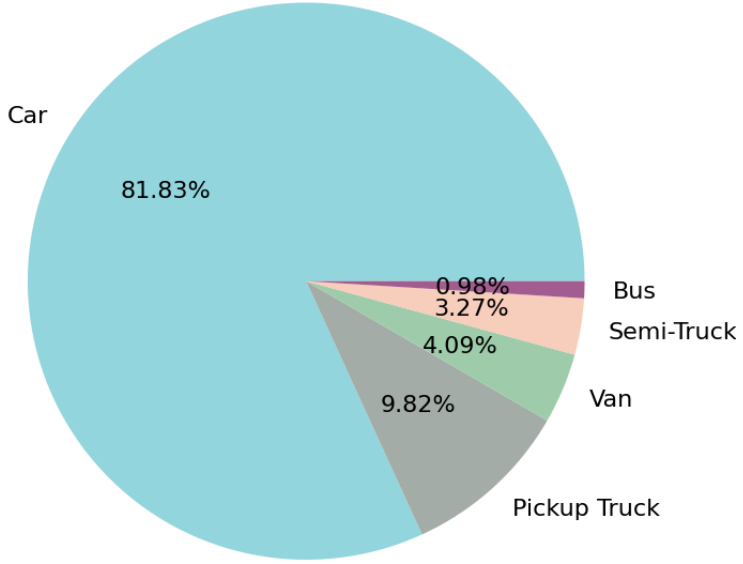


Figure 8.5: The distribution of vehicle types in collected dataset.

compensate for these errors by further integrating the translation and heading information provided by the on-vehicle GPS/IMU system and applying Kalman filter [253]. Finally, all the points in different frames are transformed onto the map coordinate to form a global point cloud map. The aggregated point cloud maps will be imported to RoadRunner [254] to produce the vector maps. The road is drawn and inferred from the intensity information visualized by distinct colors in Roadrunner. We then output the OpenDRIVE (Xodr) maps and convert them to lanelet maps [255] as the final format.

8.3.3 Data Analysis

Fig. 8.4 reveals the distribution of relative poses between the two collection vehicles across all scenarios. It can be observed that the two vehicles have a variety of relative poses, generating diverse view combinations of scenes. As Fig. 8.5 describes, most of the objects in V2V4Real belong to the Car class, while Pickup Truck ranks second. The number of Vans and Semi-Trucks are similar, while Bus has the least quantities. Fig. 8.6 shows the LiDAR points density distribution inside different objects bounding boxes and the bounding boxes' size distribution. As we may see in the left figure, when there is only one vehicle (Tesla) scanning the environment, the number of LiDAR points within bounding boxes drops dramatically as the radial distance increases. Enhanced by the shared visual information from the other vehicle (Ford Fusion), the LiDAR point density of each object increases significantly and still retains at a high level even when the distance reaches 100 m. This validates the great benefits that cooperative perception can bring to the system. As the right figure reveals, the annotated objects have diverse bounding box sizes, with lengths ranging from 2.5 m to 23 m, widths ranging from 1.5 m to 4.5 m, and heights ranging from 1 m to 4.5 m, demonstrating the diversity of our data.

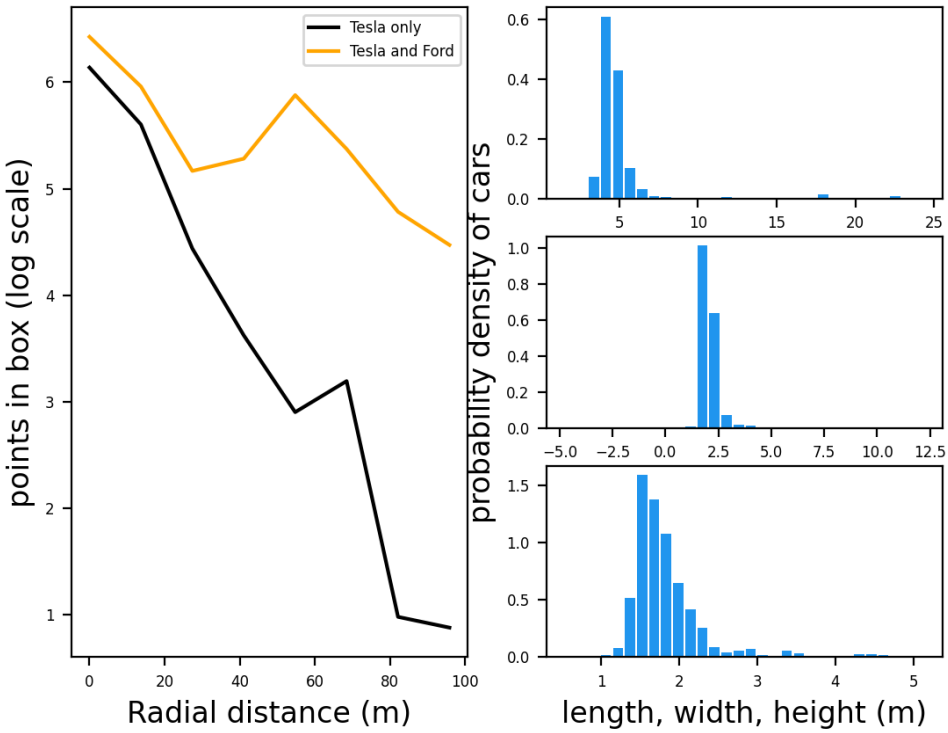


Figure 8.6: **Left**: Number of LiDAR points (e-based log scale) within ground truth bounding boxes with respect to radial distance from the ego vehicle. **Right**: Bounding box size distributions.

8.4 Tasks

Our dataset supports multiple cooperative perception tasks, including detection, tracking, prediction, localization, etc. In this paper, we focus on cooperative detection, tracking, and Sim2Real transfer learning tasks.

8.4.1 Cooperative 3D Object Detection

Scope. The V2V4Real detection task requires users to leverage multiple LiDAR views from different vehicles to perform 3D object detection on the ego vehicle. Compared to the single-vehicle detection task, cooperative detection has several domain-specific challenges:

- **GPS error:** There exists unavoidable error in the relative pose of the collaborators, which can produce global misalignments when transforming the data into a unified coordinate system.
- **Asynchronicity:** The sensor measurements of collaborators are usually not well-synchronized, which is caused by the asynchrony of the distinct sensor systems as well as the communication delay during the data transmission process [19].
- **Bandwidth limitation:** Typical V2V communication technologies require restricted bandwidth, which limits the transmitted data size [96, 19, 15]. Therefore, cooperative detection algorithms must consider the trade-off between accuracy and bandwidth requirements.

The major mission of this track is to design efficient cooperative detection methods to handle the above challenges.

Groundtruth. During training or testing, one of the two collection vehicles will be selected as the ego vehicle, and the other will transform its annotated bounding boxes to the ego's coordinate. In this way, the groundtruth is defined in a unified (the ego) coordinate system. Note that in the training phase, the ego vehicle is randomly picked, while during testing, we fix Tesla as ego. Due to asynchronicity and localization errors, the bounding boxes from two vehicles corresponding to the same object have some offsets. In such a case, we select the one annotated in the ego vehicle as the groundtruth.

Evaluation. The evaluation range in x and y direction are $[-100, 100]$ m and $[-40, 40]$ m with respect to the ego vehicle. Similar to DAIR-V2X [234], we categorize different vehicle types as the same class and focus only on vehicle detection. We use the Average Precision (AP) at Intersection-over-Union (IoU) 0.5 and 0.7 as the metric to evaluate the performance of vehicle detection. To assess the transmission cost, Average MegaByte (AM) is employed, which represents the transmitted data size specified by the algorithm. Following [234, 19], we evaluate all the models under two settings: 1) *Sync* setting, under which the data trans-

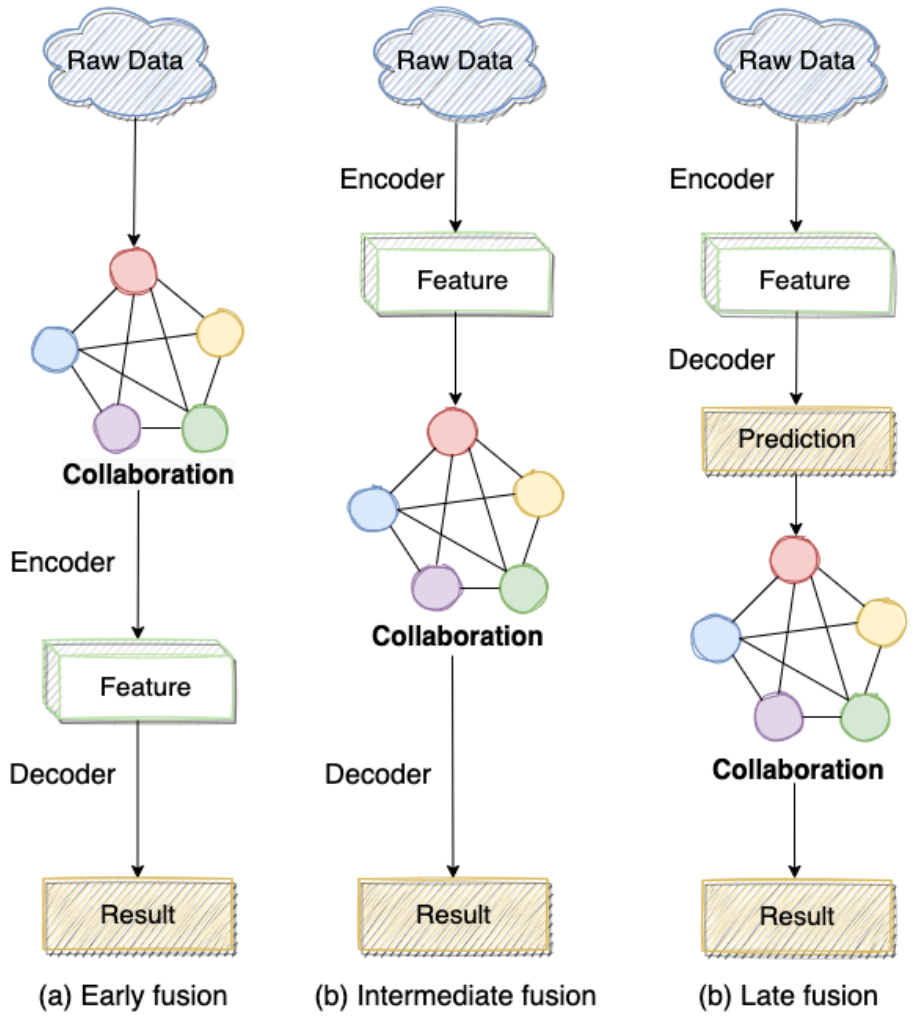


Figure 8.7: **The three different fusion strategies:** (a) Early Fusion, (b) Intermediate Fusion, and (c) Late Fusion.

mission is regarded as instantaneous, whereas the asynchrony is only induced by the distinct cycles of the sensor systems. 2) *Async* setting, where we consider the data transmission delay as 100 ms. We simulate such communication delay by retrieving the LiDAR data from the previous timestamp from the non-ego vehicle.

Benchmarking methods. We evaluate most commonly adopted fusion strategies as Fig. 8.7 demonstrated for cooperative perception with state-of-the-art methods in the domain. In total, four fusion strategies are considered:

Method	**Sync** (AP@IoU=0.5/0.7)				**Async** (AP@IoU=0.5/0.7)			
	Overall	0-30m	30-50m	50-100m	Overall	0-30m	30-50m	50-100m
No Fusion	39.8/22.0	69.2/42.6	29.3/14.4	4.8/1.6	39.8/22.0	69.2/42.6	29.3/14.4	4.8/1.6
Late Fusion	55.0/26.7	73.5/36.8	43.7/22.2	36.2/17.3	50.2/22.4	70.7/34.2	41.0/19.8	26.1/7.8
Early Fusion	59.7/32.1	76.1/46.3	42.5/20.8	47.6/**21.1**	52.1/25.8	74.6/43.6	34.5/16.3	30.2/**9.5**
F-Cooper [93]	60.7/31.8	80.8/46.9	45.6/23.6	32.8/13.4	53.6/26.7	79.0/44.1	38.7/19.5	18.1/6.0
V2VNet [15]	64.5/34.3	80.6/51.4	52.6/26.6	42.6/14.6	56.4/28.5	78.6/48.0	44.2/21.5	25.6/6.9
AttFuse [6]	64.7/33.6	79.8/44.1	**53.1**/**29.3**	43.6/19.3	57.7/27.5	78.6/41.4	**45.5**/**23.8**	27.2/9.0
V2X-ViT [19]	64.9/**36.9**	82.0/**55.3**	51.7/26.6	43.2/16.2	55.9/29.3	79.7/**50.4**	43.3/21.1	24.9/7.0
CoBEVT [20]	**66.5**/36.0	**82.3**/51.1	52.1/28.2	**49.1**/19.5	**58.6**/**29.7**	**80.3**/48	44.7/22.8	**30.5**/8.7

Table 8.3: **Cooperative 3D object detection benchmark.**

- *No Fusion:* Only ego vehicle's point cloud is used for visual reasoning. This strategy serves as the baseline.
- *Late Fusion:* Each vehicle detects 3D objects utilizing its own sensor observations and delivers the predictions to others. Then the receiver applies Non-maximum suppression to produce the final outputs.
- *Early Fusion:* The vehicles will directly transmit the raw point clouds to other collaborators and the ego vehicle will aggregate all the point clouds to its own coordinate frame, which preserves complete information but requires large bandwidths.
- *Intermediate Fusion:* The collaborators will first project their LiDAR to the ego vehicle's coordinate system and then extract intermediate features using a neural feature extractor. Afterward, the encoded features are compressed and broadcasted to the ego vehicle for cooperative feature fusion. We benchmark a number of leading intermediate methods, including AttFuse [6], F-Cooper [93], V2VNet [15], V2X-Vit [19], and CoBEVT [20] (see Sec. 8.2.3 for detail descriptions). Similar to previous works [20, 6, 19], we train a simple auto-encoder to compress the intermediate features by 16× to save bandwidth and decompress them to the original size on the ego side.

Method	AMOTA(↑)	AMOTP(↑)	sAMOTA(↑)	MOTA(↑)	MT(↑)	ML(↓)
No Fusion	16.08	41.60	53.84	43.46	29.41	60.18
Late Fusion	29.28	51.08	71.05	59.89	45.25	31.22
Early Fusion	26.19	48.15	67.34	60.87	40.95	32.13
F-Cooper [93]	23.29	43.11	65.63	58.34	35.75	38.91
AttFuse [6]	28.64	50.48	73.21	63.03	46.38	28.05
V2VNet [15]	30.48	54.28	75.53	**64.85**	**48.19**	27.83
V2X-ViT [19]	30.85	54.32	74.01	64.82	45.93	**26.47**
CoBEVT [20]	**32.12**	**55.61**	**77.65**	63.75	47.29	30.32

Table 8.4: **Cooperative Tracking benchmark**. All numbers represent percentages.

8.4.2 Object Tracking

Scope. In this track, we study whether and how object tracking models can obtain benefits from the cooperative system. There are two major approaches to tracking algorithms: joint detection and tracking and tracking by detection. In this paper, we focus on the second class.

Evaluation. We employ the same evaluation metrics in [231, 3] for object tracking, including 1) Multi Object Tracking Accuracy (MOTA), 2) Mostly Tracked Trajectories (MT), 3) Mostly Lost Trajectories (ML), 4) Average Multiobject Tracking Accuracy (AMOTA), 5) Average Multiobject Tracking Precision (AMOTP), and 6) scaled Average Multiobject Tracking Accuracy (sAMOTA). Specifically, the AMOTA and AMOTP average MOTA and MOTP across all recall thresholds, which takes into account the prediction confidence, compared to traditional MOTA and MOTP metrics. sAMOTA is proposed by [3] to guarantee a more linear span over the entire $[0, 1]$ range significantly difficult tracking tasks.

Baselines tracker. We implement AB3Dmot tracker [231] as our baseline tracker. Given the detection results from the cooperative detection models, AB3Dmot combines the 3D Kalman Filter with Birth and Death Memory technique to achieve an efficient and robust tracking performance.

8.4.3 Sim2Real Domain Adaptation

Scope. Data labeling is time-consuming and expensive for the perception system [212]. When it comes to cooperative perception, the cost can dramatically expand as the labelers need to annotate multiple sensor views, which is impossible to scale up. A potential solution is to employ infinite and inexpensive simulation data. However, it is known that there is a significant domain gap between simulated and real-world data distributions. Therefore, this track investigates how to utilize domain adaptation methods to reduce domain discrepancy in the cooperative 3D detection task.

Training. We define the target domain as the V2V4Real dataset and the source domain as a large-scale open simulated OPV2V dataset [6]. The training data consists of two parts: the OPV2V training set with provided annotations, and V2V4Real training set's LiDAR point cloud without access to the labels. Participants should leverage domain adaption algorithms to enable the cooperative detection models to generate domain-invariant features.

Evaluation. The evaluation will be conducted on the test set of V2V4Real dataset under the *Sync* setting, and the assessment protocol is the same as the cooperative 3D object detection track.

Evaluated methods. The baseline method is to train the detection models on OPV2V and directly test on V2V4Real without any domain adaptation. To demonstrate the effectiveness of domain adaptation, we implement a similar method as in [203], which applies two domain classifiers for feature-level and object-level adaption and utilizes gradient reverse layer (GRL) [204] to backpropagate the gradient to assist the model for generating domain-invariant features.

8.5 Experiments

8.5.1 Implementation Details

The dataset is split into the train/validation/test set with 14,210/2,000/3,986 frames, respectively, for all three tasks. All the detection models employ PointPillar [1] as the backbone to extract 2D features from the point cloud. we train all models with 60 epochs, a batch size of 4 per GPU (RTX3090), a learning rate of 0.001 , and we decay the learning rate with a cosine annealing [158]. Early stopping is used to find the best epoch. We also add normal point cloud data augmentations for all experiments, including scaling, rotation, and flip [1]. We employ AdamW [82] with a weight decay of 1×10^{-2} to optimize our models. For the tracking task, we take the previous 3 frames together with the current frame as the inputs.

8.5.2 3D LiDAR Object Detection

Tab. 8.3 demonstrates the quantitive comparison between various cooperative 3D detection models on our V2V4Real dataset. We can observe that:

- Compared to the single-vehicle perception baseline, all cooperative perception methods can significantly boost performance by at least 15.2% in terms of overall AP at IoU 0.5. Furthermore, the accuracy of all evaluation ranges is improved, whereas long-range detection has the most benefits with a minimum of 28.0% and 11.8% gain for AP@0.5 and AP@0.7, respectively.
- Under both *Sync* and *Async* settings, intermediate fusion methods achieve the best trade-off between accuracy and transmission cost. Among all the intermediate fusion methods, CoBEVT has the best performance in terms of AP@0.5, 1.6% higher than the second best model V2X-Vit, 6.8% higher than *Early Fusion*, and 11.5% higher than *Late Fusion* in the *Sync* setting.
- Except for *No Fusion*, all other methods' AP dropped significantly when the com-

munication delay was introduced. For instance, CoBEVT, V2X-ViT, and V2VNet drops 6.3%, 7.6%, and 5.8% at AP@0.7, respectively. This observation highlighted the importance of robustness to the asynchrony for cooperative perception methods.

8.5.3 3D Object Tracking

Tab. 8.4 shows the benchmark results for cooperative tracking. It can be seen that when AB3Dmot combines with cooperative detection, the performance is dramatically better than the single-vehicle tracking method. Similar to the cooperative detection track, CoBEVT [20] achieves the best performance in most of the evaluation metrics, including AMOTA (16.04% higher than baseline), sAMOTA (23.81% higher than baseline), and AMOTP (14.01% better than baseline).

8.5.4 Sim2Real Domain Adaptation

As Tab. 8.5 reveals, there exist serious domain gaps between the simulated dataset OPV2V and our real-world dataset V2V4Real. Without any domain adaptation, only seeing the simulated data will decrease the accuracy of the detection models by 42.2%, 37.1%, 41.3%, 37.5%, 33.9 for AttFuse, F-Cooper, V2VNet, V2X-ViT, and CoBEVT. Applying the domain adaption technique alleviates the performance drop by an average of 7.46%. Furthermore, the strongest model, CoBEVT, can reach 40.2% after employing the domain adaptation, which is higher than the *No Fusion* baseline method that uses real-world data for training.

8.6 Conclusion

We present V2V4Real, a large-scale real-world dataset that covers up to 410 km driving areas, contains 20K LiDAR frames, 40K RGB images, and are annotated with 240K bounding boxes as well as HDMaps, to promote V2V cooperative perception research. We further

Method	AP@IoU=0.5	AP drop
AttFuse [6]	22.5	42.2
AttFuse **w/ D.A.**	23.4 (+0.9)	41.3
F-Cooper [93]	23.6	37.1
F-Cooper **w/ D.A.**	37.3 (+13.7)	23.4
V2VNet [15]	23.2	41.3
V2VNet **w/ D.A.**	26.3 (+3.1)	38.2
V2X-ViT [19]	27.4	37.5
V2X-ViT **w/ D.A.**	39.5 (+12.1)	25.4
CoBEVT [20]	32.6	33.9
CoBEVT **w/ D.A.**	40.2 (+7.6)	26.3

Table 8.5: **Domain Adaptation benchmark**. The number in the bracket indicates the precision gain when using domain adaptation. AP drop refers to the precision gap compared to directly training on the V2V4Real dataset.

introduce three V2V perception benchmarks involving 3D object detection, object tracking, and Sim2Real domain adaptation, which opens up the possibility for future task development. V2V4Real will be made fully available to the public to accelerate the progress of this new field. We will also release the benchmarks and baseline models for camera images and HDMap learning tasks in the next version.

Broader impact. Although the proposed benchmark covers various driving scenes for V2V perception, there may still exist extremely challenging scenarios that do not appear in our training set. In such cases, the models should be trained more carefully in order not to hinder generalization abilities. Out-of-distribution detection is also an important topic that has not been investigated within the scope of this paper. These issues should be taken care of by future related research for robust and safe autonomous perception.

8.7 Acknowledgement

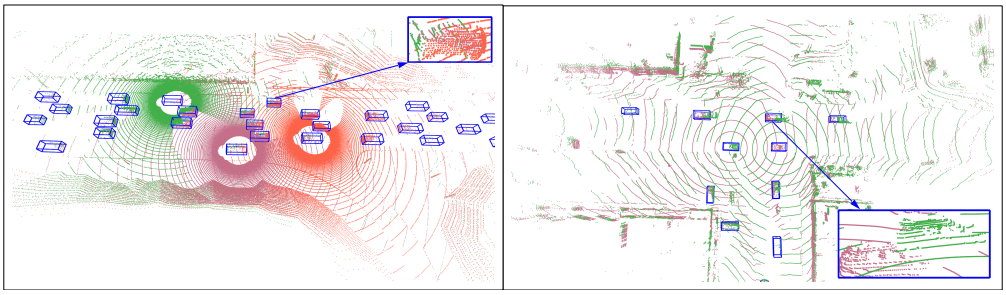
The project belongs to OpenCDA ecosystem [228] and is funded in part by the Federal Highway Administration project and California RIMI Program. Special thanks go to Transportation Research Center Inc for their collaboration in experimental data collection and

processing.

CHAPTER 9

Towards the Optimistic Sim2Real Training Strategy for Cooperative Perception

Cooperative perception in autonomous driving often grapples with the significant costs of real-world data collection and labeling. One practical solution is to employ large-scale economic simulation data for training deep learning models. However, the divergence between simulation and real-world data frequently compromises model performance in real-world scenarios. This paper addresses a critical question: With a large-scale labeled cooperative perception simulation dataset and a mini-scale labeled real-world dataset, what is the optimal training strategy to enhance performance in real-world scenarios? To investigate this, we conducted extensive experiments to explore several categories of training strategies using the simulated OPV2V dataset and the real-world V2V4Real dataset, employing state-of-the-art models. More importantly, we design a domain-tailored plug-in training module named Homogeneous Training Augmenter (HTA), comprising a homogeneous aligner coupled with a data augmenter specially tailored for cooperative perception. Extensive experiments demonstrate that our HTA can significantly improve all training strategies. Remarkably, by utilizing only simulation data and a modest amount of labeled real-world data, our optimal training strategy enabled the state-of-the-art model CoBEVT to attain 55% AP@0.5 and 28% AP@0.7 in the 3D object detection tasks. This achievement surpasses some methods trained on labeled large-scale real-world data, and crucially, it decreases the sim2real 3D object detection disparity to under 8% at AP@0.7.



(a) A snapshot from simulated dataset OPV2V [6]

(b) A snapshot from real-world dataset V2V4Real [23]

Figure 9.1: The gap between simulated and real-world cooperative perception dataset. The real-world dataset has fewer agents and suffers from sensing information misalignment on the same object caused by relative pose error and asynchornization.

9.1 INTRODUCTION

Recently, cooperative perception has attracted significant attention in academics and industry fields for its potential to enhance autonomous driving safety [6, 256, 257, 258, 234]. Despite the potential, the large-scale deployment of cooperative perception remains a considerable challenge. A primary obstacle is the high cost of real-world data collection and labeling. As pointed out in previous studies [23, 234], cooperative perception data collection and labeling costs could double or even triple compared to single vehicle datasets like the Waymo Open Dataset [4] or nuScenes [3], due to the need for a fleet operation.

Researchers have attempted to circumvent this bottleneck by solely utilizing simulated cooperative perception datasets for model training, which can lead to substantial cost savings. In this approach, models are trained exclusively on large-scale, labeled simulation datasets, and domain adaption techniques are employed to bridge the gap between simulation and real-world data [23, 259, 260]. However, simulated and real-world usually have large disparity. Figure 9.1 underscores these discrepancies, particularly the augmented number of agents in simulations and their perfectly aligned sensing information. Therefore, the domain-adaption approaches have a performance deficiency compared to models trained on large-scale real-

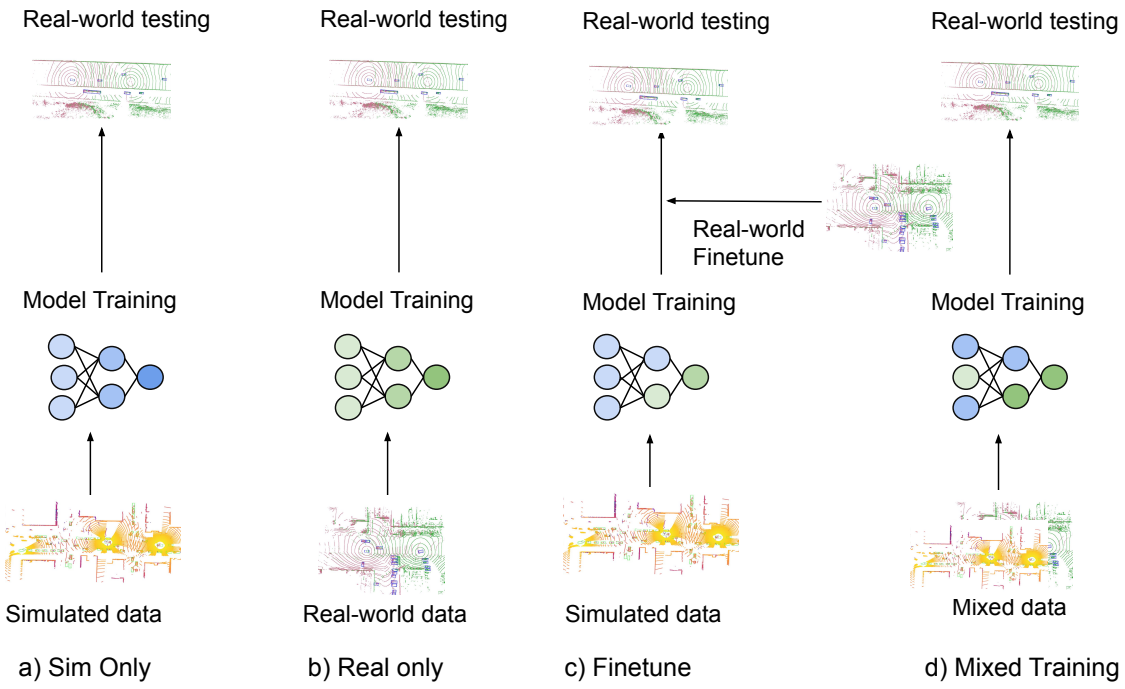


Figure 9.2: **Four basic Sim2Real training strategies for cooperative perception.**

world datasets and fall short in their application as onboard models or for offboard automatic labeling.

In this study, we propose a more practical alternative. Instead of relying solely on simulation data, we label a mini-scale real-world dataset at a negligible cost. Our focus deviates from the domain adaptation algorithms that previous studies have emphasized [23, 259, 260]. Instead, we concentrate on uncovering the optimal training strategy given a large-scale simulation dataset and a mini-scale labeled real-world data. Ideally, the model trained by the best strategy should have the deficient performance to serve as an onboard model for online inference or an auto-labeling model to accelerate the data labeling.

As depicted in Figure 9.2, we investigate several unique combinations of labeled simulated and real-world data. Yet, we assume these basic training strategies will only lead to subopti-

mal results due to two significant disparities between simulated and real-world data: 1) There are usually more agents showing in the simulated data, as the cost of increasing the number of collaborators in simulation is negligible while that for real-world can be significant; 2) Simulated data is usually perfect with the absence of localization and synchronization issues in real-world data. More importantly, since we only have minor-scaled real-world training data, the simulated data occupies a much larger quantitative ratio, which will have a dominant influence on the model and lead to unsatisfied performance on real-world testing. Drawing from these insights, we unveil a domain-tailored plug-in training module named Homogeneous Training Augmenter (HTA), illustrated in Figure 9.3. HTA contains two critical components: Homogeneous Aligner and Data Augmenter. The Homogeneous Aligner will harmonize the number of agents in simulation data during training, mirroring the real-world scenario. Furthermore, it will inject the real-world noises by introducing Gaussian localization noise and transmission time delay, akin to [19], into the simulation data. The Data Augmenter will bolster the ratio of real-world data through randomized data augmentation and repeated instances, thereby counterbalancing its initial minority status.

We conduct experiments on the simulated data OPV2V [6] and real-world data V2V4Real [23] with the employment of SOTA models such as CoBEVT [20] and V2X-ViT [19]. By leveraging our HTA, all training strategies gain significant improvement. More importantly, when we combine HTA and mixed training strategy together, the CoBEVT model can achieve 55% AP@0.5 and 28% AP@0.7 in the 3D object detection task on V2V4Real test set, which even outperforms the late fusion method that trained on V2V4Real large-scale train set. More importantly, the HAMT decreases the Sim2Real gap to around 7% at AP@0.7, outperforming the previous methods that only trained on simulation data by 11.1%. Our contributions can be summarized as the following:

- We extensively explore different training approaches for cooperative perception Sim2Real training that integrates both a large-scale simulation dataset and a mini-scale real-

world dataset. Unlike prior works that mainly emphasize domain adaptation, our focus is on determining the optimal training strategy, aiming to develop a model suitable for onboard online inference and auto-labeling.

- We present the novel Homogeneous Training Augmenter (HTA), a dedicated plug-in training module for cooperative perception, consisting of the Homogeneous Aligner and Data Augmenter. The Homogeneous Aligner adjusts the collaborator volume in simulation data and introduces realistic imperfections to mirror real-world scenarios, while the Data Augmenter increases the mini-scale real-world data's prominence.

- Through extensive evaluations using state-of-the-art models, we showcase the efficacy of our HTA module. When combined with the mixed training strategy, our approach achieves outstanding accuracy in 3D object detection tasks, substantially narrowing the Sim2Real performance gap.

9.2 Related Work

Cooperative Perception Methodology: The advent of cooperative perception has greatly attracted recent research and development, largely owing to its potential to enhance the perceptual capabilities of autonomous vehicles. Within the diverse array of methods, intermediate fusion [6, 19, 15], marked by the transmission of intermediate neural features for collaboration, becomes the most popular choice. It is popular for achieving an optimal balance between bandwidth consumption and performance enhancement. Pioneered by V2VNet [15], graph convolution neural networks are first used to fuse the neural features across different vehicles. DiscoNet [98] subsequently advanced this concept, introducing distillation techniques to augment performance. Following this trajectory, AttFuse [6] innovated the fusion process by incorporating attention mechanisms. V2X-Vit [19] and CoBEVT [20] further improve performance by introducing efficient vision transformers.

Where2Comm [261] pioneered a spatial-confidence-aware communication strategy, utilizing a learned spatial confidence map to highlight critical transmission features, thus enhancing performance while conserving bandwidth.

Cooperative Perception Dataset: Owing to the inherent challenges in extracting real-world data, simulators have become a prevalent source for generating collaborative perception datasets. Among all of the simulated datasets, the most popular two are V2X-Sim [14] and OPV2V [6]. V2X-Sim [14] utilizes SUMO [262] for traffic flow simulation, paired with CARLA [8] for the collection of sensor streams. Gathering data from both roadside units and multiple vehicles, V2X-Sim is versatile, accommodating both Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) scenarios, particularly at intersections. In contrast, OPV2V [6] narrows its scope to concentrate on V2V collaborations exclusively, while expanding the diversity of road types to encompass both highways and city streets. Leveraging OpenCDA [7] and CARLA, OPV2V has gathered data from 73 intriguing scenes, culminating in 11,464 frames and 232,193 annotated 3D vehicle bounding boxes. The connective fabric between vehicles varies per frame, ranging from 2 to 7, with an average approximation of 3. Recognizing the discrepancies that often exist between simulated data and real-world conditions, new real-world datasets and benchmarks have been developed. DAIR-V2X [234], the pioneering large-scale real-world dataset for V2I cooperative perception, encompasses three components: DAIR-V2X-C, DAIR-V2X-I, and DAIR-V2X-V, where DAIR-V2X-C integrates both vehicle and infrastructure sensor information, accounting for $38,845$ camera and LiDAR frames and nearly $464,000$ 3D bounding boxes across 10 classes. Meanwhile, V2V4Real [23] comes out as the first large-scale real-world multimodal dataset designed specifically for V2V perception. Featuring 20K LiDAR frames, 40K RGB frames, and 240K annotated 3D bounding boxes across 5 classes, V2V4Real presents four distinct road types, including intersections, highway entrance ramps, highway straight roads, and city straight roads, captured in Columbus, Ohio, USA. Furthermore, V2V4Real supports three core cooperative perception tasks, namely 3D object detection, 3D object tracking,

and Sim2Real domain adaptation, marking a significant stride in advancing cooperative perception research. In this paper, we select OPV2V as our simulated dataset and V2V4Real as the real-world dataset.

Sim2Real in Cooperative Perception: V2V4Real [23] pioneers the construction of a Sim2Real benchmark in the field of cooperative perception, addressing a critical challenge in bridging simulated and real-world data. In this approach, the training process is exclusively exposed to labeled simulated data from OPV2V and unlabeled real-world data. The model leverages a domain adaptation technique from [203], employing a feature domain discriminator combined with an adversarial gradient reverse layer. This mechanism enhances the cross-domain feature representation, mitigating the Sim2Real performance degradation and achieving an average improvement of 7.46%. Further advancements are made by SR-ViT [259], which contemplates both the implementation gap and the feature gap, while DUSA [260] intelligently deconstructs the collaborative Sim2Real domain adaptation challenge into two interconnected sub-problems: Sim2Real adaptation and inter-agent adaptation, achieving enhanced performance. Nevertheless, despite these technological strides in minimizing the divergence between simulation and real-world data, the results remain suboptimal, and the models produced are not yet suited for practical deployment. Distinct from previous efforts, this paper opts for a hybrid approach, augmenting the exclusive use of simulation datasets with a minimally expensive set of labeled real-world data. The focus here shifts towards uncovering an optimal training strategy within this new paradigm, with the primary goal of delivering a model with practical applications.

9.3 Methodology

In the context of this study, we work with four distinct datasets: a large-scale labeled simulated dataset denoted as S; a small-scale labeled real-world dataset, R_{train}; a labeled real-world validation set, R_{val} for model training's early stopping; and the real-world testing

set, R_{test}. Our goal is to identify an optimal training strategy, that enables the cooperative perception model, M, to maximize the Average Precision (AP) performance on R_{test}. The forthcoming sections will systematically unfold our approach. Section 9.3.1 provides an exhaustive overview of basic training strategies germane to this task, while a quantitative examination of the disparities between simulated and real-world datasets is detailed in Section 9.3.2. Building on these foundational insights, we subsequently present our domain-tailored training strategy in Section 9.3.3, one that is both practical and acutely attuned to the unique challenges of cooperative perception.

9.3.1 Overview of All Training Strategies

Given the simulated dataset S, the real-world training and validation dataset R_{train} and R_{val}, and the real-world testing dataset R_{test}, there are 4 basic training strategies as Fig 9.2 demonstrates: simulated data only training, real-world data only training, two-stage fine-tuning strategy, and mixed data training strategy.

Sim Only: Only the simulated dataset S is used for training. The validation dataset R_{val} can be used for unsupervised domain adaptation.

Real Only: Only the real-world training dataset R_{train} is used for training. This strategy explores whether relying solely on real-world data can achieve better performance.

Two-stage Fine-tuning Strategy: This strategy consists of two stages. In the first stage, the model is trained on the simulated dataset S using the same strategy as Sim Only. In the second stage, the pre-trained model is fine-tuned on the real-world training dataset R_{train}.

Mixed Data Training Strateg: This strategy mixes the simulated dataset S and the real-world training dataset R_{train} together with shuffling operations and sends them to the model for training simultaneously.

While these strategies form the groundwork, they do not fully capture the nuances of

cooperative perception. A deeper analysis of the differences between simulated and real-world data is essential for domain-tailored training strategy.

9.3.2 Analysis of Dataset Gap

In this section, we employ the OPV2V [6] and V2V4Real [23] datasets as representatives of simulated and real-world data, respectively. Our goal is to undertake a quantitative evaluation of the discrepancies between them. We structure this analysis around four principal dimensions:

Number of Agents: We measure the average number of collaborating agents throughout the entirety of each dataset.

Number of Void Feature Values: The varying agent counts lead to differences in shared feature density when using fusion models. To gauge this, we determine the count of zeros in shared features, as generated by the PointPillar [1], before sending it to the fusion model.

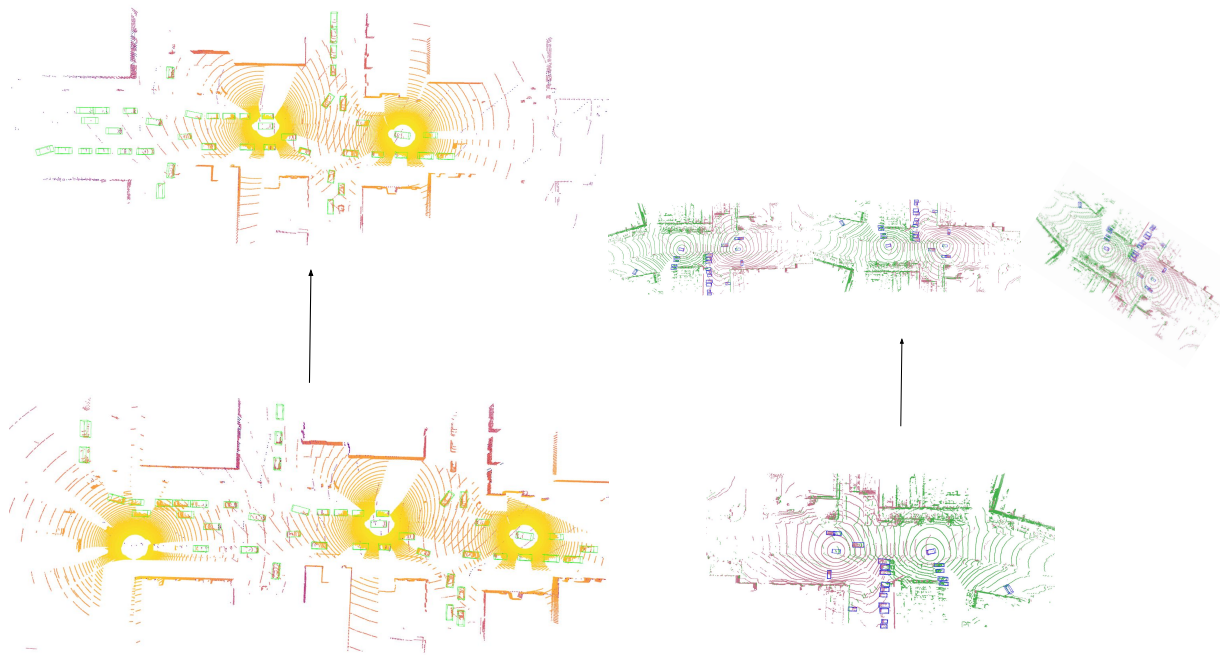
Pose Error: Given the inherent localization inaccuracies in autonomous driving systems, discrepancies in relative poses between agents are anticipated. To quantify this, we project annotated bounding boxes from individual agent coordinates to a standardized coordinate system. By assessing the bounding box center distances for identical objects annotated under varying coordinates, we derive an average indicative of pose error.

Asynchronization Error: Asynchronization discrepancies between distinct real-world sensor systems are inevitable. We measure this by determining the average time difference between two system timestamps.

As detailed in table 9.1, V2V4Real and OPV2V present marked differences across these four metrics. Specifically, OPV2V boasts a higher count of agents, resulting in a notably diminished zero count relative to V2V4Real — a difference of 33.86%. Additionally, V2V4Real registers an average pose error of $0.3961m$ and a time asynchronization of $48ms$. In con-

Dataset	# of Agents	# of Void Feature Values	Pose Noise	Async
V2V4Real	2	74M	0.23/0.18/0.1	48 ms
OPV2V	2.89	55M (-33.86%)	0 /0 /0	0 ms

Table 9.1: The data gap between OPV2V and V2V4Real. The numbers in pose noise represent the mean error on the x, and y direction, and the variance of the Euclidean distance.



(a) Homogeneous Aligner (b) Data Augmenter

Figure 9.3: Two major components of HTA. Homogeneous Aligner will align simulated data with real-world data by reducing the number of agents in simulated data and injecting localization and asynchronization noise. The Data Augmenter will replicate the mini-scale real-world training data and apply rotation and flipping to the point cloud for augmentation.

trast, OPV2V exhibits no such discrepancies. Collectively, these distinctions culminate in a domain gap, posing a challenge to the fusion model's adaptability.

9.3.3 Homogeneous Training Augmenter

In the previous section, we identified several key differences that arise from variations in agent numbers, relative pose noise, and system synchronization. Additionally, models may

overly rely on simulated data and neglect real-world data, especially when the volume of real-world data is limited. Recognizing these challenges, we introduce the Homogeneous Training Augmenter (HTA), comprising a Homogeneous Aligner and a Data Augmenter, as depicted in Fig.9.3.

Homogeneous Aligner. This component aims to reduce the discrepancies between real-world and simulated data. Instead of using all agents' sensor data for training, the homogeneous aligner randomly selects an equivalent number of agents as found in the real-world data. This ensures a consistent feature density. Additionally, the aligner introduces Gaussian noise to the relative pose between agents and adds time delay noise ranging from 0-200 ms. These adjustments are guided by the mean and variance observed in our quantitative analysis (table 9.1), emulating real-world synchronization disparities and localization errors.

Data Augmenter. Given the data-driven nature of deep learning models, reliance on primarily simulated data could lead to performance issues in real-world applications. To counter this, the Data Augmenter serves to enhance the real-world data's presence in the training set. This is achieved through techniques such as copy-paste operations and random point cloud transformation, enabling the limited real-world data to occupy a more substantial portion of the training data.

9.3.4 HTA Integration

Our Homogeneous Training Augmenter (HTA) is designed with flexibility and can be adapted to work with all four primary categories of training strategies.

Sim Only. In this strategy, where only simulated data without labeled real-world data is used, the Homogeneous Aligner is applied to the simulated training data.

Real Only. Since the focus is solely on real-world data without any simulated input, only the Data Augmenter is used to enhance the real-world training dataset.

Two-stage Fine-tuning Strategy. This strategy involves two stages: in the first stage, the Homogeneous Aligner is utilized, while the second stage employs the Data Augmenter to fine-tune the model.

Mixed Data Training Strategy. This balanced approach applies the Homogeneous Aligner to the simulated data and the Data Augmenter to the labeled real-world data, allowing for a harmonized integration of both types of information.

9.4 Experiments

9.4.1 Experiment Setup

We employ the simulated dataset OPV2V [6] and the real-world dataset V2V4Real [23] for our study. From OPV2V, we extract 8745 frames from its training and validation sets, denoted as S. An additional 600 frames, covering 3 distinct scenarios, are labeled from V2V4Real to form R_{train}. We directly source R_{val} and R_{test} from the pre-labeled V2V4Real dataset. The evaluation range is consistent with that of V2V4Real: $x \in [-100, 100]m$ and $y \in [-40, 40]m$. We evaluate model performance using Average Precision (AP) with Intersection over Union (IoU) thresholds of 50% and 70%. We integrate CoBEVT [20] and V2X-ViT [19] for a holistic evaluation. Each model is trained across the four strategies: *Sim Only*, *Real Only*, *Two-Stage Fine-tuning*, and *Mixed Training*, both with and without our HTA module. An additional *Upper Bound* strategy is used for comparison, training the models on V2V4Real's extensive labeled set.

9.4.2 Training Details

Our models are trained on dual RTX-A6000 GPUs using the Adam optimizer [82]. We employ the cosine annealing learning rate scheduler [159] consistently over a 60-epoch duration for all strategies. The best-performing checkpoint is chosen based on the epoch showcasing

the lowest validation loss. With HTA integration, Gaussian noise (mean $0.2m$ on the x and y direction, variance $0.1m^2$) is added to the relative pose. Additionally, uniform noise distribution (0 to 200 ms) is applied to timestamps. Augmenting real-world data bolsters the dataset to 2,400 frames, achieving a 3:10 ratio with the simulated data, a ratio confirmed optimal through subsequent ablation tests. The entire training sequence concludes within three days.

9.4.3 Quantitative Evaluation

Table 9.2 indicates the efficacy of the HTA across all training strategies. With V2X-ViT [19], HTA boosts AP@0.5 for the four primary strategies by respective margins of 2.4%, 1.3%, 4.9%, and 7.3%. For CoBEVT [20], the enhancements are seen by 3.7%, 4.4%, 5.0%, and 6.4%. This demonstrates the versatility and generality of HTA in enhancing various Sim2Real strategies. Notably, the most effective approach is the *Mixed Training* combined with our HTA, scoring 29.5% and 28.8% at AP@0.7, a mere 7.4% and 7.2% below the *Upper Bound*. In V2V4Real, the foundational cooperative perception model late fusion, when trained using *Upper Bound*, secures 26.7% on AP@0.7, suggesting that our optimal Sim2Real strategy is already suitable for real-world applications.

9.4.4 Ablation Study

Given that the CoBEVT model, when coupled with the mixed training strategy, manifests superior performance, it becomes the subject of our ablation studies.

Components Analysis of HTA. We delve into the individual components of HTA to investigate their respective impacts. Table 9.3 illustrates that both the Homogeneous Aligner and the Data Augmenter play pivotal roles in enhancing the model's performance. Specifically, the Homogeneous Aligner elevates the AP@0.5 by 5.4% and the AP@0.7 by 5.2%. Introducing the Data Augmenter can further augment these metrics by 2.1% and 2.2% for

Training Strategy	V2X-ViT		CoBEVT	
	AP@0.5	AP@0.7	AP@0.5	AP@0.7
Upper Bound	64.9	36.9	66.5	36.0
Sim Only	39.5	15.6	40.2	**17.7**
Sim Only w/ HTA	**41.9**	**16.8**	**43.9**	17.4
Real Only	26.3	9.1	21.4	**7.9**
Real Only w/ HTA	**27.0**	**11.1**	**25.8**	7.3
Fine-tune	42.1	19.5	44.0	21.1
Fine-tune w/HTA	**47.0**	**22.1**	**49.0**	**24.4**
Mixed Training	43.5	24.2	45.8	21.4
Mixed Training w/HTA	**50.8**	**29.5**	**53.3**	**28.8**

Table 9.2: Main Results

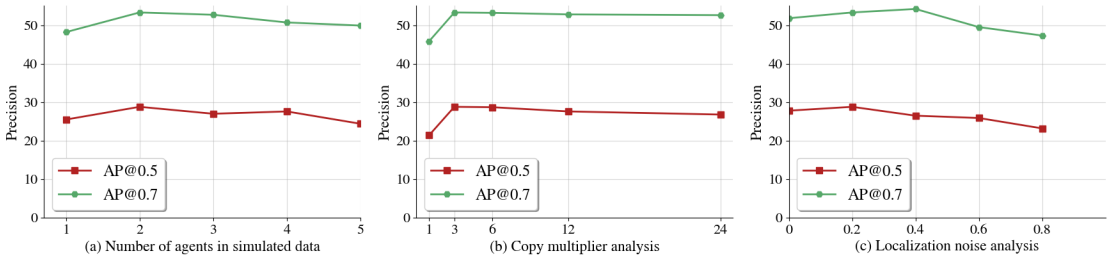
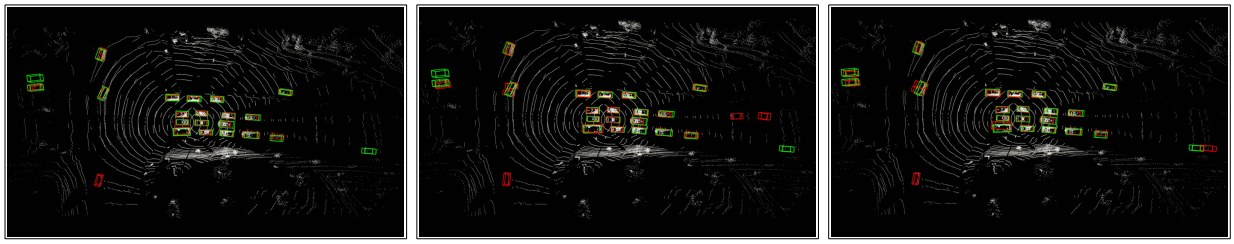


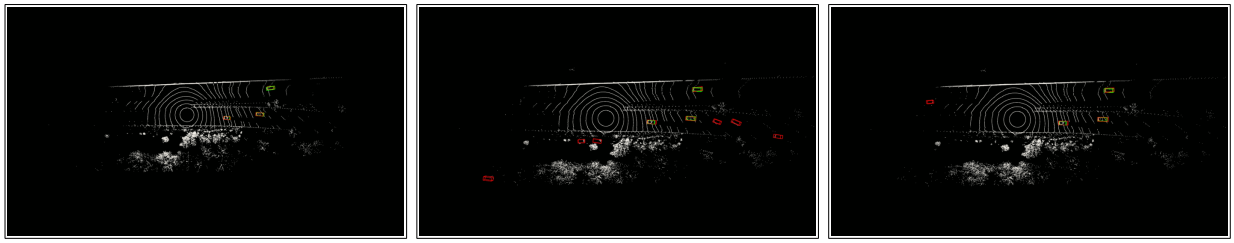
Figure 9.4: Ablation studies. (a) The influence of the number of agents selected by the Homogeneous Aligner on the AP. (b) The influence of different copy multipliers applied by the Data Augmenter. (c) AP vs. localization error added by Homonegeous Aligner.

AP@0.5 and AP@0.7, respectively.

Agent Number Effects. As section 9.3.2 demonstrates, an inconsistent number of agents can introduce very distinct feature distribution. Therefore, one of the primary tasks executed by HTA is ensuring that the number of agents in the simulated data aligns with that in the real-world data. Consequently, understanding the influence of agent numbers on the simulated data becomes important. Figure 9.4 a) reveals that the model's detection accuracy



(a) Upper Bound in intersection. (b) Mixed Training in intersection. (c) Mixed Training w/HTA in intersection



(d) Upper Bound in highway. (e) Mixed Training in highway. (f) Mixed Training w/HTA in highway

Figure 9.5: **The 3D detection visualizations of CoBEVT trained with different strategies.** Green and red 3D bounding boxes represent the ground truth and prediction respectively. With our HTA module, the detection results are clearly improved and close to Upper Bound.

Homogeneous Aligner	Data Augmenter	AP@0.5	AP@0.7
		45.8	21.4
✓		51.2	26.6
✓	✓	53.3	28.8

Table 9.3: Components Analysis of HTA integrated with CoBEVT Mixed Training

peaks when the agent counts in OPV2V match that in V2V4Real (specifically, 2 agents).

Copy-paste Ratio. The Data Augmenter enhances the dataset by replicating and integrating mini-scale real-world data into the larger mixed dataset. We evaluate the optimal number of copies required to attain optimal performance. Figure 9.4 b) shows that a copy multiplier of 3 achieves the best trade-off: any further increment does not lead to substantial gains. Hence, 3 emerges as the optimal multiplier, balancing accuracy with training efficiency.

Localization Error. This study aims to determine the ideal magnitude of localization error to introduce into the pose within OPV2V. Observations from figure 9.4 c) highlight that emulating the pose error consistent with real-world data statistics yields optimal results.

Extra Labeling Cost. Given our incorporation of a labeled mini-scale real-world dataset, it's critical to quantify the additional costs relative to a simulated-only dataset. Engaging a labeler at an hourly rate of 20$ for three hours to label the 600 frames resulted in an additional expenditure of 60$. While this added expense may seem marginal, the resultant boost in Sim2Real performance is notably significant.

9.4.5 Qualitative Analysis

Figure 9.5 demonstrates the 3D detection visualization of CoBEVT trained by different strategies on V2V4Real. It is visually clear that when HTA is employed in mixed training, there are much fewer false positives and negatives, and the visual results are very close to *Upper Bound*.

9.5 Conclusion

In this paper, we explore the best Sim2Real training strategy for cooperative perception given a large-scale labeled simulated dataset and a mini-scale real-world dataset. We propose

Homogeneous Training Augmenter, which is tailored to fill the gap between simulation to realword in cooperative perception and can be used as a plug-in training module for different training strategies. Through extensive experiments, we demonstrate that mixed training strategy combine with HTA can achieve the best performance, largely reduce the Sim2Real gap.

CHAPTER 10

Conclusion and Future Work

With the rapid development of automated driving, Cooperative Driving Assistance (CDA) will play a critical role in revolutionizing the transportation system. My thesis constructs the foundation for prototyping and developing a CDA system. To date, more than 70 institutions worldwide have adopted my simulation framework, open dataset, and proposed algorithms in cooperative perception.

However, there still exists a large space for exploration in this field. For instance, leveraging the recent popular multi-modal foundation models to enhance the robustness of CDA systems is an exciting topic. Furthermore, as more researchers focus on end-to-end driving systems, revising the current CDA architecture to support this trend is another intriguing topic.

Bibliography

[1] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019.

[2] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020.

[4] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020.

[5] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019.

[6] Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and Jiaqi Ma. Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In *IEEE International Conference on Robotics and Automation*, pages 2583–2589. IEEE, 2022.

[7] Runsheng Xu, Yi Guo, Xu Han, Xin Xia, Hao Xiang, and Jiaqi Ma. Opencda: an open cooperative driving automation framework integrated with co-simulation. In *IEEE Intelligent Transportation Systems Conference*, pages 1155–1162. IEEE, 2021.

[8] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of The 6th Conference on Robot Learning*, volume 78, pages 1–16, 2017.

[9] Philip Koopman and Michael Wagner. Challenges in autonomous vehicle testing and validation. *SAE International Journal of Transportation Safety*, 4(1):15–24, 2016.

[10] Bart Van Arem, Cornelie JG Van Driel, and Ruben Visser. The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Transactions on intelligent transportation systems*, 7(4):429–436, 2006.

[11] Jackeline Rios-Torres and Andreas A Malikopoulos. A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1066–1077, 2016.

[12] Jeffery B Greenblatt and Samveg Saxena. Autonomous taxis could greatly reduce greenhouse-gas emissions of us light-duty vehicles. *nature climate change*, 5(9):860–863, 2015.

[13] Daniel J Fagnant and Kara Kockelman. Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice*, 77:167–181, 2015.

[14] Yiming Li, Ziyan An, Zixun Wang, Yiqi Zhong, Siheng Chen, and Chen Feng. V2x-sim: A virtual collaborative perception dataset for autonomous driving. *arXiv preprint arXiv:2202.08449*, 2022.

[15] Tsun-Hsuan Wang, Sivabalan Manivasagam, Ming Liang, Bin Yang, Wenyuan Zeng, and Raquel Urtasun. V2vnet: Vehicle-to-vehicle communication for joint perception and prediction. In *European Conference on Computer Vision*, pages 605–621. Springer, 2020.

[16] Zhenghao Peng, Quanyi Li, Ka Ming Hui, Chunxiao Liu, and Bolei Zhou. Learning to simulate self-driven particles system with coordinated policy optimization. *Advances in Neural Information Processing Systems*, 34:10784–10797, 2021.

[17] Ke Ma, Hao Wang, and Tiancheng Ruan. Analysis of road capacity and pollutant emissions: Impacts of connected and automated vehicle platoons on traffic flow. *Physica A: Statistical Mechanics and its Applications*, 583:126301, 2021.

[18] On-Road Automated Driving (ORAD) committee. Sae j3216 standard: Taxonomy and definitions for terms related to cooperative driving automation for on-road motor vehicles. In *SAE International*, 2020.

[19] Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. *European Conference on Computer Vision*, 2022.

[20] Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, and Jiaqi Ma. Cobevt: Cooperative bird's eye view semantic segmentation with sparse transformers. In *Proceedings of The 6th Conference on Robot Learning*, volume 205, pages 989–1000, 2023.

[21] Runsheng Xu, Weizhe Chen, Hao Xiang, Xin Xia, Lantao Liu, and Jiaqi Ma. Model-agnostic multi-agent perception framework. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1471–1478. IEEE, 2023.

[22] Runsheng Xu, Jinlong Li, Xiaoyu Dong, Hongkai Yu, and Jiaqi Ma. Bridging the domain gap for multi-agent perception. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6035–6042. IEEE, 2023.

[23] Runsheng Xu, Xin Xia, Jinlong Li, Hanzhao Li, Shuo Zhang, Zhengzhong Tu, Zonglin Meng, Hao Xiang, Xiaoyu Dong, Rui Song, et al. V2v4real: A real-world large-scale dataset for vehicle-to-vehicle cooperative perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13712–13722, 2023.

[24] A. Stevens and J. Hopkin. Benefits and deployment opportunities for vehicle/roadside cooperative its. In *IET and ITS Conference on Road Transport Information and Control (RTIC 2012)*, pages 1–6, 2012.

[25] Taylor Lochrane, Laura Dailey, and Corrina Tucker. Carma℠: Driving innovation. *Public Roads*, 83(4), 2020.

[26] Nicholas Hyldmar, Yijun He, and Amanda Prorok. A fleet of miniature cars for experiments in cooperative driving. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3238–3244, 05 2019.

[27] G. vanRossum and J. Deboer. Interactively testing remote servers using the python programming language. *CWI quarterly*, 4:283–304, 1991.

[28] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.

[29] George F. Riley and Thomas R. Henderson. The ns-3 network simulator. In Klaus Wehrle, Mesut Günes, and James Gross, editors, *Modeling and Tools for Network Simulation*, pages 15–34. Springer, 2010.

[30] Mechanical Simulation . Carsim, 2021.

[31] Liu Hao, Xingan David Kan Lin Xiao, Xiao-Yun Lu Steven E. Shladover, Wouter Schakel Meng Wang, and Bart van Arem. Using cooperative adaptive cruise control (cacc) to form high-performance vehicle streams.final report. 2018.

[32] E SHLADOVER STEVEN, C Nowakowski, XY Lu, and R Ferlis. Cooperative adaptive cruise control (cacc) definitions and operating concepts. In *2015 TRB ANNUAL MEETING*, volume 1, pages 1–16, 2015.

[33] Yi Guo and Jiaqi Ma. Leveraging existing high-occupancy vehicle lanes for mixed-autonomy traffic management with emerging connected automated vehicle applications. *Transportmetrica A: Transport Science*, 16(3):1375–1399, 2020.

[34] Jiaqi Ma, Edward Leslie, Amir Ghiasi, and Yi Guo. Empirical analysis of a freeway bundled connected-and-automated vehicle application using experimental data. *Journal of Transportation Engineering*, 146, 04 2020.

[35] Amir Ghiasi, Jiaqi Ma, Fang Zhou, and Xiaopeng Li. Speed harmonization algorithm using connected autonomous vehicles. In *96th Annual Meeting of the Transportation Research Board*, number 17-02568, 2017.

[36] Amir Ghiasi, Xiaopeng Li, and Jiaqi Ma. A mixed traffic speed harmonization model with connected autonomous vehicles. *Transportation Research Part C: Emerging Technologies*, 104:210–233, 2019.

[37] Jiaqi Ma, Xiaopeng Li, Steven Shladover, Hesham A. Rakha, Xiao-Yun Lu, Ramanujan Jagannathan, and Daniel J. Dailey. Freeway speed harmonization. *IEEE Transactions on Intelligent Vehicles*, 1(1):78–89, 2016.

[38] Alireza Talebpour, Hani S. Mahmassani, and Samer H. Hamdar. Speed harmonization: Evaluation of effectiveness under congested conditions. *Transportation Research Record*, 2391(1):69–79, 2013.

[39] Fang Zhou, Xiaopeng Li, and Jiaqi Ma. Parsimonious shooting heuristic for trajectory design of connected automated traffic part i: Theoretical analysis with generalized time geography. *Transportation Research Part B Methodological*, 95, 11 2015.

[40] Yiheng Feng, Larry Head, Shayan Khoshmagham, and Mehdi Zamanipour. A real-time adaptive signal control in a connected vehicle environment. *Transportation Research Part C: Emerging Technologies*, 55, 01 2015.

[41] Chunhui Yu, Yiheng Feng, Henry X. Liu, Wanjing Ma, and Xiaoguang Yang. Integrated optimization of traffic signals and vehicle trajectories at isolated urban intersections. *Transportation Research Part B: Methodological*, 112:89–112, 2018.

[42] Michele Segata, Stefan Joerer, Bastian Bloessl, Christoph Sommer, Falko Dressler, and Renato Lo Cigno. Plexe: A platooning extension for veins. *IEEE Vehicular Networking Conference, VNC*, 2015, 12 2014.

[43] C. Sommer, R. German, and F. Dressler. Bidirectionally coupled network and road traffic simulation for improved ivc analysis. *IEEE Transactions on Mobile Computing*, 10(1):3–15, 2011.

[44] Cathy Wu, Aboudy Kreidieh, Kanaad Parvate, Eugene Vinitsky, and Alexandre M Bayen. Flow: A modular learning framework for autonomy in traffic, 2020.

[45] Epic Games. Unreal engine.

[46] C. Olaverri-Monreal, Javier Errea-Moreno, Alberto Díaz-Álvarez, Carlos Biurrun-Quel, Luis Serrano-Arriezu, and Markus Kuba. Connection of the sumo microscopic traffic simulator and the unity 3d game engine to evaluate v2x communication-based systems. *Sensors (Basel, Switzerland)*, 18, 2018.

[47] Arne Kesting, Martin Treiber, and Dirk Helbing. Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1928):4585–4605, 2010.

[48] U.S. Department of Transportation Federal Highway Administration. Next generation simulation (ngsim) vehicle trajectories and supporting data, 2016.

[49] Jiaqi Ma, Fang Zhou, Zhitong Huang, Christopher Melson, Rachel James, and Xiaoxiao Zhang. Hardware-in-the-loop testing of connected and automated vehicle applications: A use case for queue-aware signalized intersection approach and departure. *Transportation Research Record Journal of the Transportation Research Board*, 2672, 01 2018.

[50] OpenDRIVE. Opendrive — Wikipedia, the free encyclopedia, 2005.

[51] FBX. Fbx file format — Wikipedia, the free encyclopedia, 2006.

[52] YAML. Yaml — Wikipedia, the free encyclopedia, 2001.

[53] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, ChristopherSTAN, Liu Changyu, Laughing, tkianai, Adam Hogan, lorenzomammana, yxNONG, AlexWang1900, Laurentiu Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Francisco Ingham, Frederik, Guilhen, Hatovix, Jake Poznanski, Jiacong Fang, Lijun Yu, changyu98, Mingyu Wang, Naman Gupta, Osama Akhtar, PetrDvoracek, and Prashant Rai. ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements, October 2020.

[54] Xin Xia, Ehsan Hashemi, Lu Xiong, Amir Khajepour, and Nan Xu. Autonomous vehicles sideslip angle estimation: Single antenna gnss/imu fusion with observability analysis. *IEEE Internet of Things Journal*, 2021.

[55] Lu Xiong, Xin Xia, Yishi Lu, Wei Liu, Letian Gao, Shunhui Song, and Zhuoping Yu. Imu-based automated vehicle body sideslip angle and attitude estimation aided by gnss using parallel adaptive kalman filters. *IEEE Transactions on Vehicular Technology*, 69(10):10668–10680, 2020.

[56] Anoop Sathyan, Jiaqi Ma, and Kelly Cohen. Decentralized cooperative driving automation: A reinforcement learning framework using genetic fuzzy systems. *Transportmetrica B*, 07 2021.

[57] Francesco Bella and Roberta Russo. A collision warning system for rear-end collision: a driving simulator study. *Procedia - Social and Behavioral Sciences*, 20:676–686, 12 2011.

[58] Gerrit Naus, Rene Vugts, Jeroen Ploeg, M.J.G. Molengraft, and M. Steinbuch. String-stable cacc design and experimental validation: A frequency-domain approach. *Vehicular Technology, IEEE Transactions on*, 59:4268 – 4279, 12 2010.

[59] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 82–92, 2019.

[60] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[61] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[62] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[63] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10526–10535, 2020.

[64] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7345–7353, 2019.

[65] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 663–678, Cham, 2018. Springer International Publishing.

[66] Q. Chen, S. Tang, Q. Yang, and S. Fu. Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 514–524, Los Alamitos, CA, USA, jul 2019. IEEE Computer Society.

[67] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

[68] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[69] Tsun-Hsuan Wang, Sivabalan Manivasagam, Ming Liang, Binh Yang, Wenyuan Zeng, J. Tu, and R. Urtasun. V2vnet: Vehicle-to-vehicle communication for joint perception and prediction. In *ECCV*, 2020.

[70] Zaydoun Yahya Rawashdeh and Zheng Wang. Collaborative automated driving: A machine learning-based method to enhance the accuracy of shared information. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3961–3966, 2018.

[71] Zijian Zhang, Shuai Wang, Yuncong Hong, Liangkai Zhou, and Qi Hao. Distributed dynamic map fusion via federated learning for intelligent networked vehicles. *ArXiv*, abs/2103.03786, 2021.

[72] Andreas Rauch, Felix Klanner, Ralph Rasshofer, and Klaus Dietmayer. Car2x-based perception in a high-level fusion architecture for cooperative perception systems. In *2012 IEEE Intelligent Vehicles Symposium*, pages 270–275, 2012.

[73] Zaydoun Yahya Rawashdeh and Zheng Wang. Collaborative automated driving: A machine learning-based method to enhance the accuracy of shared information. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3961–3966, 2018.

[74] Qi Chen, Xu Ma, Sihai Tang, Jingda Guo, Qing Yang, and Song Fu. F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, SEC '19, page 88–100, New York, NY, USA, 2019. Association for Computing Machinery.

[75] Sivabalan Manivasagam, Shenlong Wang, Kelvin Wong, Wenyuan Zeng, Mikita Sazanovich, Shuhan Tan, Bin Yang, Wei-Chiu Ma, and Raquel Urtasun. Lidarsim: Realistic lidar simulation by leveraging the real world. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11167–11176, 2020.

[76] Ehsan Emad Marvasti, Arash Raftari, Amir Emad Marvasti, Yaser P Fallah, Rui Guo, and Hongsheng Lu. Cooperative lidar object detection via feature sharing in deep networks. In *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, pages 1–7. IEEE, 2020.

[77] Roadrunner: Design 3d scenes for automated driving simulation.

[78] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015.

[79] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[80] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

[81] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018.

[82] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[83] Fabio Arena and Giovanni Pau. An overview of vehicular communications. *Future Internet*, 11(2):27, 2019.

[84] Michael Treml, José Arjona-Medina, Thomas Unterthiner, Rupesh Durgesh, Felix Friedmann, Peter Schuberth, Andreas Mayr, Martin Heusel, Markus Hofmarcher, Michael Widrich, et al. Speeding up semantic segmentation for autonomous driving. In *NeurIPS workshop MLITS*, 2016.

[85] Khaled El Madawi, Hazem Rashed, Ahmad El Sallab, Omar Nasr, Hanan Kamel, and Senthil Yogamani. Rgb and lidar fusion based 3d semantic segmentation for autonomous driving. In *IEEE Intelligent Transportation Systems Conference*, pages 7–12. IEEE, 2019.

[86] Zhongkai Zhou, Xinnan Fan, Pengfei Shi, and Yuanxue Xin. R-msfm: Recurrent multi-scale feature modulation for monocular depth estimating. In *IEEE International Conference on Computer Vision*, pages 12777–12786, 2021.

[87] Xinnan Fan, Zhongkai Zhou, Pengfei Shi, Yuanxue Xin, and Xuan Zhou. Rafm: Recurrent atrous feature modulation for accurate monocular depth estimating. *IEEE Signal Processing Letters*, pages 1–5, 2022.

[88] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *European Conference on Computer Vision*, pages 641–656. Springer, 2018.

[89] Zhao Zelin, Wu Ze, Zhuang Yueqing, Li Boxun, and Jia Jiaya. Tracking objects as pixel-wise distributions. *arXiv preprint arXiv:2207.05518*, 2022.

[90] Zhaoxin Fan, Yazhi Zhu, Yulin He, Qi Sun, Hongyan Liu, and Jun He. Deep learning on monocular object pose detection and tracking: A comprehensive overview. *ACM Computing Surveys (CSUR)*, 2021.

[91] Zixu Zhang and Jaime F Fisac. Safe occlusion-aware autonomous driving via game-theoretic active perception. *arXiv preprint arXiv:2105.08169*, 2021.

[92] Qi Chen, Sihai Tang, Qing Yang, and Song Fu. Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds. In *IEEE 39th International Conference on Distributed Computing Systems*, pages 514–524. OPTorganization, 2019.

[93] Qi Chen, Xu Ma, Sihai Tang, Jingda Guo, Qing Yang, and Song Fu. F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pages 88–100, 2019.

[94] Nicholas Vadivelu, Mengye Ren, James Tu, Jingkang Wang, and Raquel Urtasun. Learning to communicate and correct pose errors. *arXiv preprint arXiv:2011.05289*, 2020.

[95] Andreas Rauch, Felix Klanner, Ralph Rasshofer, and Klaus Dietmayer. Car2x-based perception in a high-level fusion architecture for cooperative perception systems. In *IEEE Intelligent Vehicles Symposium*, pages 270–275. OPTorganization, 2012.

[96] Zaydoun Yahya Rawashdeh and Zheng Wang. Collaborative automated driving: A machine learning-based method to enhance the accuracy of shared information. In *IEEE Intelligent Transportation Systems Conference*, pages 3961–3966. IEEE, 2018.

[97] Ian Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *International Conference on Machine Learning*, pages 1319–1327. PMLR, 2013.

[98] Yiming Li, Shunli Ren, Pengxiang Wu, Siheng Chen, Chen Feng, and Wenjun Zhang. Learning distilled collaboration graph for multi-agent perception. *Advances in Neural Information Processing Systems*, 34:29541–29552, 2021.

[99] Yanghui Mo, Peilin Zhang, Zhijun Chen, and Bin Ran. A method of vehicle-infrastructure cooperative perception based vehicle state information fusion using improved kalman filter. *Multimedia Tools and Applications*, pages 1–18, 2021.

[100] Xiangmo Zhao, Kenan Mu, Fei Hui, and Christian Prehofer. A cooperative vehicle-infrastructure based urban driving environment perception method using a ds theory-based credibility map. *Optik*, 138:407–415, 2017.

[101] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019.

[102] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020.

[103] Yuanxin Zhong, Minghan Zhu, and Huei Peng. Vin: Voxel-based implicit network for joint 3d object detection and segmentation for lidars. *arXiv preprint arXiv:2107.02980*, 2021.

[104] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1951–1960, 2019.

[105] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[106] Zhaoxin Fan, Zhenbo Song, Hongyan Liu, Zhiwu Lu, Jun He, and Xiaoyong Du. Svt-net: Super light-weight sparse voxel transformer for large scale place recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 551–560, 2022.

[107] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10012–10022, 2021.

[108] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12124–12134, 2022.

[109] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *Advances in Neural Information Processing Systems*, 34:9355–9366, 2021.

[110] Zhendong Wang, Xiaodong Cun, Jianmin Bao, and Jianzhuang Liu. Uformer: A general u-shaped transformer for image restoration. *arXiv preprint arXiv:2106.03106*, 2021.

[111] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxim: Multi-axis mlp for image processing. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5769–5780, 2022.

[112] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12894–12904, 2021.

[113] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *European Conference on Computer Vision*, pages 459–479, 2022.

[114] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of The Web Conference*, pages 2704–2710, 2020.

[115] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, 2015.

[116] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

[117] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2736–2746, 2022.

[118] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *IEEE International Conference on Computer Vision*, pages 3464–3473, 2019.

[119] Xin Xia, Peng Hang, Nan Xu, Yanjun Huang, Lu Xiong, and Zhuoping Yu. Advancing estimation accuracy of sideslip angle by fusing vehicle kinematics and dynamics information with fuzzy logic. *IEEE Transactions on Vehicular Technology*, 2021.

[120] You Li, Yuan Zhuang, Xin Hu, Zhouzheng Gao, Jia Hu, Long Chen, Zhe He, Ling Pei, Kejie Chen, Maosong Wang, et al. Toward location-enabled iot (le-iot): Iot positioning techniques, error sources, and error mitigation. *IEEE Internet of Things Journal*, 8(6):4035–4062, 2020.

[121] Rt3000. https://www.oxts.com/products/rt3000-v3. Accessed: 2021-11-11.

[122] John B Kenney. Dedicated short-range communications (dsrc) standards in the united states. *Proceedings of the IEEE*, 99(7):1162–1182, 2011.

[123] Manabu Tsukada, Takaharu Oi, Akihide Ito, Mai Hirata, and Hiroshi Esaki. Autoc2x: Open-source software to realize v2x cooperative perception among autonomous vehicles. In *Vehicular Technology Conference*, pages 1–6. OPTorganization, 2020.

[124] Andreas Rauch, Felix Klanner, and Klaus Dietmayer. Analysis of v2x communication parameters for the development of a fusion architecture for cooperative perception systems. In *IEEE Intelligent Vehicles Symposium*, pages 685–690. OPTorganization, 2011.

[125] Mong H Ng, Kaahan Radia, Jianfei Chen, Dequan Wang, Ionel Gog, and Joseph E Gonzalez. BEV-Seg: Bird's eye view semantic segmentation using geometry and semantic point cloud. *arXiv preprint arXiv:2006.11436*, 2020.

[126] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

[127] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15273–15282, 2021.

[128] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European Conference on Computer Vision*, pages 194–210. Springer, 2020.

[129] Syed Ammar Abbas and Andrew Zisserman. A geometric approach to obtain a bird's eye view from an image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[130] Youngseok Kim and Dongsuk Kum. Deep learning based vehicle position and orientation estimation via inverse perspective mapping image. In *2019 IEEE Intelligent Vehicles Symposium*, pages 317–323, 2019.

[131] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *European Conference on Computer Vision*, 2022.

[132] Weixiang Yang, Qi Li, Wenxi Liu, Yuanlong Yu, Yuexin Ma, Shengfeng He, and Jia Pan. Projecting your view attentively: Monocular road scene layout estimation via cross-view transformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15536–15545, 2021.

[133] Avishkar Saha, Oscar Mendez Maldonado, Chris Russell, and Richard Bowden. Translating images into maps. *arXiv preprint arXiv:2110.00966*, 2021.

[134] Yunshuang Yuan, Hao Cheng, and Monika Sester. Keypoints-based deep feature fusion for cooperative vehicle detection of autonomous driving. *IEEE Robotics and Automation Letters*, 2022.

[135] Hanspeter A Mallot, Heinrich H Bülthoff, JJ Little, and Stefan Bohrer. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological cybernetics*, 64(3):177–185, 1991.

[136] Lennart Reiher, Bastian Lampe, and Lutz Eckstein. A sim2real deep learning approach for the transformation of images from multiple vehicle-mounted cameras to a semantically segmented image in bird's eye view. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7. IEEE, 2020.

[137] Tom Bruls, Horia Porav, Lars Kunze, and Paul Newman. The right (angled) perspective: Improving the understanding of road scenes using boosted inverse perspective mapping. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 302–309. IEEE, 2019.

[138] Minghan Zhu, Songan Zhang, Yuanxin Zhong, Pingping Lu, Huei Peng, and John Lenneman. Monocular 3d vehicle detection using uncalibrated traffic cameras through homography. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3814–3821. IEEE, 2021.

[139] Bowen Pan, Jiankai Sun, Ho Yin Tiga Leung, Alex Andonian, and Bolei Zhou. Cross-view semantic segmentation for sensing surroundings. *IEEE Robotics and Automation Letters*, 5(3):4867–4873, 2020.

[140] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

[141] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021.

[142] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. *arXiv preprint arXiv:2107.00641*, 2021.

[143] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34, 2021.

[144] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In *European Conference on Computer Vision*, pages 108–126. Springer, 2020.

[145] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846, 2021.

[146] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Improved multiscale vision transformers for classification and detection. *arXiv preprint arXiv:2112.01526*, 2021.

[147] Juhong Min, Yucheng Zhao, Chong Luo, and Minsu Cho. Peripheral vision transformer. *arXiv preprint arXiv:2206.06801*, 2022.

[148] Ali Hatamizadeh, Hongxu Yin, Jan Kautz, and Pavlo Molchanov. Global context vision transformers. *arXiv preprint arXiv:2206.09959*, 2022.

[149] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021.

[150] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6824–6835, 2021.

[151] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. *Advances in Neural Information Processing Systems*, 34:28522–28535, 2021.

[152] Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *Advances in neural information processing systems*, 34, 2021.

[153] Hanxiao Liu, Zihang Dai, David So, and Quoc Le. Pay attention to mlps. *Advances in Neural Information Processing Systems*, 34, 2021.

[154] James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms. *arXiv preprint arXiv:2105.03824*, 2021.

[155] Zilong Huang, Youcheng Ben, Guozhong Luo, Pei Cheng, Gang Yu, and Bin Fu. Shuffle transformer: Rethinking spatial shuffle for vision transformer. *arXiv preprint arXiv:2106.03650*, 2021.

[156] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[157] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International conference on artificial neural networks*, pages 52–59. Springer, 2011.

[158] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[159] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[160] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. *arXiv preprint arXiv:1811.08188*, 2018.

[161] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[162] Min Hua, Guoying Chen, Buyang Zhang, and Yanjun Huang. A hierarchical energy efficiency optimization control strategy for distributed drive electric vehicles. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 233(3):605–621, 2019.

[163] Guoying Chen, Xuanming Zhao, Zhenhai Gao, and Min Hua. Dynamic drifting control for general path tracking of autonomous vehicles. *IEEE Transactions on Intelligent Vehicles*, 2023.

[164] Zixing Lei, Shunli Ren, Yue Hu, Wenjun Zhang, and Siheng Chen. Latency-aware collaborative perception. In *European Conference on Computer Vision*, 2022.

[165] Wei Liu, Xin Xia, Lu Xiong, Yishi Lu, Letian Gao, and Zhuoping Yu. Automated vehicle sideslip angle estimation considering signal measurement characteristic. *IEEE Sensors Journal*, 21(19):21675–21687, 2021.

[166] Rui Song, Liguo Zhou, Venkatnarayanan Lakshminarasimhan, Andreas Festag, and Alois Knoll. Federated learning framework coping with hierarchical heterogeneity in cooperative its. *arXiv preprint arXiv:2204.00215*, 2022.

[167] Rui Song, Anupama Hegde, Numan Senel, Alois Knoll, and Andreas Festag. Edge-aided sensor data sharing in vehicular communication networks. In *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, pages 1–7, 2022.

[168] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations*, 2016.

[169] Johan Thunberg, Galina Sidorenko, Katrin Sjöberg, and Alexey Vinel. Efficiently bounding the probabilities of vehicle collision at intelligent intersections. *IEEE Open Journal of Intelligent Transportation Systems*, 2:47–59, 2021.

[170] Hua Xie, Yunjia Wang, Xieyang Su, Shengchun Wang, and Liang Wang. Safe driving model based on v2v vehicle communication. *IEEE Open Journal of Intelligent Transportation Systems*, 3:449–457, 2022.

[171] Qiang Han, Yong-Shuai Zhou, Yu-Xin Tang, Xian-Guo Tuo, and Ping He. Event-triggered finite-time sliding mode control for leader-following second-order nonlinear multi-agent systems. *IEEE Open Journal of Intelligent Transportation Systems*, 3:570–579, 2022.

[172] Steven E. Shladover. Opportunities and challenges in cooperative road vehicle automation. *IEEE Open Journal of Intelligent Transportation Systems*, 2:216–224, 2021.

[173] Rodolfo Valiente, Behrad Toghi, Ramtin Pedarsani, and Yaser P Fallah. Robustness and adaptability of reinforcement learning-based cooperative autonomous driving in mixed-autonomy traffic. *IEEE Open Journal of Intelligent Transportation Systems*, 3:397–410, 2022.

[174] Simeon C Calvert and Giulio Mecacci. A conceptual control system description of cooperative and automated driving in mixed urban traffic with meaningful human control for design and evaluation. *IEEE Open Journal of Intelligent Transportation Systems*, 1:147–158, 2020.

[175] Rahi Avinash Shet and Shengyue Yao. Cooperative driving in mixed traffic: An infrastructure-assisted approach. *IEEE Open Journal of Intelligent Transportation Systems*, 2:429–447, 2021.

[176] Antonella Ferrara, Gian Paolo Incremona, Eugeniu Birliba, and Paola Goatin. Multi-scale model based hierarchical control of freeway traffic via platoons of connected and automated vehicles. *IEEE Open Journal of Intelligent Transportation Systems*, 2022.

[177] Yasser H Khalil and Hussein T Mouftah. Licanet: Further enhancement of joint perception and motion prediction based on multi-modal fusion. *IEEE Open Journal of Intelligent Transportation Systems*, 3:222–235, 2022.

[178] Sou Kitajima, Hanna Chouchane, Jacobo Antona-Makoshi, Nobuyuki Uchida, and Jun Tajima. A nationwide impact assessment of automated driving systems on traffic safety using multiagent traffic simulations. *IEEE Open Journal of Intelligent Transportation Systems*, 3:302–312, 2022.

[179] Hanjiang Hu, Zuxin Liu, Sharad Chitlangia, Akhil Agnihotri, and Ding Zhao. Investigating the impact of multi-lidar placement on object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2550–2559, 2022.

[180] Sanbao Su, Yiming Li, Sihong He, Songyang Han, Chen Feng, Caiwen Ding, and Fei Miao. Uncertainty quantification of collaborative detection for self-driving. *arXiv preprint arXiv:2209.08162*, 2022.

[181] Zelin Zhao, Ze Wu, Yueqing Zhuang, Boxun Li, and Jiaya Jia. Tracking objects as pixel-wise distributions. *arXiv preprint arXiv:2207.05518*, 2022.

[182] Zelin Zhao and Jiaya Jia. End-to-end view synthesis via nerf attention. *arXiv preprint arXiv:2207.14741*, 2022.

[183] Zelin Zhao, Karan Samel, Binghong Chen, and lee song. Proto: Program-guided transformer for program-guided tasks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 17021–17036. Curran Associates, Inc., 2021.

[184] Wei Liu, Karoll Quijano, and Melba Crawford. Yolov5-tassel: Detecting tassels in rgb uav imagery with improved yolov5 based on transfer learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2022.

[185] Shaoyue Song, Hongkai Yu, Zhenjiang Miao, Jianwu Fang, Kang Zheng, Cong Ma, and Song Wang. Multi-spectral salient object detection by adversarial domain adaptation. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 12023–12030, 2020.

[186] Wei Shao, Sichen Zhao, Zhen Zhang, Shiyu Wang, Mohammad Saiedur Rahaman, Andy Song, and Flora D Salim. Fadacs: A few-shot adversarial domain adaptation architecture for context-aware parking availability sensing. In *2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–10. IEEE, 2021.

[187] Zhekai Du, Jingjing Li, Hongzu Su, Lei Zhu, and Ke Lu. Cross-domain gradient discrepancy minimization for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3937–3946, 2021.

[188] Jinlong Li, Zhigang Xu, Lan Fu, Xuesong Zhou, and Hongkai Yu. Domain adaptation from daytime to nighttime: A situation-sensitive vehicle detection and traffic flow parameter estimation framework. *Transportation Research Part C: Emerging Technologies*, 124:102946, 2021.

[189] Lan Fu, Hongkai Yu, Felix Juefei-Xu, Jinlong Li, Qing Guo, and Song Wang. Let there be light: Improved traffic surveillance via detail preserving night-to-day transfer. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[190] Tongkun Xu, Weihua Chen, WANG Pichao, Fan Wang, Hao Li, and Rong Jin. Cdtrans: Cross-domain transformer for unsupervised domain adaptation. In *International Conference on Learning Representations*, 2021.

[191] Xingxu Yao, Sicheng Zhao, Pengfei Xu, and Jufeng Yang. Multi-source domain adaptation for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3273–3282, 2021.

[192] Zhaoxin Fan, Yulin He, Zhicheng Wang, Kejian Wu, Hongyan Liu, and Jun He. Reconstruction-aware prior distillation for semi-supervised point cloud completion. *arXiv preprint arXiv:2204.09186*, 2022.

[193] Ling Yang, Liangliang Li, Zilun Zhang, Xinyu Zhou, Erjin Zhou, and Yu Liu. Dpgn: Distribution propagation graph network for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13390–13399, 2020.

[194] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

[195] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016.

[196] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017.

[197] Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5031–5040, 2019.

[198] Hanjiang Hu, Zhijian Qiao, Ming Cheng, Zhe Liu, and Hesheng Wang. Dasgil: Domain adaptation for semantic and geometric-aware image-based localization. *IEEE Transactions on Image Processing*, 30:1342–1353, 2020.

[199] Ling Yang and Shenda Hong. Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion. In *International Conference on Machine Learning*, pages 25038–25054. PMLR, 2022.

[200] Zhengfa Liu, Guang Chen, Zhijun Li, Yu Kang, Sanqing Qu, and Changjun Jiang. Psdc: A prototype-based shared-dummy classifier model for open-set domain adaptation. *IEEE Transactions on Cybernetics*, pages 1–14, 2022.

[201] Sanqing Qu, Guang Chen, Jing Zhang, Zhijun Li, Wei He, and Dacheng Tao. Bmd: A general class-balanced multicentric dynamic prototype strategy for source-free domain adaptation. In *European conference on computer vision*, 2022.

[202] Hossein Talebi and Peyman Milanfar. Learning to resize images for computer vision tasks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 497–506, 2021.

[203] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018.

[204] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[205] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Yingxia Shao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *arXiv preprint arXiv:2209.00796*, 2022.

[206] Yantao Lu, Xuetao Hao, Shiqi Sun, Weiheng Chai, Muchenxuan Tong, and Senem Velipasalar. Raanet: Range-aware attention network for lidar-based 3d object detection with auxiliary density level estimation, 2021.

[207] Peixi Xiong, Xuetao Hao, Yunming Shao, and Jerry Yu. Adaptive attention model for lidar instance segmentation. In *International Symposium on Visual Computing*, pages 141–155. Springer, 2019.

[208] Heming Du, Xin Yu, and Liang Zheng. Learning object relation graph and tentative policy for visual navigation. In *European Conference on Computer Vision*, pages 19–34. Springer, 2020.

[209] Anwesan Pal, Yiding Qiu, and Henrik Christensen. Learning hierarchical relationships for object-goal navigation. In *Conference on Robot Learning*, pages 517–528. PMLR, 2021.

[210] Ya Wu, Ariyan Bighashdel, Guang Chen, Gijs Dubbelman, and Pavol Jancura. Continual pedestrian trajectory learning with social generative replay. *IEEE Robotics and Automation Letters*, 8(2):848–855, 2023.

[211] Wei Liu, Lu Xiong, Xin Xia, Yishi Lu, Letian Gao, and Shunhui Song. Vision-aided intelligent vehicle sideslip angle estimation based on a dynamic model. *IET Intelligent Transport Systems*, 14(10):1183–1189, 2020.

[212] Hao Xiang, Runsheng Xu, Xin Xia, Zhaoliang Zheng, Bolei Zhou, and Jiaqi Ma. V2xp-asg: Generating adversarial scenes for vehicle-to-everything perception. *arXiv preprint arXiv:2209.13679*, 2022.

[213] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 850–855. IEEE, 2006.

[214] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

[215] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[216] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[217] Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[218] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1, pages 609–616. Citeseer, 2001.

[219] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[220] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699, 2002.

[221] Ananya Kumar, Percy S Liang, and Tengyu Ma. Verified uncertainty calibration. *Advances in Neural Information Processing Systems*, 32, 2019.

[222] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017.

[223] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. Learning non-maximum suppression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4507–4515, 2017.

[224] Songtao Liu, Di Huang, and Yunhong Wang. Adaptive nms: Refining pedestrian detection in a crowd. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6459–6468, 2019.

[225] Rasmus Rothe, Matthieu Guillaumin, and Luc Van Gool. Non-maximum suppression for object detection by passing messages between windows. In *Asian conference on computer vision*, pages 290–306. Springer, 2014.

[226] Meelis Kull, Telmo Silva Filho, and Peter Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *Artificial Intelligence and Statistics*, pages 623–631. PMLR, 2017.

[227] Ponnambalam Kumaraswamy. A generalized probability density function for double-bounded random processes. *Journal of hydrology*, 46(1-2):79–88, 1980.

[228] Runsheng Xu, Hao Xiang, Xu Han, Xin Xia, Zonglin Meng, Chia-Ju Chen, Camila Correa-Jullian, and Jiaqi Ma. The opencda open-source ecosystem for cooperative driving automation research. *IEEE Transactions on Intelligent Vehicles*, pages 1–13, 2023.

[229] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, pages 180–191. PMLR, 2022.

[230] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2397–2406, 2022.

[231] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3d multi-object tracking: A baseline and new evaluation metrics. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10359–10366. IEEE, 2020.

[232] Zelin Zhao, Ze Wu, Yueqing Zhuang, Boxun Li, and Jiaya Jia. Tracking objects as pixel-wise distributions. In *European Conference on Computer Vision*, pages 76–94. Springer, 2022.

[233] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[234] Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, et al. Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21361–21370, 2022.

[235] Simon Suo, Sebastian Regalado, Sergio Casas, and Raquel Urtasun. Trafficsim: Learning to simulate realistic multi-agent behaviors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10400–10409, 2021.

[236] Sivabalan Manivasagam, Shenlong Wang, Kelvin Wong, Wenyuan Zeng, Mikita Sazanovich, Shuhan Tan, Bin Yang, Wei-Chiu Ma, and Raquel Urtasun. Lidarsim: Realistic lidar simulation by leveraging the real world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11167–11176, 2020.

[237] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[238] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3234–3243, 2016.

[239] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020.

[240] Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, et al. Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 21361–21370, 2022.

[241] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8555–8564, 2021.

[242] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021.

[243] Yue Wang, Alireza Fathi, Abhijit Kundu, David A Ross, Caroline Pantofaru, Tom Funkhouser, and Justin Solomon. Pillar-based object detection for autonomous driving. In *European Conference on Computer Vision*, pages 18–34. Springer, 2020.

[244] Lue Fan, Ziqi Pang, Tianyuan Zhang, Yu-Xiong Wang, Hang Zhao, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Embracing single stride 3d object detector with sparse transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8458–8468, 2022.

[245] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018.

[246] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4604–4612, 2020.

[247] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7077–7087, 2021.

[248] Yingwei Li, Adams Wei Yu, Tianjian Meng, Ben Caine, Jiquan Ngiam, Daiyi Peng, Junyang Shen, Yifeng Lu, Denny Zhou, Quoc V Le, et al. Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17182–17191, 2022.

[249] Jiaxun Cui, Hang Qiu, Dian Chen, Peter Stone, and Yuke Zhu. Coopernaut: end-to-end driving with cooperative perception for networked vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17252–17262, 2022.

[250] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021.

[251] Hang Qiu, Pohan Huang, Namo Asavisanu, Xiaochen Liu, Konstantinos Psounis, and Ramesh Govindan. Autocast: Scalable infrastructure-less cooperative perception for distributed collaborative driving. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '22, 2022.

[252] E Li, Shuaijun Wang, Chengyang Li, Dachuan Li, Xiangbin Wu, and Qi Hao. Sustech points: A portable 3d point cloud interactive annotation platform system. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1108–1115. IEEE, 2020.

[253] Charles K Chui, Guanrong Chen, et al. *Kalman filtering*. Springer, 2017.

[254] Valter Crescenzi, Giansalvatore Mecca, Paolo Merialdo, et al. Roadrunner: Towards automatic data extraction from large web sites. In *VLDB*, volume 1, pages 109–118, 2001.

[255] Philipp Bender, Julius Ziegler, and Christoph Stiller. Lanelets: Efficient map representation for autonomous driving. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 420–425. IEEE, 2014.

[256] Kun Yang, Dingkang Yang, Jingyu Zhang, Mingcheng Li, Yang Liu, Jing Liu, Hanqi Wang, Peng Sun, and Liang Song. Spatio-temporal domain awareness for multi-agent collaborative perception. *arXiv preprint arXiv:2307.13929*, 2023.

[257] Binglu Wang, Lei Zhang, Zhaozhong Wang, Yongqiang Zhao, and Tianfei Zhou. Core: Cooperative reconstruction for multi-agent perception. *arXiv preprint arXiv:2307.11514*, 2023.

[258] Yonglin Tian, Jiangong Wang, Yutong Wang, Chen Zhao, Fei Yao, and Xiao Wang. Federated vehicular transformers and their federations: Privacy-preserving computing and cooperation for autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2022.

[259] Jinlong Li, Runsheng Xu, Xinyu Liu, Baolu Li, Qin Zou, Jiaqi Ma, and Hongkai Yu. S2r-vit for multi-agent cooperative perception: Bridging the gap from simulation to reality. *arXiv preprint arXiv:2307.07935*, 2023.

[260] Xianghao Kong, Wentao Jiang, Jinrang Jia, Yifeng Shi, Runsheng Xu, and Si Liu. Dusa: Decoupled unsupervised sim2real adaptation for vehicle-to-everything collaborative perception. In *Proceedings of the 30th ACM International Conference on Multimedia*, 2023.

[261] Yue Hu, Shaoheng Fang, Zixing Lei, Yiqi Zhong, and Siheng Chen. Where2comm: Communication-efficient collaborative perception via spatial confidence maps. *Advances in Neural Information Processing Systems*, 35:4874–4886, 2022.

[262] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *IEEE Intelligent Transportation Systems Conference*, pages 2575–2582. IEEE, 2018.