

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Collaborative Perception Models for Enhanced Situational Awareness in Autonomous Vehicles

Bruno Filipe Torres Costa



Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Prof. Andry Maykol Gomes Pinto

June 30, 2025

Resumo

O impulso moderno para a automação total colocou a condução autónoma na vanguarda dos avanços tecnológicos, prometendo um transporte mais seguro e eficiente, com benefícios como maior segurança, redução do consumo de combustível e diminuição do congestionamento do tráfego. Entre as várias áreas científicas da investigação em condução autónoma, esta dissertação foca-se na perceção. A perceção é um domínio crítico que permite aos veículos autónomos interpretar o seu ambiente e tomar decisões informadas. O desenvolvimento de métodos de perceção de alta qualidade é vital para garantir a segurança humana e melhorar a fiabilidade dos sistemas autónomos.

Um desafio significativo nesta área reside nas limitações dos sistemas de perceção de agente único, que sofrem com oclusão de sensores, campos de visão limitados e obstruções ambientais. Estas limitações comprometem a segurança e a eficácia, particularmente em cenários de condução complexos e reais. A perceção colaborativa, que permite a múltiplos veículos autónomos partilhar e integrar dados sensoriais, apresenta-se como uma solução promissora ao melhorar a perceção do ambiente e a capacidade de tomada de decisão.

Esta dissertação explora o potencial da perceção colaborativa para melhorar os sistemas de condução autónoma, abordando as limitações fundamentais da perceção de um único agente, como a oclusão de sensores, os campos de visão limitados e as obstruções ambientais. Este trabalho apresenta o PerceptNet-V2X, uma nova estrutura colaborativa que incorpora uma estratégia de colaboração intermédia centrada numa representação Full Perspective View recentemente introduzida que codifica dados de sensores espaciais 3D num formato 2D denso. Esta representação, embora concebida para compatibilidade com arquiteturas de deteção 2D, preserva a estrutura geométrica crítica, permitindo assim uma perceção multiagente eficiente e eficaz. Para demonstrar as suas capacidades, a estrutura é instanciada utilizando detetores de objetos 2D de última geração, especificamente Real Time Detection Transformer e You Only Look Once, num ambiente multiagente.

A estrutura é avaliada em dois conjuntos de dados de referência: OPV2V, um conjunto de dados sintéticos de grande escala, e V2X-Real, um conjunto de dados do mundo real. Os resultados experimentais demonstram melhorias significativas no desempenho de deteção em relação às representações tradicionais, particularmente em condições desafiantes, como oclusões, tráfego intenso e ambientes urbanos dinâmicos. No conjunto de dados OPV2V, o método proposto atinge um AP@0,5 de 95,63%, e um F1@0,5 de 94,36%, representando ganhos substanciais de até 5,02 pontos percentuais em AP e 3,15 pontos percentuais em F1 em comparação com a representação tradicional Bird's-Eye-View. No conjunto de dados V2X-Real, atinge um mAP@0,3 de 61,55%, representando uma melhoria relativa superior a 16 pontos percentuais em comparação com as abordagens anteriores. Estes resultados realçam a eficácia prática e a forte capacidade de generalização do método proposto em cenários de perceção colaborativa sintéticos e reais. No geral, este trabalho contribui com uma estrutura de perceção colaborativa escalável, modular e eficaz, aproximando o estado da arte de uma condução autónoma multi-agente segura e fiável.

Abstract

The modern drive for full automation has placed autonomous driving at the forefront of technological advancements, promising safer, more efficient transportation with benefits such as enhanced security, reduced fuel consumption, and decreased traffic congestion. Among the various scientific areas in autonomous driving research, this dissertation focuses on perception. Perception is a critical domain that enables autonomous vehicles to interpret their surroundings and make informed decisions. The development of high-quality perception methods is vital for ensuring human safety and advancing the reliability of autonomous systems.

A significant challenge in this field lies within the limitations of single-agent perception systems, which suffer from sensor occlusion, restricted fields of view, and environmental obstructions. These limitations compromise safety and effectiveness, particularly in complex, real-world driving scenarios. Collaborative perception, which allows multiple autonomous vehicles to share and integrate sensory data, presents a promising solution by enhancing situational awareness and decision-making capabilities.

This dissertation explores the potential of collaborative perception to enhance autonomous driving systems by addressing the fundamental limitations of single-agent perception, such as sensor occlusion, limited fields of view, and environmental obstructions. This work presents PerceptNet-V2X, a novel collaborative framework, incorporating an intermediate collaboration strategy centered on a newly introduced Full Perspective View representation that encodes 3D spatial sensor data into a dense 2D format. This representation, while designed for compatibility with 2D detection architectures, preserves critical geometric structure, thereby enabling efficient and effective multi-agent perception. To demonstrate its capabilities, the framework is instantiated using state-of-the-art 2D object detectors, specifically Real Time Detection Transformer and You Only Look Once, within a multi-agent setting.

The framework is evaluated on two benchmark datasets: OPV2V, a large-scale synthetic dataset, and V2X-Real, a real-world dataset. Experimental results demonstrate significant improvements in detection performance over traditional representations, particularly in challenging conditions such as occlusions, dense traffic, and dynamic urban environments. On the OPV2V dataset, the proposed method achieves an AP@0.5 of 95.63%, and an F1@0.5 of 94.36%, representing substantial gains up to 5.02 percentage points in AP and 3.15 percentage points in F1 compared to the traditional Bird’s-Eye-View representation. On the V2X-Real dataset, it achieves a mAP@0.3 of 61.55% representing a relative improvement exceeding 16 percentage points compared to prior state-of-the-art approaches. These results highlight the practical effectiveness and strong generalization capability of the proposed method across both synthetic and real-world collaborative perception scenarios. Overall, this work contributes with a scalable, modular and effective collaborative perception framework that moves the field closer to safe and reliable multi-agent autonomous driving.

UN Sustainable Development Goals

The United Nations Sustainable Development Goals (SDGs) provide a global framework to achieve a better and more sustainable future for all. It includes 17 goals to address the world's most pressing challenges, including poverty, inequality, climate change, environmental degradation, peace, and justice. This work focuses on Autonomous Driving (AD), an area where the integration of Deep Learning (DL) techniques plays a crucial role in enhancing both efficiency and safety. These advancements align closely with multiple SDGs, demonstrating how technological innovation can contribute meaningfully to sustainable development. The specific SDGs mentioned have the following names:

SDG 3 Ensure healthy lives and promote well-being for all at all ages.

SDG 9 Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation.

SDG 11 Make cities and human settlements inclusive, safe, resilient and sustainable.

SDG	Target	Contribution	Performance Indicators and Metrics
3	3.6	Enhancement of road safety through improved perception in AD systems.	Reduction in vehicle collision rates; Decrease in road traffic fatalities; Improved detection of vulnerable road users.
9	9.1	Development of sustainable and resilient transport infrastructure enabled by collaborative perception systems.	Increase in deployment of smart mobility infrastructures; Improved reliability of perception under diverse environmental conditions.
	9.5	Advancement of technological capabilities in intelligent transportation systems.	Number of novel architectures developed; Performance benchmarks against state-of-the-art models.
11	11.2	Improved perception accuracy for urban mobility applications.	Enhanced safety and efficiency in urban traffic scenarios; Detection accuracy for urban-specific obstacles (e.g., pedestrians).

Agradecimentos

Ao meu orientador, Prof. Andry Pinto, expresso o meu profundo agradecimento pela sua valiosa mentoria ao longo desta dissertação. A sua experiência e conhecimento foram determinantes para a concretização deste trabalho.

Ao meu co-orientador, Celso Pereira, que tive o enorme prazer de conhecer, agradeço sinceramente a constante disponibilidade para discutir ideias e encontrar soluções. A sua abertura e colaboração, em qualquer momento, foram essenciais para o desenvolvimento e sucesso deste projecto.

À malta do MOTUS, agradeço pelo ambiente acolhedor e pelo espírito de entreajuda que sempre demonstraram. Criaram as condições ideais para um trabalho produtivo e motivador, contribuindo de forma significativa para os resultados alcançados.

Deixo também um agradecimento especial aos meus amigos, que me acompanharam ao longo desta longa caminhada e com quem partilhei momentos e memórias que levo para a vida.

Por fim, e não menos importante, à minha família. Ao meu pai e à minha mãe, pelo apoio incondicional, sem o qual nada disto teria sido possível. Aos meus irmãos, agradeço por me terem ensinado, muitas vezes sem saberem, a cultivar a paciência, uma qualidade que se revelou essencial para a conclusão desta dissertação.

Bruno Filipe Torres Costa

Acknowledgements

This work was supported by European Structural and Investment Funds in the FEDER component, through the Operational Competitiveness and Internationalization Programme, ATLAS Project [Project n014632; Funding Reference: POCI-FEDER-014632]. Also, by the TALOS project funded by European Commission under the Horizon Europe framework with no 101119744.

Bruno Filipe Torres Costa

“There are many paths to the top of the mountain, but the view is always the same.”

Chinese Proverb

Contents

Resumo	i
Abstract	ii
Agradecimentos	iv
Acknowledgements	v
1 Introduction	1
1.1 Context	1
1.2 Motivation	2
1.3 Objectives	3
1.4 Contributions	3
1.5 Document Structure	4
2 State-of-the-Art	5
2.1 Introduction	5
2.2 Object Detection	6
2.3 Multi-Modality	10
2.4 Collaborative Perception Systems	13
2.5 Spatial Representations	19
2.6 Critical Review	22
3 Deep Learning Architecture for Collaborative Perception in Autonomous Driving	26
3.1 Introduction	26
3.2 Data Sharing	28
3.3 Data Representation	30
3.4 Augmentation Techniques	32
3.5 Network Components	34
4 Results	42
4.1 Introduction	42
4.2 System Specifications	42
4.3 Evaluation Metrics	43
4.4 Model Evaluation in synthetic environments	45
4.5 Model Evaluation in real-world environments	55
5 Conclusions and Future Work	63

CONTENTS

viii

References

65

A Additional Metrics

69

List of Figures

2.1	Evolution of CNNs for Object Detection. LeNet-5 [9], AlexNet [10], GoogleNet [11], VGG [12], ResNet [13].	6
2.2	Object Detector Taxonomy. R-CNN [14], Fast R-CNN [15], Faster R-CNN [16], YOLO [17], RT-DETR [8].	7
2.3	Type and positioning of sensors in an automated vehicle to enable the vehicle perception of its surrounding - <i>The Perception Belt</i> . Reprinted from [27].	10
2.4	Representation of each data fusion technique: Early, Late and Intermediate. . . .	12
2.5	Example of collaborative perception system collaboration scheme. (a) Single agent perception framework. (b) Early collaboration perception framework. (c) Intermediate collaboration perception framework. (d) Late collaboration perception framework. Redesigned from [32].	14
2.6	Collaboration modules in the collaborative perception system. Adapted from [32].	15
2.7	Evolution of collaborative perception architectures in AD. Copper [33], Coop3D [34], F-Cooper [35], V2VNet [36], AttFusion [37], V2X-VIT [38], CoAlign [39], MKD-Cooper [40], CoMamba [41], CollaMamba [42], COPERA [43].	16
2.8	Types of spatial representations in a perception system	19
2.9	BEV representation of a 3D point cloud.	20
2.10	RV representation of a 3D point cloud.	21
2.11	Voxel-based representation of a 3D point cloud.	21
2.12	Point-based representation of a 3D point cloud.	22
3.1	Overview of the PerceptNet-V2X architecture.	27
3.2	FPV data sharing between nodes for the PerceptNet-V2X architecture.	28
3.3	FPV projection of a V2X-Real dataset point cloud.	30
3.4	Impact of augmentation techniques visualized on a BEV maximum height projection sample of an OPV2V dataset point cloud.	32
3.5	Effect of local pixel expansion on vertical FPV projections from the V2X-Real dataset. Each pair shows the original and augmented views for front, back, left, and right directions.	34
3.6	Full Perspective View (FPV) Block common to all nodes in the PerceptNet-V2X architecture.	35
3.7	Sparse Encoder-Decoder block common to all FPV Blocks in the PerceptNet-V2X architecture.	36
3.8	BEV Reprojection Block common to all nodes in the PerceptNet-V2X architecture.	38
3.9	BEV Fusion Block of the PerceptNet-V2X architecture.	39
4.1	Sensor setup for each CAV in the OPV2V dataset [37].	45

4.2	Training results in the OPV2V dataset.	48
4.3	Impact of localization errors in the Default Carla Towns.	51
4.4	Impact of localization errors in the Culver City Digital Town.	52
4.5	OPV2V Default Carla Towns dataset qualitative results. Green boxes indicate ground-truth bounding boxes, while red boxes represent car model predictions. The ego vehicle is identified with a blue bounding box.	53
4.6	OPV2V Digital Culver City dataset qualitative results. Green boxes indicate ground-truth bounding boxes, while red boxes represent car model predictions. The ego vehicle is identified with a blue bounding box.	54
4.7	V2X-Real smart intersection and data acquisition example [52].	55
4.8	Number of annotations per class in log scale in V2X-Real Dataset [52].	57
4.9	Training results in the V2X-Real dataset.	59
4.10	V2X-Real dataset qualitative results. Green boxes represent ground-truth bounding boxes, while red, purple, and orange boxes indicate predicted cars, pedestrians, and trucks, respectively. The ego vehicle is identified with a blue bounding box.	61

List of Tables

2.1	State-of-the-Art Collaborative Perception Methods.	23
2.2	Comparison of state-of-the-art 3D object detection models on the OPV2V dataset.	24
2.3	State-of-the-Art 3D Object Detection results on the V2X-Real Dataset under the VC setting.	25
4.1	System Specifications.	43
4.2	Sensor specifications for each CAV in the OPV2V dataset [37].	45
4.3	Scenario types, distribution and assigned difficulty levels in the OPV2V dataset [37].	46
4.4	Hyperparameter Specifications for the OPV2V dataset.	47
4.5	Test results on the OPV2V dataset.	49
4.6	Comparison of PerceptNet-V2X with state-of-the-art object detection models in the OPV2V dataset.	52
4.7	Key Sensor Specifications in V2X-Real Dataset [52].	56
4.8	Hyperparameter Specifications for the V2X-Real dataset.	58
4.9	Test Results in the V2X-Real Dataset under the Vehicle Centric (VC) setting.	59
4.10	Comparison of PerceptNet-V2X with state-of-the-art object detection models in the V2X-Real dataset.	60

Abbreviations and Symbols

2D	Two Dimensional
3D	Three Dimensional
AD	Autonomous Driving
AI	Artificial Intelligence
AP	Average Precision
AV	Autonomous Vehicle
BEV	Bird's-Eye-View
CAF	Collaborative Attention Fusion
CAV	Connected and Automated Vehicle
CNNs	Convolutional Neural Networks
CPS	Collaborative Perception Space
CSAF	Cross-View Spatial Attention Fusion
DoF	Degrees of Freedom
DL	Deep Learning
FOV	Field of View
FPV	Full Perspective View
GIoU	Generalized Intersection over Union
GNN	Graph Neural Networks
GPS	Global Positioning System
GPU	Graphics Processing Unit
I2I	Infrastructure to Infrastructure
IC	Infrastructure Centric
IMU	Inertial Measurement Unit
IoU	Intersection over Union

LiDAR	Light Detection and Ranging
mAP	mean Average Precision
ML	Machine Learning
NMS	Non-Maximum Suppression
RADAR	Radio Detection and Ranging
RAM	Random Access Memory
ReLU	Rectified Linear Unit
RPN	Region Proposal Network
RT-DETR	Real Time Detection Transformer
RV	Range View
SDGs	United Nations Sustainable Development Goals
SFF	Spatial Feature Fusion
SSD	Single-Shot Detector
SSM	State Space Model
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
VC	Vehicle Centric
VFE	Voxel Feature Encoding
VFF	Voxel Feature Fusion
ViT	Vision Transformer
VRAM	Video Random Access Memory
YOLO	You Only Look Once

Chapter 1

Introduction

1.1 Context

Recently, Autonomous Driving (AD) has emerged as a prominent and multifaceted topic within the field of technology. The concept encompasses a vast range of subjects, from the latest advances in engineering to the social, cultural, economic, legal, and even philosophical implications of this technology [1]. In fact, the advancement of Autonomous Vehicles (AVs), whether partially or fully autonomous, has the potential to usher in a multitude of benefits.

It is anticipated that road safety will improve as the responsibility for vehicle control shifts from the driver to the vehicle side. The World Health Organization (WHO) indicates that approximately 1.19 million deaths are attributed each year to road traffic accidents [2]. Additionally, numerous studies on road accident causation have found that human error is responsible for over 90% of road accidents [3]. According to the norms defined by the Society of Automotive Engineers (SAE) [4], most currently available commercial solutions fall within levels 2 and 3 of vehicle automation [5]. Hence, fully AVs have yet to match the driving proficiency of an attentive human driver. However, they hold significant potential to improve road safety through the integration of monitoring, alerting, and driver assistance technologies. In line with the European Commission's "Vision Zero" program [6], which aims to eliminate road fatalities by 2050, significant investments have been made in technologies that enhance vehicle autonomy and road safety.

From another perspective, a significant benefit of AVs lies in improving traffic flow. As highlighted in [3], analyzes based on the available traffic flow models has concluded that the existing transportation infrastructure can be utilized in a more efficient manner. This would result in a reduction of traffic congestion and a minimization of time spent on the road by motorists. Furthermore, AVs contribute to lower fuel consumption and provide enhanced comfort and convenience for passengers.

In the context of AD, the ability of each agent to self-localize and accurately interpret its surroundings is crucial for ensuring overall vehicle safety. A robust and reliable perception system is, therefore, indispensable.

Modern perception systems typically rely on sensory data from various sources, including ultrasonic sensors, cameras, Light Detection and Ranging (LiDAR), and Radio Detection and Ranging (RADAR). The ego vehicle, referring to the autonomous vehicle in motion that is actively perceiving its environment, can employ different combinations of these sensors or rely on a single type. This perception process is further enhanced by fusing information from multiple sensors using advanced sensor fusion techniques, which improve the system's accuracy and robustness in interpreting the environment.

1.2 Motivation

AVs have shown great promise in improving road safety and traffic efficiency. However, existing systems still face critical challenges in perception and situational awareness. Single-agent perception systems, even when equipped with advanced sensors and fusion algorithms, are limited by occlusions, restricted sensor range, and variability in real-world environments. These limitations compromise their reliability in complex scenarios, such as dense urban traffic or intersections with obstructed views.

To address these challenges, a shift toward collaborative perception systems is necessary. Through the use of Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication, collaborative perception enables vehicles to share and aggregate sensor data, thereby extending situational awareness beyond the capabilities of any individual vehicle. This approach enhances safety and efficiency by mitigating blind spots, improving early hazard detection, and enabling more coordinated traffic flow. For instance, shared data can provide early warnings of unseen obstacles or enhance collaboration in cluttered environments. Complementing this shift is the transformative potential of Machine Learning (ML), especially DL, which has enabled significant progress in object detection, semantic segmentation, and sensor fusion. These advancements suggest a promising path for building intelligent and adaptable collaborative perception systems.

Nevertheless, robust sensor fusion remains a major challenge, especially due to the sparsity of data representation in real-world settings. This limitation affects perception accuracy and, by extension, compromises road safety. The complexity of fusing, often sparse representations, both within a vehicle and across multiple agents, demands innovative representation and fusion strategies.

This dissertation addresses these challenges by introducing an improved data representation specifically designed to handle sparsity effectively. Through the use of collaborative perception frameworks and DL capabilities, the proposed approach aims to overcome key barriers in multi-agent sensor fusion. Ultimately, this research contributes to advancing safer, more intelligent, and reliable transportation systems.

1.3 Objectives

This dissertation proposes the development of a collaborative perception system, aimed at validating the hypothesis that integrating sensor data from multiple Connected and Automated Vehicles (CAVs) can significantly enhance the robustness and reliability of AD. While the framework is designed to address data sparsity in Bird's-Eye-View (BEV) representations, its capabilities are demonstrated using state-of-the-art Two Dimensional (2D) object detectors, such as You Only Look Once (YOLO) [7] and Real Time Detection Transformer (RT-DETR) [8], which operate on collaboratively fused representations derived from multiple agents. The methodology incorporates advanced DL models and intermediate feature fusion strategies to enable effective multi-agent collaboration and maintain high detection performance across diverse and dynamic environments. Evaluation is conducted using publicly available AD datasets from both synthetic and real-world driving scenarios. These datasets provide a comprehensive foundation for training, validating, and testing the proposed system under realistic and challenging conditions.

The specific objectives of this dissertation are as follows:

- To improve collaborative perception by addressing challenges such as data sparsity in BEV representations, thereby enhancing detection accuracy through improved data representation.
- To design and implement an intermediate collaborative architecture for improved perception in complex driving scenarios.
- To analyze relevant open-source datasets for collaborative perception in AD systems, with a focus on both real-world and synthetic driving scenarios.
- To train and evaluate the proposed architectures, benchmarking their performance on collaborative perception datasets and comparing the results with current state-of-the-art methods.

1.4 Contributions

This dissertation was conducted at the Faculdade de Engenharia da Universidade do Porto, with the primary objective of advancing research in AD. The work led to the preparation of a state-of-the-art research paper, currently under review, entitled "COPERA: Collaborative Perception for Real-Time Object Detection in Autonomous Driving," submitted to the IEEE Transactions on Intelligent Transportation Systems Journal. Additional works produced as part of this research, all currently under review, include "FuseNet-V2X: A Collaborative Perception Architecture for Autonomous Driving Scenarios," submitted to ROBOT 2025, and "PerceptNet-V2X: Perception Network for Vehicle to Everything Scenarios in Autonomous Driving," submitted to IEEE Access 2025.

1.5 Document Structure

This dissertation is structured into four main chapters. Chapter 2 introduces the fundamental concepts and provides a comprehensive review of the state-of-the-art in object detection, multi-modality, collaborative perception systems and spatial representations. It also includes a critical analysis of the existing research landscape, highlighting the challenges and gaps that this dissertation aims to address. Chapter 3 describes the proposed methodology and outlines the technical solutions developed for designing a robust collaborative perception system for AD. Chapter 4 presents and analyzes the experimental results, evaluating the performance of the proposed solutions in comparison with current state-of-the-art architectures. Finally, Chapter 5 summarizes the key conclusions drawn from the research, discusses the broader implications of the findings, and outlines directions for future work.

Chapter 2

State-of-the-Art

This chapter provides a comprehensive review of the current literature and technological advancements in AD, focusing on key components such as perception systems, collaborative perception models, and foundational technologies like advanced DL architectures and sensor fusion techniques. The chapter starts with Section 2.1, which introduces essential concepts that underpin AD technologies, providing clarity on overarching principles applicable across various methods and systems. Section 2.2 discusses relevant Two Dimensional (2D) and Three Dimensional (3D) object detection models, highlighting architectural advancements, their strengths, and their applications in AD. Section 2.3 discusses multi-modal perception systems, which integrate data from multiple sensor modalities to enhance perception and understanding. Section 2.4 explores collaborative perception systems, analyzing their potential advantages, challenges, and the innovations they bring to AD. In Section 2.5, the types of spatial representations used in AD are discussed, including both 2D and 3D approaches, with a comparison of their respective benefits and drawbacks. The chapter concludes with Section 2.6, which provides a critical analysis of the current state of collaborative AD technologies, outlining the remaining challenges and areas requiring further research and development.

2.1 Introduction

Recent advancements in perception systems, communication networks, and Artificial Intelligence (AI) have driven significant improvements in the capabilities of AV in areas such as perception, navigation, and decision-making. Despite these advancements, individual vehicle perception systems continue to face inherent limitations. Challenges such as restricted sensor range, occlusions caused by obstacles or other vehicles, and adverse environmental conditions such as rain, fog, or reduced lighting conditions can significantly compromise the performance of AV, particularly in complex and dynamic scenarios.

To address these limitations, collaborative perception has emerged as a transformative approach. In collaborative perception, multiple vehicles and infrastructure nodes work together,

sharing sensor data and processed information to construct a comprehensive and unified understanding of their surroundings. This collective pooling of data enhances situational awareness, enabling autonomous systems to detect and respond to occluded objects, navigate congested environments, and make safer decisions in real time.

This Chapter examines the state-of-the-art perception systems for AV, with a focus on collaborative approaches. It also explores key technologies, including data fusion methods, advanced ML models, and the integration of multi-modal sensors, which are instrumental in advancing the perception capabilities of AV. The analysis highlights both the potential and the challenges of these technologies, providing a foundation for understanding their role in shaping the future of AD.

2.2 Object Detection

Object detection is a computer vision and perception task that involves identifying and localizing objects of interest within a 2D or 3D representation of the environment. It is a foundational capability in many fields, including AD, where systems must detect and classify objects such as vehicles, pedestrians, traffic signs, and obstacles in real-time. Accurate object detection is critical for AV to interpret and interact with their surroundings safely and effectively.

At the core of object detection are Convolutional Neural Networks (CNNs), which have revolutionized the field with their ability to extract hierarchical features from images. This revolutionary engineering tool plays an indispensable role in enabling AD to achieve reliable perception in diverse and dynamic environments.

Historical Context

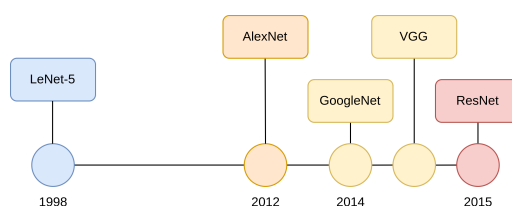


Figure 2.1: Evolution of CNNs for Object Detection. LeNet-5 [9], AlexNet [10], GoogleNet [11], VGG [12], ResNet [13].

Over the years, as illustrated in 2.1, CNNs architectures have undergone significant innovations to enhance efficiency, accuracy, and scalability for object detection tasks. One of the pioneering CNNs architectures introduced by *Lecun et al.* (1998) was LeNet-5 [9]. This type network was designed for handwritten digit recognition and introduced fundamental building blocks such as convolutional layers, pooling layers, and fully connected layers. Moreover *Krizhevsky et al.* (2012) presented a groundbreaking architecture called AlexNet [10] that popularized DL. AlexNet

utilized Rectified Linear Unit (ReLU) activations, dropout for regularization and Graphics Processing Unit (GPU) acceleration for faster training, winning the ImageNet Large Scale Visual Recognition Challenge and ushering in the era of deep neural networks. GoogleNet [11], also known as Inception v1, introduced by *Christian Szegedy et al.* (2014) is known for its use of Inception modules, which allow parallel processing at multiple scales. These modules utilize various filter sizes within a single layer, concatenating their outputs to create a richer feature representation. GoogleNet achieved significant success in the ImageNet Large-Scale Visual Recognition Challenge in 2014. VGG [12] developed by *Karen Simonyan et al.* (2014) is known for its simplicity and depth, VGG employed 3×3 convolutional kernels and introduced models with 16 or 19 layers. It demonstrated that deeper networks with scriptsizeer filters could capture complex spatial patterns effectively. Ultimately, ResNet [13] addressed the vanishing gradient problem by introducing residual learning through skip connections. Authored by *He et al.* (2015) this revolutionary network for the time enabled the development of extremely deep networks like ResNet-50 and ResNet-101 without degradation in performance.

Object Detector Taxonomy

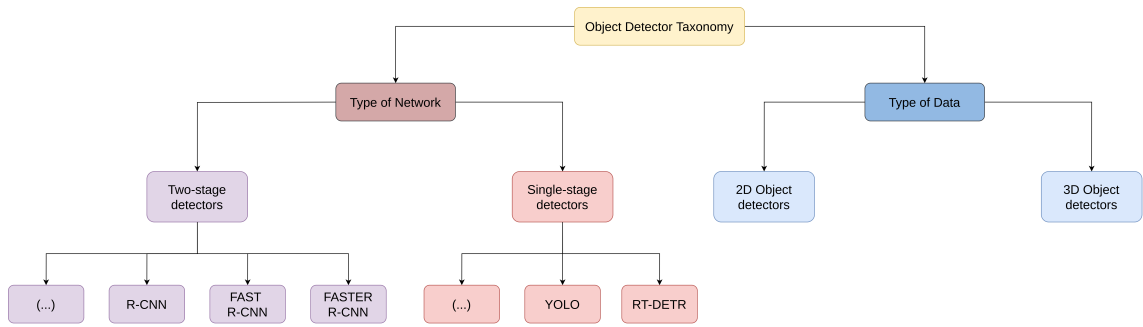


Figure 2.2: Object Detector Taxonomy. R-CNN [14], Fast R-CNN [15], Faster R-CNN [16], YOLO [17], RT-DETR [8].

Figure 2.2 represents the current state of the object detection field, which can be categorized based on two primary dimensions: the type of network and the type of data utilized.

Regarding the type of network, object detectors are generally divided into two main classes: two-stage detectors and single-stage detectors. Two-stage detectors [14][15][16] operate by first generating region proposals that likely contain objects and then by classifying the objects in a second step. This two-step approach allows for high precision, making two-stage models particularly effective for tasks that demand accuracy in detecting objects in cluttered scenes or complex backgrounds. However, the computational cost of processing each region proposal independently makes these methods slower and less suited for real-time applications such as AD.

In contrast, single-stage detectors [17] [7] [18] [8] eliminate the need for a separate region proposal phase. These models predict object locations and classifications directly in a single pass through the neural network. As a result, single-stage detectors are significantly faster and more

efficient, making them ideal for real-time applications. However, this speed advantage can sometimes come at the expense of reduced accuracy, particularly for detecting scriptsize or overlapping objects in dense scenes.

In addition to the type of network, object detectors can also be categorized based on the type of data they process. These categories include 2D object detectors and 3D object detectors. Each category of methods, whether based on network type or data type, is designed to address specific challenges, balancing trade-offs between accuracy, speed, and computational complexity. The taxonomy outlined in Figure 2.2 highlights the diversity of approaches within the field of object detection.

2D Object Detection

2D object detectors operate on 2D data representation, where the focus is on identifying and localizing objects within the plane. These detectors are typically built using CNNs and excel in applications where only visual data from a single image is available. Despite their effectiveness in many scenarios, they are inherently limited in understanding depth and spatial relationships between objects. Current state-of-the-art object detectors in the context of AD rely on single-stage detectors [7] [18] [8].

The YOLO family [19], first introduced by *Redmon et al.* (2015) [17], remains a state-of-the-art single-stage detector. The YOLO architecture is built around three key components. First, the backbone is responsible for extracting features from the raw input data using convolutional layers, converting it into multi-scale feature maps. Next, the neck acts as an intermediary processing layer, consolidating and refining features across different scales to improve representation. Finally, the head generates the final outputs for object detection, including both localization and classification, based on the enhanced feature maps. Building on this established architecture, YOLOV12 [18], presented by *Ultralytics* (2025), extends and enhances the foundation laid by the previous versions [19], introducing architectural innovations and parameter optimizations to achieve superior detection performance.

Introduced by *Yian Zhao et al.* (2023) RT-DETR [8] is also a state-of-the-art real-time 2D object detector. Unlike traditional CNNs based approaches, RT-DETR integrates the strengths of the attention mechanisms [20] to process image data, capturing long-range dependencies and complex spatial relationships. This transformative approach provides a substantial boost in detection accuracy while maintaining real-time performance, making it particularly well-suited for AD scenarios. By taking advantage of detection transformers [21], RT-DETR sets a new solution for robust and reliable 2D object detection in dynamic environments. RT-DETR addresses the limitations of traditional detectors such as YOLO, that typically require Non-Maximum Suppression (NMS) for post-processing, by integrating end-to-end detection capabilities with real-time performance. Central to its architecture is an efficient hybrid encoder, which accelerates the processing of multi-scale features by decoupling intra-scale interactions and cross-scale fusion. Additionally, this architecture incorporates an uncertainty-minimal query selection mechanism, enabling the decoder to start with high-quality initial queries, thereby increasing detection accuracy.

3D Object Detection

On the other hand, 3D object detectors extend the detection process into 3D space, operating directly on the spatial dimensions of the objects in the scene. This category includes approaches that analyze the geometry and structure of objects in 3D. These detectors are crucial in applications that require a detailed understanding of the spatial environment, such as AD, robotics, and augmented reality systems. While 3D object detectors provide richer and more accurate information about object positions and orientations, their reliance on high-dimensional data results in increased computational demands, presenting challenges for deployment in resource-constrained environments. Concerning AD, numerous methods have been developed for detecting 3D objects mostly from point clouds, which can broadly be divided into two main categories: point-based methods [22] [23] where it operates directly on the point cloud as a set of N points (x, y, z) , optionally with features like intensity or color, and voxel-based methods [24] [25] [26] where the 3D space is divided into regular grids (voxels), and features from points within each voxel are aggregated.

Regarding point-based methods, PointNet [22] introduced by *Charles et al.* (2016) presents a novel type of neural network that directly consumes point clouds and respects the permutation invariance of points in the input. This model provides a unified architecture for applications ranging from object classification, part segmentation, to scene semantic parsing. Moreover, introduced by *Shi et al.* (2018) PointRCNN [23] also proposes 3D object detection from raw point cloud. The whole framework is composed of two stages: stage one for the bottom-up 3D proposal generation and stage two for refining proposals in the canonical coordinates to obtain the final detection results.

On the other hand, the voxel-based methodology separates the point clouds into a predetermined number of voxels and employs DL technologies for voxel feature extraction. The research made by *Yin Zhou et al.* (2017) culminated in the development of VoxelNet [24]. This architecture presents a generic 3D detection network that unifies feature extraction and bounding box prediction into a single stage, end-to-end trainable deep network. Specifically, VoxelNet divides a point cloud into equally spaced 3D voxels and transforms a group of points within each voxel into a unified feature representation through the newly introduced Voxel Feature Encoding (VFE) layer. In this way, the point cloud is encoded as a descriptive volumetric representation, which is then connected to a Region Proposal Network (RPN) to generate detections. Furthermore, presented by *Yan et al.* (2018) SECOND [25] investigates an improved sparse convolution method for voxel-based 3D convolutional networks, which significantly increases the speed of both training and inference. This method also introduces a new form of angle loss regression to improve the orientation estimation performance and a new data augmentation approach that can enhance the convergence speed and performance. Finally, PointPillars [26] presented by *Lang et al.* (2018), introduces a novel encoder which utilizes PointNets to learn a representation of point clouds organized in vertical columns called "Pillars". This framework transforms point clouds into a sparse pseudo-image, enabling 2D networks for feature learning using a Single-Shot Detector (SSD) head

for bounding box regression, improving computational speed significantly.

2.3 Multi-Modality

The environments in which autonomous cars operate are highly complex. Similar to humans, autonomous agents can benefit from extended and multi-modal perception by integrating data from multiple sensors, such as cameras, LiDAR and RADAR. To achieve comprehensive environmental awareness, most automobile manufacturers today equip their vehicles with an extensive array of sensors, commonly referred to as the *Perception Belt*, as illustrated in Figure 2.3.

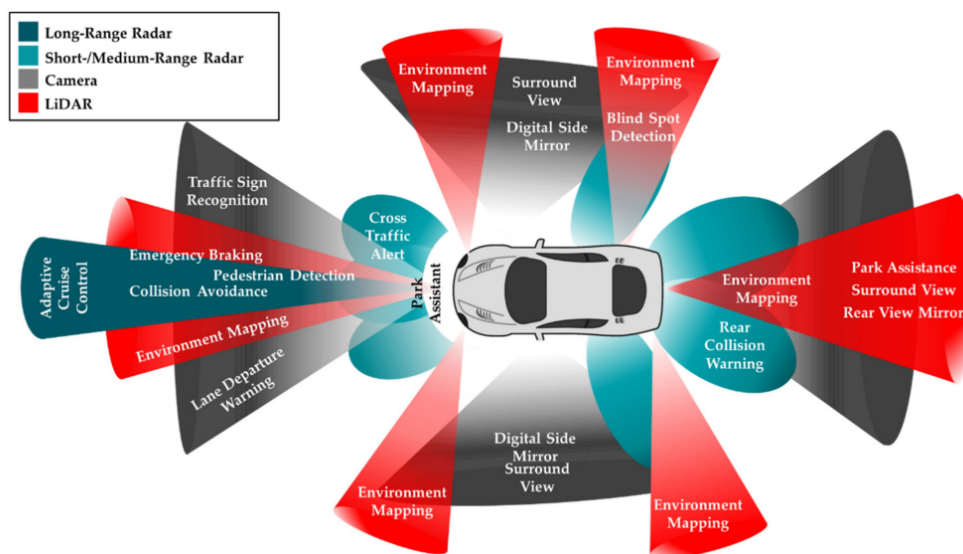


Figure 2.3: Type and positioning of sensors in an automated vehicle to enable the vehicle perception of its surrounding - *The Perception Belt*. Reprinted from [27].

In the automotive sector, cameras are one of the most widely adopted technologies for environmental perception in AV. They operate by detecting light from the surroundings using a photosensitive surface (image plane) through a camera lens, producing clear, high-resolution images [27]. This type of technology is relatively inexpensive, versatile and capable of identifying both static and dynamic objects in their Field of View (FOV). With appropriate software as previously explained in 2.2, they can recognize road signs, traffic lights, lane markings, and obstacles, making them indispensable in various applications that require texture acquisition. Camera systems in AV can utilize monocular or stereo configurations. Monocular cameras use a single lens to capture a series of 2D images, but they lack native depth information. Advanced monocular cameras, employing algorithms and dual-pixel autofocus hardware, can infer depth under specific conditions. Stereo cameras, or binocular systems, consist of two lenses separated by a baseline, mimicking the depth perception mechanism of human vision. By calculating the disparity between two slightly different images, stereo cameras generate depth maps, enabling 3D spatial awareness. Despite their capabilities, cameras have limitations. They rely on sufficient ambient light and are

less effective in adverse weather conditions like fog, rain, or snow. Their inability to inherently measure distance with high precision makes them dependent on sensor fusion with technologies like LiDAR or RADAR for depth information. Moreover, LiDAR sensors use laser pulses to measure distances by calculating the time taken for emitted infrared beams to reflect off objects and return to the sensor. These sensors create a detailed 3D representation of the environment in the form of a point cloud. This type of infrared-based sensors offer high accuracy and a broad FOV, making it a cornerstone technology for AV. Modern LiDAR sensors can generate up to 200,000 points per second, covering 360° horizontal and a vertical FOV up to 30°. They operate at wavelengths typically around 905 nm, offering better performance in adverse weather conditions than older 1550 nm systems [27]. While LiDAR excels in spatial mapping, it lacks the ability to provide color or texture information, necessitating integration with camera data. Ultimately, RADAR operates by emitting electromagnetic waves and analyzing the reflected signals to detect objects and measure their velocity, range, and angle. By taking advantage of the Doppler effect, RADAR excels in determining the relative speed of moving objects, making it particularly valuable for adaptive cruise control and collision avoidance. This type of sensor functions reliably in all lighting and most weather conditions, including fog, rain, and snow. Additionally, it can operate day and night, unlike cameras, which require adequate environmental conditions. However, RADAR coarse resolution limits its ability to identify and classify objects accurately, making it less suitable for object detection tasks. For instance, RADAR may struggle to differentiate between a stationary object like a metal guardrail and a dynamic one such as a moving vehicle.

In summary, while the integration of multiple sensor modalities enhances perception in AV, LiDAR stands out as a critical component due to its high-precision spatial mapping and robustness in varying environmental conditions. Unlike cameras, which are sensitive to lighting, and RADAR, which lacks fine resolution, LiDAR delivers detailed and accurate 3D representations essential for reliable navigation and obstacle detection. Its ability to generate rich geometric information makes it indispensable for achieving the level of environmental understanding required for AD.

Sensor Fusion

In AD, sensor fusion techniques integrates inputs from various sensors to create a more detailed and accurate representation of the environment. This approach improves object detection by minimizing uncertainties and making the system more reliable. A well-implemented sensor fusion system can handle failures from individual sensors, enhancing safety and efficiency in AD. There are three main strategies for sensor fusion in object detection: early fusion, late fusion, and intermediate fusion, as illustrated in Figure 2.4.

Early fusion involves combining sensor data at a raw stage before significant processing occurs. For instance, LiDAR point cloud data can be aligned with a camera's image plane to merge LiDAR depth information with the camera's visual details. This approach is advantageous because it has low computational requirements. However, it is highly dependent on accurate calibration between sensors to ensure spatial and temporal alignment [27]. Any miscalibration can lead to

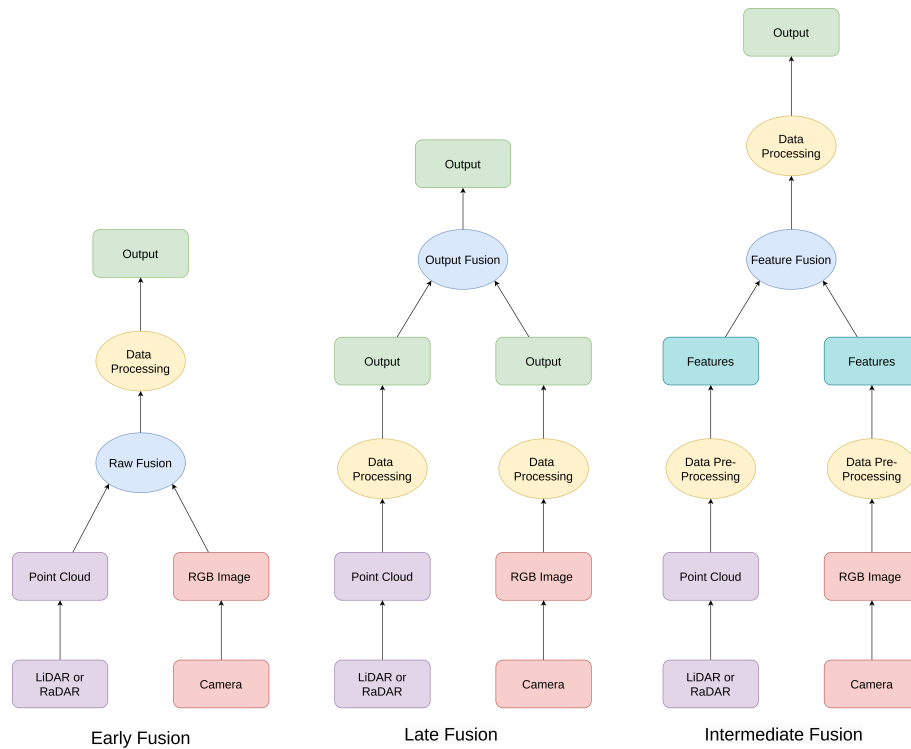


Figure 2.4: Representation of each data fusion technique: Early, Late and Intermediate.

errors in the fused data. Early fusion is ideal for scenarios requiring quick processing but can be less reliable due to its sensitivity to calibration issues. On the other hand, late fusion combines sensor data after it has been individually processed. For example, a camera might detect objects using vision algorithms, while a RADAR system measures their speed, and the results are then merged. This method allows for independent processing of each sensor using specialized techniques, making it easier to modularize the system. However, it can be computationally intensive, as each sensor's data requires separate processing. Additionally, late fusion may lose intermediate details, like object contours or depth gradients, that could enhance detection accuracy. Its reliance on pre-processed outputs makes it less adaptable to fast-changing environments. Ultimately, intermediate fusion, also called feature-level fusion, combines data after some initial processing but before final decision-making. For example, feature maps from a camera and depth information from LiDAR can be merged to take advantage of the strengths of both. This method provides a balance between efficiency and the ability to utilize complementary sensor features. It is highly adaptable and can be customized to suit various system architectures. Nevertheless, implementing intermediate fusion can be challenging, as it requires carefully designed algorithms and precise synchronization between sensors.

2.4 Collaborative Perception Systems

Although individual perception systems have achieved remarkable advancements with the integration of DL techniques [28] [29] [30] [31], they are inherently constrained by several limitations. These include challenges such as occlusions, restricted sensor resolutions, and limited FOV, which can undermine their ability to provide a complete and accurate understanding of the surrounding environment.

Collaborative perception systems extend the capabilities of individual sensors by enabling V2V and V2I communication to share and fuse perception data. This approach, often referred to as Vehicle-to-Everything (V2X) perception, allows AV to gain access to a richer, more comprehensive view of their surroundings. Collaborative perception is particularly effective in mitigating the limitations of occlusions and FOV constraints, as multiple perspectives from distributed sources contribute to a shared understanding of the environment.

Collaboration Scheme

A typical collaborative AD perception system is composed of several key stages: input, base network, feature, perception head, and output. Based on the data sharing and collaboration stage, collaborative perception schemes can be broadly categorized into early collaboration, intermediate collaboration and late collaboration [32], as shown in Figure 2.5 (b-d). In order to facilitate understanding the relationship between individual perception and the three collaborative schemes, individual perception framework is also illustrated in Figure 2.5 (a).

Early collaboration involves raw data fusion at the input stage of the network, commonly referred to as data-level fusion or low-level fusion, as illustrated in Figure 2.5 (b). In an AD context, the ego vehicle receives raw sensor data from other agents, applies necessary transformations, and subsequently aggregates the transformed data onboard. This approach benefits from the rich and comprehensive nature of raw data, which provides detailed descriptions of the agents and their surroundings. As a result, early collaboration effectively addresses challenges such as occlusion and long-range detection, significantly enhancing individual perception and overall system performance. However, the reliance on high-bandwidth data transmission poses a critical limitation. The substantial communication overhead required for transmitting raw data makes it difficult to achieve efficient real-time edge computing, which is vital in dynamic and latency-sensitive environments like AD. Future advancements in data compression, network optimization, and distributed processing may help mitigate these challenges, enabling early collaboration to become a more viable option in edge-computing scenarios.

To address the high bandwidth demands of early collaboration, some studies propose intermediate collaborative perception methods as a compromise, effectively balancing the performance-bandwidth trade-off. In intermediate collaboration, as highlighted in Figure 2.5 (c), other agents typically transmit deep semantic features to the ego vehicle, which then fuses these features to generate the final predictions. This approach has gained popularity due to its flexibility, making

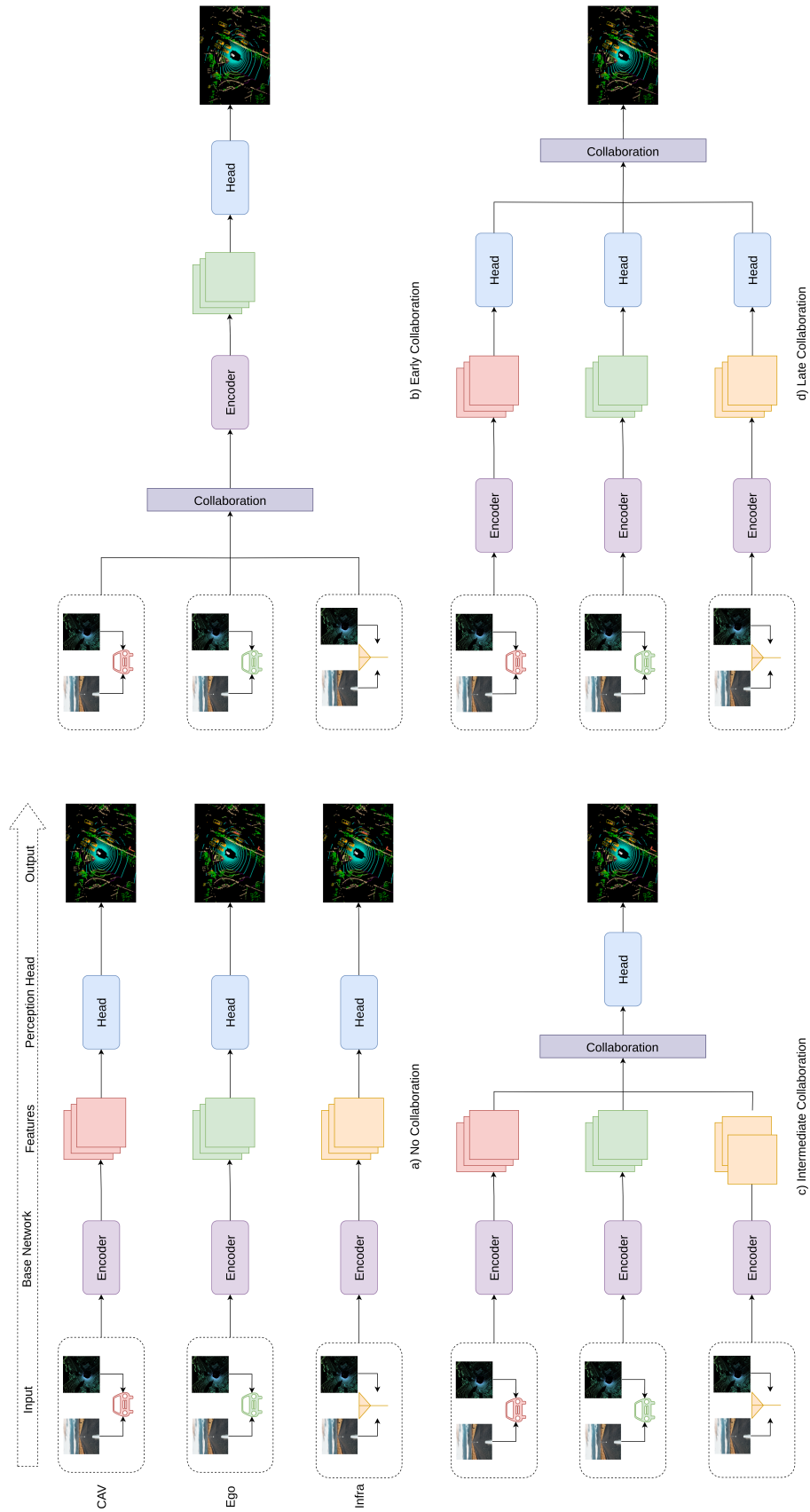


Figure 2.5: Example of collaborative perception system collaboration scheme. (a) Single agent perception framework. (b) Early collaboration perception framework. (c) Intermediate collaboration perception framework. (d) Late collaboration perception framework. Redesigned from [32].

it the most widely adopted choice for multi-agent collaborative perception. By transferring processed features rather than raw data, intermediate collaboration significantly reduces bandwidth requirements, improving the feasibility of real-time applications. However, this method introduces challenges related to information loss during feature extraction and redundancy from transmitting unnecessary data. These issues have spurred ongoing research into optimizing feature selection and fusion strategies to ensure that only the most relevant and high-quality features are shared and utilized, enhancing overall system efficiency and effectiveness.

Finally, late or object-level collaboration focuses on prediction fusion at the network output, as depicted in Figure 2.5 (d). In this scheme, each agent independently trains its network and shares the outputs with other agents. The ego vehicle then spatially transforms these outputs and merges them after postprocessing. This approach is notably more bandwidth-efficient and simpler compared to early and intermediate collaboration, as it involves sharing processed predictions rather than raw data or intermediate features. These advantages make late collaboration appealing for scenarios with limited communication resources or computational constraints. However, late collaboration comes with significant drawbacks. The outputs generated by individual agents can be noisy or incomplete, leading to degraded performance when the predictions are fused. As a result, late collaboration typically delivers the poorest perception performance among the three schemes. This trade-off underscores the need for robust noise-handling and output enhancement techniques to mitigate its limitations and improve reliability in practical applications.

Collaboration Modules

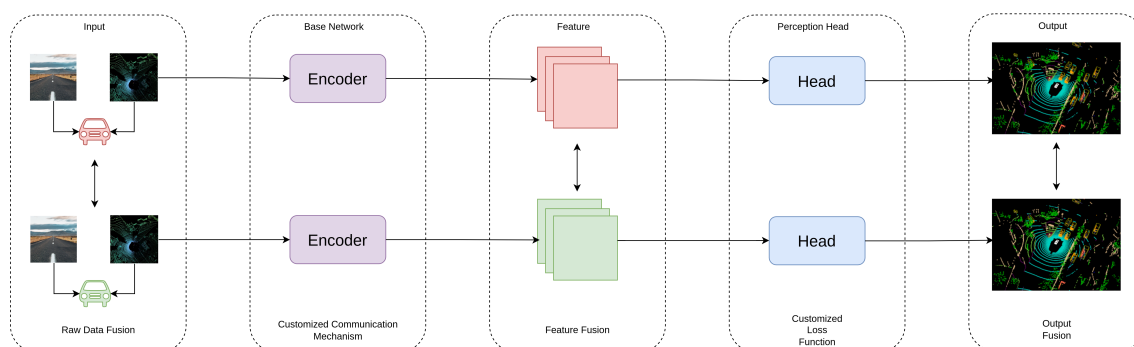


Figure 2.6: Collaboration modules in the collaborative perception system. Adapted from [32].

In collaborative perception systems, researchers typically design dedicated collaboration modules at various stages of the network to enhance both collaborative efficiency and perception accuracy. These modules are essential for aggregating multi-agent observations and correspond to three primary collaboration schemes, as illustrated in Figure 2.6, based on the taxonomy presented in [32]. Early collaboration is characterized by raw data fusion at the input stage, where unprocessed sensor data from multiple agents is integrated to form a rich, comprehensive environmental representation. Intermediate collaboration involves feature fusion during the feature extraction

process, where deep semantic features are shared and aggregated across agents to improve contextual understanding. Late collaboration, on the other hand, performs output fusion at the final stage, where predictions from individual agents are merged to generate the final perceptual output. Beyond these core schemes, some studies propose customized communication mechanisms, primarily integrated within intermediate fusion architectures. These mechanisms aim to optimize bandwidth usage and transmission efficiency by selectively exchanging information from the most relevant agents and transmitting only the most informative features. While effective, this specific topic is beyond the scope of this dissertation. Additionally, customized loss functions are frequently introduced at the perception head level to direct the network’s learning process toward capturing the most salient features for perception tasks. Such loss functions help the system focus on task-critical information, ultimately enhancing performance and computational efficiency. Nevertheless, this topic is also beyond the scope of this dissertation, as it primarily focuses on the design of collaborative perception architectures rather than the specific loss functions employed within them.

Figure 2.7 presents the evolution of collaborative perception architectures in AD, highlighting the transition from early-stage raw data fusion approaches to more advanced feature-level collaboration techniques.

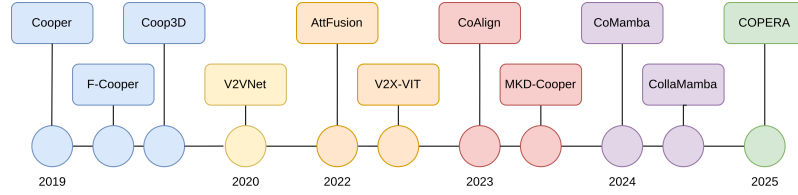


Figure 2.7: Evolution of collaborative perception architectures in AD. Copper [33], Coop3D [34], F-Cooper [35], V2VNet [36], AttFusion [37], V2X-VIT [38], CoAlign [39], MKD-Cooper [40], CoMamba [41], CollaMamba [42], COPERA [43].

Raw Data Fusion

Raw data fusion occurs at the input stage of the network, in early collaboration for perception systems. The irregular structure of point clouds allows for direct aggregation, which has driven the development of techniques centered on point cloud fusion.

The first system to introduce early collaborative perception, Cooper [33], proposed by *Qi Chen et al.* (2019), selects LiDAR data as the fusion target. To minimize data size, Cooper compresses point clouds by preserving only positional coordinates and reflection intensity values. After information exchange between agents, Cooper reconstructs the received point clouds using a transformation matrix and combines them with the ego vehicle’s point cloud data for final predictions. Expanding on Cooper’s approach, Coop3D [34], developed by *Eduardo Arnold et al.* (2019), enhances point cloud fusion techniques in early collaboration. Rather than using straightforward concatenation, Coop3D applies spatial transformation techniques for more effective sensor data integration. Moreover, Coop3D replaces Cooper’s onboard V2V communication model with a

centralized system to merge sensor data from multiple sources. This centralized architecture facilitates the sharing of sensor and processing costs, improving efficiency and reducing overall expenses. However, due to the high bandwidth requirements of raw data communication between vehicles, raw data fusion approaches, are generally not considered optimal solutions compared to other state-of-the-art architectures.

Feature Fusion

Recent advancements in DL have sparked significant interest in feature fusion techniques within the research community. These techniques aim to enhance collaborative perception by merging extracted features at an intermediate stage of processing. Feature fusion methods can be broadly categorized into four main types: Traditional, Graph, Attention or State Space Model (SSM) based fusion.

At the initial stage of collaborative perception, traditional feature fusion methods, such as concatenation, summation, and linear weighting, were widely utilized due to their simplicity and ability to achieve fast inference. Building upon these approaches, the first intermediate collaborative perception framework, F-Cooper, introduced by *QI Chen et al.* (2019) [35], focuses on extracting and integrating two types of features: low-level voxel features and deep spatial features. F-Cooper employs two feature fusion strategies: Voxel Feature Fusion (VFF) and Spatial Feature Fusion (SFF). Both methods utilize element-wise maxout operations to merge features in overlapping regions. Since voxel features are closer to raw data, VFF achieves a performance level comparable to raw data fusion, particularly for near-object detection. Meanwhile, SFF takes advantage of channel selection techniques to reduce transmission overhead while preserving detection accuracy, offering a balanced approach to efficiency and precision.

Despite the simplicity of traditional intermediate fusion methods, they fail to account for the underlying relationships among multiple agents and lack the ability to reason about the messages exchanged between senders and receivers. Graph Neural Networks (GNN) have proven effective in addressing these shortcomings by enabling the propagation and aggregation of messages between neighboring nodes, demonstrating their potential in perception tasks. V2VNet [36], presented by *T. Wang et al.* (2020), takes advantage of a spatially-aware GNN for modeling communication among agents. In its message-passing phase, V2VNet employs a variational image compression algorithm to compress features, ensuring efficient data transmission. During the cross-vehicle aggregation phase, the framework compensates for time delays to establish an initial state for each node. It then spatially transforms and warps compressed features from neighboring agents to align with the ego vehicle's perspective. These operations are confined to overlapping FOV. For the feature fusion phase, V2VNet adopts an averaging operation to aggregate features and updates the node states using a convolutional gated recurrent unit. While V2VNet demonstrates significant performance improvements through its use of GNN, its reliance on scalar-valued collaboration weights limits its ability to capture the varying importance of different spatial regions. CoAlign, proposed by *Yifan Lu et al.* [39] (2023), is a hybrid collaboration framework specifically designed to address pose estimation errors in multi-agent environments. The framework utilizes

agent-object pose graph modeling to improve pose consistency and employs a multi-scale feature fusion strategy to increase robustness. Unlike prior approaches that rely on ground-truth pose data for supervision, CoAlign is more practical as it operates without such supervision and makes no specific assumptions about pose error distributions.

Furthermore, attention mechanisms have emerged as powerful tools for modeling feature relationships in collaborative perception systems. Recent research on attention modules [20] and Vision Transformers (ViTs) [44] has demonstrated their superior performance in state-of-the-art architectures. A notable advancement in this domain was the introduction of OPV2V by *Runsheng Xu et al.* (2022) [37]. This work proposed an attention-based intermediate fusion framework, named AttFusion, which combines data from multiple connected vehicles using a PointPillars [26] backbone. In addition to the architecture, OPV2V presented the first large-scale simulated dataset dedicated to V2V collaborative perception, significantly enriching the field's research resources. To address the challenges of robustness and flexibility in V2X scenarios, V2X-ViT [38] introduced by *Runsheng Xu et al.* (2022), is a vision transformer-based framework that integrates a holistic attention mechanism. This architecture alternates between heterogeneous multi-agent self-attention and multi-scale window self-attention, facilitating effective inter-agent fusion while mitigating issues such as pose errors and sensor modality discrepancies. Building on these advancements, *Zhiyuan Li et al.* (2023) introduced MKD-Cooper [40], a cooperative perception framework based on multi-teacher knowledge distillation for 3D object detection. Central to this architecture is the Collaborative Attention Fusion (CAF) module, which dynamically captures inter-vehicle interactions. By integrating deep feature representations from neighboring vehicles, MKD-Cooper constructs a context-rich fused feature map. Furthermore, it introduces an adaptive knowledge distillation approach that assigns weights to each teacher model based on performance, thereby facilitating efficient and effective knowledge transfer to the student model.

More recently, to address the computational bottlenecks inherent in attention-based mechanisms, particularly their quadratic complexity, SSM mechanisms have recently gained traction as a promising alternative for collaborative perception in AD systems. *Jinlong Li et al.* (2024) introduced CoMamba [41], a real-time cooperative 3D object detection framework that takes advantage of bidirectional SSMs. By replacing attention modules with state-space modeling, CoMamba significantly reduces computational overhead, thus improving scalability and enabling efficient real-time inference in onboard vehicle systems. Building on this paradigm, *Yang Li et al.* (2024) proposed CollaMamba [42], a multi-agent collaborative perception framework based on spatial-temporal SSM. This architecture features a spatial SSM backbone designed to capture cross-agent causal dependencies with linear complexity, alongside a history-aware temporal feature enhancement module that enriches the model's contextual understanding. Together, these components enable effective feature refinement while maintaining a lightweight and computationally efficient design, making CollaMamba particularly suitable for deployment in resource-constrained and real-time operational environments.

Building upon the advances in attention mechanisms, *Celso Pereira et al.* (2025) introduced COPERA [43], an architecture designed for 2D object detection in AD scenarios. COPERA de-

defines a Collaborative Perception Space (CPS) that allows multiple CAVs to collaboratively share relative poses and feature representations through V2V communication. This architecture based in the BEV representation with three input channels, range, intensity and height, is designed to operate in real-time on constrained devices, such as CAVs onboard stack. Furthermore, the architecture incorporates variants based on state-of-the-art 2D object detectors, including RT-DETR and YOLO11, to demonstrate its flexibility and performance scalability. Central to COPERA is the Cross-View Spatial Attention Fusion (CSAF) block, which employs a multi-head attention mechanism to model inter-agent spatial relationships and selectively fuse features from neighboring CAVs, thereby improving detection robustness in occluded and sensor-limited conditions.

Output Fusion

Contrasting with raw data fusion, output fusion merges data during the postprocessing phase by combining the outputs from multiple agents. In applications like 3D object detection, it commonly uses postprocessing techniques to remove overlapping or low-confidence predictions.

However, late fusion methods encounter significant challenges, such as spatial misalignment, which arises from discrepancies in the positioning of outputs, and temporal misalignment, caused by differences in data collection or processing times. These challenges limit the effectiveness of late fusion approaches, making them less commonly used in collaborative perception systems.

2.5 Spatial Representations

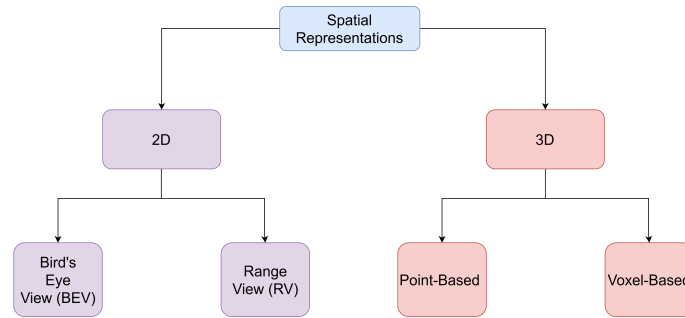


Figure 2.8: Types of spatial representations in a perception system

The spatial representation of 3D data plays a critical role in determining the performance and computational efficiency of DL architectures used in AD systems. 3D data can be expressed in multiple forms, primarily categorized as 2D projections or 3D representations [45], as illustrated in Figure 2.8. In 2D representations, the inherently 3D data obtained from sensors like LiDAR and RADAR is projected onto a 2D plane. This approach results in two commonly used formats, Bird's-Eye-View (BEV) and Range View (RV). On the other hand, 3D representations preserve the spatial integrity of the point cloud data and are generally divided into two primary methodologies, Point-Based methods and Voxel-Based methods.

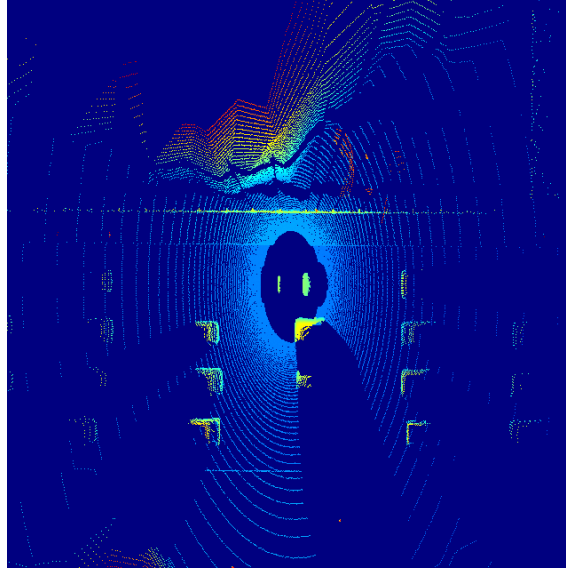


Figure 2.9: BEV representation of a 3D point cloud.

Concerning 2D representation, BEV methods project the 3D point cloud onto a 2D top-down representation, offering a comprehensive overview of the environment as illustrated in Figure 2.9. This projection is particularly useful in AD, as it provides an intuitive, global perspective of both static and dynamic objects in the scene. BEV representations are highly effective at capturing the spatial relationships between objects, which is crucial for tasks such as object detection and motion planning. One of the main advantages of the BEV representation is its ability to present the spatial layout in a compact and memory-efficient manner, while retaining important contextual information. This makes it well-suited for real-time processing, with the application of efficient 2D convolutions. Furthermore, by reducing the complexity of working directly with raw 3D point clouds, BEV enables faster and more scalable processing for downstream tasks. However, BEV representations do have some drawbacks. The transformation of 3D data into a 2D plane can result in the loss of fine-grained details, especially for objects that are far from the sensor or partially occluded. Additionally, BEV suffers from sparsity, as many areas may not contain relevant data, leading to gaps or sparse regions in the 2D projection. This sparsity can be problematic for detection and classification tasks, particularly in environments with low point cloud density. Moreover, the BEV format may not fully capture vertical structures or intricate 3D interactions between objects. To address these limitations, recent methods have focused on refining the BEV representation. For example, BirdNet+ [46] infers oriented 3D bounding boxes solely from BEV images by utilizing a two-stage object detector enhancing the representation of vertical structures. Furthermore, some approaches integrate additional 3D information with the BEV to improve detection accuracy, especially in challenging scenarios where fine-grained localization and the handling of sparsity is crucial.

Instead of a 2D top-down representation as present in Figure 2.10, RV methods utilize a spherical projection to map a 3D point cloud onto a 2D image plane, resulting in what is commonly

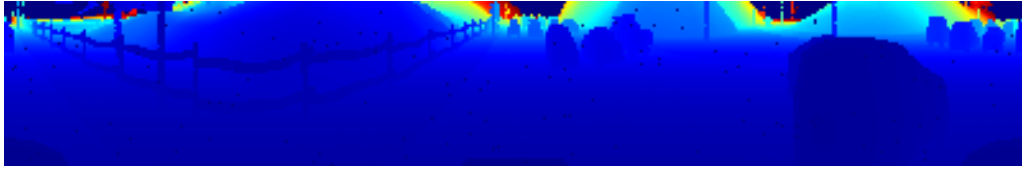


Figure 2.10: RV representation of a 3D point cloud.

referred to as a "range image." This technique offers several notable advantages. It is highly memory-efficient, as the range image minimizes the presence of empty pixels, optimizing memory utilization. Additionally, it inherently encodes spatial information about the environment, including both occupied regions and free space, making it particularly well-suited for representing spatial occupancy. RV methods are also computationally efficient, enabling the use of dense 2D convolutions for rapid image processing. Despite their advantages, the adoption of RV representations for object detection remains limited, particularly in collaborative scenarios. Compared to other methods, raw or intermediate RV data often contributes little to collaborative perception due to its limited information density. Additionally, these representations are highly sensitive to scale variations, as they depend on the LiDAR resolution, which can differ significantly across CAVs. Recent advancements aim to address the limitations of RV methods in object detection. For instance, RCD [47] presented by *Alex Bewley et al.* (2020), introduces a range-conditioned dilation layer that dynamically adjusts the dilation rate as a function of the measured range, effectively mitigating scale sensitivity issues. Similarly, RangeDet [48] introduced by *Lue Fan et al.* (2021), tackles challenges such as scale variations between near and far objects, as well as the inconsistency between the 2D range image coordinates used during feature extraction and the 3D Cartesian coordinates required for the final output.

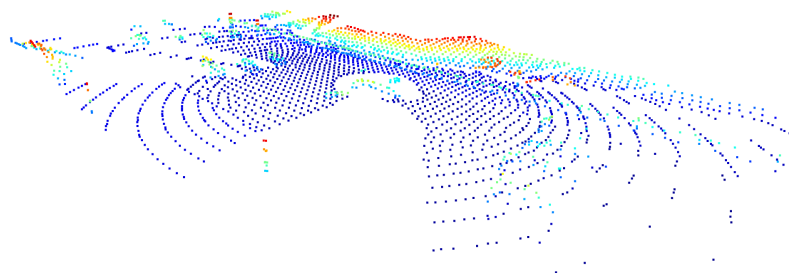


Figure 2.11: Voxel-based representation of a 3D point cloud.

Contrarily, 3D representation like Voxel-based methods as showcased in Figure 2.11, use a grid-based approach to divide the 3D space into scriptsize, discrete voxel cells. These methods often face challenges in areas of high point density, where multiple LiDAR points may fall into the same voxel cell, leading to collisions. Some techniques address these collisions by selecting

the nearest point, while others, like PointPillars [26] presented by *Lang et al.* (2018), apply a max-pooling strategy to extract features.

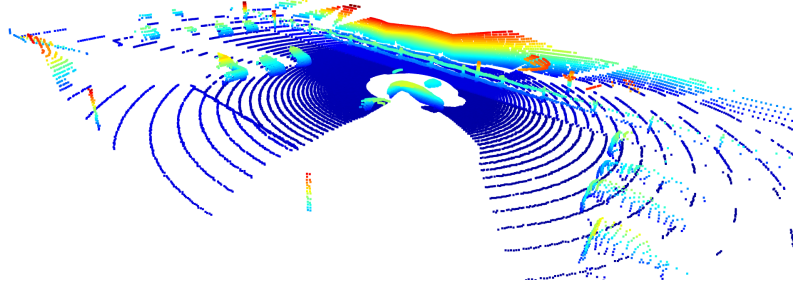


Figure 2.12: Point-based representation of a 3D point cloud.

Ultimately as illustrated in Figure 2.12, Point-based methods process the entire point cloud directly, avoiding the need to project it into a lower-dimensional space. Techniques like PointNet [22], presented by *Charles et al.* (2016), introduced permutation-invariant networks, allowing the processing of point clouds without depending on the specific order of points. Additionally, PointNet++ [49], also introduced by *Charles et al.* (2017), enhanced these methods but was primarily limited to structured indoor environments like object detection tasks. Although PointRCNN [23], presented by *Shaoshuai Shi et al.* (2018), extended their use to urban settings like AD, point-based methods struggle to scale effectively with larger point clouds and extensive scenes, making them unsuitable for real-time, safety-critical applications due to high latency.

2.6 Critical Review

Up to this point, a comprehensive overview of state-of-the-art collaborative perception systems within the context of AD has been presented. This review encompasses critical areas such as object detection, sensor multi-modality and data representation, all of which are pivotal for the advancement of effective and reliable AD systems.

Table 2.1 provides a summary of key state-of-the-art architectures in collaborative perception, highlighting their approaches regarding scheme, data fusion, feature fusion and data representation. The analysis of this table reveals that most contemporary methods rely on intermediate collaboration. Early collaboration was initially preferred due to its simpler implementation. However, the high bandwidth demands of this approach render it unsuitable for real-time applications. As a result, there has been a shift toward intermediate collaboration methods, which achieve a balance between efficiency and accuracy by sharing features and incorporating attention and SSM mechanisms. These advancements make intermediate methods more suitable for real-time applications. Regarding data representation, voxel-based and BEV representations dominate the field due to their computational efficiency and ease of processing compared to other approaches.

Table 2.1: State-of-the-Art Collaborative Perception Methods.

Method	Year	Modality	Scheme	Data Fusion	Feat Fusion	Data Representation
Cooper [33]	2019	LiDAR	E	Raw	-	Voxel
F-Cooper [35]	2019	LiDAR	I	-	Trad	Voxel
Coop3D [34]	2019	LiDAR	E	Raw	-	Voxel
V2VNet [36]	2020	LiDAR	I	-	Graph	Voxel
AttFusion [37]	2022	LiDAR	I	-	Atten	Voxel
V2X-VIT [38]	2022	LiDAR	I	-	Atten	Voxel
CoALign [39]	2023	LiDAR	I	-	Graph	Voxel
MKD-Cooper [40]	2023	LiDAR	I	-	Atten	Voxel
CoMamba [41]	2024	LiDAR	I	-	SSM	Voxel
CollaMamba [42]	2024	LiDAR	I	-	SSM	BEV
COPERA [43] (*)	2025	LiDAR	I	-	Atten	BEV

Overall: not addressed by the method (-).

Scheme: early (E), intermediate (I), or late (L) collaboration.

Data Fusion: raw data (Raw) or output (Out) fusion.

Feature Fusion: traditional (Trad), graph (Graph), attention (Atten) or state-space-model (SSM) fusion.

Data Representation: voxel-based (Voxel), point-based (Point), range view (RV), or bird’s-eye view (BEV).

(*): 2D Object Detector

The OPV2V [37] dataset is a widely adopted large-scale benchmark developed to support research in collaborative perception for AD. It was generated using the CARLA simulator [50] in conjunction with the OpenCDA co-simulation framework [51], enabling high-fidelity, multi-agent interactions in diverse urban scenarios. Spanning over 70 driving scenes across nine environments, including eight distinct CARLA towns and a high-resolution digital replica of Culver City, Los Angeles, OPV2V contains 11,464 frames and 232,913 annotated 3D vehicle bounding boxes. Data collection utilizes a comprehensive multi-modal sensor suite consisting of a 64 beam LiDAR (120-meter range), four RGB cameras, and Global Positioning System (GPS)/Inertial Measurement Unit (IMU) sensors, all synchronized at 10 Hz to emulate real-world CAV configurations. This dataset supports a variety of cooperative perception tasks such as 3D object detection, BEV semantic segmentation, object tracking, and motion prediction. Its scale, diversity, and sensor realism make it a robust benchmark for evaluating multi-agent perception systems under realistic and challenging conditions. Performance on OPV2V is commonly assessed using Average Precision (AP)@0.5 and AP@0.7, which correspond to AP scores at Intersection over Union (IoU) thresholds of 0.5 and 0.7, respectively. These metrics reflect both the detection accuracy and the localization precision required for safety-critical AD applications.

Table 2.2 provides a comparative analysis of state-of-the-art collaborative perception models evaluated on the OPV2V dataset using the AP@0.5 metric. The evaluation spans two distinct environments: the Default CARLA Towns and the Culver City Digital Town. These scenarios offer varying levels of environmental complexity, enabling a comprehensive assessment of each model’s robustness and generalization capabilities. All models in Table 2.2 adopt intermediate collaboration strategies, where agents exchange high-level feature representations rather than raw sensor data or final predictions. This approach strikes a balance between the communication bandwidth of early fusion and the limited contextual awareness of late fusion, allowing models to benefit

Table 2.2: Comparison of state-of-the-art 3D object detection models on the OPV2V dataset.

Model	Default CARLA Towns AP@0.5	Culver City Digital Town AP@0.5
F-Cooper [35]	86.5	77.0
V2V-Net [36]	88.1	86.8
AttFusion [37]	90.8	85.4
CoMamba [41]	91.9	87.4
V2X-ViT [38]	94.4	85.4
MKD-Cooper [40]	96.5	89.5
CoAlign [39]	96.6	N/A
CollaMamba [42]	97.5	N/A
COPERA [43] (*)	97.85	94.63

AP@0.5: AP at an IoU threshold of 0.5.

N/A: Not available.

(*): 2D Object Detector

from shared contextual information while maintaining scalability. Recent advancements indicate a growing momentum behind SSM-based architectures, such as CoMamba and CollaMamba, which deliver competitive performance by modeling temporal dynamics with linear computational complexity. Attention-based models, like AttFusion, V2X-ViT, and MKD-Cooper, also demonstrate strong results. Their ability to focus on salient spatial features using multi-head attention mechanisms gives them a significant advantage over traditional fusion methods like F-Cooper and earlier graph-based techniques such as V2V-Net. Among all evaluated models, the most notable breakthrough comes from COPERA, a collaborative perception architecture designed for 2D object detection. COPERA achieves the highest performance on both benchmarks, reaching an AP@0.5 of 97.85% in the Default CARLA Towns and 94.63% in the Culver City Digital Town. Its effectiveness stems from the novel CSAF block, which utilizes multi-head attention to model spatial relationships between CAVs, thereby enhancing collaborative feature fusion. Although COPERA operates in the 2D domain, predicting object locations on the BEV plane rather than producing full 3D bounding boxes, is sufficient for many applications. Tasks such as obstacle avoidance, path planning, and intersection handling can often rely solely on accurate BEV-level detections. As such, COPERA’s performance illustrates a shift towards lighter, attention 2D models that can match or exceed the capabilities of more complex 3D frameworks, especially in real time scenarios.

Introduced by *Xiang et al.* in late 2024, the V2X-Real [52] dataset is a large-scale, real-world benchmark designed to facilitate research in cooperative perception for AD, with a particular focus on interactions between CAV and smart roadside infrastructure. Data was collected in complex urban environments using two CAV and two intelligent infrastructure units, each equipped with synchronized multi-modal sensors. The dataset comprises over 33,000 LiDAR frames and 171,000 RGB images, annotated with more than 1.2 million bounding boxes spanning 10 object categories. These annotations support a wide range of perception tasks, including object detection under various contextual conditions. Each vehicle was equipped with a 128 beam LiDAR (200-meter range), four RGB-D cameras, all synchronized at 10 Hz and high-frequency GPS/IMU sensors operating at 1000 Hz. The infrastructure units featured either 128 beam or 64 beam LiDARs (40-meter range), two RGB cameras operating at 30 Hz, and GPS modules updating at 5 Hz. Elevated view-

points provided by the infrastructure enable enhanced scene coverage, especially in occluded or congested scenarios. Together, these sensor-rich platforms create a realistic and diverse testbed for evaluating cooperative perception algorithms. The V2X-Real dataset stands out as a valuable resource for developing scalable and robust V2X perception systems applicable to real-world AD environments. Performance on this benchmark is evaluated using mAP@0.3 and mAP@0.5, which measure mean Average Precision (mAP) at IoU of 0.3 and 0.5, over three superclasses, car, pedestrian, and truck. These metrics assess both the detection accuracy and the localization precision of predicted bounding boxes under practical urban conditions.

Table 2.3: State-of-the-Art 3D Object Detection results on the V2X-Real Dataset under the VC setting.

Model	mAP@0.3	mAP@0.5
F-Cooper [35]	38.1	29.8
AttFusion [37]	42.5	33.8
V2X-ViT [38]	44.8	35.8

AP@X: AP at an IoU threshold of X.

mAP@X: mAP at an IoU threshold of X.

The results in Table 2.3 demonstrate the continued dominance of attention-based methods in real-world cooperative perception tasks. Traditional fusion approaches like F-Cooper lag significantly behind, achieving a mAP@0.5 of just 29.8%, whereas attention-based models such as AttFusion and V2X-ViT perform substantially better, with V2X-ViT reaching the highest scores, 44.8% at mAP@0.3 and 35.8 at mAP@0.5. These results validate the effectiveness of attention mechanisms in prioritizing relevant features and suppressing noise across diverse sensor modalities and agent viewpoints. However, it is important to note that models based in SSM frameworks, such as CoMamba and CollaMamba, which show strong performance in simulated environments like OPV2V, have yet to be evaluated in real-world datasets like V2X-Real. As such, their true applicability and robustness in complex, real-world scenarios remains uncertain.

In conclusion, while significant progress has been made in collaborative perception, particularly with attention-based fusion strategies, still remains a considerable gap between synthetic benchmarks and real-world deployment. The relatively modest mAP values, even for state-of-the-art models, highlight the ongoing challenges of achieving reliable, high-precision perception in dynamic and cluttered urban environments. Furthermore, the discrepancy between simulation and reality underscores the need for more extensive real-world testing and the development of robust, scalable architectures capable of generalizing beyond controlled settings. These findings point toward a pressing need for continued innovation in perception algorithms, data diversity, and evaluation metrics to bridge the gap toward fully autonomous and collaborative driving systems.

Chapter 3

Deep Learning Architecture for Collaborative Perception in Autonomous Driving

This chapter presents the deep learning architecture designed to support advanced collaborative perception in multi-agent AD systems. It begins with a high-level overview of the proposed architecture in Section 3.1, outlining the core components and design. The mechanisms for sharing information between collaborative agents are explained in Section 3.2. Section 3.3 details the data representation strategies designed for collaborative scenarios. Section 3.4 describes the data augmentation methods employed to improve the model generalization and robustness. Finally, Section 3.5 provides details about specific network components.

3.1 Introduction

Intermediate collaboration, where agents exchange intermediate feature maps instead of raw point clouds or final detection outputs, has emerged as an effective strategy for collaborative perception. This approach provides a strong balance between perception accuracy, communication efficiency, and system scalability. Within this paradigm, the proposed PerceptNet-V2X architecture is specifically designed for AD scenarios, combining multi-perspective LiDAR processing with intermediate feature collaboration through V2X communication. As shown in Figure 3.1, each agent first projects its local point cloud into the ego vehicle’s reference frame, when applicable. The transformed point cloud is then passed into the FPV block, which generates a novel Full Perspective View (FPV) representation that simultaneously captures vertical and BEV perspectives of the environment.

Within the FPV block, two outputs are produced: a vertical feature map of size $640 \times 640 \times 4$, denoted as Vertical Features, and a raw BEV feature map of size $640 \times 640 \times 2$, denoted as Raw BEV. The Vertical Features are forwarded to the BEV Reprojection Block, where they are

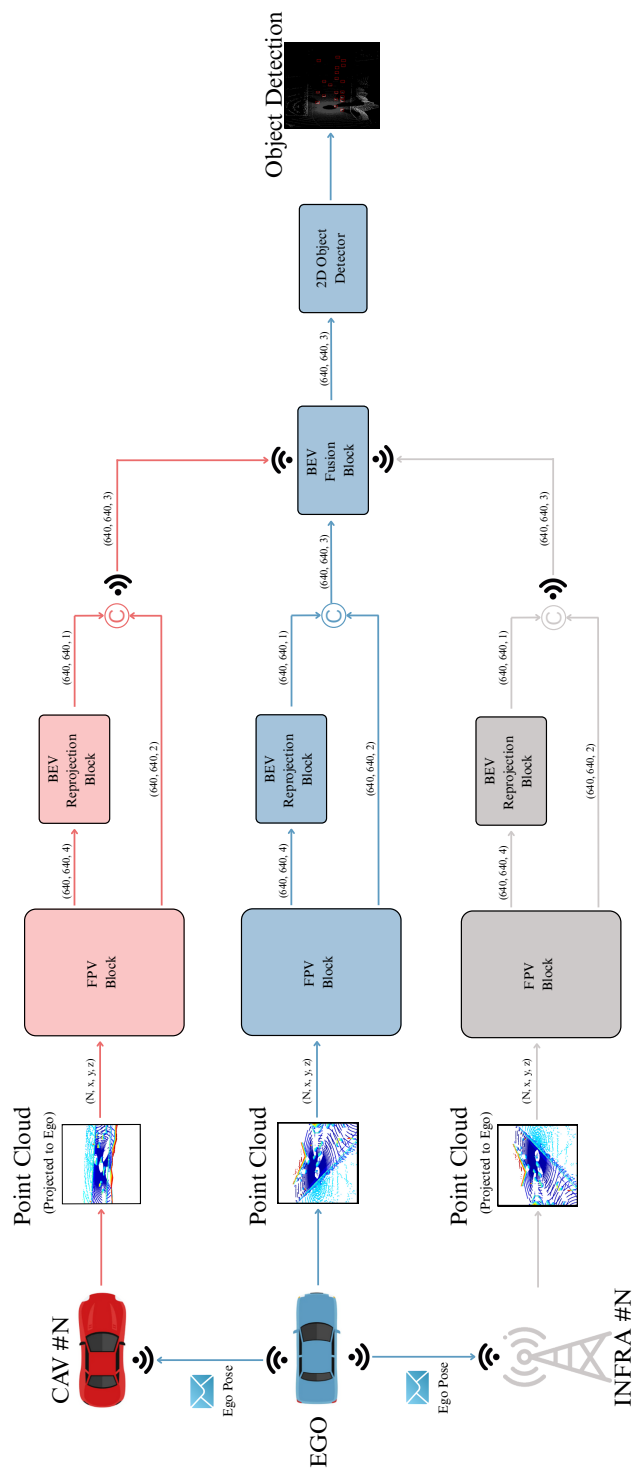


Figure 3.1: Overview of the PerceptNet-V2X architecture.

transformed into a BEV-aligned feature map referred to as the BEV Vertical Features of shape $640 \times 640 \times 1$.

The raw BEV features generated inside the FPV block are preserved and later concatenated with the BEV Vertical features. This results in a unified $640 \times 640 \times 3$ BEV feature map that captures both geometric structure and vertical contextual information. Moreover, these intermediate BEV feature maps are transmitted to the ego vehicle via V2X communication. At the ego node, the received maps are integrated using the BEV Fusion Block, which performs a grid-wise aggregation. Specifically, for each grid cell, it selects the maximum from the BEV feature maps received from each agent.

To evaluate the effectiveness of the PerceptNet-V2X fusion strategy, the resulting fused BEV tensor is processed by a 2D object detection module. Two high-performance detectors, YOLO and RT-DETR, are used to benchmark detection accuracy and validate the benefits of the proposed collaborative perception framework. While the current implementation is validated using 2D object detectors, the architecture is designed to be semi-agnostic, allowing adaptation to a range of perception tasks, such as BEV segmentation.

3.2 Data Sharing

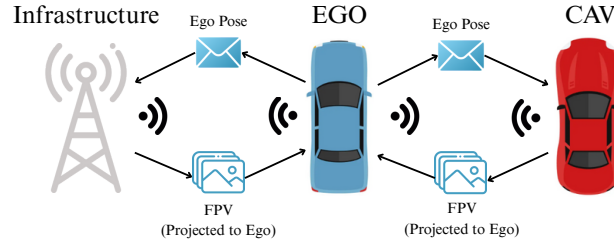


Figure 3.2: FPV data sharing between nodes for the PerceptNet-V2X architecture.

Building upon the architectural overview, sharing data between collaborative nodes is essential for enhancing the perception capabilities of AVs. Whether sharing raw point cloud data, intermediate feature maps, or detection outputs, it is critical that all shared information be accurately spatially aligned within a common reference frame, typically the ego vehicle’s frame, to ensure consistent interpretation and fusion.

Figure 3.2 illustrates the data sharing and alignment process used in the PerceptNet-V2X architecture. This mechanism enables the ego vehicle to extend its perceptual field by incorporating spatially aligned information from surrounding CAVs and infrastructure nodes. To facilitate this, the ego vehicle first broadcasts its current pose to all nearby collaborative nodes. Each node, whether a CAV or infrastructure unit, uses the received ego pose to project its locally captured 3D point cloud data into the ego vehicle’s coordinate frame, ensuring all shared data is spatially coherent.

This transformation relies on a sequence of rigid-body operations represented through homogeneous transformation matrices. The pose of each entity, either a node or the ego vehicle, is

described by six parameters: a position vector (x, y, z) and an orientation specified by roll (ϕ) , pitch (θ) , and yaw (ψ) . Together, these parameters define the full 6-DoF configuration of each platform in the global 3D space. From these, the necessary transformation matrices are constructed to convert coordinates from a node's local frame into the ego frame, enabling accurate and consistent multi-agent data fusion. From these parameters, the homogeneous transformation matrix $H_{\text{node}}^{\text{world}}$ is constructed by combining a rotation matrix $R_{\text{node}}^{\text{world}}$ and a translation vector $T_{\text{node}}^{\text{world}}$. The translation vector is simply:

$$T_{\text{node}}^{\text{world}} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (3.1)$$

The rotation matrix $R_{\text{node}}^{\text{world}}$ is computed from the Euler angles, roll, pitch and yaw:

$$R_{\text{node}}^{\text{world}} = \begin{bmatrix} \cos \theta \cos \psi & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & -\cos \psi \sin \theta \cos \phi - \sin \psi \sin \phi \\ \cos \theta \sin \psi & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & -\sin \psi \sin \theta \cos \phi + \cos \psi \sin \phi \\ \sin \theta & -\cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix}. \quad (3.2)$$

The full homogeneous transformation matrix from the world to the node frame is defined as:

$$H_{\text{node}}^{\text{world}} = \begin{bmatrix} R_{\text{node}}^{\text{world}} & T_{\text{node}}^{\text{world}} \\ \mathbf{0}^\top & 1 \end{bmatrix}. \quad (3.3)$$

An analogous transformation matrix $H_{\text{ego}}^{\text{world}}$ is constructed for the ego vehicle. To transform a point from the node's local frame into the ego vehicle's local frame, a two-step process is performed. The complete transformation between the node's frame and the ego vehicle's frame is:

$$H_{\text{node}}^{\text{ego}} = (H_{\text{ego}}^{\text{world}})^{-1} \cdot H_{\text{node}}^{\text{world}}. \quad (3.4)$$

Given a 3D point in the node's local frame $p^{\text{node}} = (x, y, z)^\top$, it is first transformed to homogeneous coordinates:

$$\tilde{p}^{\text{node}} = \begin{bmatrix} p^{\text{node}} \\ 1 \end{bmatrix}. \quad (3.5)$$

The transformation into the ego vehicle's frame is then expressed as:

$$\tilde{p}^{\text{ego}} = H_{\text{node}}^{\text{ego}} \cdot \tilde{p}^{\text{node}}. \quad (3.6)$$

This transformation ensures that spatial data shared between nodes is geometrically consistent within the ego vehicle's coordinate system. Once transformed, these aligned 3D points are projected into 2D utilizing the FPV representation presented in Figure 3.3. This step is crucial in

PerceptNet-V2X, as it ensures that collaborative data is correctly fused, allowing for enhanced perception even in scenarios involving occlusions or restricted FOV.

3.3 Data Representation

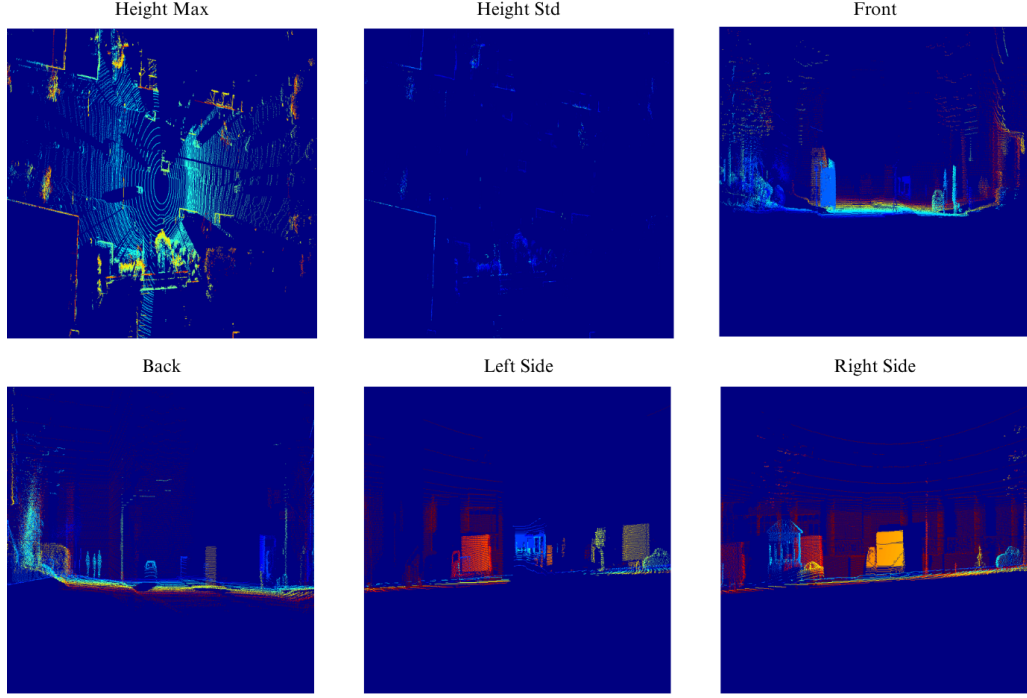


Figure 3.3: FPV projection of a V2X-Real dataset point cloud.

A critical factor in achieving strong performance in DL models, particularly for the PerceptNet-V2X architecture, is the precise data representation of the model input. In the context of the PerceptNet-V2X architecture, a central innovation is the introduction of the FPV representation. This representation provides a structured and compact 2D encoding of raw 3D point cloud data by projecting it onto six distinct 2D spaces, thereby preserving essential geometric features while enabling efficient neural processing. As illustrated in Figure 3.3, the FPV includes projections onto the BEV plane, as well as vertical projections capturing the front, back, left, and right perspectives relative to the agent. Let the raw point cloud be denoted as: $P = \{(x_i, y_i, z_i)\}_{i=1}^N$ where each point represents a 3D coordinate. All projections are rendered at a fixed resolution of 640×640 , with pixel coordinates computed using specific scaling factors s_x, s_y, s_z and positional offsets L_1, W_1, H_1 to align physical coordinates with image space. Let G_{uv} denote the set of points that project into the pixel located at (u, v) .

The six FPV projections are defined as follows:

- BEV: Defined over the horizontal (x, y) plane. Pixel indices are computed as:

$$u = (x - L_1) \cdot s_x, \quad v = (y - W_1) \cdot s_y. \quad (3.7)$$

This view encodes two features:

- Height Max:

$$I_{\max}(u, v) = \max_{(x_i, y_i) \in G_{uv}} z_i, \quad (3.8)$$

- Height Std:

$$I_{\text{std}}(u, v) = \sqrt{\frac{1}{|G_{uv}|} \sum_{(x_i, y_i) \in G_{uv}} (z_i - \bar{z}_{uv})^2}, \quad (3.9)$$

where z_i is the height of each of the points in G_{uv} , $\bar{z}_{uv}$ is the mean height and $|G_{uv}|$ is the number of points in the cell.

- Front View: Captures the forward perspective by projecting onto the (y, z) plane, using only points where $x > 0$. Pixel indices:

$$u = (y - W_1) \cdot s_y, \quad v = (z - H_1) \cdot s_z. \quad (3.10)$$

Feature value encodes the minimum forward distance:

$$I_{\text{Front}}^{\text{depth}}(u, v) = \min_{(y_i, z_i) \in G_{uv}} x_i. \quad (3.11)$$

- Back View: Similar to front view, but for points where $x < 0$. Same projection plane and pixel mapping:

$$u = (y - W_1) \cdot s_y, \quad v = (z - H_1) \cdot s_z. \quad (3.12)$$

The feature is the absolute backward distance (to maintain positivity):

$$I_{\text{Back}}^{\text{depth}}(u, v) = \min_{(y_i, z_i) \in G_{uv}} |x_i|. \quad (3.13)$$

- Left View: Projects points with $y < 0$ onto the (x, z) plane. Pixel coordinates:

$$u = (x - L_1) \cdot s_x, \quad v = (z - H_1) \cdot s_z. \quad (3.14)$$

The encoded value is the absolute lateral distance:

$$I_{\text{Left}}^{\text{depth}}(u, v) = \min_{(x_i, z_i) \in G_{uv}} |y_i|. \quad (3.15)$$

- Right View: Uses points where $y > 0$ and projects them onto the (x, z) plane. Pixel computation:

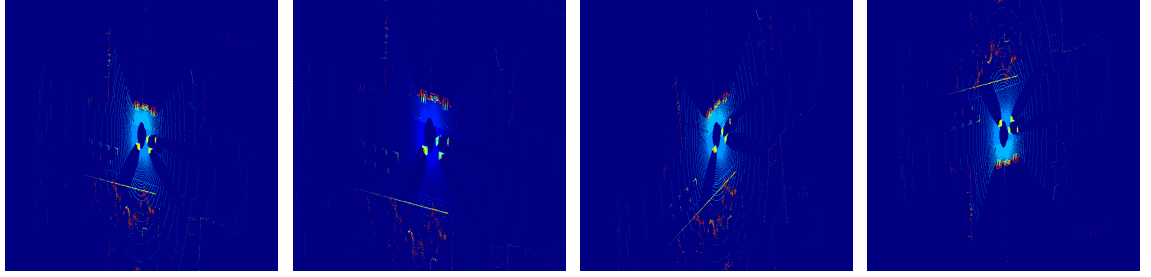
$$u = (x - L_1) \cdot s_x, \quad v = (z - H_1) \cdot s_z. \quad (3.16)$$

The feature value is the lateral distance:

$$I_{\text{Right}}^{\text{depth}}(u, v) = \min_{(x_i, z_i) \in G_{uv}} y_i. \quad (3.17)$$

Together, these six projections form a rich spatial tensor of shape $640 \times 640 \times 6$, which preserves both horizontal and vertical structural information from multiple fixed viewpoints. This representation enables efficient processing using 2D convolutional networks while retaining critical 3D geometric information inherent in the original point cloud.

3.4 Augmentation Techniques



(a) BEV normal projection. (b) BEV scaled projection. (c) BEV rotated projection. (d) BEV flipped projection.

Figure 3.4: Impact of augmentation techniques visualized on a BEV maximum height projection sample of an OPV2V dataset point cloud.

Connected to the data representation, the augmentation techniques employed in PerceptNet-V2X are designed to enhance the model’s robustness and generalization capabilities. These techniques introduce controlled random variability into the training data, enabling the model to learn more robust and invariant features that generalize better to unseen scenarios.

In this work, a set of augmentation techniques, as illustrated in Figure 3.4, is applied at the 3D point cloud level, which is particularly important given the model’s reliance on spatial and geometric features for accurate perception. These augmentations help simulate diverse geometrical conditions and sensor variations, enhancing the model’s robustness and generalization without the need for collecting additional data. The augmentation strategies employed are as follows.

Let $p = (x, y, z)^\top$ represent a point in the point cloud defined as $P = \{(x_i, y_i, z_i)\}_{i=1}^N$. Each augmentation transforms p into a new point p' using a specific randomized operation, described below:

- **Random Rotation:** The point cloud is randomly rotated around the vertical (z) axis by an angle ψ . The rotation angle ψ is sampled from a continuous uniform distribution over the interval $[-\pi, \pi]$, expressed as:

$$\psi \sim U(-\pi, \pi). \quad (3.18)$$

The rotated point p' is then computed as:

$$p' = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} p. \quad (3.19)$$

This simulates varying heading directions of the ego vehicle and surrounding agents, helping the model become invariant to rotational transformations commonly encountered during driving scenarios.

- **Random Scaling:** The point cloud is uniformly scaled by a factor s , where s is drawn from a continuous uniform distribution over $[0.5, 1.5]$:

$$s \sim U(0.5, 1.5). \quad (3.20)$$

The scaled point p' is given by:

$$p' = sp = (sx, sy, sz)^\top. \quad (3.21)$$

This augmentation simulates variations in object sizes, improving the model's robustness to real-world scale discrepancies and sensor artifacts.

- **Random Flipping:** The point cloud is randomly flipped along the y-axis with a probability of 50%. The flipping decision is modeled as a Bernoulli random variable F :

$$F \sim \text{Bernoulli}(0.5). \quad (3.22)$$

If flipped (i.e., $F = 1$), the transformed point p' is defined as:

$$p' = (x, -y, z)^\top. \quad (3.23)$$

This mimics mirrored driving environments, such as opposite-direction traffic or urban layouts in different regions. It encourages the model to recognize objects and scenes regardless of spatial orientation, thereby enhancing generalization across diverse scenarios.

Additionally, to address the inherent sparsity in each of the vertical projections, the FPV representation is further augmented using a local pixel expansion technique. Due to the discrete nature of the projection and the inherent sparsity of LiDAR measurements, many pixels remain unassigned or isolated. To address this its applied a kernel-based value propagation on each vertical view, that expands each non-zero pixel across a fixed neighborhood. For a given projection $I_{\text{view}}^{\text{depth}}(i, j)$, an augmented projection $I'_{\text{view}}^{\text{depth}}(i, j)$, where $\text{view} \in \{\text{Front}, \text{Back}, \text{Left}, \text{Right}\}$, is defined by applying the following update rule:

$$I'_{\text{view}}^{\text{depth}}(i, j) = \min \left(I_{\text{view}}^{\text{depth}}(i, j), I'_{\text{view}}^{\text{depth}}(i, j) \right), \quad \forall (i, j) \in N_k(u, v), \quad (3.24)$$

where (u, v) is any pixel such that $I_{\text{view}}^{\text{depth}}(i, j) > 0$ and $N_k(u, v)$ is the square neighborhood centered at (u, v) with kernel size $k \times k$, where $k = 3$.

The use of the min operator ensures that in overlapping regions, the closest structure is preserved, in line with the depth in the FPV vertical views. This leads to a more connected and

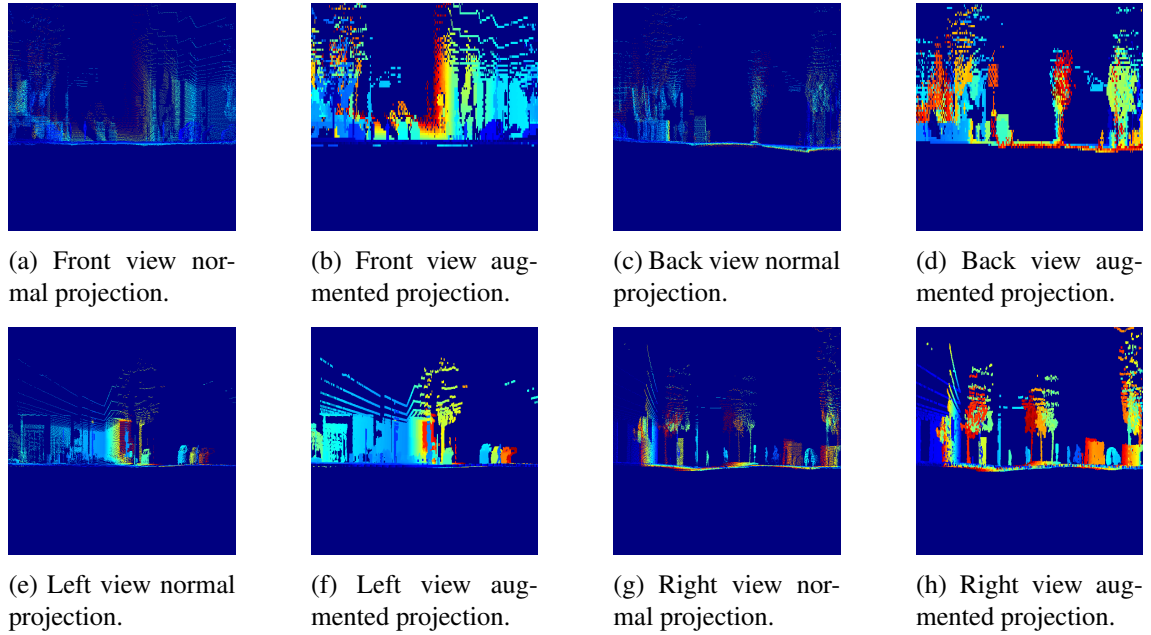


Figure 3.5: Effect of local pixel expansion on vertical FPV projections from the V2X-Real dataset. Each pair shows the original and augmented views for front, back, left, and right directions.

informative projection as illustrated in Figure 3.5, particularly in areas where LiDAR returns are sparse or discontinuous due to occlusions. This sparsity augmentation aims to enhance the continuity of directional features and improve semantic encoding through the Sparse Encoder-Decoder network.

3.5 Network Components

After the high-level introduction of PerceptNet-V2X in section 3.1, it is essential to examine the internal components that constitute the architecture. This subsection explains how the FPV Block is connected to the BEV Reprojection Block. Furthermore, it details how the BEV Fusion Block aggregates intermediate feature maps from multiple agents and produces a BEV fused representation, which is subsequently passed to 2D object detection modules for final inference.

FPV Block

The FPV Block, shared across all nodes in the PerceptNet-V2X architecture and illustrated in Figure 3.6, plays a central role in transforming raw 3D point cloud data into a structured 2D representation that captures both vertical and BEV perspectives of the environment. The input point cloud is first encoded into the novel FPV representation, which is then processed through two parallel pathways. In the vertical view pathway, the vertical projections, front, back, left, and right, are aggregated into a feature map of size $640 \times 640 \times 4$ and passed through a Sparse Encoder-Decoder network to extract semantic features. In the BEV pathway, two statistical maps representing the maximum height and height standard deviation per grid cell are preserved as

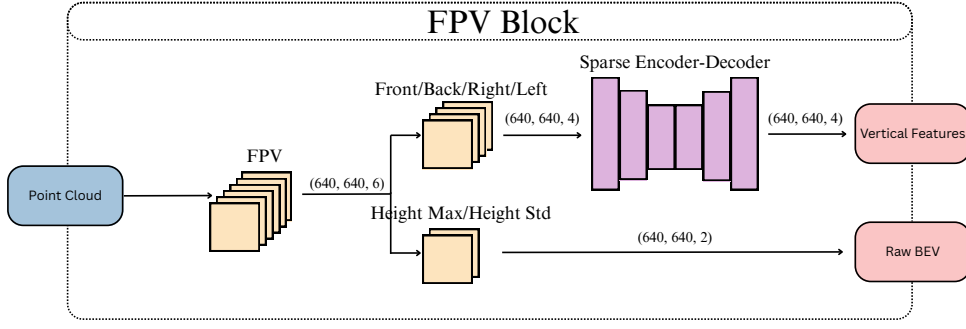


Figure 3.6: FPV Block common to all nodes in the PerceptNet-V2X architecture.

raw BEV features with dimensions $640 \times 640 \times 2$. The FPV Block thus produces two outputs: a semantically enriched vertical feature map of size $640 \times 640 \times 4$, denoted as Vertical Features and Raw BEV features of size $640 \times 640 \times 2$. These outputs together form a compact representation of the scene for further processing.

Sparse Encoder-Decoder

The Sparse Encoder-Decoder, as illustrated in Figure 3.7 and inspired by U-Net [53] architectures, constitutes a foundational processing module of the FPV Block. This architecture is engineered to accommodate and process spatially sparse input modalities, with a primary focus on vertically projected views such as the front, back, left, and right views derived from LiDAR point clouds. These inputs are inherently sparse, even after pixel expansion augmentation, and often contain significant regions of missing or occluded data due to the nature of 3D sensing and environmental conditions. The network takes as input a single-channel sparse image $I_{\text{view}}^{\text{depth}}(i, j)$, where $\text{view} \in \{\text{Front, Back, Left, Right}\}$, and automatically generates a binary validity mask $M_{\text{view}}^{\text{depth}}(i, j)$ by identifying non-zero pixel locations. This approach ensures that all subsequent operations are explicitly conditioned on the spatial validity of the input data.

The encoder pathway $Enc1 \rightarrow Enc4$ consists of four hierarchical blocks that progressively extract higher-level semantic features. Each encoder block follows the same structural pattern:

- **Convolution:** Applies masked convolutions to extract features from valid regions with batch normalization and ReLU activation function.
- **Mask Dilation:** Expands the validity mask to accommodate the convolution receptive field.
- **Activation:** Creates an activation mask based on feature magnitudes to identify valid responses.
- **Update:** Updates both features and masks based on the activation mask.
- **Downsampling:** Reduces spatial resolution by half while preserving essential information.

The bottleneck, $Enc5$, captures compact semantic features at the lowest resolution. This block consists of:

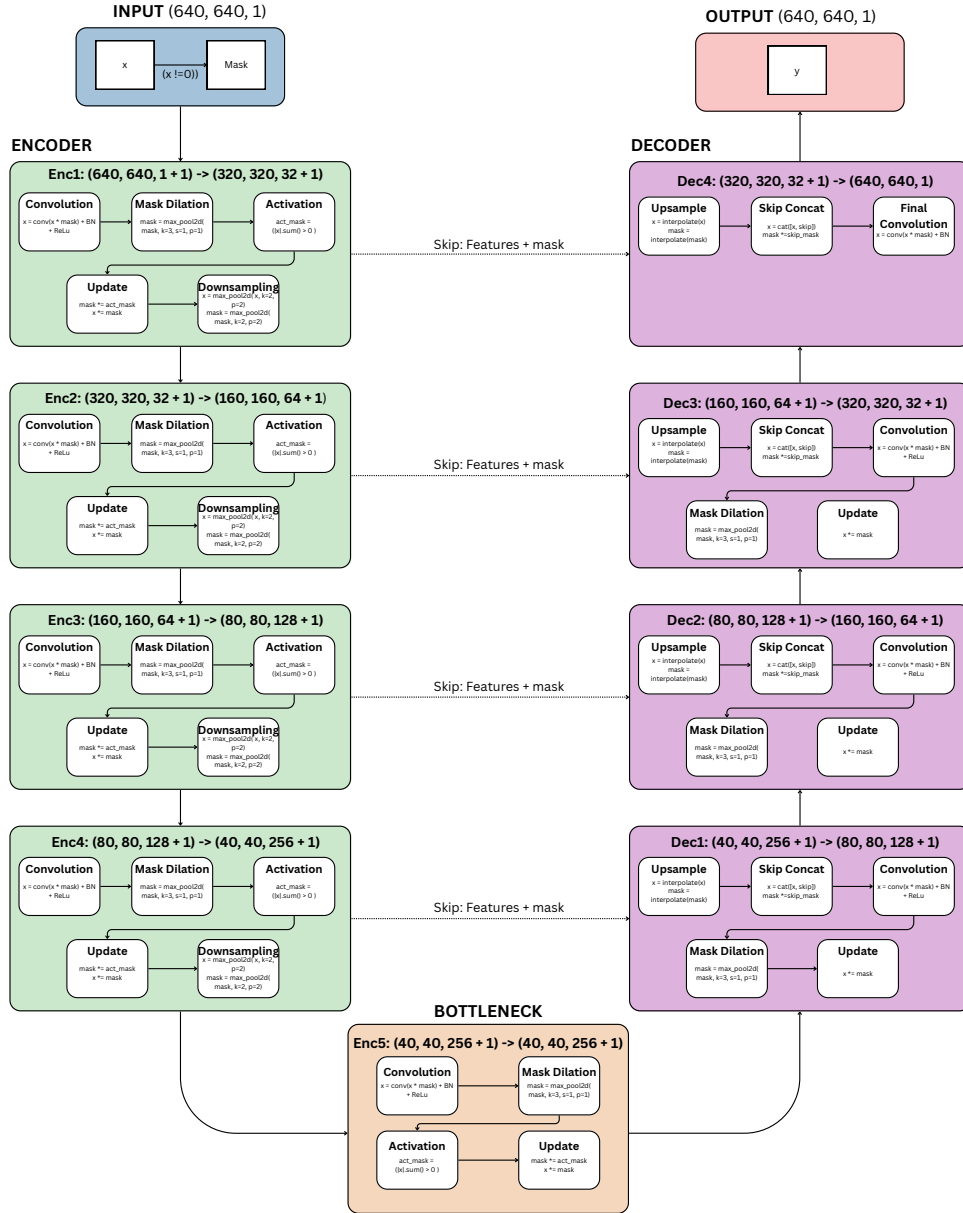


Figure 3.7: Sparse Encoder-Decoder block common to all FPV Blocks in the PerceptNet-V2X architecture.

- Convolution: Final feature extraction at the bottleneck, with masked convolutions, batch normalization and ReLU activation function.
- Mask Dilation: Expands the validity mask for context aggregation.
- Activation: Generates an activation mask from bottleneck features.
- Update: Updates the latent representation and associated mask.

The decoder pathway Dec1 $\rightarrow$ Dec4 follows a symmetric U-Net structure with progressively decreasing channel dimensions $256 \rightarrow 128 \rightarrow 64 \rightarrow 32 \rightarrow 1$. The decoder aims to restore full

spatial resolution while incorporating fine details through skip connections. Each decoder block operates as follows:

- **Upsampling:** Both feature maps and validity masks are upsampled using bilinear interpolation for features and nearest-neighbor interpolation for masks to match the target resolution of the corresponding encoder stage.
- **Skip Concat:** Features from the symmetric encoder stage are concatenated channel-wise, while masks are combined multiplicatively to ensure validity constraints from both pathways are respected.
- **Convolution:** The concatenated features undergo masked convolution to fuse multi-scale information while preserving sparsity constraints. Standard normalization and ReLU activation are applied, except for the final decoder layer which omits ReLU to allow negative feature values.
- **Mask Dilation:** Expands the validity mask for context aggregation.
- **Update:** Updates the feature representation for the next layer based on the validity mask.

The final decoder block Dec4 restores the full input resolution (640×640) and outputs dense feature maps through a final masked convolutional layer. This output represents various semantic features extracted from the input sparse image, conditioned on the validity mask.

In general, this architecture provides several key benefits through its mask-aware operations that ensure resilience to missing or irregular data. Geometric and semantic consistency is retained via skip connections and mask-guided refinement. The network automatically generates and maintains validity masks from input sparsity patterns, eliminating the need for external mask computation. Additionally, the design offers high modularity as a backbone adaptable to various vision tasks in both single-agent and collaborative multi-agent settings. Within the PerceptNet-V2X framework, the Sparse Encoder-Decoder module serves as the fundamental feature extraction backbone, transforming raw, sparse LiDAR observations into semantically meaningful and spatially dense representations. These representations provide a robust foundation for multi-agent collaboration and enable effective perception in complex real-world environments where data sparsity and occlusions are inherent challenges.

BEV Reprojection Block

After vertical semantic features, denoted as Vertical Features, are extracted by the Sparse Encoder-Decoder, they are passed to the next stage in the PerceptNet-V2X architecture: the BEV Reprojection Block. This module receives the extracted features from the vertical views and reprojects them into a common BEV coordinate frame. The process involves associating each extracted vertical feature with its corresponding 3D point and then projecting it onto the BEV plane while preserving geometric and semantic coherence.

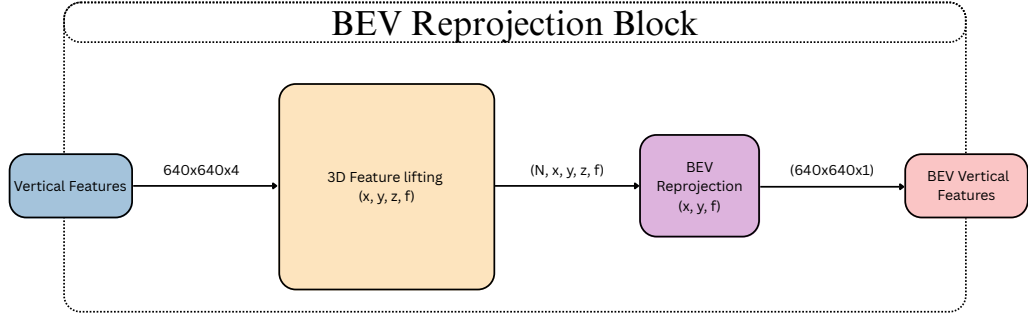


Figure 3.8: BEV Reprojection Block common to all nodes in the PerceptNet-V2X architecture.

The transformation takes two sequential operations: a 3D Feature Lifting step, followed by a BEV Projection step as illustrated in Figure 3.8. Each vertical feature map produced by the Sparse Encoder-Decoder, corresponding to the front, back, left, and right views, encodes semantic features with a spatial resolution of $640 \times 640 \times 4$. These vertical feature maps are denoted as $I_{\text{Front}}^{\text{feature}}(u, v)$, $I_{\text{Back}}^{\text{feature}}(u, v)$, $I_{\text{Left}}^{\text{feature}}(u, v)$, $I_{\text{Right}}^{\text{feature}}(u, v)$. Let each pixel in a feature map be associated with a corresponding 3D point through an auxiliary depth map $I_{\text{Front}}^{\text{depth}}(u, v)$, $I_{\text{Back}}^{\text{depth}}(u, v)$, $I_{\text{Left}}^{\text{depth}}(u, v)$, $I_{\text{Right}}^{\text{depth}}(u, v)$. Then, for each view, a lifting operation maps the encoded 2D feature map back into the 3D space. This lifting is view-dependent and follows the geometric inversion of the original projection. For each pixel (u, v) in a vertical view, the mapping to a 3D point (x, y, z) and its semantic feature f is defined as follows:

- Front View: projected over the (y, z) plane with $x > 0$:

$$x = I_{\text{Front}}^{\text{depth}}(u, v), \quad y = \frac{u}{s_y} + W_1, \quad z = \frac{v}{s_z} + H_1, \quad f = I_{\text{Front}}^{\text{feature}}(u, v). \quad (3.25)$$

- Back View: projected over the (y, z) plane with $x < 0$:

$$x = -I_{\text{Back}}^{\text{depth}}(u, v), \quad y = \frac{u}{s_y} + W_1, \quad z = \frac{v}{s_z} + H_1, \quad f = I_{\text{Back}}^{\text{feature}}(u, v). \quad (3.26)$$

- Left View: projected over the (x, z) plane with $y < 0$:

$$y = -I_{\text{Left}}^{\text{depth}}(u, v), \quad x = \frac{u}{s_x} + L_1, \quad z = \frac{v}{s_z} + H_1, \quad f = I_{\text{Left}}^{\text{feature}}(u, v). \quad (3.27)$$

- Right View: projected over the (x, z) plane with $y > 0$:

$$y = I_{\text{Right}}^{\text{depth}}(u, v), \quad x = \frac{u}{s_x} + L_1, \quad z = \frac{v}{s_z} + H_1, \quad f = I_{\text{Right}}^{\text{feature}}(u, v). \quad (3.28)$$

Each lifted point and its feature vector forms a tuple (x, y, z, f) , resulting in a 3D semantic feature point cloud $P = \{(x_i, y_i, z_i, f_i)\}_{i=1}^N$ over all views. The final step of the Reprojection Block consists in projecting the lifted 3D features back onto the BEV plane and creating an additional BEV vertical feature representation.

- **BEV Vertical Features:** Defined over the horizontal (x, y) plane, where each grid cell aggregates the lifted 3D features. Pixel indices are computed as:

$$u = (x - L_1) \cdot s_x, \quad v = (y - W_1) \cdot s_y. \quad (3.29)$$

The aggregated BEV feature for each cell is then computed as the average over all point features within that cell:

$$I_{\text{features}}(u, v) = \frac{1}{|G_{uv}|} \sum_{i \in G_{uv}} f_i, \quad (3.30)$$

where $|G_{uv}|$ is the number of lifted points falling into the cell (u, v) .

This operation effectively reprojects high-dimensional features into a dense 2D BEV representation, preserving semantic and spatial context while enabling subsequent processing via 2D convolutional layers.

BEV Fusion Block

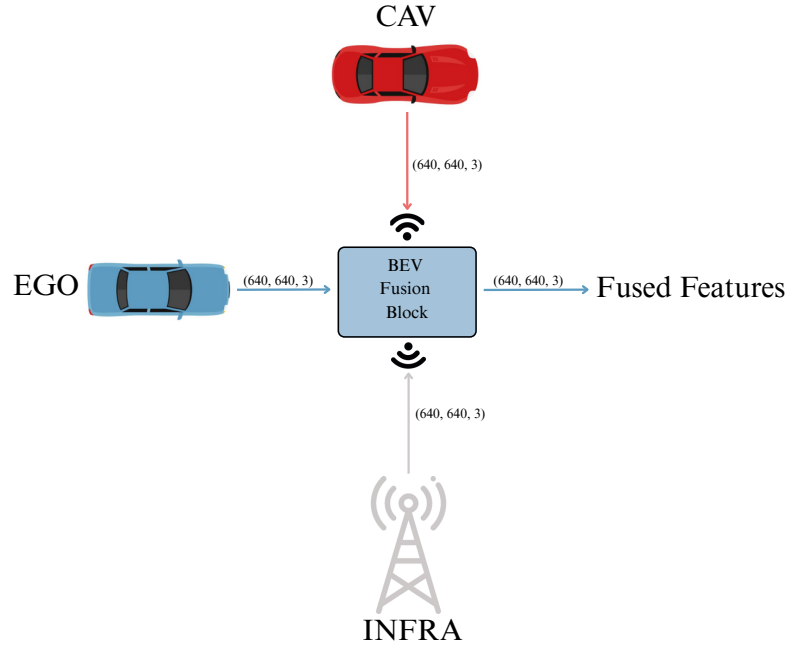


Figure 3.9: BEV Fusion Block of the PerceptNet-V2X architecture.

The BEV Fusion Block, highlighted in Figure 3.9, is responsible for combining the BEV-based concatenated features from the ego vehicle and external collaborating agents to form a unified BEV representation of the environment.

Let the ego vehicle's feature maps be denoted as $I_{\text{max}}^{\text{ego}}(u, v)$, $I_{\text{std}}^{\text{ego}}(u, v)$, and $I_{\text{features}}^{\text{ego}}(u, v)$, corresponding respectively to the maximum height, the standard deviation of height, and the lifted 3D feature representation aggregated in each grid cell G_{uv} . Similarly, let the external agent's maps be $I_{\text{max}}^{\text{node}}(u, v)$, $I_{\text{std}}^{\text{node}}(u, v)$, and $I_{\text{features}}^{\text{node}}(u, v)$.

The fusion is applied cell-wise across the three feature maps. For the maximum height map, the fused value is given by:

$$I_{\max}^{\text{fused}}(u, v) = \max(I_{\max}^{\text{ego}}(u, v), I_{\max}^{\text{node}}(u, v)). \quad (3.31)$$

This formulation ensures that in the absence of ego information, the external agent's data fills the gap. In overlapping regions, the fusion prioritizes the taller structure to preserve vertical scene geometry. For the standard deviation map, which encodes local structural variability, the fusion follows a similar logic:

$$I_{\text{std}}^{\text{fused}}(u, v) = \max(I_{\text{std}}^{\text{ego}}(u, v), I_{\text{std}}^{\text{node}}(u, v)). \quad (3.32)$$

By using the maximum of the two values, the fused representation remains sensitive to complex surface structures that may be captured more distinctly by one of the agents. Finally, for the lifted BEV Vertical Features aggregated in BEV cells, a max operation is applied across corresponding feature channels to preserve the most salient semantic and spatial signals:

$$I_{\text{features}}^{\text{fused}}(u, v) = \max(I_{\text{features}}^{\text{ego}}(u, v), I_{\text{features}}^{\text{node}}(u, v)), \quad (3.33)$$

where the max is applied element-wise.

This BEV fusion strategy produces a coherent, high-fidelity spatial representation that integrates complementary observations from multiple viewpoints, while maintaining robustness in the presence of occlusions or partial observations.

2D Object Detectors

To validate the effectiveness of the proposed PerceptNet-V2X pipeline, variants of two state-of-the-art object detectors, RT-DETR [8] and YOLOV12 [18], are employed. These enhanced models, on their largest versions, denoted as RT-DETR-X and YOLOV12-X, are particularly well-suited to take advantage of the rich spatial and semantics available in the fused feature representations. Their integration ensures a comprehensive and fair evaluation of detection performance under various collaborative perception settings.

Both models are trained using their original multi-component loss formulations as proposed by their respective authors. These formulations combine classification and localization objectives, along with model-specific regularization terms. Each component is scaled by a dedicated hyperparameter, denoted by λ , allowing precise control over the learning dynamics and task prioritization during training. The total loss for the RT-DETR-X model comprises four components:

$$L_{\text{RT-DETR}} = L_{\text{main}} + L_{\text{aux}} + L_{\text{dn}} + L_{\text{aux-dn}}. \quad (3.34)$$

These terms represent:

- L_{main} : main detection loss,
- L_{aux} : auxiliary loss from intermediate decoder layers,

- L_{dn} : denoising loss,
- $L_{\text{aux-dn}}$: auxiliary denoising loss.

Each component loss L_i , where $i \in \{\text{main}, \text{aux}, \text{dn}, \text{aux-dn}\}$, is composed of a weighted combination of classification, bounding box regression, and generalized IoU terms:

$$L_i = \lambda_{\text{class}} L_{\text{class}} + \lambda_{\text{bbox}} L_{\text{bbox}} + \lambda_{\text{giou}} L_{\text{giou}}. \quad (3.35)$$

Here, L_{class} denotes the classification loss, L_{bbox} is the bounding box regression loss, and L_{giou} refers to the Generalized Intersection over Union (GIoU) loss, which enhances spatial alignment between predicted and ground-truth boxes. Hyperparameters $\lambda_{\text{class}}, \lambda_{\text{bbox}}, \lambda_{\text{giou}}$ allow for adaptive weighting during optimization.

The loss function used to train the YOLOV12-X architecture is similarly multi-faceted:

$$L_{\text{YOLO}} = \lambda_{\text{cls}} L_{\text{cls}} + \lambda_{\text{box}} L_{\text{box}} + \lambda_{\text{dff}} L_{\text{dff}}. \quad (3.36)$$

In this formulation:

- L_{cls} : classification loss,
- L_{box} : bounding box regression loss,
- L_{dff} : Distribution Focal Loss.

As with RT-DETR-X, the individual loss terms of YOLOV12-X are modulated by the respective weights $\lambda_{\text{cls}}, \lambda_{\text{box}}, \lambda_{\text{dff}}$, enabling a balanced training process that can adapt to the demands of collaborative perception scenarios. Overall, the use of these two powerful detectors ensures that the quality of the fused representations is rigorously assessed through detection performance.

Chapter 4

Results

This chapter presents a comprehensive evaluation of the proposed collaborative perception framework across both synthetic and real-world driving scenarios. Section 4.1 provides a high-level overview of the chapter objectives. Section 4.2 details the computational environment, including the hardware and software configurations used during model training and evaluation. Section 4.3 defines the performance metrics used to assess detection accuracy, robustness, and overall system effectiveness. Section 4.4 presents experimental results on the OPV2V synthetic dataset, and section 4.5 reports the corresponding evaluation on the real-world V2X-Real dataset.

4.1 Introduction

The objective of this chapter is to assess the proposed collaborative perception model performance across both synthetic and real-world scenarios using the OPV2V and V2X-Real datasets. These datasets provide a robust foundation for analyzing the behavior of collaborative perception systems in varying environmental and operational contexts. The evaluation focuses on the accuracy and robustness of the model under diverse driving conditions, including scenarios with significant occlusions, restricted sensor visibility, and heterogeneous agent configurations. Such conditions closely reflect real-world challenges faced by AVs in complex urban settings. The evaluation process is divided into two principal components. The first part introduces the system specifications and the evaluation metrics employed to quantify detection performance. The second part presents detailed experimental results obtained from the benchmark datasets, offering comparative insights against existing state-of-the-art methods.

4.2 System Specifications

To ensure reproducibility and transparency, the computational environments used for both training and inference phases are outlined in Table 4.1. These environments differ in performance capabilities, reflecting both high-end cloud computing resources and more constrained, consumer grade hardware setups.

Table 4.1: System Specifications.

Component	Specification
General Software	
Operating System	Ubuntu 22.04
Python Version	3.10
PyTorch Version	2.6.0
CUDA Version	12.4
Inference Environment	
CPU	Intel Core i5-13420H
GPU	NVIDIA GeForce RTX 4050 Laptop GPU
VRAM	6 GB
RAM	16 GB
Training Environment	
CPU	Intel Xeon Bronze 3204
GPU	NVIDIA Tesla A100
VRAM	40 GB
RAM	128 GB

The training process was conducted on a high-performance server equipped with an NVIDIA Tesla A100 GPU with 40 GB of Video Random Access Memory (VRAM) and 128 GB of Random Access Memory (RAM). This setup facilitated efficient training of DL models with large batch sizes and complex architectures. In contrast, inference and model testing were carried out locally on a laptop featuring an Intel Core i5 processor and an NVIDIA RTX 4050 GPU with 6 GB of VRAM and 16 GB of RAM. This demonstrates the practical applicability of the model in scenarios with limited computational resources, such as onboard vehicle hardware.

The consistency of the software stack across environments using Ubuntu 22.04, Python 3.10, PyTorch 2.6.0, and CUDA 12.4, ensured compatibility and stability during development, experimentation and deployment.

4.3 Evaluation Metrics

The evaluation of the proposed collaborative perception model is based on three widely employed object detection metrics: AP, mAP and the F1 Score. These metrics quantify the accuracy and robustness of the detection pipeline across different conditions and classes. To support the understanding of these metrics, the core concepts of precision, recall and IoU are explained in Appendix A.

- **Average Precision (AP):** AP is a widely used metric in object detection and segmentation tasks. It quantifies the area under the precision-recall curve, offering a scalar value that summarizes the trade-off between precision and recall over varying confidence thresholds. To construct this curve, detections are ranked by their classification confidence scores. By iteratively lowering the score threshold, it is generated multiple pairs of precision and recall values, where low-confidence predictions are progressively treated as false positives. Each point on the curve reflects the precision and recall computed at a specific threshold. The

resulting precision-recall curve is typically non-convex and sometimes may be smoothed before integration. The AP is then calculated as the area under this curve:

$$AP = \int_0^1 p(r) dr, \quad (4.1)$$

where $p(r)$ represents precision as a function of recall r . AP is typically computed with respect to a specific IoU threshold, which determines whether a predicted bounding box is classified as a True Positive. A detection is considered correct if its IoU with the corresponding ground truth object exceeds the defined threshold, otherwise, it is treated as a False Positive. For instance, in COCO [54] evaluation, the final reported score, AP@0.5:0.95, is the average of AP values computed at different IoU thresholds (from 0.5 to 0.95 in 0.05 increments), providing a more comprehensive assessment of detection quality.

- **Mean Average Precision (mAP):** mAP is the average of the AP values across all object classes. It serves as a comprehensive metric to evaluate multi-class object detection performance by aggregating the detection quality per class into a single scalar value. The mAP is computed as:

$$mAP = \frac{1}{C} \sum_{c=1}^C AP_c, \quad (4.2)$$

where C denotes the total number of classes and AP_c is AP for class c . This metric reflects both the precision and recall of the model across different object categories and is widely used in benchmark challenges like PASCAL VOC [55].

- **F1-Score:** F1 score is defined as the harmonic mean of precision and recall, offering a single, balanced metric that simultaneously accounts for both false positives and false negatives. It is particularly advantageous when dealing with imbalanced datasets, where relying solely on accuracy or either precision or recall may provide misleading conclusions. Mathematically, the F1 score is expressed as:

$$F1\text{-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (4.3)$$

A high F1 score signifies that the model achieves a strong balance between precision and recall, thus effectively minimizing both types of classification errors. This makes the F1 score especially valuable for evaluating detection systems where the trade-off between detecting true positives and avoiding false alarms is critical. Similar to the AP metric, the F1 score is generally computed with respect to a specific IoU threshold. This threshold determines whether a predicted bounding box sufficiently overlaps with a ground truth box to be considered a True Positive, directly impacting the precision and recall values used in the F1 score calculation.

4.4 Model Evaluation in synthetic environments

In order to validate the performance of the proposed collaborative perception architecture on a simulated setting, the OPV2V [37] dataset, which is a widely used synthetic dataset designed for evaluating collaborative perception systems in AD scenarios is used.

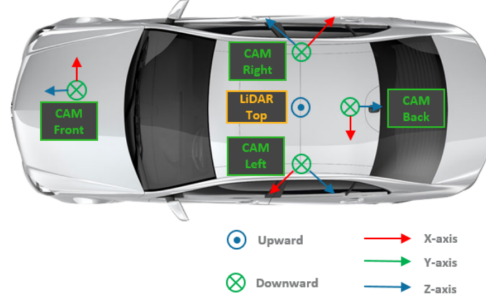


Figure 4.1: Sensor setup for each CAV in the OPV2V dataset [37].

OPV2V Dataset

OPV2V is the first large-scale open benchmark designed specifically for collaborative perception tasks among CAV. The dataset comprises over 11,464 frames across 70 interesting scenes, with more than 230,000 annotated 3D bounding boxes, captured in eight CARLA simulator towns and a digital replica of Culver City, Los Angeles. This unique setup enables the simulation of realistic and diverse traffic scenarios with varying levels of occlusion and vehicle density.

Table 4.2: Sensor specifications for each CAV in the OPV2V dataset [37].

Sensor Type	Specification
Cameras (×4)	800 × 600 pixels, 110° FOV, 360° combined
LiDAR	64 channels, 1.3M pts/s, (120 ± 0.2) m range, vertical FOV: −25° to 5°
GPS/IMU	20 mm positional error, 2° heading error

Each CAV in the OPV2V dataset is equipped with a comprehensive and well calibrated sensor suite designed to facilitate high-fidelity multi-modal perception. The sensor configuration is summarized in Table 4.2, and visually illustrated in Figure 4.1. Specifically, each vehicle is equipped with four RGB cameras arranged to provide a 360° FOV, a 64 channel LiDAR sensor capable of generating dense point clouds with high spatial resolution, and GPS/IMU modules to provide accurate localization and orientation data. The cameras operate at a resolution of 800×600 pixels with a 110° horizontal FOV each, collectively enabling full surround vision. The LiDAR sensor captures up to 1.3 million points per second with a maximum effective range of approximately 120 meters and a vertical FOV spanning from -25° to 5°, sufficient to detect objects in both close and distant proximity. The GPS/IMU system provides positional accuracy within 20 mm and orientation precision up to 2°, ensuring reliable ego-vehicle pose estimation and synchronization across

CAV. This sensor setup recorded at 10 Hz, allows the OPV2V dataset to simulate realistic collaborative perception scenarios, capturing diverse viewpoints and enabling effective fusion strategies for tasks such as object detection and BEV segmentation.

Table 4.3: Scenario types, distribution and assigned difficulty levels in the OPV2V dataset [37].

Road Type	Percentage (%)	Difficulty Level
4-way Intersection	24.5	++
T Intersection	24.1	++
Straight Segment	20.7	+
Curvy Segment	23.3	++
Midblock	4.7	+
Entrance Ramp	2.7	+++

Additionally, the OPV2V dataset covers a wide range of realistic driving scenarios, categorized into six road types. Table 4.3 presents the distribution of these scenarios along with their assigned difficulty levels. The percentage column reflects the proportion of each road type across the full dataset, providing insights into how frequently each scenario appears. Difficulty levels are qualitatively estimated based on factors such as traffic complexity, vehicle maneuvering space, and occlusion likelihood. Notably, entrance ramps, although representing only 2.7% of the data, are considered highly challenging due to rapid merging behavior and high-speed interactions, hence the ‘+++’ rating. In contrast, straight segments, which are more prevalent and simpler in nature, are labeled with a single ‘+’.

Implementation Details

The dataset is partitioned into training, validation, and test sets, comprising 6764, 1981 and 2719 frames, respectively. The test set is further divided into two subsets to evaluate performance across diverse environments: (i) the Default CARLA Towns and (ii) the Culver City Digital Town, a high-fidelity digital replica of a real-world urban environment. Each frame contains between two and seven CAVs. The ego vehicle in each frame is defined as the CAV with the lowest identifier among all active participants. Detection performance is assessed within a spatial region centered around the ego vehicle, specifically covering a range of $[-140.8, 140.8]$ meters along the longitudinal axis (x) and $[-40, 40]$ meters along the lateral axis (y), thereby constraining evaluation to the ego vehicle’s operationally relevant surroundings. Additionally, only messages transmitted by CAVs located within a 70 meter radius of the ego vehicle are considered, aligning with the typical communication range of V2X systems. For the input data of the model, min max normalization is applied to the FPV representation, ensuring that all features are scaled to a range of $[0, 1]$. The augmentations employed are the ones explained in Section 3.4. Performance evaluation for the overall detected cars is conducted using AP and F1 score metrics, each calculated at IoU thresholds of 0.5 and 0.7.

Model Optimization

In this subsection, the optimization process of the developed architecture is presented, highlighting the hyperparameter configurations and training setup across different model variants. This analysis serves to validate the effectiveness of the proposed collaborative perception framework. To ensure a consistent and controlled evaluation environment, all models were trained using the same set of hyperparameters.

Table 4.4: Hyperparameter Specifications for the OPV2V dataset.

Hyperparameter	Value
General	
Input Shape	$640 \times 640 \times 3$
Seed	1
Batch Size	8
Number of Epochs	30
Learning Rate	1×10^{-3}
Optimizer	AdamW
Weight Decay	1×10^{-4}
Confidence Threshold	0.25
IoU Threshold	0.45
Number of Classes	1
YOLOV12-X-Specific	
Class Loss Weight (λ_{cls})	0.50
Bbox Loss Weight (λ_{box})	7.50
DFL Loss Weight (λ_{dfl})	1.50
RT-DETR-X-Specific	
Number of Object Queries	300
Class Loss Weight (λ_{class})	1.00
α Loss Weight	0.75
γ Loss Weight	2.00
Bbox Loss Weight (λ_{bbox})	5.00
GIoU Loss Weight (λ_{giou})	2.00
Projection-Specific	
Meter to pixel in x (s_x)	2.273
Meter to pixel in y (s_y)	8
Meter to pixel in z (s_z)	160
Offset in x (L_1)	-140.8
Offset in y (W_1)	-40
Offset in z (H_1)	-3

Table 4.4 outlines the core training parameters, detector-specific and projection-specific configurations applied to both YOLOV12-X and RT-DETR-X variants. The AdamW [56] optimizer was selected for its efficient convergence behavior, effective regularization via decoupled weight decay and strong generalization performance in DL applications. Throughout training, both in single-agent and collaborative frameworks, AdamW is used in conjunction with a linear learning rate warm-up strategy applied over the first three epochs. This prevents abrupt weight updates during the early stages of training and facilitates stable convergence. After the warm-up phase, a Reduce on Plateau learning rate scheduler is adopted. This scheduler dynamically reduces the learning rate when the validation performance stagnates, helping the model escape local minima and promoting continued performance gains. A fixed random seed was employed across all ex-

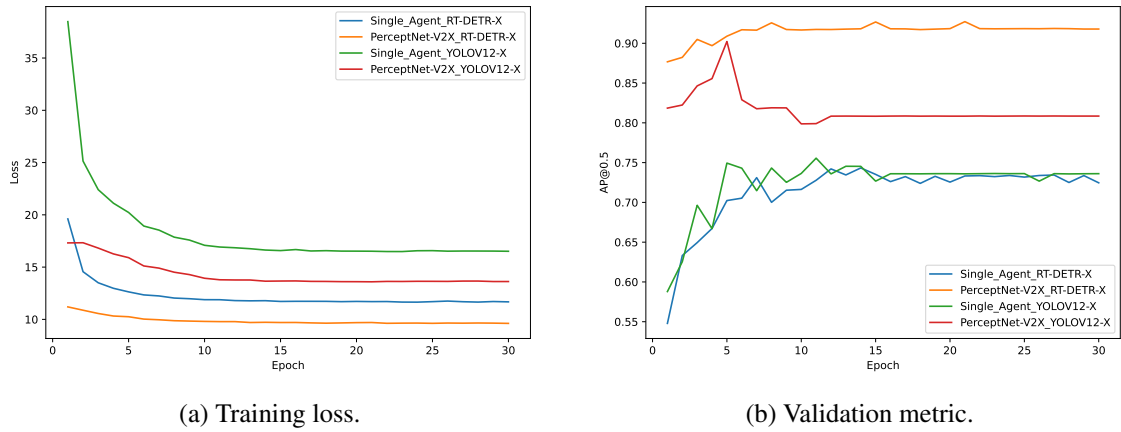


Figure 4.2: Training results in the OPV2V dataset.

periments to ensure reproducibility and enable consistent, comparable evaluation of the different model variants.

Figure 4.2 displays the training loss and validation performance over 30 epochs for the four evaluated frameworks on the OPV2V dataset. In Figure 4.2a, all models show a steady decrease in training loss, suggesting effective convergence across architectures. The collaborative models, PerceptNet-V2X, exhibit consistently lower training loss than their single-agent counterparts, due to the initialization with pretrained weights from the corresponding single-agent configurations. This weight reuse accelerates convergence and stabilizes the learning process in the collaborative setting. Contrarily, the single-agent variants are initialized with the pretrained weights from COCO [54] dataset. RT-DETR-X-based models demonstrate a significantly lower loss than those based on YOLOV12-X. This is attributed to differences in architectural design and training losses. Figure 4.2b illustrates the validation performance, AP@0.5, computed on the entire validation set at each epoch. The RT-DETR-X variant of PerceptNet-V2X achieves the highest AP@0.5 across all epochs, showing superior detection capabilities on the validation set. Additionally, both PerceptNet-V2X models outperform their single-agent versions, reaffirming the effectiveness of collaborative perception.

Quantitative Results

This subsection presents the quantitative evaluation of the proposed PerceptNet-V2X framework on the OPV2V. The results are reported using standard IoU thresholds of 0.5 and 0.7. Performance is assessed using both mAP and F1 score, which together offer a more comprehensive picture of detection accuracy and balance between precision and recall.

Table 4.5 presents a comparative evaluation of detection performance on the OPV2V dataset, focusing on the single-agent baselines and the proposed PerceptNet-V2X with the FPV representation using two different object detectors: YOLOV12-X and RT-DETR-X. In the Default Carla Town setting, PerceptNet-V2X significantly improves detection performance over its single-agent counterparts. With the RT-DETR-X baseline, the collaborative model achieves an AP@0.5 of

Table 4.5: Test results on the OPV2V dataset.

Model	Baseline	Representation	Default Carla Towns		Culver City Digital Town		Time (ms)	Params (M)	Size (MB)
			AP@0.5 / F1@0.5	AP@0.7 / F1@0.7	AP@0.5 / F1@0.5	AP@0.7 / F1@0.7			
Single-Agent	YOLOV12-X	FPV	80.96 / 85.57	72.73 / 74.90	72.98 / 78.95	59.61 / 65.00	154.7	72.7	289.3
Single-Agent	RT-DETR-X	FPV	85.96 / 78.09	66.93 / 60.19	71.99 / 72.73	55.23 / 51.70	230.99	138.39	284.6
PerceptNet-V2X	YOLOV12-X	FPV	92.86 / 94.36	88.43 / 88.04	86.03 / 88.61	75.34 / 77.18	221.35	72.7	289.3
PerceptNet-V2X	RT-DETR-X	FPV	95.63 / 90.76	88.91 / 81.14	87.55 / 85.55	71.70 / 69.43	312.63	138.39	284.6
PerceptNet-V2X	YOLOV12-X	BEV	90.38 / 91.47	85.71 / 83.99	81.01 / 85.46	68.61 / 72.81	68.24	68.67	275.2
PerceptNet-V2X	RT-DETR-X	BEV	94.60 / 90.58	86.67 / 81.00	86.54 / 85.50	68.76 / 68.18	141.48	134.86	270.4

AP@X: AP at an IoU threshold of X.

F1@X: F1 score at an IoU threshold of X.

Time (ms): Average time to process a single sample on a RTX 4050 Laptop GPU.

Params (M): Number of trainable parameters in millions.

Size (MB): Model size in megabytes.

95.63%, representing an absolute improvement of 9.67 percentage points over the single-agent counterpart. The corresponding F1@0.5 score shows an improvement of 12.67 percentage points. Similarly, with the YOLOV12-X baseline, PerceptNet-V2X achieves an AP@0.5 of 92.86%, reflecting an improvement of 11.9 percentage points over the single-agent model. The F1@0.5 score also improves by 8.79 percentage points. In the Culver City Digital Town scenario, which presents more complex urban layouts and increased occlusions, the benefits of collaborative perception become even more pronounced. Using the RT-DETR-X object detector, PerceptNet-V2X surpasses the single-agent model by 15.56 percentage points in AP@0.5, along with an F1@0.5 score gain of 12.82 percentage points. With the YOLOV12-X object detector, AP@0.5 improves by 13.05 percentage points, while F1@0.5 score increases by 9.66 percentage points. For higher IoU thresholds, the improvements are even more pronounced. These results demonstrate that the advantages of collaborative perception are amplified in complex, dynamic environments where individual agents face visibility limitations. Moreover, a comparison between the two detectors reveals that while RT-DETR-X generally achieves the highest AP performance, YOLOV12-X consistently delivers higher F1 scores across both environments when used within PerceptNet-V2X. This indicates that RT-DETR-X may be better suited for applications requiring superior detection accuracy, whereas YOLOV12-X offers more balanced precision-recall performance.

From a computational perspective, while the collaborative perception approach delivers substantial performance improvements, these gains come at the expense of increased inference time while maintaining identical model complexity. For YOLOV12-X, the inference time increases from 154.7 ms in single-agent mode to 221.35 ms in the collaborative PerceptNet-V2X configuration, representing a 43.1% increase. Similarly, RT-DETR-X shows an inference time increase from 230.99 ms to 312.63 ms, corresponding to a 35.3% overhead. Since the fusion mechanism in PerceptNet-V2X operates without additional trainable parameters, both the parameter count and model size remain unchanged between single-agent and collaborative configurations. YOLOV12-X maintains 72.7M parameters and 289.3 MB model size, while RT-DETR-X preserves 138.39M parameters and 284.6 MB model size. This demonstrates that the collaborative perception benefits come purely from information sharing and fusion processes rather than increased model capacity, making the approach particularly attractive for deployment scenarios where model size constraints are critical. Additionally, YOLOV12-X offers a more lightweight architecture with 72.7M param-

eters compared to RT-DETR-X’s 138.39M parameters, while maintaining comparable model sizes (289.3 MB vs 284.6 MB), making it more suitable for constrained environments.

Table 4.5 also includes a comparison with traditional BEV representation, which is commonly used in collaborative perception frameworks. This evaluation aims to assess the impact of the proposed FPV representation compared to the traditional BEV approach within the PerceptNet-V2X framework. To ensure a fair comparison, two variants of PerceptNet-V2X were trained: one using only the standard BEV representation and the other employing the novel FPV representation. Specifically, in the BEV collaborative configuration of PerceptNet-V2X, the Vertical views from the FPV Block, the Sparse Encoder-Decoder and the BEV Reprojection Block are removed, utilizing only the raw BEV features from the original model as described in Section 3.3. In contrast, the FPV PerceptNet-V2X employs the full implementation presented in Section 3.1, incorporating all architectural components. All other model components and training settings were kept identical across both variants.

The results demonstrate that the FPV representation consistently outperforms the traditional BEV approach across both environments and detectors. In the Default Carla Town scenario, the FPV variant with RT-DETR-X achieves an AP@0.5 of 95.63%, surpassing its BEV counterpart by 1.03 percentage points. Similarly, the YOLOV12-X configuration shows more substantial improvements, with FPV achieving 92.86% AP@0.5 compared to 90.38% for BEV, representing a 2.48 percentage points gain. The F1@0.5 scores follow a similar pattern, with YOLOV12-X showing the most pronounced improvement from 91.47% to 94.36%, a 2.89 percentage points increase. In the more challenging Culver City Digital Town environment, the advantages of FPV become more pronounced. The RT-DETR-X configuration achieves an AP@0.5 of 87.55% with FPV compared to 86.54% with BEV, while YOLOV12-X shows a larger improvement from 81.01% to 86.03%, representing a 5.02 percentage points increase. The F1@0.5 improvements are even more significant, with YOLOV12-X increasing from 85.46% to 88.61%, leading to a 3.15 percentage points improvement. These results demonstrate that the FPV representation offers significant advantages over traditional BEV approaches in complex urban environments characterized by frequent occlusions and dense multi-agent interactions. Specifically, FPV effectively addresses the inherent data sparsity challenges that affect BEV representations, where critical spatial information is often lost or inadequately captured due to the top-down perspective’s limitations in complex 3D scenes.

However, these performance improvements come with substantial computational trade-offs. The FPV representation introduces significant overhead due to the additional Sparse Encoder-Decoder module and the 3D Feature Lifting process. For YOLOV12-X, the inference time increases from 68.24 ms with BEV to 221.35 ms with FPV. Similarly, RT-DETR-X shows an increase from 141.48 ms to 312.63 ms. The additional architectural complexity also results in increased model parameters, with YOLOV12-X growing from 68.67M to 72.7M parameters and RT-DETR-X increasing from 134.86M to 138.39M parameters. Model sizes also increase correspondingly, from 275.2 MB to 289.3 MB for YOLOV12-X and from 270.4 MB to 284.6 MB for RT-DETR-X. The choice between FPV and BEV representations ultimately depends on the

specific application requirements, with FPV offering superior accuracy at the cost of increased computational complexity, while BEV provides a more efficient alternative with acceptable performance for constrained scenarios.

To further validate the robustness of the proposed collaborative perception architecture, Figures 4.3 and 4.4 illustrate stress tests conducted on the FPV-based PerceptNet-V2X model. These tests simulate localization noise by introducing perturbations to the positions and orientation, explained in Section 3.2, of each CAV in the world coordinate frame before computing their transformation matrices relative to the ego vehicle. Accurate transformations are essential for effective feature fusion and are typically derived from each vehicle’s localization system, which is inherently subject to positional and orientation errors. In this simulation, Gaussian noise is added to both the position (x, y, z) and heading (ϕ, θ, ψ) components of each CAV’s pose in the world frame. Specifically, zero-mean normal distributions are used with standard deviations ranging from 0 to 0.5 meters for the positional components, and from 0 to 1 degree for the orientation components. These perturbations reflect realistic localization inaccuracies commonly encountered in AD scenarios.

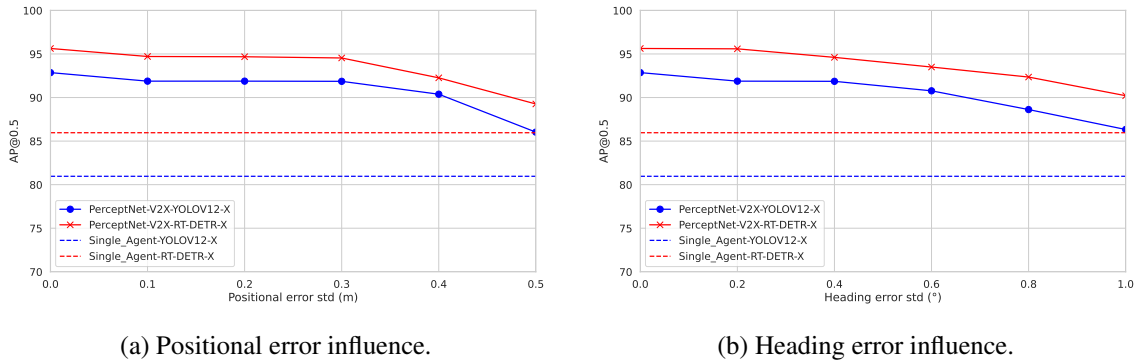


Figure 4.3: Impact of localization errors in the Default Carla Towns.

Figure 4.3 illustrates the impact of localization errors in the Default Carla Town setting. Subfigure 4.3a shows that the performance of PerceptNet-V2X begins to decline under positional noise, with noticeable degradation occurring at a standard deviation of approximately 0.3 meters for both the RT-DETR-X and YOLOV12-X detectors. Despite this, the models consistently outperform their respective single-agent baselines, even under the highest tested noise level of 0.5 meters. Subfigure 4.3b depicts the influence of heading noise, where a gradual drop in performance is observed beginning at around 0.6 degrees of standard deviation. Nonetheless, even at 1 degree of orientation noise, both collaborative variants maintain an advantage over their single-agent counterparts.

Moreover, Figure 4.4 presents the corresponding results for the Culver City Digital Town environment. In subfigure 4.4a, both collaborative models demonstrate robustness to positional noise up to 0.2 meters, beyond which a decline in performance becomes evident. However, similar to the Default setting, their performance at 0.5 meters still remains above that of the single-agent baselines. Subfigure 4.4b shows PerceptNet-V2X robustness to heading noise in the Culver City

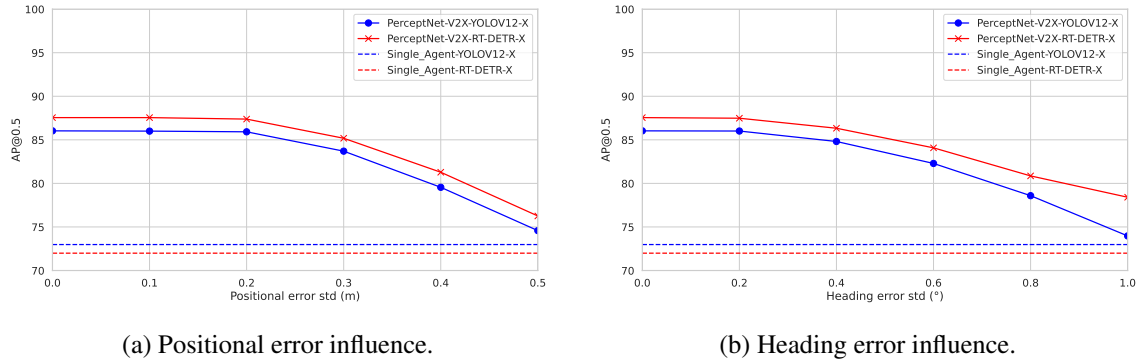


Figure 4.4: Impact of localization errors in the Culver City Digital Town.

Digital Town environment, with performance degradation starting at approximately 0.4 degrees of standard deviation. Despite this, both collaborative models sustain superior precision relative to their single-agent counterparts, reaffirming the robustness of PerceptNet-V2X to both positional and orientation uncertainties in more complex urban layouts. In conclusion, these results indicate that the proposed collaborative perception system demonstrates a degree of resilience to realistic localization noise.

Comparison with State-of-the-Art Architectures

Table 4.6: Comparison of PerceptNet-V2X with state-of-the-art object detection models in the OPV2V dataset.

Model	Default CARLA Towns AP@0.5	Culver City Digital Town AP@0.5
F-Cooper [35]	86.5	77.0
V2V-Net [36]	88.1	86.8
AttFusion [37]	90.8	85.4
CoMamba [41]	91.9	87.4
V2X-ViT [38]	94.4	85.4
MKD-Cooper [40]	96.5	89.5
CoAlign [39]	96.6	N/A
CollaMamba [42]	97.5	N/A
COPERA [43] (*)	97.85	93.91
PerceptNet-V2X (*)	95.63	87.55

AP@0.5: AP at an IoU threshold of 0.5.

N/A: Not available.

(*): 2D Object Detector.

Table 4.6 compares PerceptNet-V2X to leading collaborative perception methods on the OPV2V dataset. Using RT-DETR-X, PerceptNet-V2X achieves 95.63% AP@0.5 in Default CARLA Towns and 87.55% in Culver City Digital Town, placing it among the top performing detection models. While COPERA achieves the highest reported scores, PerceptNet-V2X outperforms most prior methods, including V2X-ViT, CoMamba, AttFusion, V2V-Net and F-Cooper, especially in the more complex Culver City scenario. This highlights its strong generalization and robustness in dynamic and dense environments. Overall, PerceptNet-V2X offers a competitive alternative for

collaborative perception, demonstrating the effectiveness of structured 2D projections in simulated AD settings.

Qualitative Results

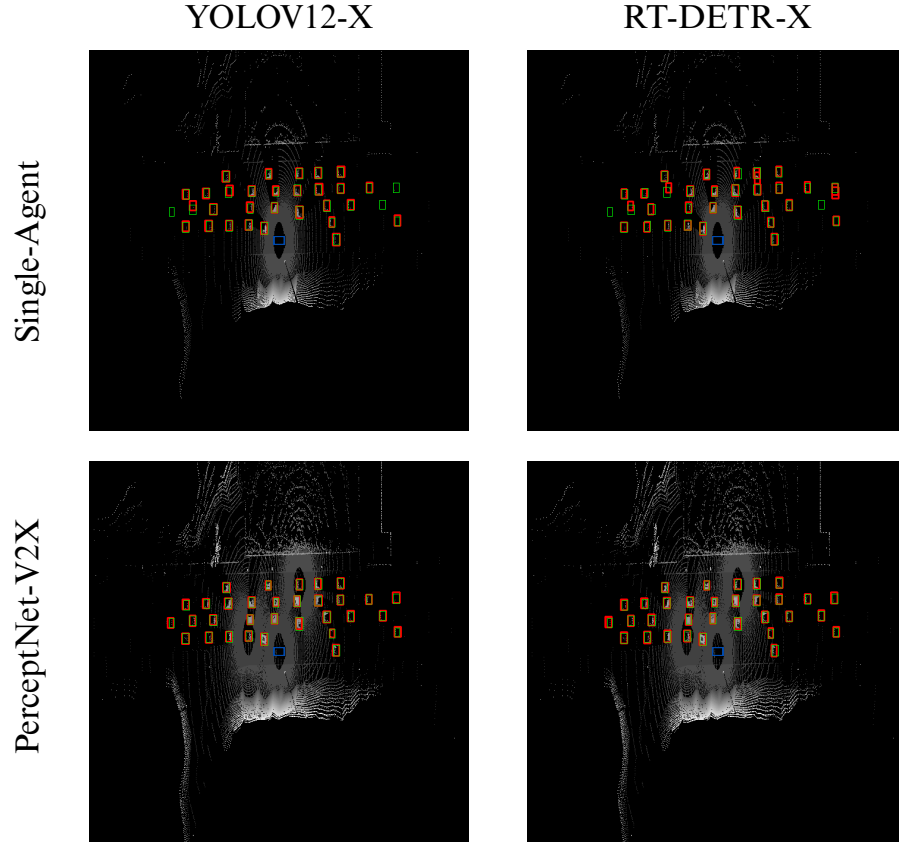


Figure 4.5: OPV2V Default Carla Towns dataset qualitative results. Green boxes indicate ground-truth bounding boxes, while red boxes represent car model predictions. The ego vehicle is identified with a blue bounding box.

Figure 4.5 provides a comprehensive qualitative comparison between PerceptNet-V2X (bottom row) and its single-agent counterpart (top row), using a representative sample from the OPV2V Default Carla Towns dataset. The visualization employs a standardized color scheme where green bounding boxes denote ground-truth annotations, red bounding boxes represent model predictions and the blue bounding box identifies the ego vehicle. The collaborative perception framework demonstrates substantial improvements across multiple dimensions: enhanced detection coverage, increased spatial accuracy of bounding box localization and superior performance in identifying vehicles under challenging visibility conditions. The most significant improvements are observed for vehicles that are partially occluded or positioned outside the ego vehicle’s direct FOV, which represent critical scenarios where single-agent perception typically fails due to limited sensor coverage. Under the collaborative setup, these previously undetected or poorly localized vehicles are consistently identified with high confidence and precise spatial boundaries. This enhancement

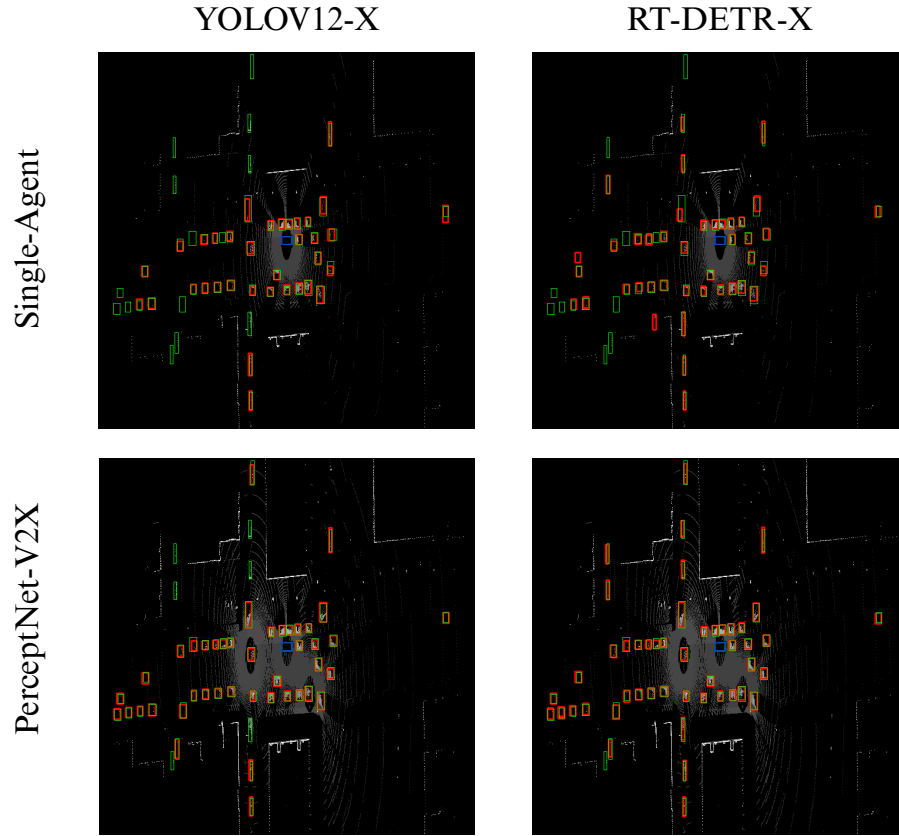


Figure 4.6: OPV2V Digital Culver City dataset qualitative results. Green boxes indicate ground-truth bounding boxes, while red boxes represent car model predictions. The ego vehicle is identified with a blue bounding box.

directly validates the effectiveness of the proposed V2X fusion strategy in aggregating complementary viewpoints to overcome individual agent limitations.

Figure 4.6 extends this analysis to the more challenging OPV2V Digital Culver City dataset, which features dense traffic patterns and complex multi-agent urban interactions. The results demonstrate consistent collaborative advantages even under these complexity conditions, with particularly pronounced improvements in heavily occluded scenarios. The comparative analysis between detection architectures reveals distinct performance characteristics: RT-DETR-X exhibits superior detection accuracy and robustness in collaborative settings, particularly excelling in highly occluded environments, while YOLOV12-X demonstrates more conservative prediction behavior with reduced false positive rates, resulting in higher overall F1 scores. These qualitative observations corroborate the quantitative analysis and highlight the complementary nature of different detection architectures within collaborative perception frameworks.

4.5 Model Evaluation in real-world environments

To validate the performance of the proposed collaborative perception model in a real-world setting, the V2X-Real [52] dataset is utilized, an introduced benchmark specifically designed for evaluating collaborative perception systems in AD scenarios.

V2X-Real Dataset

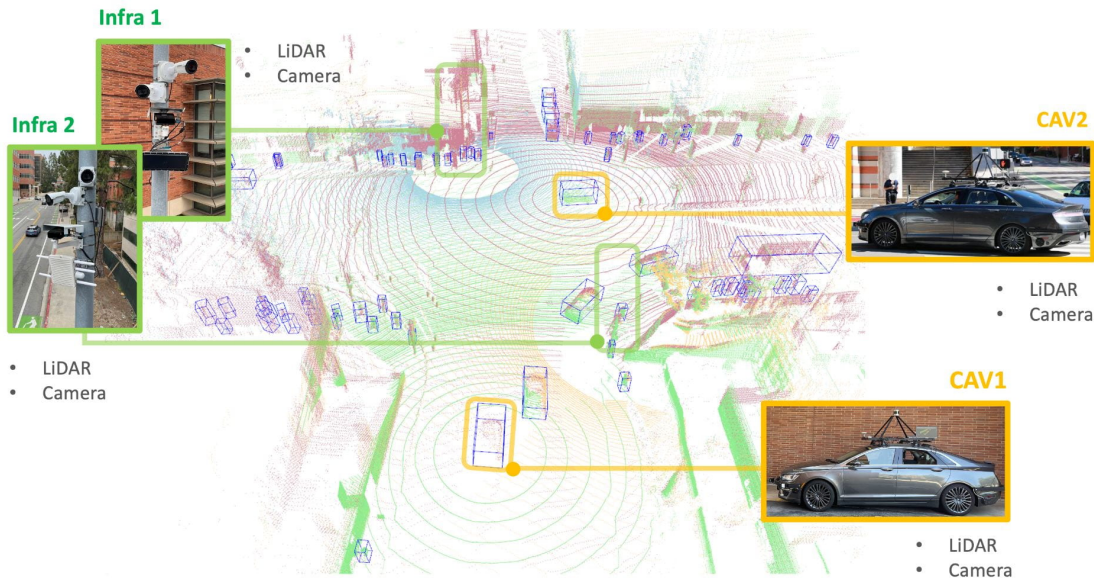


Figure 4.7: V2X-Real smart intersection and data acquisition example [52].

The V2X-Real dataset is among the first to offer large-scale V2X perception data collected from real CAVs and intelligent infrastructure units deployed in dense urban environments. It comprises around 33,000 LiDAR frames and 171,000 camera images, with over 1.2 million annotated 3D bounding boxes covering 10 object categories. These annotations enable rigorous training and evaluation for perception tasks in challenging real-world conditions. The data was collected over three months, amounting to more than 75 hours of driving logs. It covers a wide spectrum of traffic interactions, including 50 intersection scenarios (e.g., left/right turns, signal waits) and 10 corridor-based V2V scenarios (e.g., overtaking, car following). The data captures diverse illumination and weather conditions, providing a robust benchmark for studying collaborative perception systems. To support different collaboration paradigms in V2X research, the dataset is divided into four specialized subsets:

- **Vehicle Centric (VC):** One of the AVs is designated as the ego agent, while both infrastructure units and other vehicles serve as collaborators. This configuration simulates a common V2X use case where the vehicle integrates information from multiple sources to enhance its local perception.

- **Infrastructure Centric (IC):** In this setting, the infrastructure node acts as the ego agent, aggregating sensor data from nearby vehicles and other infrastructure units. This variant is intended for studies focused on traffic monitoring or smart city infrastructure.
- **Vehicle-to-Vehicle (V2V):** This subset includes only collaborations between vehicles. Additional V2V-specific data, such as highway or corridor scenarios, expands the diversity beyond intersections.
- **Infrastructure to Infrastructure (I2I):** This configuration isolates cooperative perception between infrastructure units, an important setting for traffic systems that do not rely on CAVs.

In this work, only the VC subset is used, as it most closely reflects practical deployment scenarios for AD. The VC setting allows direct evaluation of how collaboration with infrastructure and nearby vehicles can enhance a vehicle’s perception capability, an essential aspect of real-world V2X applications.

Table 4.7: Key Sensor Specifications in V2X-Real Dataset [52].

Agent	Sensor	Sensor Model	Details
Infrastructure	LiDAR	Ouster OS1-128/64 ($\times 1$)	128/64 beams, 10Hz capture frequency, 360° horizontal FOV, -16.6 to $+16.6$ vertical FOV (64), -22.5 to $+22.5$ (128), range up to 40m
	Camera	Axis-P14555 ($\times 2$)	RGB, 30Hz, 1920×1080 resolution, 76° horizontal FOV, 48° vertical FOV, JPEG compressed
	GPS	Garmin 18x LVC ($\times 1$)	5Hz update rate
Vehicle	LiDAR	RoboSense Ruby Plus ($\times 1$)	128 beams, 10Hz, 360° horizontal FOV, -25 to $+15$ vertical FOV, range up to 200m
	Camera	ZED 2i ($\times 4$)	Stereo RGBD, 10Hz, 1920×1080 resolution, 120° horizontal FOV, JPEG compressed
	GPS/IMU	Gongji GPS System ($\times 1$)	1000Hz update rate, double precision

The sensor configuration used during data collection consists of two CAV and two infrastructure nodes as illustrated in Figure 4.7. Each vehicle is outfitted with a high-resolution 128 beam LiDAR sensor, four ZED 2i stereo cameras positioned to achieve full 360° coverage and a GPS/IMU system for precise localization and motion estimation. Infrastructure nodes are equipped with either 64 or 128 beam LiDAR sensors, high-resolution network cameras and synchronized GPS modules. A detailed breakdown of the sensor specifications is shown in Table 4.7.

Additionally, Figure 4.8 presents the number of annotations per object class on a logarithmic scale, providing insight into the distribution and frequency of different categories within the dataset. On average, each scene contains approximately 36 annotated objects, demonstrating a substantial object density that reflects realistic and busy urban traffic conditions. The minimum and maximum number of objects per scene are 9 and 115, respectively, indicating considerable variability in scene complexity. The dataset is annotated across 10 distinct classes, representing a wide range of road users and obstacles encountered in urban driving environments. V2X-Real includes a rich representation of vulnerable road users such as pedestrians, scooters, motorcycles and bicycles, in addition to more common vehicle categories. This diversity is crucial for training and evaluating perception systems that must accurately detect and classify a broad spectrum of objects, especially those that are of reduced dimension or less predictable in their movements. The

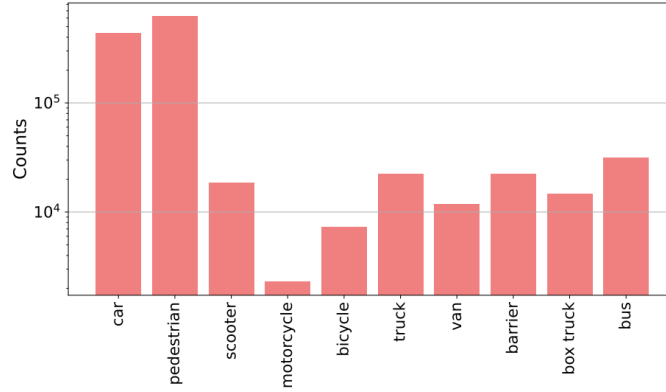


Figure 4.8: Number of annotations per class in log scale in V2X-Real Dataset [52].

logarithmic scale used in the Figure 4.8 emphasizes the variation in annotation counts, highlighting that some classes, such as cars and pedestrians, are much more frequently encountered, while others occur less often but remain essential for comprehensive perception. This class imbalance is a realistic characteristic of traffic data and poses additional challenges for detection algorithms, making V2X-Real a valuable benchmark for robust multi-class object detection.

Implementation Details

To facilitate more effective model training and evaluation, the original 10 object classes provided in the dataset are grouped into three super-classes: pedestrian, car and truck, as proposed by the dataset authors [52]. The dataset is split into training, validation, and test sets, containing 23379, 2770 and 6850 frames, respectively. Each frame involves information sharing between two CAV and two infrastructure nodes, each providing sensor data. Within the VC collaboration setting, the ego vehicle is defined as the vehicle with the lowest identifier among all active participants in the scene. Detection evaluation is restricted to a spatial region centered on the ego vehicle, covering $[-100, 100]$ meters longitudinally (x) and $[-40, 40]$ meters laterally (y). This region represents the operationally relevant environment for the ego vehicle. Furthermore, only sensor messages from CAV or infrastructure nodes located within a 50 meter radius of the ego vehicle are considered, corresponding to the typical communication range of V2X systems. For the input data of the model, min max normalization is applied to the FPV representation, ensuring that all features are scaled to a range of $[0, 1]$. The augmentations employed are the ones explained in Section 3.4. The dataset encompasses a broad spectrum of object sizes, ranging from pedestrians with bounding boxes of 0.5 meters to large buses exceeding 20 meters in length, which makes precise estimation of object poses and dimensions particularly challenging. To accommodate this variability and ensure a fair evaluation, the performance metrics use reduced IoU thresholds of 0.3 and 0.5 for AP and F1 scores calculations. This adjustment allows the evaluation to more accurately capture the model's perception capabilities across diverse and complex urban environments.

Model Optimization

Table 4.8: Hyperparameter Specifications for the V2X-Real dataset.

Hyperparameter	Value
General	
Input Shape	$640 \times 640 \times 3$
Seed	1
Batch Size	8
Number of Epochs	30
Learning Rate	1×10^{-3}
Optimizer	AdamW
Weight Decay	1×10^{-4}
Confidence Threshold	0.25
IoU Threshold	0.45
Number of Classes	3
YOLOV12-X-Specific	
Class Loss Weight (λ_{cls})	0.50
Bbox Loss Weight (λ_{bbox})	7.50
DFL Loss Weight (λ_{dfl})	1.50
RT-DETR-X-Specific	
Number of Object Queries	300
Class Loss Weight (λ_{class})	1.00
α Loss Weight	0.75
γ Loss Weight	2.00
Bbox Loss Weight (λ_{bbox})	5.00
GIoU Loss Weight (λ_{giou})	2.00
Projection-Specific	
Meter to pixel in x (s_x)	3.2
Meter to pixel in y (s_y)	8
Meter to pixel in z (s_z)	32
Offset in x (L_1)	-100
Offset in y (W_1)	-40
Offset in z (H_1)	-10

This subsection outlines the optimization process of the proposed architecture on real-world data, detailing the hyperparameter configuration and training setup across different model variants. The hyperparameters, as listed in Table 4.8, are consistent with those employed for the OPV2V dataset, with the sole exception being the number of object classes. For the V2X-Real dataset, the model was configured to detect three distinct object categories: car, pedestrian and truck, whereas only a single class is considered in the OPV2V dataset setup.

Figure 4.9 presents the training results on the V2X-Real dataset, utilizing the hyperparameter configurations outlined in Table 4.8. The subplot in Figure 4.9a illustrates the training loss across all model variants. Similar to the training results observed on the OPV2V dataset, the training loss steadily decreases throughout the epochs, demonstrating the models capacity to effectively learn from real-world data. Notably, the PerceptNet-V2X models consistently exhibit lower training losses, highlighting the benefit of initializing from pretrained single-agent weights.

The subplot in Figure 4.9b depicts validation performance using mAP@0.3. Consistent with the OPV2V results, PerceptNet-V2X integrated with RT-DETR-X achieves the highest validation performance, confirming its robustness under real-world V2X conditions. Both collaborative

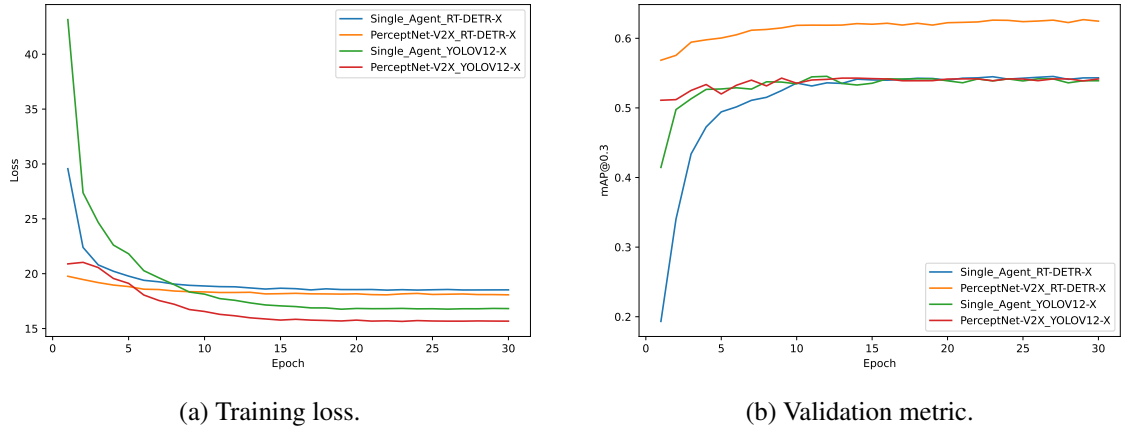


Figure 4.9: Training results in the V2X-Real dataset.

variants featuring YOLOV12-X and RT-DETR-X, surpass their respective single-agent baselines, highlighting the substantial performance gains enabled by multi-agent feature collaboration. It is particularly important to note that although the validation curves for the YOLOV12-X collaborative and single-agent models appear similar in absolute terms, this masks a critical distinction: the collaborative models are evaluated against scene-level ground truths, encompassing all agents within the evaluation range, whereas single-agent models are assessed solely using ego vehicle annotations. As such, the comparable validation scores actually represent improved detection performance in the collaborative case, as identifying more objects across a wider spatial context is inherently more challenging.

Quantitative Results

Table 4.9: Test Results in the V2X-Real Dataset under the VC setting.

Model	Baseline	mAP@0.3/F1@0.3	mAP@0.5/F1@0.5
Single-Agent	YOLOV12-X	44.66 / 36.77	44.29 / 35.93
Single-Agent	RT-DETR-X	47.8 / 48.23	42.46 / 37.97
PerceptNet-V2X	YOLOV12-X	50.26 / 39.84	49.92 / 38.9
PerceptNet-V2X	RT-DETR-X	61.55 / 55.24	53.72 / 41.83

AP@X: AP at an IoU threshold of X.

F1@X: F1 score at an IoU threshold of X.

Table 4.9 reports the detection performance on the V2X-Real dataset under the VC setting. For models based on YOLOV12-X, the single-agent setup achieves a mAP@0.3 of 44.66% and an F1@0.3 of 36.77%. Incorporating collaborative perception via PerceptNet-V2X achieves a notable improvement, reaching 50.26% mAP and 39.84% F1, an increase of 5.6 and 3.07 percentage points, respectively. With RT-DETR-X as the base detector, the gains are even more substantial. The single-agent version achieves a mAP@0.3 of 47.8% and F1@0.3 of 48.23%, while its collaborative PerceptNet-V2X variant improves these to 61.55% and 55.24%, corresponding to enhancements of 13.75 and 7.01 percentage points. These results clearly demonstrate the effectiveness

of collaborative feature fusion. In contrast to the OPV2V dataset, where YOLOV12-X outperformed RT-DETR-X in F1 score, the V2X-Real dataset reveals the opposite: RT-DETR-X models now surpass YOLOV12-X in F1 performance. This reversal highlights the stronger real-world generalization and robustness of RT-DETR-X. Overall, these findings confirm that the proposed collaborative perception framework significantly improves detection accuracy over single-agent baselines and that detectors like RT-DETR are especially well-suited for collaborative settings in complex real-world environments.

Comparison with State-of-the-Art Architectures

Table 4.10: Comparison of PerceptNet-V2X with state-of-the-art object detection models in the V2X-Real dataset.

Model	mAP@0.3	mAP@0.5
F-Cooper [35]	38.1	29.8
AttFusion [37]	42.5	33.8
V2X-ViT [38]	44.8	35.8
PerceptNet-V2X (*)	61.55	53.72

AP@X: AP at an IoU threshold of X.

mAP@X: mAP at an IoU threshold of X.

(*): 2D Object Detector

Following the comprehensive evaluation of PerceptNet-V2X across different object detectors and single-agent baselines, Table 4.10 presents a performance comparison against several state-of-the-art collaborative perception architectures on the V2X-Real dataset. Utilizing the RT-DETR-X object detector, PerceptNet-V2X achieves a mAP@0.3 of 61.55% and a mAP@0.5 of 53.72%, significantly outperforming existing methods such as F-Cooper, AttFusion and V2X-ViT by over 16 percentage points. This substantial improvement highlights the robustness of PerceptNet-V2X in complex real-world driving environments, particularly under challenging conditions such as dense traffic, dynamic actors, and frequent occlusions scenarios in which traditional methods often struggle.

The modest mAP scores when compared to OPV2V dataset AP results also reflect the inherent performance gap between simulated and real-world scenarios, a known challenge in AD research. Nevertheless, the introduction of PerceptNet-V2X, featuring the novel FPV representation delivers a notable improvement, marking a significant advancement toward narrowing this gap and enhancing real-world perception robustness.

Qualitative Results

Figure 4.10 illustrates qualitative results on a representative scene from the V2X-Real dataset, highlighting differences in detection performance across single-agent and collaborative models. The visualization employs a class-specific color scheme where green bounding boxes represent ground-truth annotations, while red, purple, and orange bounding boxes indicate predicted cars, pedestrians, and trucks, respectively. The ego vehicle is identified with a blue bounding box. This real-world scenario features dense urban clutter and multiple partially occluded objects, conditions

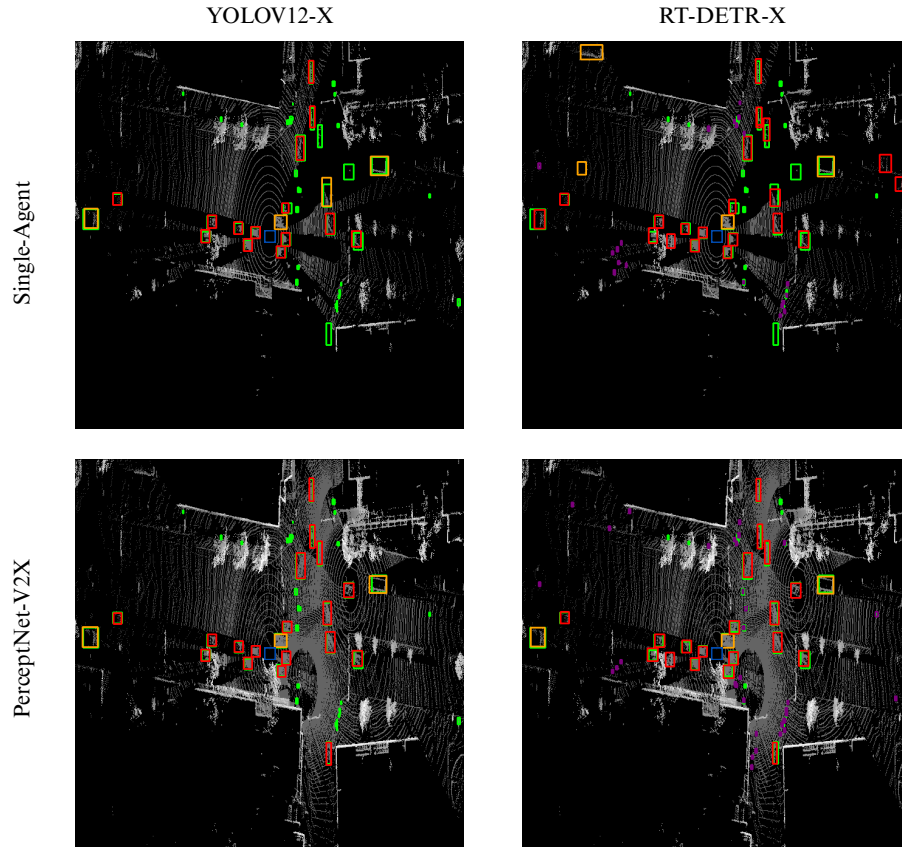


Figure 4.10: V2X-Real dataset qualitative results. Green boxes represent ground-truth bounding boxes, while red, purple, and orange boxes indicate predicted cars, pedestrians, and trucks, respectively. The ego vehicle is identified with a blue bounding box.

under which single-agent models typically struggle to maintain reliable detection performance. As demonstrated in the top-left and top-right panels of the figure, both YOLOV12-X and RT-DETR-X single-agent variants exhibit significant limitations in consistently detecting small and occluded objects. The sparse distribution of colored prediction boxes compared to green ground-truth annotations clearly illustrates these detection failures. In contrast, the bottom panels depicting collaborative PerceptNet-V2X predictions demonstrate substantially enhanced spatial coverage and object localization accuracy. PerceptNet-V2X, particularly when employing RT-DETR-X as the underlying detector, exhibits tighter and more spatially accurate bounding boxes, including more successful identification of pedestrians and partially occluded vehicles that remain undetected by single-agent baselines. These visual improvements directly corroborate the quantitative gains reported in Table 4.9, reinforcing the framework’s enhanced robustness under complex real-world conditions.

A detailed comparative analysis of the detection architectures integrated within PerceptNet-V2X reveals a consistent performance advantage for RT-DETR-X over YOLOV12-X, particularly evident in pedestrian detection scenarios. The qualitative results demonstrate that YOLO-based models frequently fail to identify multiple pedestrian instances, even when collaborative percep-

tion is employed, as evidenced by the absence of purple bounding boxes around clearly visible pedestrian ground-truth annotations. This systematic detection failure indicates that the detector has effectively learned to suppress the pedestrian class, a consequence of the limited visual footprint pedestrians occupy within the standard 640×640 input resolution. When pedestrian point clouds are projected from LiDAR data to the image plane, they are typically represented by sparse pixel clusters, significantly reducing their learnability during training. This limitation persists across both single-agent and collaborative settings, suggesting that the root cause lies not in insufficient contextual information or inadequate feature fusion mechanisms, but rather in the architectural sensitivity of YOLO models to low-resolution, sparsely represented object classes. To validate this hypothesis, additional experiments were conducted using an increased input resolution of 800×800 . These higher-resolution tests revealed substantial improvements in pedestrian detection for YOLOV12-X, as evidenced by increased bounding box coverage and enhanced spatial accuracy in the resulting outputs. This finding confirms that the observed limitation is not inherent to YOLO's core architecture, but is fundamentally tied to input resolution constraints and the density of point-level features for small or distant objects such as pedestrians.

Figure 4.10 therefore highlights two critical insights for collaborative perception in AD. First, the substantial advantage of collaborative frameworks in handling real-world visual complexity and occlusion scenarios. Second, the critical importance of carefully optimizing both object detector selection and input resolution parameters to ensure reliable detection of challenging object classes, particularly pedestrians, in safety-critical AD applications.

Chapter 5

Conclusions and Future Work

This dissertation has successfully explored and addressed the limitations of single-agent perception systems in AVs by introducing an intermediate collaborative perception framework. Challenges such as sensor occlusion, restricted FOV and environmental obstructions critically reduce situational awareness and decision-making capacity in real-world AD scenarios. To overcome these challenges, the proposed framework, PerceptNet-V2X, enables multi-agent collaboration via intermediate feature sharing, powered by the novel FPV representation. FPV is a new 2D encoding of 3D point cloud data that preserves key geometric information while enabling efficient and robust feature fusion across agents. This design significantly addresses the sparsity issues in conventional BEV representations and ensures integration with 2D object detectors such as RT-DETR and YOLO, demonstrating plug-and-play flexibility across different detection models.

The architecture was rigorously evaluated on two benchmark datasets: OPV2V, a large-scale synthetic dataset, and V2X-Real, a real-world dataset. Experimental results demonstrate consistent and significant improvements across both synthetic and real-world evaluation environments. On the synthetic OPV2V dataset, PerceptNet-V2X achieved an AP@0.5 of 95.63% and an F1@0.5 of 94.36%, representing substantial gains up to 5.02 percentage points in AP and 3.15 percentage points in F1 compared to the traditional BEV representation. Moreover, on the challenging V2X-Real dataset, the method reached a mAP@0.3 of 61.55%, representing a relative improvement exceeding 16 percentage points over prior state-of-the-art approaches. These results on the V2X-Real dataset significantly reduce the real-to-sim gap, demonstrating that the PerceptNet-V2X architecture effectively bridges the performance divide between synthetic training environments and real-world deployment scenarios. The performance improvements across both synthetic and real-world datasets validate the effectiveness of the proposed collaborative perception approach in addressing the fundamental challenges of single-agent systems and traditional BEV representations. The consistent gains demonstrate that the intermediate feature fusion strategy, combined with the FPV representation, successfully takes advantage of multi-agent collaboration to enhance situational awareness and detection accuracy in complex driving environments.

In conclusion, this work fulfills its goal of advancing collaborative perception for multi-agent AD systems. It presents a scalable and modular architecture that integrates intermediate features

from distributed agents, significantly improves detection accuracy and builds a strong foundation for future research in real-world, collaborative autonomous systems. The research contributes both theoretical insights through the novel FPV representation and practical solutions through the PerceptNet-V2X framework, advancing the state-of-the-art in collaborative AD perception.

Building on these findings, several avenues can further elevate the performance and practicality of collaborative perception systems in AVs:

- The 3D lifting process can be optimized through vectorized operations or lookup tables, potentially reducing computational overhead and enabling real-time performance on resource-constrained embedded systems commonly used in AVs.
- Investigating more effective methods for multi-agent feature fusion, potentially incorporating attention mechanisms, confidence weighting or graph-based representations. These approaches could better capture the relationships between agents and improve the quality of collaborative perception.
- Extending the framework to support prediction of object orientation, enabling improved object detection and localization capabilities. This would provide richer semantic understanding for more sophisticated AD decision-making.
- Developing adaptive communication protocols that dynamically regulate data exchange based on contextual factors such as occlusion severity, scene complexity and latency constraints. This would optimize the trade-off between perception accuracy and communication overhead in resource-constrained environments.
- Expanding the training and evaluation to include additional datasets that capture diverse environmental conditions, such as rain, fog or snow to improve robustness and generalization in adverse real-world scenarios. This would ensure reliable performance across the full spectrum of AD conditions.

These future research directions would further strengthen the proposed collaborative perception framework and accelerate its deployment in real-world AD systems, ultimately contributing to safer and more reliable transportation infrastructure.

References

- [1] L. R. Agostinho, *Visual odometry system using multi-modal information in autonomous driving scenarios*, Accessed: 2025-05-28, 2022. [Online]. Available: <https://hdl.handle.net/10216/142639>.
- [2] World Health Organization, *Road traffic injuries*, Accessed: 2025-05-31, 2025. [Online]. Available: https://www.who.int/health-topics/road-safety#tab=tab_1.
- [3] M. Maurer, J. Gerdes, B. Lenz, and H. Winner, *Autonomous Driving. Technical, Legal and Social Aspects*. May 2016, ISBN: 9783662488454. DOI: [10.1007/978-3-662-48847-8](https://doi.org/10.1007/978-3-662-48847-8).
- [4] Society of Automotive Vehicles, *Levels of driving automation*, Accessed: 2024-12-10, 2024. [Online]. Available: https://www.zf.com/master/media/magazine/2022/22_04/2022_04_29_6_levels_of_automated_driving_/sae-j3016-visual-chart_5.3.21.pdf.
- [5] Imagination, *What are the six levels of autonomous driving technology*, Accessed: 2024-12-10, 2024. [Online]. Available: <https://www.imaginationtech.com/future-of-automotive/when-will-autonomous-cars-be-available/what-are-the-levels-of-autonomy-in-self-driving-cars/#::~:~:text=The%20current%20state%20of%20play&text=The%20SAE%20and%20the%20ISO,set%20to%20become%20more%20important..>
- [6] VisionZero, *What is vision zero*, Accessed: 2024-12-10, 2024. [Online]. Available: <https://visionzeronetwork.org/about/what-is-vision-zero/>.
- [7] G. Jocher and J. Qiu, *Ultralytics yolo11*, version 11.0.0, 2024. [Online]. Available: <https://github.com/ultralytics/ultralytics>.
- [8] Y. Zhao et al., *Detrs beat yolos on real-time object detection*, 2023. arXiv: [2304.08069](https://arxiv.org/abs/2304.08069) [cs.CV].
- [9] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. DOI: [10.1109/5.726791](https://doi.org/10.1109/5.726791).
- [10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.
- [11] C. Szegedy et al., *Going deeper with convolutions*, 2014. arXiv: [1409.4842](https://arxiv.org/abs/1409.4842) [cs.CV]. [Online]. Available: <https://arxiv.org/abs/1409.4842>.
- [12] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, 2015. arXiv: [1409.1556](https://arxiv.org/abs/1409.1556) [cs.CV]. [Online]. Available: <https://arxiv.org/abs/1409.1556>.

- [13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.
- [14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation*, 2014. arXiv: [1311.2524 \[cs.CV\]](https://arxiv.org/abs/1311.2524). [Online]. Available: <https://arxiv.org/abs/1311.2524>.
- [15] R. Girshick, *Fast r-cnn*, 2015. arXiv: [1504.08083 \[cs.CV\]](https://arxiv.org/abs/1504.08083). [Online]. Available: <https://arxiv.org/abs/1504.08083>.
- [16] S. Ren, K. He, R. Girshick, and J. Sun, *Faster r-cnn: Towards real-time object detection with region proposal networks*, 2016. arXiv: [1506.01497 \[cs.CV\]](https://arxiv.org/abs/1506.01497). [Online]. Available: <https://arxiv.org/abs/1506.01497>.
- [17] J. Redmon, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
- [18] Y. Tian, Q. Ye, and D. Doermann, “Yolov12: Attention-centric real-time object detectors,” *arXiv preprint arXiv:2502.12524*, 2025.
- [19] Ultralytics, *Models supported by ultralytics*, Accessed: 2024-12-20, 2024. [Online]. Available: <https://docs.ultralytics.com/models/>.
- [20] A. Vaswani et al., *Attention is all you need*, 2023. arXiv: [1706.03762 \[cs.CL\]](https://arxiv.org/abs/1706.03762). [Online]. Available: <https://arxiv.org/abs/1706.03762>.
- [21] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, *End-to-end object detection with transformers*, 2020. arXiv: [2005.12872 \[cs.CV\]](https://arxiv.org/abs/2005.12872). [Online]. Available: <https://arxiv.org/abs/2005.12872>.
- [22] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 77–85. DOI: [10.1109/CVPR.2017.16](https://doi.org/10.1109/CVPR.2017.16).
- [23] S. Shi, X. Wang, and H. Li, “Pointnet: 3d object proposal generation and detection from point cloud,” Jun. 2019, pp. 770–779. DOI: [10.1109/CVPR.2019.00086](https://doi.org/10.1109/CVPR.2019.00086).
- [24] Y. Zhou and O. Tuzel, “Voxelnet: End-to-end learning for point cloud based 3d object detection,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4490–4499. DOI: [10.1109/CVPR.2018.00472](https://doi.org/10.1109/CVPR.2018.00472).
- [25] Y. Yan, Y. Mao, and B. Li, “Second: Sparsely embedded convolutional detection,” *Sensors*, vol. 18, no. 10, 2018, ISSN: 1424-8220. DOI: [10.3390/s18103337](https://doi.org/10.3390/s18103337). [Online]. Available: <https://www.mdpi.com/1424-8220/18/10/3337>.
- [26] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, “Pointpillars: Fast encoders for object detection from point clouds,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 689–12 697. DOI: [10.1109/CVPR.2019.01298](https://doi.org/10.1109/CVPR.2019.01298).
- [27] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, *Sensor and sensor fusion technology in autonomous vehicles: A review*, Feb. 2021. DOI: [10.20944/preprints202102.0459.v1](https://doi.org/10.20944/preprints202102.0459.v1).
- [28] X. Wang, Y. Jin, C. Li, Y. Cen, and Y. Li, “Vsl: View-aware sphere learning network for cross-view vehicle re-identification,” *International Journal of Intelligent Systems*, vol. 37, Mar. 2022. DOI: [10.1002/int.22857](https://doi.org/10.1002/int.22857).

- [29] H. Zhang, G. Luo, J. Li, and F.-Y. Wang, “C2fda: Coarse-to-fine domain adaptation for traffic object detection,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12 633–12 647, 2022. DOI: [10.1109/TITS.2021.3115823](https://doi.org/10.1109/TITS.2021.3115823).
- [30] H. Zhang, Y. Tian, K. Wang, W. Zhang, and F.-Y. Wang, “Mask ssd: An effective single-stage approach to object instance segmentation,” *IEEE Transactions on Image Processing*, vol. 29, pp. 2078–2093, 2020. DOI: [10.1109/TIP.2019.2947806](https://doi.org/10.1109/TIP.2019.2947806).
- [31] X. Wu, D. Sahoo, and S. C. Hoi, “Recent advances in deep learning for object detection,” *Neurocomputing*, vol. 396, pp. 39–64, 2020, ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2020.01.085>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231220301430>.
- [32] Y. Han, H. Zhang, H. Li, Y. Jin, C. Lang, and Y. Li, “Collaborative perception in autonomous driving: Methods, datasets, and challenges,” *IEEE Intelligent Transportation Systems Magazine*, vol. 15, no. 6, pp. 131–151, Nov. 2023, ISSN: 1941-1197. DOI: [10.1109/imits.2023.3298534](https://doi.org/10.1109/imits.2023.3298534). [Online]. Available: <http://dx.doi.org/10.1109/MITS.2023.3298534>.
- [33] Q. Chen, S. Tang, Q. Yang, and S. Fu, *Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds*, 2019. arXiv: [1905.05265](https://arxiv.org/abs/1905.05265) [cs.CV]. [Online]. Available: <https://arxiv.org/abs/1905.05265>.
- [34] E. Arnold, M. Dianati, R. de Temple, and S. Fallah, “Cooperative perception for 3d object detection in driving scenarios using infrastructure sensors,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 1852–1864, Mar. 2022, ISSN: 1558-0016. DOI: [10.1109/tits.2020.3028424](https://doi.org/10.1109/tits.2020.3028424). [Online]. Available: <http://dx.doi.org/10.1109/TITS.2020.3028424>.
- [35] Q. Chen, *F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds*, 2019. arXiv: [1909.06459](https://arxiv.org/abs/1909.06459) [cs.CV]. [Online]. Available: <https://arxiv.org/abs/1909.06459>.
- [36] T.-H. Wang et al., *V2vnet: Vehicle-to-vehicle communication for joint perception and prediction*, 2020. arXiv: [2008.07519](https://arxiv.org/abs/2008.07519) [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2008.07519>.
- [37] R. Xu, H. Xiang, X. Xia, X. Han, J. Li, and J. Ma, *Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication*, 2022. arXiv: [2109.07644](https://arxiv.org/abs/2109.07644) [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2109.07644>.
- [38] R. Xu, H. Xiang, Z. Tu, X. Xia, M.-H. Yang, and J. Ma, *V2x-vit: Vehicle-to-everything cooperative perception with vision transformer*, 2022. arXiv: [2203.10638](https://arxiv.org/abs/2203.10638) [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2203.10638>.
- [39] Y. Lu et al., “Robust collaborative 3d object detection in presence of pose errors,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 4812–4818.
- [40] Z. Li, H. Liang, H. Wang, M. Zhao, J. Wang, and X. Zheng, “Mkd-cooper: Cooperative 3d object detection for autonomous driving via multi-teacher knowledge distillation,” *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 1490–1500, 2024. DOI: [10.1109/TIV.2023.3310580](https://doi.org/10.1109/TIV.2023.3310580).
- [41] J. Li et al., *Comamba: Real-time cooperative perception unlocked with state space models*, 2024. arXiv: [2409.10699](https://arxiv.org/abs/2409.10699) [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2409.10699>.

- [42] Y. Li et al., *Collamamba: Efficient collaborative perception with cross-agent spatial-temporal state space model*, 2024. arXiv: 2409.07714 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2409.07714>.
- [43] C. Pereira, B. Costa, and A. M. Pinto, “Copera - collaborative perception for real-time object detection in autonomous driving,” *IEEE Transactions on Intelligent Transportation Systems*, 2025.
- [44] A. Dosovitskiy et al., *An image is worth 16x16 words: Transformers for image recognition at scale*, 2021. arXiv: 2010.11929 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2010.11929>.
- [45] L. Kong et al., *Rethinking range view representation for lidar segmentation*, 2023. arXiv: 2303.05367 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2303.05367>.
- [46] A. Barrera, C. Guindel, J. Beltrán, and F. García, *Birdnet+: End-to-end 3d object detection in lidar bird’s eye view*, 2020. arXiv: 2003.04188 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2003.04188>.
- [47] A. Bewley, P. Sun, T. Mensink, D. Anguelov, and C. Sminchisescu, *Range conditioned dilated convolutions for scale invariant 3d object detection*, 2021. arXiv: 2005.09927 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2005.09927>.
- [48] L. Fan, X. Xiong, F. Wang, N. Wang, and Z. Zhang, “Rangedet: In defense of range view for lidar-based 3d object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 2918–2927.
- [49] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, *Pointnet++: Deep hierarchical feature learning on point sets in a metric space*, 2017. arXiv: 1706.02413 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/1706.02413>.
- [50] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An open urban driving simulator,” in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.
- [51] R. Xu, Y. Guo, X. Han, X. Xia, H. Xiang, and J. Ma, “Opencda: An open cooperative driving automation framework integrated with co-simulation,” in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2021, pp. 1155–1162.
- [52] H. Xiang et al., “V2x-real: A large-scale dataset for vehicle-to-everything cooperative perception,” *arXiv preprint arXiv:2403.16034*, 2024.
- [53] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation*, 2015. arXiv: 1505.04597 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/1505.04597>.
- [54] T.-Y. Lin et al., *Microsoft coco: Common objects in context*, 2015. arXiv: 1405.0312 [cs.CV].
- [55] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *Int. J. Comput. Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010, ISSN: 0920-5691. DOI: 10.1007/s11263-009-0275-4. [Online]. Available: <https://doi.org/10.1007/s11263-009-0275-4>.
- [56] I. Loshchilov and F. Hutter, *Decoupled weight decay regularization*, 2019. arXiv: 1711.05101 [cs.LG]. [Online]. Available: <https://arxiv.org/abs/1711.05101>.

Appendix A

Additional Metrics

- Precision: Precision is the ratio of true positive detections to all positive detections made by the model. It reflects how accurate the positive predictions are:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (\text{A.1})$$

where TP is the number of True Positives and FP is the number of False Positives.

- Recall: Recall, also known as sensitivity, measures the ratio of true positives to all actual positives. It indicates how well the model identifies all relevant instances:

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (\text{A.2})$$

where FN is the number of False Negatives.

- Intersection over Union (IoU): IoU is a fundamental metric used to quantify the overlap between two regions, typically a predicted bounding box and a ground truth bounding box. Mathematically, it is defined as the ratio of the area of the intersection between the predicted and ground truth bounding boxes to the area of their union:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}. \quad (\text{A.3})$$

An IoU of 1 indicates a perfect match, while an IoU of 0 indicates no overlap. Intermediate values reflect varying degrees of spatial alignment. In object detection, a prediction is considered a True Positive if its IoU with the ground truth exceeds a pre-defined threshold.