Towards Domain Invariant Real-time Point Cloud Perception

by

Yaodong Cui

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2023

© Yaodong Cui 2023

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Simon Yang Professor, School of Engineering, University of Guelph
Supervisor(s):	Amir Khajepour Professor, Dept. of MME, University of Waterloo
Internal Member:	Pan Zhao Assistant Professor, Dept. of MME, University of Waterloo
Internal-External Member:	Alexander Wong Professor, Dept. of SYDE, University of Waterloo
Internal-External Member:	Xuemin (Sherman) Shen Professor, Dept. of ECE, University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Yaodong Cui is the sole author of Chapters 1, 4, 5, and 6 which were written under the supervision of Prof. Amir Khajepour and were not published. This thesis consists of parts of two manuscript written for publication.

Research presented in Chapter 2:

Cui, Yaodong, Ren Chen, Wenbo Chu, Long Chen, Daxin Tian, Ying Li, and Dongpu Cao. "Deep learning for image and point cloud fusion in autonomous driving: A review." IEEE Transactions on Intelligent Transportation Systems 23, no. 2 (2021): 722-739.

Research presented in Chapter 3:

Cui, Yaodong, Minghao Ning, Wenbo Li, Dongpu Cao, and Amir Khajapour. "Task-attentive LiDAR Filter: Adaptive ROI and Semantic-based Ground Segmentation." In 2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI), pp. 1-6. IEEE, 2022. As lead author of these two chapters, I was responsible for contributing to conceptualizing study design, carrying out data collection and analysis, and drafting and submitting manuscripts. My coauthors provided guidance during each step of the research and provided feedback on draft manuscripts.

Abstract

In recent years, autonomous driving has witnessed substantial advancements, owing in part to the rapid advancement of 3D sensor technologies, particularly Light Detection and Ranging (LiDAR) sensors.Despite these advancements, the challenge of open world autonomous driving remains a pressing issue that requires resolution. This encompasses the need for dependable and resilient 3D perception across a multitude of environments. Particularly, point cloud is irregular, unordered, and continuous with large data size, which presents unique challenges for its real-time processing. Another major challenge came from the perception system's inability to adapt to different environments, known as the domain adaptation problem. This is due in part to a lack of diverse and representative datasets, and in part to existing models' insufficient generalization ability. To tackle this challenge, this dissertation conducts a thorough investigation into the domain adaptation challenges associated with real-time point cloud perception.

This dissertation addresses these challenges associated with the deployment and training of point cloud perception systems in a self-contained manner.To ensure sufficient real-time capability during deployment, a novel approach called task-attentive 3D perception is proposed. It incorporates HD-map, vehicle states, and emergency breaking distance to dynamically remove task un-related point cloud for driving safety and computation efficiency. Furthermore, this thesis looks at the problem of driving in varied domains from both the data and the model standpoint. First, a novel method for creating the target real-world domains in the simulator using real-world prior is presented. Using this method, a noval large-scale multi-task dataset called Domains-3D is developed, which comprises both real-world and synthetic domains. Finally, using this dataset, a novel Plug-and-Play (PnP) domain adaptation algorithm for 3D point clouds that minimizes domain shifts is presented. This algorithm improves a model's performance for both cross-scene domain adaptation and intra-domain 3D object detection.

Acknowledgements

I would like to extend my heartfelt gratitude to my supervisor, Prof. Amir Khajepour, for his unparalleled support and guidance during my Ph.D. program at MVS Lab. His mentorship and expertise have been instrumental in shaping my thesis and ensuring its successful completion. I am also grateful for the numerous opportunities he has provided me, including participating in WATonoBus, which have expanded my understanding of real-world engineering challenges and offered me valuable hands-on experience. In addition to his professional guidance, I would like to acknowledge Prof. Amir's encouragement and support on a personal level. His belief in me has been a constant source of motivation and inspiration. Furthermore, his involvement in our start-up, LoopX.ai, has been integral to its success and I am deeply appreciative of his unwavering support and contributions.

I am also deeply thankful to Professor Dongpu Cao for his invaluable contributions to the early stages of my Ph.D. journey. Despite his limited time at Waterloo, he provided me with a new perspective on both research and life. I am particularly grateful to him for introducing me to the field of autonomous driving, which has since become a passion and focus of mine.

To my dear friends Boni Li, Chen Sun, Jinwei Zhang, Jiaming Zhong, Panjian Yu, Shucheng Huang, Wenrui Ye, Yu Chao, and Zhenan Liu, words cannot express my gratitude for the support and friendship you have shown me throughout my Ph.D journey at Mechatronic Vehicle Systems (MVS) Lab, LoopX.ai, and CogDrive Lab. Your encouragement and camaraderie have made this experience truly unforgettable. I would also like to extend a special thanks to Shucheng Huang. Your generosity in offering your advice, support and even letting me use your computer has been invaluable. I would also like to express my sincere gratitude to Assylbek Dakibay, Neel P. Bhatt, and Ehsan Hashemi for their invaluable contribution in creating a realistic CARLA map. I am proud to have had the opportunity to work and learn alongside such talented and supportive individuals and I will always cherish the memories and bonds we have formed.

Dedication

This thesis is dedicated to you, Lanqin Yao, and I dedicate it to you with all my heart.

You have been my partner in every sense of the word, always there to listen, offer advice, and pick me up when I was feeling down. Your belief in me never wavered, even on the toughest days, and your constant encouragement has been a source of inspiration and motivation.

You have made countless sacrifices and put your own dreams and aspirations on hold so that I could follow mine. Your selflessness and love have been the foundation of our life together, and I am deeply grateful for all that you have done for me.

Table of Contents

List of Figures **xii**

List of Tables **xv**

List of Algorithms **xvii**

List of Abbreviations **xviii**

1 Introduction **1**

1.1 Background and Motivation 1

1.2 Objectives and Contributions 5

1.3 Structure of The Thesis 7

2 Literature Review **11**

2.1 Deep Learning on Image 11

2.2 Deep Learning on Point Cloud 12

2.3 Evaluation Metrics 16

2.4 Autonomous Driving Dataset 18

2.4.1 Real-world Autonomous Driving Dataset: 18
2.4.2 Synthetic Driving Data Generation: 19
2.5 3D Object Detection 19
2.5.1 2D Proposal Based Sequential Models 21
2.5.2 3D Proposal Based Sequential Model 25
2.5.3 One-step Models 28
2.6 Domain Adaptation for Point Cloud Perception 30
2.6.1 Data-driven Approach 31
2.6.2 Model-driven Approach 33
2.7 Open Challenges 33
2.8 Summary 34

3 Task-attentive 3D Object Detection **36**

3.1 Background 36
3.2 Methodology 40
3.3 Experiments 48
3.3.1 Testing Platform 48
3.3.2 Dataset 48
3.4 Results and Discussion 49
3.4.1 The impacts on perception run-time 49
3.4.2 The impact on ground segmentation 51
3.5 Summary 52

4 Domains-3D: A Novel Domain Adaptation Dataset **53**

4.1 Background 55

4.2 Methodology 57

4.2.1 Problem Formulation 57

4.2.2 Synthetic Domain Generation Using Real-wold Prior 58

4.2.3 Algorithms for CARLA 60

4.3 Dataset Designs 63

4.3.1 Dataset Overview 63

4.3.2 Map Design 64

4.3.3 Sensor Suite 64

4.3.4 Annotations 67

4.4 Experiments and Meta Analysis 67

4.5 Summary 80

5 Background-Aware Point Cloud Domain Adaptation **81**

5.1 Background 81

5.2 Methodology 83

5.2.1 Problem Statement 83

5.2.2 A Meta Analysis of Background Stuff 86

5.3 Experiments 89

5.3.1 Cross-City Domain Adaptation Results 91

5.3.2 Urban-to-Rural Domain Adaptation Results 92

5.3.3 Public-to-Campus Domain Adaptation Results 94

5.3.4 Real-time performance 97
5.4 Summary 98

6 Conclusions and Outlooks **99**
6.1 Conclusions 99
6.2 Outlook And Future Works 100

Letter of Copyright Permission **102**

References **103**

List of Figures

1.1 Notable applications of LiDAR sensors in various autonomy platforms. . . 2

1.2 Examples of different conditions that could affect driving 3

1.3 The overall workflow of the proposed methods in a autonomous vehicle system. 5

1.4 An overview of the structure of this thesis 10

2.1 The structure of this literature review categorized by tasks. 12

2.2 Timeline of 3D object detection networks and their corresponding fusion levels. 15

2.3 Some typical model architectures and fusion methods for dynamic object detection. 20

2.4 A scatter plot of models' average precision on 3D object detection(AP_3D) with respect to runtime. The results are based on the public available KITTI test set. 25

2.5 Timeline of 3D semantic segmentation networks and their corresponding fusion levels. 29

2.6 The workflow of Domain-Invariant Data Representation. 31

2.7 The workflow of Domain Mapping. 32

3.1 An overview of the proposed task-attentive LiDAR filter method. 37

3.2 An example of dynamic object enters the road and potentially interact with the ego-vehicle. 40
3.3 An example of adaptive ROI filter with road area and PIA. 42
3.4 (a) An illustration of the sensor setups on the WATonoBus. (b) The WATonoBus. 43
3.5 Example of task-attentive filter on geese. 45
3.6 Boxplots of task-attentive filter on WatonoBus data 47

4.2 The University campus map based on University of Waterloo in Unreal Engine Editor. 65
4.3 A comparison between the accumulated scanning patterns of traditional mechanical LiDAR and recent solid-state LiVOX LiDAR [140]. 66
4.4 The intra-class attributes of Vehicles 69
4.5 The vehicle's temporal and spatial distributions of cars in real and simulated Boston and Singapore domains 70
4.6 The pedestrian's temporal and spatial distributions of pedestrian in real and simulated Boston and Singapore domains 71
4.7 The inter-class distribution. 72
4.8 The vehicle's temporal and spatial distributions of real and simulated urban and suburban/rural domains 74
4.9 The pedestrian's temporal and spatial distributions of real and simulated urban and suburban/rural domains 77
4.10 The vehicle's and pedestrian's temporal and spatial distributions of real and simulated campus domains 78

5.1 An instance of *scenes-0247* visualized on the HD-map, where ego-poses are illustrated in black dots and map layers are represented in different colors. 84

5.2 The distribution of road and traffic elements in two different domains. . . . 88

List of Tables

2.1 Comparative results on KITTI 3D Object Detection benchmark 23

3.1 The number of remaining points after applying different types of ROI filter. 49

3.2 The run time of different perception tasks after applying different types of ROI filter. 50

4.1 Comparison of existing autonomous driving datasets that provide point cloud data. 56

4.2 The per-frame foreground objects distribution on different domains. 67

5.1 The per-scene road structures and traffic elements distribution on different domains. 86

5.2 3D object detection results for cross-city domain adaptation. 90

5.3 3D object detection results average on all classes for cross-city domain adaptation. 91

5.4 3D object detection results on rural domain for urban to rual domain adaptation. 93

5.5 3D object detection results average on all classes for urban to rual domain adaptation. 94

5.6 3D object detection results on nuScenes for public to campus adaptation. . 95

5.7 3D object detection results average on all classes for urban to rual domain adaptation. 96

5.8 3D object detection results on nuScenes for same-location validation. . . . 97

List of Algorithms

1 Generating Spawn Points for NPC in CARLA 61
2 Rich Scenes Generation for Data Collection 62

List of Abbreviations

2D Two-dimensional 13, 15, 19

3D Three-dimensional 1, 2, 10

AP Average Precision 16

bbox Bounding Box 19, 21, 22

BEV Bird's Eye View 25–27

CAD Computer-Aided Design 24

CNN Convolutional Neural Network 2, 11, 28, 29, 89, 90

DA Domain Adaptation 30–33

DNN Deep Neural Network 2, 33

ESU Extension Spatial Upsample 27

FN False Negative 16

FP False Positive 16

GNSS Global Navigation Satellite System 48

HD High-Resolution 2, 48

HHA Horizontal disparity, Height above ground, and Angle with gravity 24

IMU Inertial Measurement Unit 48

IoU Intersection over Union 16

LiDAR Light Detection and Ranging v, xii, 1, 12, 24, 26, 27, 29, 36–40, 48, 49, 52, 66

mAAE mean Average Attribute Error 17, 91, 94, 96

mAOE mean Average Orientation Error 16, 17, 91, 94, 96

mAP Mean Average Precision 16, 91

mASE mean Average Scale Error 16, 17, 91, 94, 96

mATE mean Average Translation Error 16, 17, 91, 94, 96

mAVE mean Average Velocity Error 16, 17, 91, 94, 96

mIoU mean Intersection over Union 17

MLP Multi-Layer Perceptron 11, 14, 15, 27

NDS Nuscenes Detection Score 17, 91, 92, 94, 96

NPC non-player character 60

OA Overall Accuracy 17

PIA Potential Interactive Area 6, 42

PnP Plug-and-Play v, 6–8

RGB Red Green Blue 24, 28

RGB-D Red Green Blue Depth 27, 28

ROI Region Of Interest 6, 37–41, 46, 49, 50

SOTA state-of-the-art 7, 8, 11, 22, 26, 29, 58

TN True Negative 16

TP True Positive 16

Chapter 1

Introduction

This chapter presents an overview of the dissertation's background, motivation, objectives, contributions, and organization. Section 1.1 provides context for the field, highlighting the limitations of existing methods and consolidating them into three major challenges. Section 1.2 delves deeper into the motivation for this thesis, laying out the overarching research question, specific objectives, and key contributions. Finally, Section 1.4 introduces the thesis structure, providing an overview of the remaining chapters' organization and the logical flow of ideas throughout this thesis.

1.1 Background and Motivation

In recent years, there has been a significant advancement in the field of autonomous driving, driven in large part by the rapid development of Three-dimensional (3D) sensor technologies. Particularly, LiDAR sensors have emerged as a cost-effective method for autonomous vehicles to acquire highly accurate 3D measurements of their surroundings. LiDAR generate a dense point cloud representation of their surroundings by emitting laser beams and measuring the reflected signals. Their ability to capture accurate 3D information

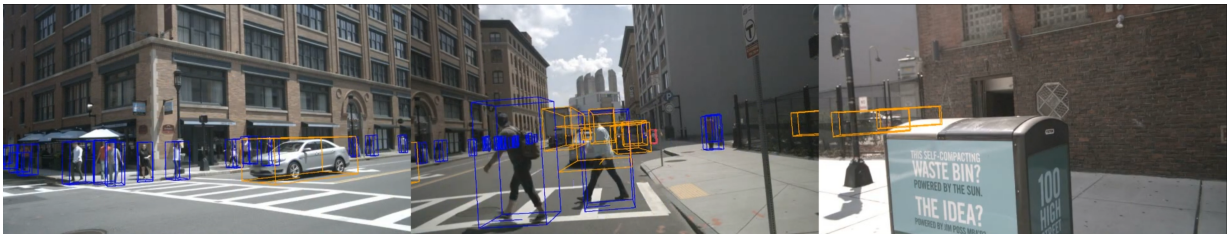


(a) Passenger Car [175] (b) Shuttle Bus [100] (c) Robot Dog [237] (d) Drone [99]

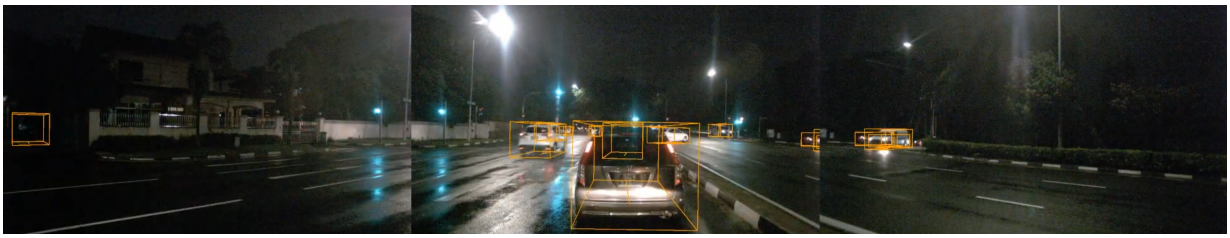
Figure 1.1: Notable applications of LiDAR sensors in various autonomy platforms.

in adverse weather conditions and low-light environments has made them a robust solution for various autonomy applications. The integration of LiDAR and deep learning in autonomous vehicles has the potential to revolutionize road safety, traffic efficiency, and personal mobility. Specifically, high-resolution point cloud data can be leveraged in three major applications for autonomous driving: real-time scene understanding, high-resolution mapping, and localization. Real-time scene understanding includes tasks such as 3D object detection and 3D semantic segmentation. The second application is the construction and update of High-Resolution (HD) map for accurate re-localization and referencing. This thesis focuses on real-time scene understanding from point cloud data.

With breakthroughs in Deep Neural Network (DNN), such as Convolutional Neural Network (CNN) and transformer architectures, real-time scene understanding has reached a level of performance that rivals that of humans under certain conditions [118] [190] [240]. However, this success has not been successfully replicated for real-time point cloud data processing [74], which poses unique challenges due to its larger data size, irregular, unordered, and continuous nature. Specifically, the irregularity of the point cloud data makes it difficult for traditional deep learning operators such as CNNs and transformers to be directly applied. To tackle this issue, various methodologies have been proposed for feature extraction from point cloud data. These include transforming point cloud data into more regular formats, such as image-like [214] [185], 3D-voxels [294] [161], or tree representations [280] [116] [191] [66] [127] , as well as models that extract features directly from



(a) Sunny Boston urban environment in the nuScenes dataset [17]



(b) Rainy Singapore rural environment in the nuScenes dataset [17]

Figure 1.2: Examples of different conditions that could affect driving

the irregular and unordered format, such as point-based [186] [21] [91] [9] [132] [70] [229] and graph-based models [24] [82] [207] [13] [15] [46]. Despite these recent advancements, their real-time performance remains a significant challenge, preventing their deployment in real-world environments [74] [40].

As a safety-critical application, autonomous vehicles operating in open world environments require the highest level of robustness across a wide range of scenarios. In such environments, any failure or delay in identifying life-threatening situations could endanger the safety of passengers and other road users. This ability to adapt to different environments is referred to as the domain adaptation problem in the field of machine learning, which still remains a major challenge [29] [86] [92] [153] [80] [88] [111] [114] [273]. More importantly, this problem is particularly pronounced in autonomous driving, where a variety of conditions could be encountered, including scenery, traffic elements and patterns, roadway types, lighting and weather conditions, and more [40] [218]. Examples of these different conditions (i.e. domains) that influence driving are illustrated in Fig.1.2. One of

the major hurdles impeding progress in solving this domain adaptation problem is the lack of diverse and representative datasets [218] [273]. This biases the model towards specific domains represented in the few available datasets, resulting in poor generalisation and performance when applied to new environments. Current research, in particular, requires large-scale datasets that accurately reflect the wide range of conditions that self-driving cars could encounter. However, gathering and annotating large amounts of such data is prohibitively time-consuming and costly. As a result, the development of cost-effective methods to create, collect, and annotate diverse and representative datasets is essential for addressing the domain adaptation problem in autonomous driving and other fields.

Another crucial aspect of addressing the domain adaptation problem is the development of novel learning algorithms capable of adapting to different domains [273] [254] [284] [187]. These models should be able to learn from a source domain while also generalizing well to the source and new target domains, and they should ideally require little or no additional target domain training data. Adversarial training [232] [64] [19] [65] and domain-invariant feature learning [64] [148] [136] [247] are two examples of such algorithms, which rely on aligning distributions or extracting domain-invariant features respectively. These methods, however, have limitations, and more research is needed to improve their performance and robustness. Therefore, the development of domain adaptation algorithms, in combination with cost-effective methods for creating diverse and representative datasets, will be crucial for achieving robust and reliable performance in open world autonomous driving.

To summarize, three major barriers to widespread deployment of point cloud-based 3D perception systems for autonomous driving have been identified:

- **Inadequate Real-time Performance:** Point-cloud-based perception systems' insufficient real-time performance limits autonomous vehicles' ability to respond quickly and safely to dynamic environments.
- **Limited Availability of Domain Adaptation Datasets:** Due to the scarcity of comprehensive domain adaptation datasets for autonomous driving, progress in domain adaptation research is hampered.

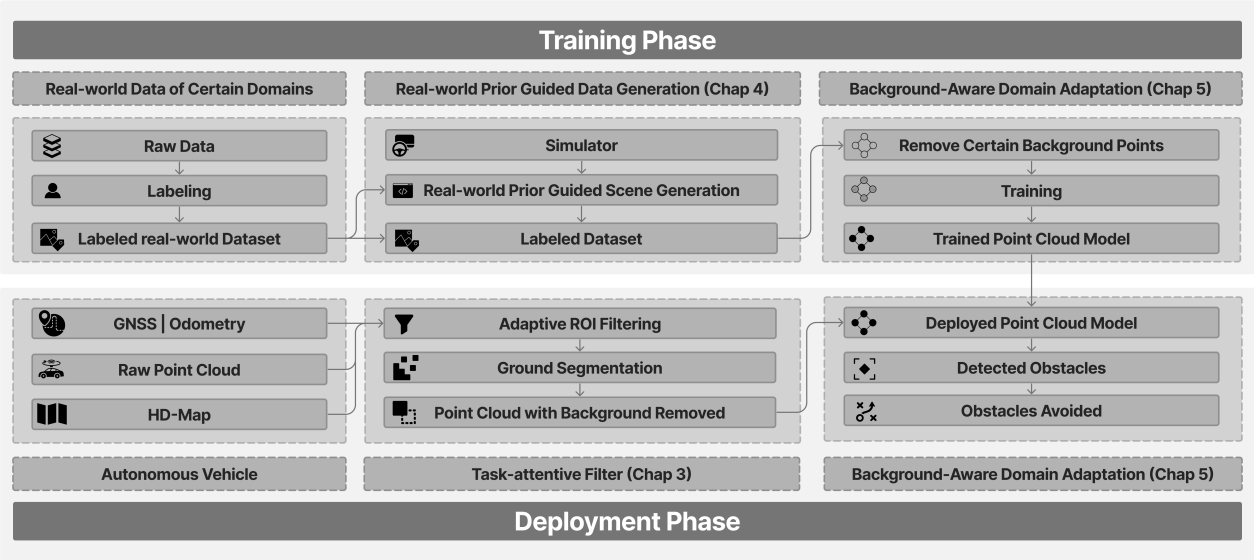


Figure 1.3: The overall workflow of the proposed methods in a autonomous vehicle system.

- **Insufficient Generalization Capabilities:** Deep neural networks with insufficient generalization capabilities, which lack the ability to generalize well to different environments or scenarios.

1.2 Objectives and Contributions

The central focus of this thesis is to tackle the computational and domain adaptation challenges faced by point-cloud-based 3D perception systems for autonomous driving. This objective is motivated by the need for accurate and efficient 3D perception in autonomous driving, as it plays a critical role in ensuring the safety and reliability of autonomous vehicles. With this in mind, the overarching research question guiding this thesis is: How to improve the real-time performance and generalization capabilities of point-cloud-based

3D perception systems for autonomous driving? To address this question and achieve the proposed objective, this thesis presents the following key contributions:

- **Task-attentive 3D Perception:** A novel adaptive Region Of Interest (ROI) filter that incorporates HD-map, vehicle states, and emergency breaking distances to estimate adaptive ROI areas for driving safety and computation efficiency. Additionally, an adaptive area, referred to as Potential Interactive Area (PIA), is introduced. PIA is area adjacent to the roadway containing dynamic objects that could interact with the ego-vehicle. The proposed approach employs HD-map semantic priors and region-wise ground segmentation for accurate ground segmentation, enabling the perception system to distinguish small obstacles near the ground surface. This is the first work to combine these techniques for autonomous vehicle perception, to the best of author's knowledge.

- **Real-World Prior-Guided Synthetic Data Generation:** A novel method for generating synthetic data that closely resembles real-world domains using real-world prior is developed. The proposed method not only improves the transfer of knowledge from synthetic to real-world domains, but also offers enhanced strategies for incorporating simulated data to enhance the performance of perception systems.

- **Domains-3D, A Novel Domain Adaptation Dataset:** A novel comprehensive domain adaptation dataset for autonomous driving is developed. With 18 cross-scene domains and 54 possible cross-domain splits, it is also the first large-scale multi-task dataset with annotated solid state lidar data, bridging a field gap as solid state lidars are expected to replace mechanical lidars in the near future. This enhancement enables the creation of 4 cross-sensor domains and 12 different cross-sensor domain splits.

- **Background-Aware Point Cloud Domain Adaptation:** A novel PnP domain adaptation algorithm for 3D point clouds that reduces the domain shift between the source and target domains. This algorithm is based on the comprehensive analysis

of the impact of background points on cross-scene domain shifts, which is the first study of its kind. Experiments demonstrate that this algorithm improves the model's performance for both cross-scene domain adaptation and intra-domain 3D object detection.

As illustrated in Fig.1.3, when taken together, these contributions form a complete and comprehensive picture of a robust and efficient real-time 3D perception system designed to operate under a variety of conditions.

1.3 Structure of The Thesis

This thesis is divided into five chapters, each addressing a different aspect of the research, as shown in Fig.1.4. The second chapter contains a review of the literature on the SOTA in deep learning and 3D perception for autonomous driving. The third chapter introduces task-attentive 3D object detection framework that reduces computation and memory complexity while improving detection of small obstacles. Chapter 4 introduces Domains-3D, a novel large domain adaptation dataset developed for autonomous driving that includes both real-world and simulated data. The fifth chapter investigates the effect of background points on domain shifts on point cloud. In addition, it presents a PnP domain adaptation algorithm to address the domain adaptation problem. Finally, chapter 6 wraps up the thesis and makes recommendations for future research.

Chapter 2: Literature Review on deep learning and 3D perception

This chapter provides a literature review on state-of-the-art (SOTA) deep learning techniques and their applications in point cloud processing for autonomous driving. It offers an introduction to deep learning and its applications on point cloud data, in addition to common methodologies and point cloud representations. It also discusses SOTA techniques for 3D object detection on point cloud data, such as sensor fusion methods for enhanced

detection performance. Furthermore, it delves into the field of 2D/3D semantic segmentation on point cloud data, discussing various approaches, their advantages and limitations, recent advancements and open challenges in the field.

Chapter 3: Task-attentive 3D Object Detection

This chapter examines a framework for task-attentive 3D object detection in autonomous driving applications. The proposed methodology for task-attentive 3D object detection, including the design of an adaptive ROI filter and semantic-based ground segmentation filter, is explained in detail. The experimental results, including quantitative and qualitative evaluations, are presented. Furthermore, a detailed analysis demonstrates the effectiveness and superiority of the proposed approach over SOTA methods.The task-attentive 3D object detection framework and its ability to improve autonomous driving systems are also discussed in the chapter.

Chapter 4: A Novel Domain Adaptation Dataset for Autonomous Driving

Domains-3D, a novel comprehensive domain adaptation dataset for self-driving cars, is developed in this chapter. The dataset, which includes 18 domains and 54 potential cross-domain splits, is designed to address the limitations of existing datasets by providing a comprehensive, representative, and challenging benchmark for evaluating the generalization capabilities of 3D perception models for autonomous driving. It is also the first large-scale multi-task dataset with annotated solid state lidar data, it bridges a field gap as solid state lidars are expected to replace mechanical lidars soon. The chapter starts with a background and challenges of synthetic data generation and domain adaptation for autonomous driving. The methodology and mathematical theorem used to calculate domain shifts between two domains in relation to the task at hand are then presented. This is then used to guide the synthetic data generation process and reduce domain shifts between the synthetic and target domains. Finally, this chapter describes how the dataset was created, beginning with an in-depth examination of the target domains built on the nuScenes dataset [17].

Chapter 5: Background-Aware Point Cloud Domain Adaptation

Background-Aware Point Cloud Domain Adaptation is a novel PnP method for improv-

ing the generalization ability of a 3D object detection model for cross-scene domain adaptation. Using mathematical proof, the method demonstrates that removing background points from a point cloud reduces the total number of domain shifts. This algorithm is tested on the Domains-3D dataset, and the results show that by removing unnecessary information from the point cloud, it improves the performance of cross-scene domain adaptation models on foreground objects as well as intra-domain 3D object detection.

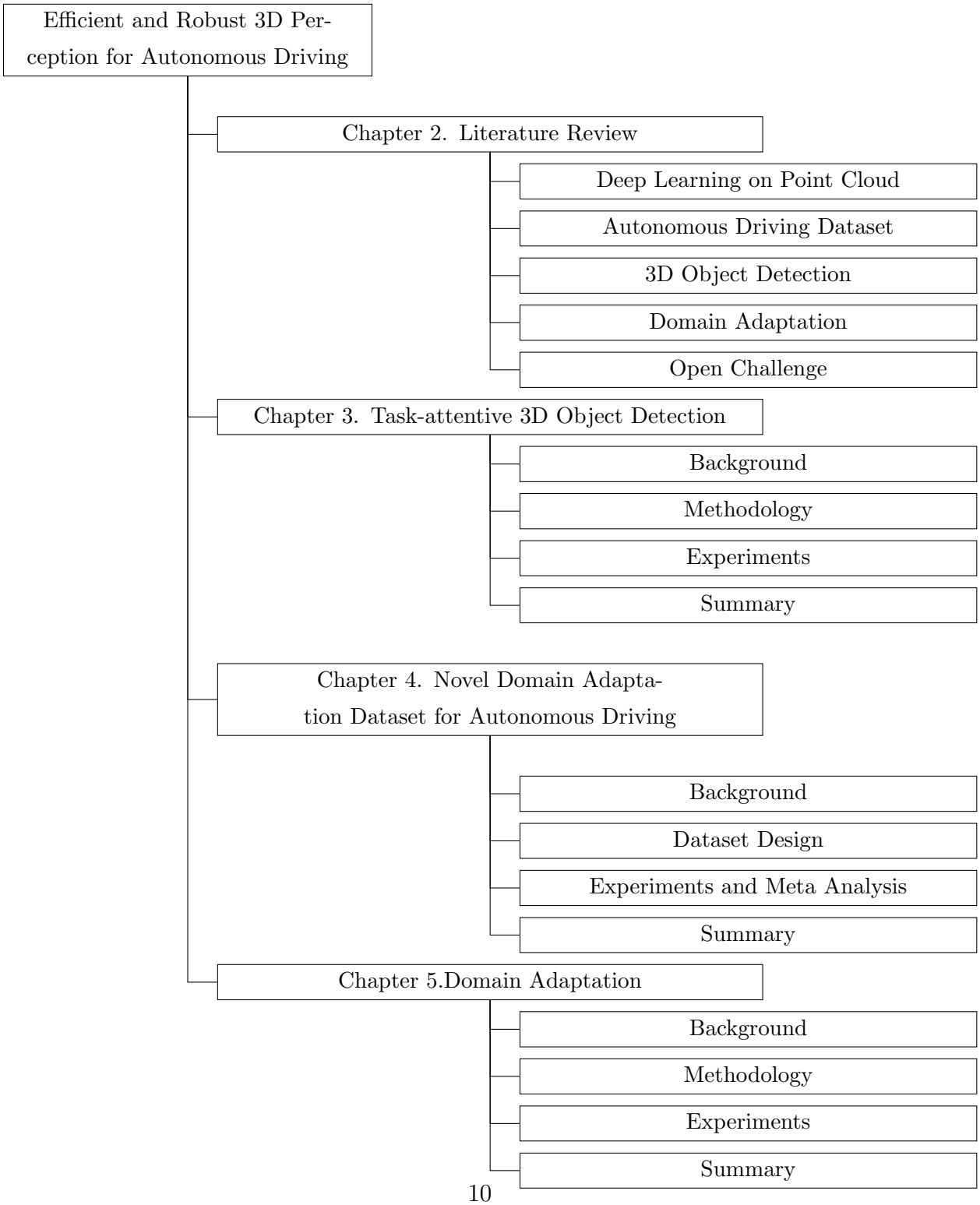


Figure 1.4: An overview of the structure of this thesis

Chapter 2

Literature Review

This chapter presents a comprehensive literature review on the SOTA deep learning techniques and their applications on point cloud processing for autonomous driving. Section 1.1 begins with an overview of deep learning and its applications on point cloud data, highlighting the most widely used methodologies and point cloud representations in the field. Section 1.2 describes the SOTA methods for 3D object detection on point cloud data, including sensor fusion methods for improved detection performance. Section 1.3 delves into the field of 2D/3D semantic segmentation on point cloud data, discussing the various approaches and their benefits and drawbacks. Finally, open challenges in these fields that could advance autonomous driving are discussed.

2.1 Deep Learning on Image

CNN are one of the most efficient and powerful deep learning models for image processing and understanding. Compared to Multi-Layer Perceptron (MLP), CNN is shift-invariant, contains fewer weights and exploits the hierarchical patterns, making it highly efficient for semantic information extraction. Typically hidden layers of a CNN consist of a hierarchy of convolutional layer, batch normalization layer, activation layer and pooling layer, allowing

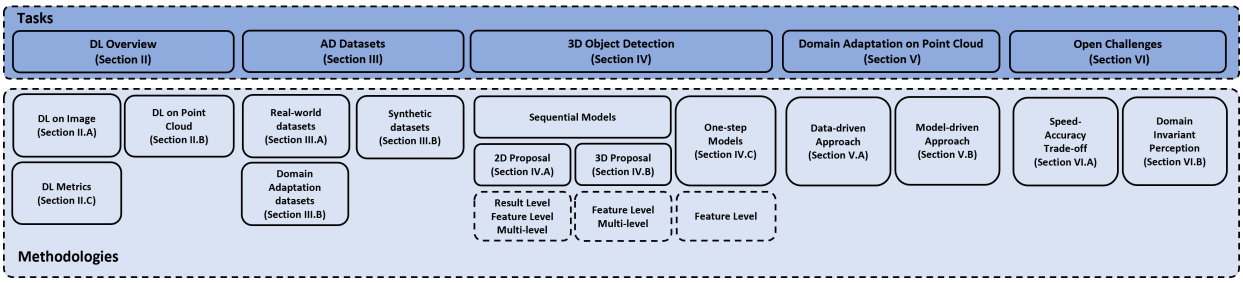


Figure 2.1: The structure of this literature review categorized by tasks.

end-to-end training. This hierarchical structure extracts image features with increasing abstract levels and receptive field, enabling the learning of high-level semantics.

2.2 Deep Learning on Point Cloud

Deep learning is a subset of artificial neural networks which leverages multiple network layers to extract features progressively. The early works of learning based artificial neural networks start in the 1940s and gained increasing attention and developments throughout the 1970s and 1980s. Some of the most important concepts, such as automatic differentiation [51], backpropagation [142] [263], max-pooling [260] are proposed during this time. During the 1990s and 2000s, researches in artificial intelligence have witnessed a slow down due to various reasons. However, the exponential growth of computational power and data, combined with advances in network architecture and training strategies, researches in artificial intelligence has since regaining momentum. Especially, deep learning based methods have demonstrated great potentials, outperforming the other methods in the ImageNet competition [196] [119] [221].

The point cloud is a set of data points, which are LiDAR's measurements of the detected object's surface. In terms of data structure, the point cloud is sparse, irregular, orderless and continuous. Point cloud encodes information in 3D structures and in per-point features (reflective intensities, color, normal, etc.,), which is invariant to scale, rigid

transformation and permutation. These characteristics made feature extractions on the point cloud challenging for existing deep learning models, which require the modifications of existing models or developing new models. Therefore, this section focuses on introducing common methodologies for point cloud processing.

Volumetric representation based: The volumetric representation partitions the point cloud into a fixed-resolution 3D grid, where features of each grid/voxel are hand-crafted or learned. This representation is compatible with standard 3D convolution [294] [161]. Several techniques have been proposed in [185] to reduce over-fittings, orientation sensitivity and capture internal structures of objects. However, the volumetric representation loses spatial resolution and fine-grained 3D geometry during voxelization which limits its performance. Furthermore, attempts to increase its spatial resolution (denser voxels) would cause computation and memory footprint to grow cubically, making it unscalable.

Index/Tree representation based: To alleviate constraints between high spatial resolution and computational costs, adapted-resolution partition methods that leverage tree-like data structures, such as kd-tree [280] [116], octree [191] [66] [127] are proposed. By dividing the point cloud into a series of unbalanced trees, regions can be partitioned based on their point densities. This allows regions with lower point densities to have lower resolutions, which reduce unnecessary computation and memory footprint. Point features are extracted along with the pre-built tree structure.

Two-dimensional (2D) views representation based: 2D views/multi-views are generated by projecting the point cloud to multiple 2D view planes. These rendered multi-view images can be processed by standard 2D convolutions and features from these views are aggregated via view-pooling layers [214]. Thus, the permutation-invariant problem is solved by transforming point cloud to images and the translation-invariant is achieved by aggregating features from different views. Qi et al. [185] combined the volumetric representation with multi-views generated via sphere rendering. Unfortunately, 2D views methods lose the 3D geometry information during the view rendering and struggle with per-point label prediction [116].

Graph representation based: Point clouds can be represented as graphs and convolution-like operation can be implemented on graphs in the spatial or spectral domain [24] [82] [207]. For graph-convolution in the spatial domain, operations are carried out by MLPs on spatially neighbouring points. Spectral-domain graph-convolutions extend convolutions as spectral filtering on graphs through the Laplacian Spectrum [13] [15] [46].

Point representation based: Point representation based methods consume the point cloud without transforming it into an intermediate data representation. Early works in this direction employ shared MLPs to process point cloud [186] [21] , while recent works concentrated on defining specialized convolution operations for points [91] [9] [132] [70] [229].

One of the pioneering works of direct learning on point clouds is the PointNet [21] [186], which employs an independent T-Net module to align point clouds and shared MLPs to process individual points for per-point feature extraction. The computation complexity of the PointNet increases linearly with the number of inputs, making it more scalable compared with volumetric based methods. To achieve permutation invariant, point-wise features are extracted by shared MLPs which are identical for all points. These features are aggregated by symmetric operations (i.e. max-pooling), which are also permutation invariant. The feature extraction process of the PointNet is defined as:

$$g\left(\{x_1, \ldots, x_n\}\right) \approx f_{sym}\left(h\left(x_1\right), \ldots, h\left(x_n\right)\right) \tag{2.1}$$

where x represents input points, h represents the per-point feature extraction function (i.e. shared MLPs), f_{sym} represents a symmetric function (i.e. max-polling) and g is a general function that one want to approximate.

However, the PointNet fails to extract local inter-point geometry at different levels. To mitigate this challenge, Qi et al. [186] extended the PointNet to extract features from different levels by grouping points into multiple sets and apply PointNets locally. To reduce the computational and memory cost of the PointNet++ [186], the RandLA-Net [89] stacked the random point sampling modules and attention-based local feature aggregation modules hierarchically to progressively increase receptive field while maintaining high efficiency.

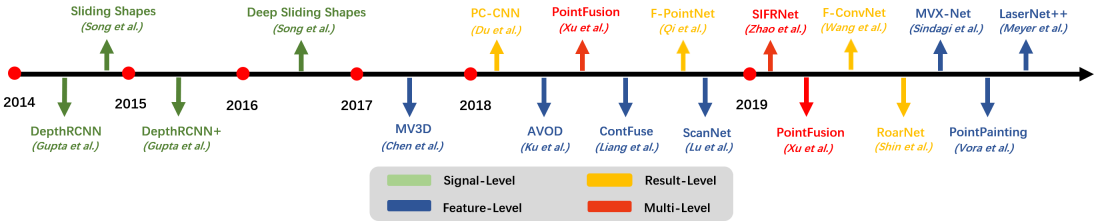


Figure 2.2: Timeline of 3D object detection networks and their corresponding fusion levels.

Unlike PointNet-based methods, the spatial relationship between points is explicitly modelled in point-wise convolutions. Point-wise convolutions aim to generalize the standard 2D discrete convolution to the continuous 3D space. The main challenge is to replace the discrete weight filter in standard convolution with a continuous weight function. This continuous weight function is approximated using MLPs in PointConv [268] and correlation functions in KPConv [229] and PCNN [9]. More specifically, the PCNN [9] defines convolution kernels as 3D points with weights. A Gaussian correlation function that takes the coordinates of the kernel point and input point is used to calculate the weighting matrix at any given 3D coordinates. The KPConv [229] follows this idea but instead uses a linear correlation function. Furthermore, KPConvs [229] are applied on local point patches hierarchically, which are similar to the concepts of standard CNNs. This general point-wise convolution $\mathcal{F}$ at an input point $x \in \mathbb{R}^3$ in 3D continuous space is defined as:

$$(\mathcal{F} * h)(x) = \sum_{x_i \in \mathcal{N}_x} h\left(x_i - x\right) f_i \tag{2.2}$$

where h is the per-point kernel function which calculates the weighting matrix given the coordinates of input points and kernel points. x_i and f_i are the ith neighboring points of x and their corresponding features (intensity, color etc.,). $\mathcal{N}_x$ are all the neighboring points of the input point x, which are determined using KNN or radius neighborhoods.

2.3 Evaluation Metrics

To thoroughly evaluate the performance of the performance detection tasks and segmentation, several metrics have been carefully selected. These metrics have been widely adopted in the literature and provide a fair basis for comparison. The specific metrics used and their corresponding descriptions are outlined below. In the task of 3D object detection on point cloud, the most commonly used metrics for evaluating accuracy are Average Precision (AP) and Mean Average Precision (mAP):

$$\begin{aligned} AP_i &= \frac{1}{N} \sum_{i \in N} \frac{TP_i}{TP_i + FP_i} \\ mAP &= \frac{1}{N} \sum_{i=1}^{N} AP_i \end{aligned} \tag{2.3}$$

Where True Positive (TP) are points correctly labeled as a specific class, True Negative (TN) are points correctly labeled as the background class, False Positive (FP) are points incorrectly labeled as a specific class, and False Negative (FN) are points that should have been labeled as a specific class but were not. This metric calculates the average of the Intersection over Union (IoU) of the predicted bounding boxes with the ground truth, for those bounding boxes that exceed a predefined IoU threshold. The use of the IoU metric also allows for a more detailed assessment for the segmentation task, as it takes into account the number of correctly classified points and penalizes for misclassifications. The IoU calculates the overlap between the target mask and the prediction output and is defined as:

$$IoU = \frac{AreaofOverlap}{AreaofUnion} \tag{2.4}$$

Other common true positives metrics that measures different aspects of 3D object detection includes mean Average Translation Error (mATE), mean Average Scale Error (mASE), mean Average Orientation Error (mAOE), mean Average Velocity Error (mAVE).

These metrics are computed over all true positive (TP) instances, considering a recall level of at least 10%. More specifically, the mATE measures the average 2D Euclidean distance (in meters) between the centers of predicted and ground-truth objects. The mASE calculates the average scale error calculated as 1 minus the Intersection over Union (IoU) after aligning object centers and orientations. And the mAOE is the average minimum yaw angle difference (in radians) between the predicted and ground-truth orientations. For speed estimation task, the mAVE calculates average absolute velocity error (in meters per second). Finally, the mean Average Attribute Error (mAAE) is the average attribute error, which is computed as 1 minus the attribute classification accuracy. For 3D object detection task, [17] proposed Nuscenes Detection Score (NDS) score, which combines these metrics and calculates a weighted sum. This score provides a comprehensive assessment of the algorithm's performance, taking into account not only the precision but also various error types related to translation, scale, orientation, velocity, and attributes.

For semantic segmentation task, the accuracy, mean Intersection over Union (mIoU), and Overall Accuracy (OA) are the most widely adopted metrics, which can be defined as:

$$\begin{aligned} \text{Accuracy } &= \frac{TP + TN}{TP + TN + FP + FN} \\ mIoU &= \frac{\sum_{i=1}^{N} IoU_i}{N} \\ OA &= \frac{\sum_{i=1}^{N} c_{ii}}{\sum_{j=1}^{N} \sum_{k=1}^{N} c_{jk}} \end{aligned} \tag{2.5}$$

Let C be an $N \times N$ confusion matrix of the segmentation results, where each entry $c_{i,j}$ represents the number of points from ground-truth class i that are predicted as class j.

For domain adaptation task, it is crucial to to quantify the discrepancy between the two domains. This is frequently accomplished through the use of $f-$divergences, a class of mathematical measurements that quantify the difference between two probability distributions. By characterizing the source and target distributions in terms of $f-$divergences, a more nuanced understanding of the domain shift can be achieved. This also enables

the design of more effective domain adaptation methods. To this end, the mathematical definition of the $f-$divergences is first discussed:

Definition 1 ($f-$divergence *Csiszár et al.* [39]) *Let p_s be absolutely continuous with respect to P_t and both be absolutely continuous with respect to a base measure d_x. Let $\phi : \mathbb{R}_+ \to \mathbb{R}$ be a convex, lower semi-continuous function that satisfies $\phi(1) = 0$*

$$D_\phi \left(P_s \| P_t\right) = \int p_t(x) \phi \left(\frac{p_s(x)}{p_t(x)} \right) dx. \tag{2.6}$$

where P_s and P_t are the source and target domain distribution functions with densities p_s and p_t respectively.

2.4 Autonomous Driving Dataset

2.4.1 Real-world Autonomous Driving Dataset:

The KITTI [11] dataset is often regarded as a seminal achievement in computer vision, offering support for a wide range of tasks such as object detection, tracking, semantic and instance segmentation, depth completion, SLAM, optical flow, and visual odometry. However, limitations such as the small sample size and mono scenes mostly captured in sunny weather conditions in one mid-sized city, Karlsruhe, Germany, prevent it from reaching its full potential. Recently, datasets such as nuScenes [17] have emerged, providing a more diverse and comprehensive representation of the real-world. It comprises larger data samples, each 20 seconds long, carefully selected to include challenging traffic scenarios and underrepresented object classes. Its data was collected in two cities, Singapore and Boston, across two different continents, and under various weather and lighting conditions. The Waymo Open Dataset [217] is another large-scale multi-modal and multi-task dataset collected across six US cities. Other datasets, such as BDD100K [279] and H3D [180], also provide valuable contributions to the field. In addition, CamVid [14], Cityscape [38] and Mapillary [173] are three popular image-based databases.

2.4.2 Synthetic Driving Data Generation:

As the demand for labeled data continues to grow in the field of autonomous driving, synthetic data has emerged as an cost effective alternative to collecting and annotating real-world data. There are three major approaches to generate synthetic dataset for perception related tasks, including the use of deep generative models [289] [130], sensor simulation-based data reconstruction and augmentation [159] [30], and computer graphics simulations [48] [194]. Previous works [48] [194] [159] [30] on computer graphics simulations are primarily focused on reducing cross-sensor domain shifts, such as photorealistic image generation. This thesis, in contrast, focuses on minimizing the cross-scene domain shifts. This is a typical and difficult challenge that occurs when synthetic data is acquired in an environment that differs from the real-world target domain. By addressing this issue, this thesis aim to improve the generalization of perception models trained on synthetic data.

2.5 3D Object Detection

3D object detection aims to locate, classify and estimate oriented bounding boxes in the 3D space. There are two main approaches for object detection: sequential and one-step. Sequential based models consist of a proposal stage and a 3D bbox regression stage in the chronological order. In the proposal stage, regions that may contain objects of interest are proposed. In the Bounding Box (bbox) regression stage, the region proposal is classified based on the region-wise features extracted from 3D information. However, the performance of sequential fusion is limited by each stage. On the other hand, one-step models consist of one stage, where 2D and 3D information is processed in a parallel manner. The timeline of 3D object detection networks and their corresponding fusion levels is shown in Figure 5. Table II and Figure 6 present comparative results of 3D object detection models on KITTI 3D Object Detection benchmark [67].

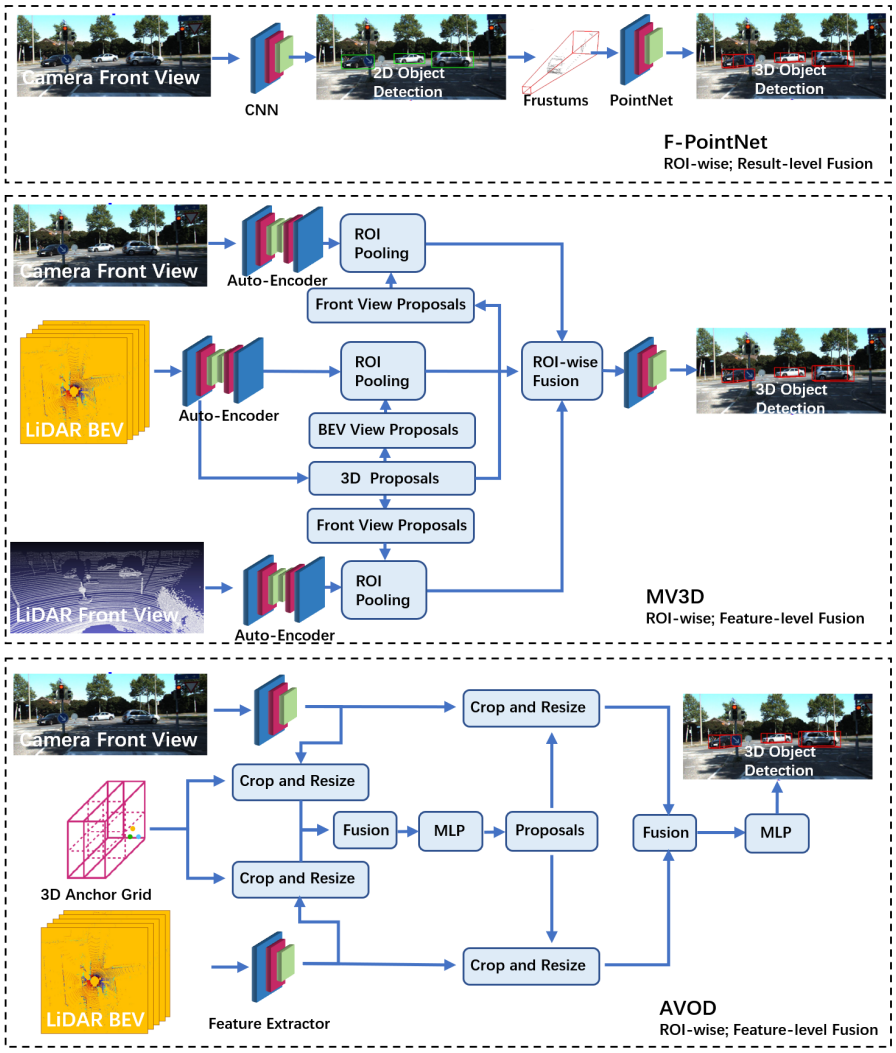


Figure 2.3: Some typical model architectures and fusion methods for dynamic object detection.

2.5.1 2D Proposal Based Sequential Models

A 2D proposal based sequential model attempts to utilize 2D image semantics in the proposal stage, which takes advantage of off-the-shelf image processing models. Specifically, these methods leverage the image object detector to generate 2D region proposals, which can be projected to the 3D space as detection seeds. There are two projection approaches to translate 2D proposals to 3D. The first one is projecting bounding boxes in the image plane to the 3D point cloud, which results in a frustum shaped 3D search space. The second method projects point cloud to the image plane, which results in the point cloud with point-wise 2D semantic information. However, far away or occluded objects are often represented by a handful of sparse points, making 3D bbox regression difficult.

Result-level Fusion

In these methods, information aggregation happens at the result-level. The intuition behind these methods is to use off-the-shelf 2D object detectors to narrow down the region of interests for the 3D object detector. The most common approach to achieve 2D bbox to 3D seed region is by reverse camera projection. This replaces the processing of the entire point cloud with multiple smaller regions of interest, which brought a significant reduction in computation and run-time. However, since the whole pipeline is depended on the results of the 2D object detector, the overall performance of this approach is very much limited by the performance of the 2D object detector. The core idea of result level fusion is not to use multimodal data to complement each other, but rather to reduce computation.

One of the early works of frustum-based method is F-PointNets [184], where 2D bounding boxes are first generated from image data and then projected to the 3D space. The resulting projected frustum proposal areas are fed into a PointNet[21] based detector for 3D object detection. Du et al. [53] extended the 2D to 3D proposals generation stage with an additional proposal refinement stage. During this refinement stage, a model fitting based method is used to filter out background points inside the seed region. Finally, the

filtered points are fed into the bbox regression network. This extra step further reduces unnecessary computation on the background point. RoarNet[203] follows a similar idea, but instead employs a neural network for the proposal refinement stage. Multiple 3D cylinder proposals are first generated based on each 2D bbox using Geometric agreement search [167], which results in smaller but more precise frustum proposals then the F-pointNet [184]. These initial cylinder proposals are then processed by a PointNet [186] based header network for final refinement. This approach outperformed SOTAs on KITTI testset, including scenarios where these sensors are not synchronized. However, these approaches assume each seed region only contains one object of interest, which is however not true for crowded scenes and small objects like pedestrians.

One possible solution towards the fore-mentioned issues is to replace the 2D object detector with 2D semantic segmentation and region-wise seed proposal with point-wise seed proposals. Intensive Point-based Object Detector (IPOD) [274] by Yang et al. is a work in this direction. In the first step, 2D semantic segmentation is used to filter out back-ground points. This is achieved by projecting points to the image plane and associated point with 2D semantic labels. The resulting foreground point cloud retains the context information and fine-grained location, which the author believes is essential for the region-proposal and bbox regression. In the following point-wise proposal generation and bbox regression stage, two PointNet++ [186] based networks are used for proposal feature extraction and bbox prediction. In addition, a novel criterion called PointsIoU is proposed to speed up training and inference. This approach has yielded significant performance advantages over other SOTA approaches in scenes with high occlusion or many objects.

Multi-level Fusion

Another possible direction of improvements is to combine result level fusion with feature level fusion, where one such work is PointFusion [272]. PointFusion first utilizes existing 2D object detector to generate 2D bboxes. These bboxes is used to select corresponding points, via projecting points to the image plane and locate points that pass through the

Table 2.1: Comparative results on KITTI 3D Object Detection benchmark (moderate difficulties)[67]. For cars the 3D bounding box IoU requirement is 70%.For pedestrians and cyclists the 3D bounding box IoU requirements are 50%.'PCR' stands for point cloud representation.FL stands for Fusion Level.FT stands for Fusion Type

Methods		FL	FT	PCR	Models	Run-time	Cars	Pedestrians	Cyclists
							AP_{3D}%	AP_{3D}%	AP_{3D}%
Sequential Models	2D Proposal based	Result	N/A	Points	F-PointNet [184]	0.17s	69.79	42.15	56.12
				Points	F-ConvNet [258]	0.47s	76.39	43.38	65.07
				Voxels	PC-CNN [53]	0.5s	73.79	-	-
				Points	RoarNet [203]	0.1s	73.04	-	-
				Points	IPOD [274]	0.2s	72.57	44.68	53.46
		Feature	Point	Multiple	PointPainting [243]	0.4s	71.70	40.97	63.78
		Multi	ROI	Points	PointFusion [272]	1.3s	63.00	28.04	29.42
			Point	Points	SIFRNet [293]	-	72.05	60.85	60.34
	3D Proposal based	Feature	ROI	2D views	MV3D [28]	0.36s	63.63	-	-
			ROI	2D views	AVOD-FPN [120]	0.08s	71.76	42.27	50.55
			ROI	2D views	SCANet [150]	0.17s	68.12	37.93	53.38
			Point	2D views	ContFuse [138]	0.06s	68.78	-	-
			Pixel	2D views	BEVF [255]	-	-	-	45.00
		Multi	Point	2D views	MMF [137]	0.08s	77.43	-	-
One-Step Models		Feature	Point	Voxels	MVX-Net [209]	0.15s	75.86	43.73	61.03
			Pixel	2D views	LaserNet++ [162]	0.04s	-	-	-

bboxs. Finally, a ResNet [79] and a PointNet[21] based network combines image and point cloud features to estimate 3D objects. In this approach, image features and point cloud features are fused per-proposal for final object detection in 3D, which facilitates 3D bbox regression. However, the proposal stage is still amodal. In SIFRNet [293], frustum proposals are first generated from an image. Point cloud features in these frustum proposals are then combined with their corresponding image features for final 3D bbox regression. To achieve scale-invariant, PointSIFT [106] is incorporated into the network. Additionally, SENet module is used to suppress less informative features.

Feature-level Fusion

Early attempts of multimodal fusion are done in pixel-wise, where 3D geometry is converted to image format or appended as additional channels of an image. The intuition is to project 3D geometry onto the image plane and leverage mature image processing methods to extract information. The resulting output, however, is also on the image plane, which is not ideal to locate objects in the 3D space. In 2014, Gupta et al. proposed DepthRCNN [76],a R-CNN [69] based architecture for 2D object detection, instance and semantic segmentation. It encodes 3D geometry from Microsoft Kinect camera in image's Red Green Blue (RGB) channels, which are Horizontal disparity, Height above ground, and Angle with gravity (HHA). Gupta et al. extended Depth-RCNN [75] in 2015 for 3D object detection by aligning 3D Computer-Aided Design (CAD) models, yielding significant performance improvement. In 2016, Gupta et al. developed a novel technique for supervised learning knowledge transfer between networks trained on image data and unseen paired image modality (depth image) [77]. In 2016, Schlosser et al. [200] further exploited learning RGB-HHA representations on 2D CNNs for pedestrian detection. However, the HHA data is generated from the LiDAR's depth instead of a depth camera. The authors also noticed that better results can be achieved if the fusion of RGB and HHA happens at deeper layers of the network.

To locate objects accurately in 3D, current works often employs point-wise fusion. In this approach, image features are appended to each point in the point cloud. This is achieved by projecting points to the image plane for the point-pixel association. PointPainting [243] proposed by Vora et al. follows the idea of projecting points to 2D semantic maps in [274], but instead performs point-wise fusion. However, instead of using 2D semantics to filter point clouds, 2D semantics is simply appended to point clouds as additional channels. The authors argued that this technique makes PointPainting flexible as it enables any point cloud based network to be applied on this fused data. To demonstrate this flexibility, the fused point cloud is fed into multiple existing point cloud detectors, which are based on PointRCNN[202], VoxelNet[294] and PointPillars[123].

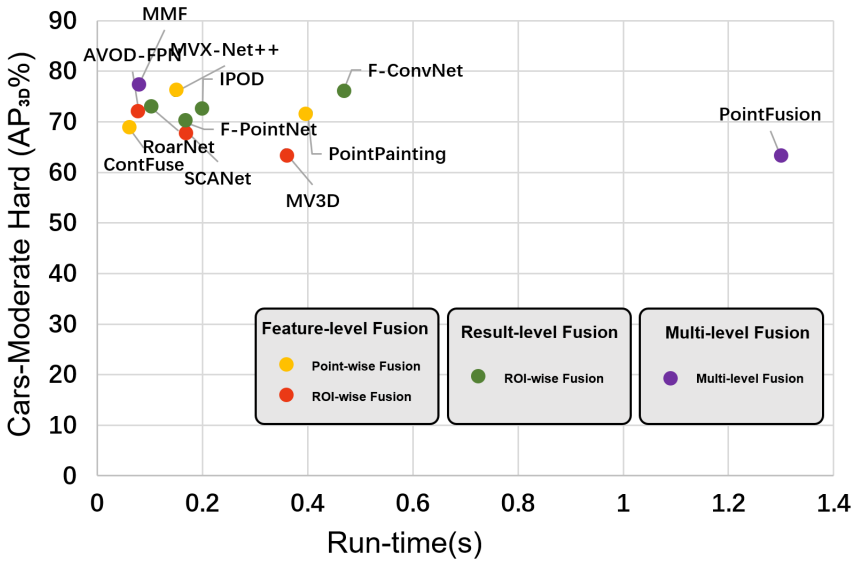


Figure 2.4: A scatter plot of models' average precision on 3D object detection(AP_3D) with respect to runtime. The results are based on the public available KITTI test set.

2.5.2 3D Proposal Based Sequential Model

In a 3D proposal based sequential Model, 3D proposals are directly generated from 2D or 3D data. The elimination of 2D to 3D transformation greatly narrows down the 3D search space for object detection. Common methods for 3D proposal generation includes the multi-view approach and the point cloud voxelization approach.

Multi-view based approach exploits the point cloud'sBird's Eye View (BEV) representation for 3D proposal generation. The BEV is the preferred viewpoint because it avoids perspective occlusions and retains the raw information of objects' orientation and x,y coordinates. These orientation and x,y coordinates information is critical for 3D object detection while making coordinate transformation between BEV and other views straightforward.

Point cloud voxelization transforms the continuous irregular data structure to a discrete regular data structure. This makes it possible to apply standard 3D discrete convolution

and leverage existing network structures to process point cloud. The drawback is the loss of some spatial resolution, which might contain fine-grained 3D structure information.

Feature-level fusion

One of the pioneering and most important works in generating 3D proposals from BEV representations is MV3D [28]. MV3D generate 3D proposals on pixelized top-down LiDAR feature map (height, density and intensity). These 3D candidates are then projected to the LiDAR front view and image plane to extract and fuse region-wise features for bbox regression. The fusion happens at region-of-interest (ROI) level via ROI pooling. The ROI can be defined as:

$$\mathrm{ROI}_v = \mathbf{T}_{3\mathrm{D}\rightarrow views}\left(p_{3\mathrm{D}}\right), views \in \{\mathrm{BV}, \mathrm{FV}, \mathrm{RGB}\} \tag{2.7}$$

where $\mathbf{T}_{3\mathrm{D}\rightarrow views}$ represents the transformation function from 3D space to BEV, front view and the image plane. The ROI-pooling can be defined as:

$$f_{views} = \mathbf{R}\left(x, \mathrm{ROI}_{views}\right), views \in \{\mathrm{BV}, \mathrm{FV}, \mathrm{RGB}\} \tag{2.8}$$

Although MV3D out-performed the SOTA models by a remarkable margin, there still are a few flaws. First, generating 3D proposals on BEV assumes that all objects of interest are captured without interference from this view-point and this LiDAR sensor. This assumption does not hold well for small object instances, such as pedestrians and bicyclists, which can be fully occluded by other large objects in the point cloud. Secondly, spatial information of small object instances is lost during the down-sample of feature maps caused by consecutive convolution operations. Thirdly, object-centric fusion combines feature maps of image and point clouds through ROI-pooling, which spoils fine-grained geometric information in the process. It is also worth noting that redundant proposals lead to repetitive computation in the bbox regression stage. To mitigate these challenges, multiple methods have been put forward to improve MV3D.

To improve small objects detection, Aggregate View Object Detection network (AVOD) [120] first improved the proposal stage in MV3D[28] with feature maps from both BEV point cloud and image. Furthermore, an auto-encoder architecture is employed to up-sample the final feature maps to its original size. This could alleviate the problem that small objects might get down-sampled to one 'pixel' with consecutive convolution operations. The proposed feature fusion Region Proposal Network (RPN) first extracts equal-length feature vectors from multiple modalities (BEV point cloud and image) with crop and resize operations. Followed by a $1X1$ convolution operation for feature space dimensionality reduction, which can reduce computational cost and boost up speed. Lu et al.[150] also utilized an encoder-decoder based proposal network with Spatial-Channel Attention (SCA) module and Extension Spatial Upsample (ESU) module. The SCA can capture multi-scale contextual information, whereas ESU recovers the spatial information.

One of the problems in object-centric fusion methods [120][28] is the loss of fine-grained geometric information during ROI-pooling. ContFuse [138] by Liang et al. tackles this information lost with point-wise fusion. This point-wise fusion is achieved with continuous convolutions [248] fusion layers which bridge image and point cloud features of different scales at multiple stages in the network. This is achieved by first extracting K nearest neighbour points for each pixel in the BEV representation of point cloud. These points are then projected to the image plane to retrieve related image features. Finally, the fused feature vector is weighted according to their geometry offset to the target 'pixel' before feeding into MLPs. However, point-wise fusion might fail to take full advantage of high-resolution images when the LiDAR points are sparse. In [137] Liang et al. further extended point-wise fusion by combining multiple fusion methodologies, such as signal-level fusion (Red Green Blue Depth (RGB-D)), feature-level fusion, multi-view and depth completion. In particular, depth completion upsamples sparse depth map using image information to generate a dense pseudo point cloud. This up-sampling process alleviates the sparse point-wise fusion problem, which facilitates the learning of cross-modality representations. Furthermore, the authors argued that multiple complementary tasks (ground estimation, depth completion and 2D/3D object detection) could assists the network achieve better

overall performance. However, point-wise/pixel-wise fusion leads to the 'feature blurring' problem. This 'feature blurring' happens when one point in the point cloud is associated with multiple pixels in the image or the other way around, which confound the data fusion. Similarly, wang et al. [255] replace the ROI-pooling in MV3D [28] with sparse non-homogeneous pooling, which enables effective fusion between feature maps from multiple modalities.

The simplest means to combine the voxelized point cloud and image is to append RGB information as additional channels of a voxel. In a 2014 paper by Song et al. [211] 3D object detection is achieved by sliding a 3D detection window on the voxelized point cloud. The classification is performed by an ensemble of Exemplar-SVMs. In this work, color information is appended to voxels by projection. Song et al. further extended this idea with 3D discrete convolutional neural networks [212]. In the first stage, the voxelized point cloud (generated from RGB-D data) is first processed by Multi-scale 3D RPN for 3D proposal generation. These candidates are then classified by joint Object Recognition Network (ORN), which takes both image and voxelized point cloud as inputs. However, volumetric representation introduces boundary artifacts and spoils fine-grained local geometry. Secondly, the resolution mismatch between image and voxelized point cloud makes fusion inefficient.

2.5.3 One-step Models

One-step models perform proposal generation and bbox regression in a single stage. By fusing the proposal and bbox regression stage into one-step, these models are often more computationally efficient. This makes them more well-suited for real-time applications on mobile computational platforms.

MVX-Net [209] presented by Sindagi et al. introduced two methods that fuse image and point cloud data point-wise or voxel-wise. Both methods employ a pre-trained 2D CNN for image feature extraction and a VoxelNet [294] based network to estimate objects from the fused point cloud. In the point-wise fusion method, the point cloud is first projected

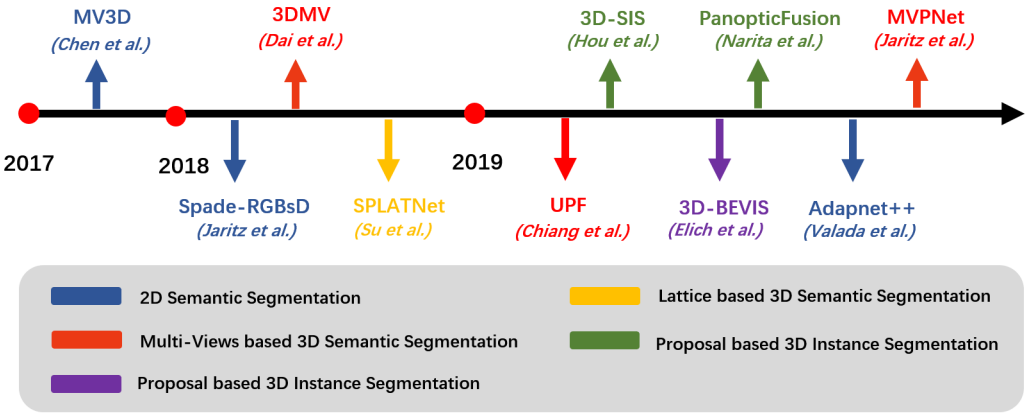


Figure 2.5: Timeline of 3D semantic segmentation networks and their corresponding fusion levels.

to image feature space to extract image features before voxelization and processed by VoxelNet. The voxel-wise fusion method first voxelized the point cloud before projecting non-empty voxels to the image feature space for voxel/region-wise feature extraction. These voxel-wise features are only appended to their corresponding voxels at a later stage of the VoxelNet. MVX-Net achieved SOTA results and out-performed other LiDAR-based methods on the KITTI benchmark while lowering false positives and false negatives rate compared to [294].

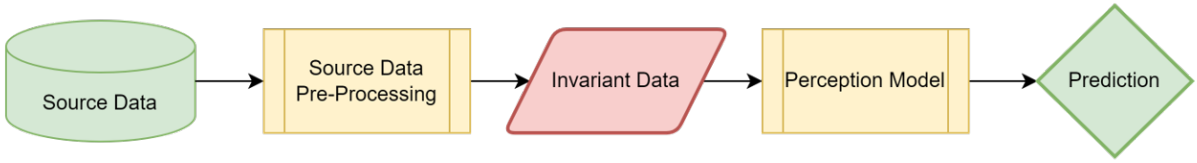
Meyer et al. [162] extended the LaserNet [163] to multi-task and multimodal network, performing 3D object detection and 3D semantic segmentation on fused image and LiDAR data. Two CNN process depth-image (generated from point cloud) and front-view image in a parallel manner and fuse them via projecting points to the image plane to associate corresponding image features. This feature map is fed into the LaserNet to predict per-point distributions of the bounding box and combine them for final 3D proposals. This method is highly efficient while achieving SOTA performance.

2.6 Domain Adaptation for Point Cloud Perception

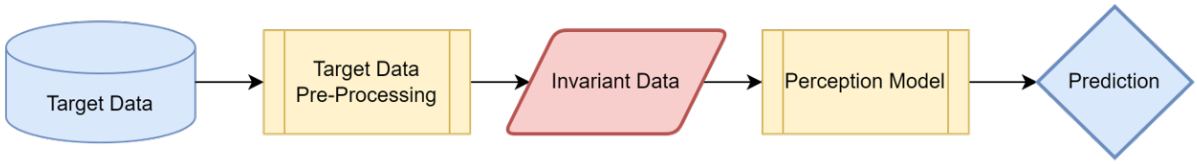
Domain Adaptation (DA) is a subset of transfer learning that aims to address the challenge of limited or absent target domain annotations. For instance, Simulation-to-reality gap is particularly prominent in autonomous driving, where annotated data from simulations is plentiful but real-world data is difficult and costly to collect and annotate. Based on the availability of the annotation, DA can be broadly classified into three categories: unsupervised transfer learning, transductive transfer learning, and inductive transfer learning. In unsupervised transfer learning, no annotated data is available for both the source and target domain. And DA is achieved through aligning the source and target domain distributions or to learning domain-invariant representations. In transductive transfer learning, annotations are only available in the source domain and the goal is to leverage this information to improve performance in the target domain through adversarial training. In inductive transfer learning, annotations are available in both the source and target domain. This allows fine-tuning a pre-trained model using target domain annotations to improve its performance.

In autonomous driving, the most common domain transfers are sim-to-real, sensor-to-sensor, and scene-to-scene. Sim-to-real domain shift occurs when models trained on synthetic data are applied to real-world data. Sensor-to-sensor DA, on the other hand, allows the same model to be applied to other sensors without retraining the model. Finally, scene-to-scene DA addresses the challenge of changing scenes and environments, where both foreground objects and background stuff could change substantially. This can be caused by variations in geographical regions, weather and lighting conditions, or traffic types.

When further narrowed down to point cloud-based perception, DA methods can be either data-driven or model driven. Data-driven approaches, such as domain-invariant data representation, domain mapping, and normalization statistics, use data transformation to align the source and target domains. These methods operate directly on the input data to reduce the discrepancy between the source and target domains. Model-driven approaches, on the other hand, focus on learning domain-invariant representations from the data. This



(a) During the training-time



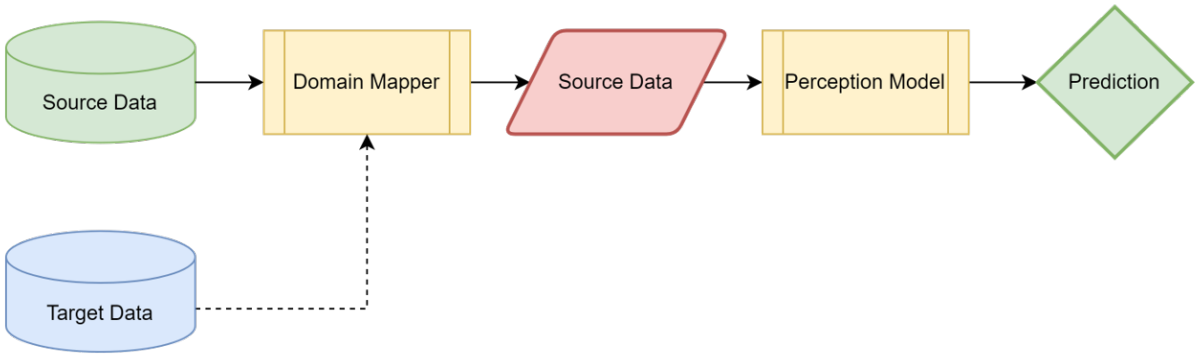
(b) During the testing-time

Figure 2.6: The workflow of Domain-Invariant Data Representation.

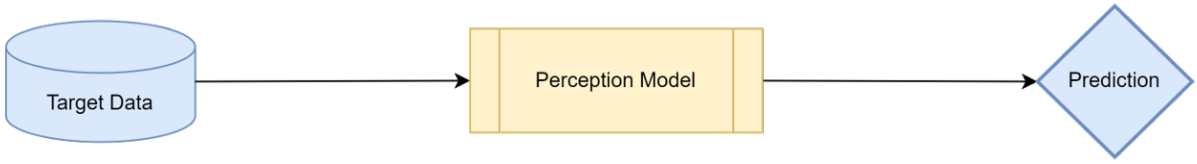
can be achieved by fine-tuning a pre-trained model, training with adversarial loss functions, or using domain-invariant feature learning. By learning domain-invariant representations, these approaches mitigate domain shift at the feature representation level of a model.

2.6.1 Data-driven Approach

Domain-invariant data representation transforms different domains into a common representation. It's workflow is shown in Fig. 2.6. In essence, it is a data pre-processing step that serves to overcome the differences between data distributions. This method is often utilised to address the sensor-to-sensor DA problem caused by different resolution and sampling patterns amongst LiDARs. To address this issue, [58] [201] [4] aligns these sampling disparities in 2D space. In particular, these methods often utilize 2d views representation of point cloud and align the disparities by either up-sampling or down-sampling. Alternatively, this domain-invariant data representation can be achieved in 3D space. [123] encodes point cloud features in vertical columns, where further analysis [182] has shown to



(a) During the training-time



(b) During the testing-time

Figure 2.7: The workflow of Domain Mapping.

be invariant against changes in LiDAR's vertical resolution. [276] exploits the underlying surface of a LiDAR scan, which is independent of sensors, to provide semantic predictions. Other methods include normalizing the input feature spaces and substituting relative coordinates encoding for absolute coordinates [192].

Domain mapping transforms the source domain data to resemble that of the target domain, commonly used in sim-to-real and dataset-to-dataset DA. An illustration of its workflow is shown in Fig. 2.7. Domain mapping transforms are often achieved by either Generative Adversarial Networks (GAN) or statistical approaches. However, most works for adversarially generating LiDAR data simply use stock image-based GANs by transforming point cloud to a BEV [198] [199] or a senor-view image [16] [170] [292].

2.6.2 Model-driven Approach

Domain-invariant feature learning models learn domain-independent feature representations by locating a common representation space for the source and target domains. There are two major approaches to domain-invariant feature learning, divergence minimization and discriminator-based. Divergence minimization methods minimize the difference between two domain representations within a DNN to produce a domain-invariant feature encoder. In [266], a DA training pipeline is proposed to reduce differences in the batch statistics between the source and target domains. [166] utilizes entropy minimization through optimal alignment of geodesic distance between source and target domains for unsupervised DA. xMUDA [105] is a cross-modal unsupervised DA method for 3D semantic segmentation. It utilizes information from both 2D images and 3D point clouds to address the heterogeneous input spaces and false predictions.The key concept behind discriminator-based techniques is to use adversarial training to induce the feature encoder to learn only domain-invariant features. In [256], the authors apply a discriminator to minimize domain gaps in the detection of far-away objects. This is accomplished by adapting the model on intermediate layers of a DNN, and is shown to be effective in dataset-to-dataset DA.

2.7 Open Challenges

How to balance speed-accuracy trade-offs?

The processing of high-resolution large-scale point clouds imposes substantial strain on existing mobile computing platforms. This leads to high power consumption and low reliability. However, the most concerning issue is the potential for frame drops, where some frames of data are not processed due to insufficient computing resources. This can significantly compromises the safety of the perception system. As such, it is crucial to strike a balance between speed and accuracy in real-world deployments of perception system.

Previous studies have attempted to solve this problem, such as frame drop detection

algorithm proposed in [98] that utilizes multiple segment fitting on camera pairs. However, simply detecting frame drops only addresses half of the issue. However, the most effective solution to the frame drop problem is to prevent it by reducing hardware workload through a careful balance of speed and accuracy [93]. This can be achieved through various techniques such as efficient model design, hardware optimization, and dynamic resource allocation.

How to mitigate domain shifts for open-world autonomous driving?

In the context of autonomous driving, domain shifts arise from shifting environmental conditions, such as weather, lighting conditions, road structures, and traffic patterns. This might resulting in sub-optimal performance of a model in previously unknown domains. Domain adaptation is often used to address the domain shift challenge. It enables a model trained on one set of data to be applied to a different collection of data. To tackle this challenge, it is critical to investigate and develop domain adaptation techniques that can effectively learn from the available training data and generalize to new domains. This includes, but is not limited to, methods such as domain randomization, domain adaptation with generative models, and transfer learning. Additionally, it is important to develop evaluation metrics and protocols that can accurately quantify the performance of the proposed methods in open-world domains.

2.8 Summary

This chapter presented an in-depth review of the most recent progress on deep learning models for 3D perception in the context of autonomous driving. Specifically, this review chapter covers topics in dataset, metrics, 3D object detection, domain adaptation. Furthermore, performance comparisons on the publicly available dataset, highlights and advantages/disadvantages of SOTA models were presented in tables. Typical model architectures

were shown in figures. Finally, open challenges that troubles real-world deployment of the autonomous driving technology is discussed.

Chapter 3

Task-attentive 3D Object Detection

This chapter proposes a novel task-attentive 3D object detection framework for autonomous driving. Section 3.1 starts with an overview of the background, including the challenges and limitations of existing methods. In Section 3.2, the methodology of the proposed adaptive ROI filter and semantic-based ground segmentation filter is explained. Section 3.3 describes the experimental settings, such as the dataset used, evaluation metrics, and implementation details. Finally, in Section 3.4, the proposed framework's experimental results and their quantitative and qualitative analysis are presented. These results demonstrate the effectiveness of our proposed approach over state-of-the-art methods. Different design choices and parameter settings of the proposed framework are also studied in this section. Overall, this chapter provides an in-depth study of the task-attentive 3D object detection framework and how it can be used to improve the performance of autonomous driving systems.

3.1 Background

LiDAR has played a central role in a variety of essential perception tasks, such as high-definition mapping, 3D localization, 3D object detection, 3D semantic segmentation, and

tracking [41] [133]. However, high-resolution LiDARs generate unprecedented amount of data, making real-time processing on mobile computing platforms extremely challenging. Furthermore, the processing delay caused by extensive point cloud data can cause serious safety issues [291] [215]. Another major challenge for point cloud processing is to detect small objects, which are often represented by a handful of points. More importantly, small objects are often closely located to the ground plane, making ground points and obstacle points challenging to distinguish. For instance, the Waterloo autonomous shuttle buses (WATonoBus) on the university campus often encounter small animals, particularly wild geese, on random road parts.

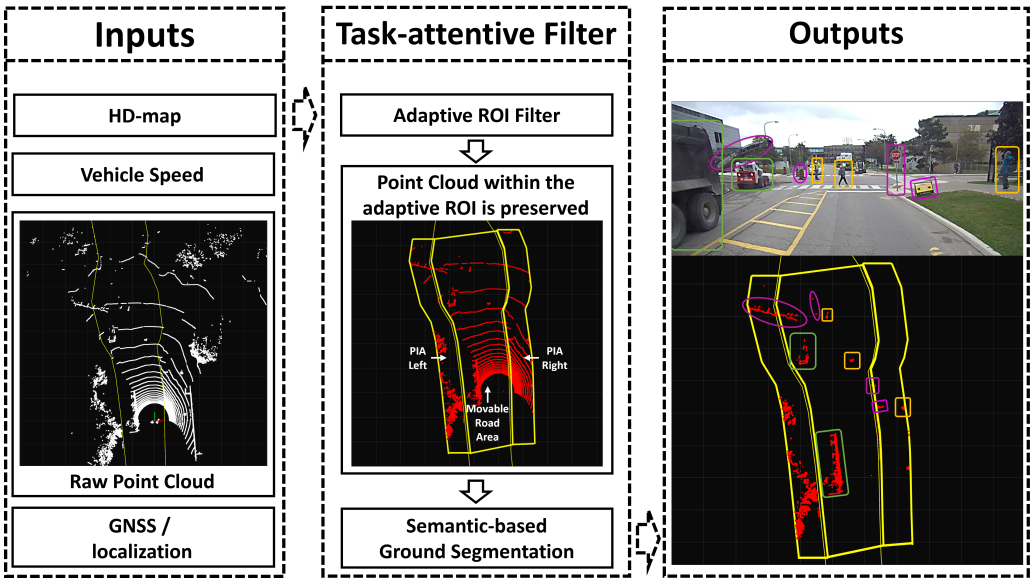


Figure 3.1: An overview of the proposed task-attentive LiDAR filter method.

To mitigate the above challenges, this chapter studies the LiDAR filter that removes unwanted point clouds located outside the ROI and belongs to the ground. More specifically, there are two main two objectives of this LiDAR filter. The first objective is to dynamically estimate ROI that reduces the number of point clouds while retaining a good perception field that ensures safety. The second objective is to accurately estimate the ground plane and separate non-ground points from ground points. The first objective reduces the com-

putation and memory workload for the perception tasks. The second objective is vital for the operation safety of the autonomous vehicle.

In this chapter, a novel LiDAR filter algorithm that combines HD-map, vehicle status and semantic-based ground segmentation to remove unwanted point clouds is proposed, including points outside ROI and ground points. Unlike existing LiDAR filters that produces a single ROI of fixed size [110] [1] [128] [158]. The proposed algorithm estimates region-of-interest (ROIs) of adaptive size and shape that consists of movable/road area and potential interactive area (PIA). After the adaptive ROI filtering, semantic-based ground segmentation is implemented. In this step, ROI is partitioned based on semantic priors from the HD map and ground planes are estimated independently for each region. Because regions with different semantics, such as road, sidewalk or green space, have drastically different surface geometry. Therefore, the proposed semantic-based ground segmentation approach can accurately estimate ground planes, preserving non-ground points from small obstacles. Furthermore, this thesis conducts detailed experiments to study the impacts of the proposed algorithm on real-world data acquired from WATonoBus projects, which operate in complex urban environments.

Researches in neurobiology suggest the human visual system is modulated by visual attention [109]. Human places visual fixation on salient regions to acquire relevant information, discarding irrelevant parts. Further studies on visual attention while driving [235] [236] [210] suggest experienced drivers allocate a majority of eye fixations on the frontal road area. This attention mechanism reduces task complexity as it saves 'computational bandwidth'. A conceptually similar concept is attention mechanism [241] [101] in machine learning, which allows the neural network to focus on a certain part of the information. However, it cannot take advantage of available prior knowledge, such as HD-map, to remove the unnecessary part of sensor inputs. This means the neural network still needs to process all the sensor inputs. Furthermore, due to the nature of supervised learning, the attention mechanism can only learn to focus on known objects that exist in the dataset [176]. As self-driving cars or autonomous robots operates in open environments, previously unknown objects might enter the road, causing unpredictable behaviour from deep learning

based perception system [68].

A more robust solution of removing irrelevant point clouds is ROI filters, which are an integral part of the most existing autonomous driving stacks. The existing ROI filters use HD-map and real-time localization to remove point cloud outside road area [110] [1] [128]. These filters are implemented using top-down 2D grids to reduce run-time. There are several drawbacks of existing ROI filters. Firstly, using road area as region-of-interest neglects dynamic objects outside the road, such as pedestrians, animals and other objects, that might suddenly enter the road. This reduces the reaction time of the planning and control system and could cause accidents. This problem could be partially solved by adding sidewalks into the ROI. However, pedestrians or animals often enter the road from non-sidewalk areas, such as green spaces, parking lots, .etc. Secondly, current ROI is one unified region that might include road surface and other terrains. The road's surface, sidewalk or green space are of different heights and flatness. Combing these different surfaces for ground estimation would cause errors where ground points are often not correctly removed [139]. These errors could cause problems for later perception tasks. Thirdly, the 2D grids based ROI filtering approach has difficulties handling complex road geometry with curves or irregular shapes.

Ground segmentation on point clouds is an essential task for autonomous driving. It is critical for (1) movable area estimation for path planning and (2) detect non-ground obstacles [84] [139]. By performing ground estimation, we can eliminate points that belong to the ground plane in a point cloud. After the ground point removal, obstacles can be easily distinguished via euclidean clustering algorithms [282] [8]. Furthermore, removing ground points also reduces computational costs for other perception tasks, such as 3D object detection, 3D semantic segmentation, and tracking [139]. To tackle complex road surface geometry, recent studies have employed region-wise ground segmentation, where the ground plane is divided into cartesian-grids [49] [7] or polar-grids/sectors [172] [84] [139]. It is worth noting that existing ground segmentation approaches partition ground region based on LiDAR scan patterns, not region-wise semantics.

3.2 Methodology

In this section, the problem definition and the designs of each individual modules of AGF is first introduced. The definition of a 3D point cloud generated by a LiDAR sensor in any given moment, is denote as $\mathbb{P}$. Each point cloud $\mathbb{P} = \{P_1, P_2, P_3, ..., P_N\}$ is a set of individual 3D points of size N. And each 3D points P_n is defined as $P_n = \{x_k, y_k, z_k, i_k\}$, where x_k, y_k, z_k is the local Cartesian coordinates centered at the LiDAR and i_k is the intensity value.

Each point cloud can be classified into two sub-sets based on their 3D location: points that are inside ROI R, and their complement R^c. In this thesis, the ROI points R consists of movable/road area R_m, left side potential interactive area R_{lp} and right side potential interactive area R_{rp}, which satisfy $R = R_m \cup R_{lp} \cup R_{rp}$. For ROI estimation, point cloud R within the adaptive ROI needs to be separated using information from HD map, real-time localization and vehicle states.

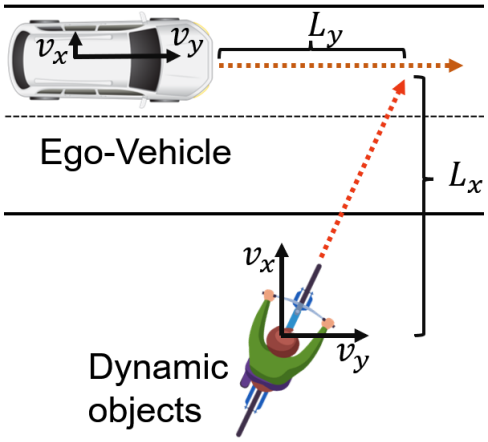


Figure 3.2: An example of dynamic object enters the road and potentially interact with the ego-vehicle.

Point cloud within the ROI R are further divided into ground points G and non-ground points G^c, which satisfy $R = G \cup G^c$. Ground points G are defined as LiDAR points that

belongs to the ground surface. And non-ground points G^c includes all objects above the ground surface, such as vehicles, pedestrian, plants .etc. The estimated non-ground points generated by ground segmentation algorithm is defined as $\hat{G}^c$, which could contain ground points by mistake. Similarly, the estimated ground points is denoted as $\hat{G}$. For HD map based ground segmentation, ground points $\hat{G}$ and non-ground points $\hat{G}^c$ with minimum errors need to be estimated.

Adaptive Region-of-Interest

In this section, an adaptive ROI filter is introuced, which estimates ROI dynamically using information from HD maps, real-time localization and vehicle states. The intuition behind the adaptive ROI filter is inspired by human driver behaviour. Existing studies [235] have suggested that experienced human driver adjusts their visual fixation area with different driving states. More specifically, the driver's main visual field is narrower and further away with an increase in the vehicle's speed. The logic behind the visual patterns of drivers is explained as follows. If the ego-vehicle travels at a faster longitudinal speed, it could afford to decrease the lateral width of ROI. This is because dynamic objects from the ego vehicle's lateral direction have less time to travel to the vehicle's path. For the same reason, the ego-vehicle should increase its ROI proportionally along its longitudinal direction.

These changes in the driver's visual field can be better understood with an example. As illustrated by Figure. 3.2, a dynamic object *obj* (cyclist) is moving towards the path of the ego-vehicle with a constant speed of v_x and V_y. the dynamic object's speed is denoted in the ego vehicle's coordinate system. It would take the cyclist $T_i = \frac{L_x}{V_x}$ to intercept with ego vehicle's path. That is, if the ego-vehicle reaches the interception point after T_i, the two objects could potentially interact with each other. Accordingly, if the ego-vehicle travels at a much faster speed and moves past this interception point before T_i, the two objects would have no opportunity to interact with each other.

Based on this intuition, this thesis proposes dynamically adjusting the size and shapes of ROI based on HD maps, real-time localization and vehicle states. This adaptive ROI

consists of road area and PIA. The vehicle states include longitudinal and lateral speed, heading angles, and breaking coefficients. The shape of the road area is acquired from the HD map. The length L_{road} of the road area is estimated in real-time and equals the length of the PIA.

potential interactive area

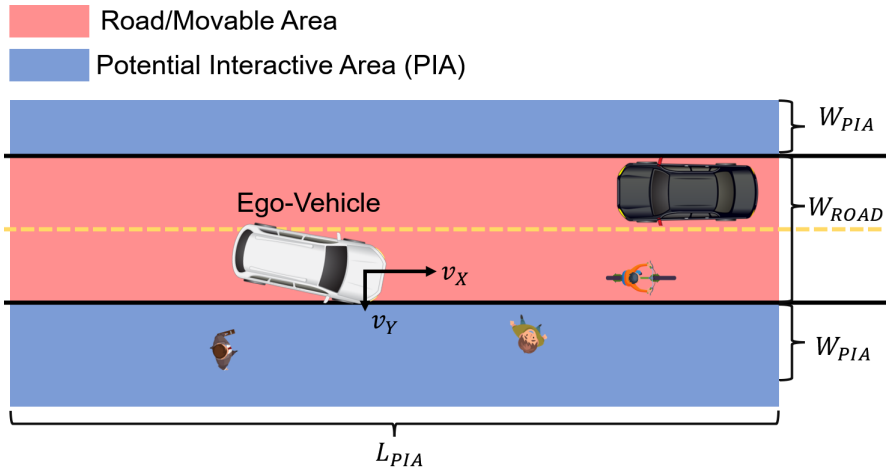


Figure 3.3: An example of adaptive ROI filter with road area and PIA.

The PIA is regions adjacent to the road area, containing dynamic objects that could enter the road and interact with the ego-vehicle. The size and shape of PIA are estimated in real-time. Because PIA is adjacent to the road area, the general shape of the PIA is similar to the road shape. This thesis denote the length and width of the PIA as L_{PIA} and W_{PIA}. The length and width of the PIA are calculated as follows:

$$L_{\text{PIA}} = abs(F\left(V_{\text{x-ego}}\right)) + L_{min} \tag{3.1}$$

where *abs* denotes a function to calculate absolute value, $F(.)$ denotes a function of the ego vehicle's speed, L_{min} is the minimum longitudinal safety distance, W_{min} is the minimum lateral safety distance. The length L_{PIA} and width W_{PIA} of the PIA is a non-linear function



(a) (b)

Figure 3.4: (a) An illustration of the sensor setups on the WATonoBus. (b) The WATonoBus.

of the vehicle's longitudinal and lateral speed. If $F(.)$ outputs a smaller value than the predefined minimum safety distance, the minimum safety distance will be employed. This predefined minimum safety distance L_{min}, W_{min} guarantees the minimum perception range. The width W_{PIA} of the PIA is defined as follows:

$$W_{\text{PIA}} = \begin{cases} F\left(V_{\text{y-ego}}\right) + W_{min}, & if F\left(V_{\text{y-ego}}\right) \geq 0 \\ W_{min}, & otherwise \end{cases} \tag{3.2}$$

where the width W_{PIA} of the left side and the right side PIA is independent and can be different. This is illustrated in Fig.3.3, where the right side PIA is wider than the left side PIA. This is because the lateral speed of the ego-vehicle is in the direction of the right side. The intuition is that the ego-vehicle needs to increase its perception field to the direction it travels to. The $F(.)$ function is defined as follows:

$$F\left(V_{\text{ego}}\right) = \begin{cases} k * h_{totoal}\left(V_{\text{ego}}\right) & if V_{\text{ego}} \geq 0 \\ -k * h_{totoal}\left(V_{\text{ego}}\right) & otherwise \end{cases} \tag{3.3}$$

where k denotes predefined adjustable coefficient (larger than 1) based on vehicle types.

The total stopping distance $h_{totoal}(.)$, denotes a function that maps the vehicle speed to the total stopping distance. The intuition behind $F(.)$ is that the perception range of the vehicle should be more extensive than its total braking distance. This distance enables the ego-vehicle to stop when an obstacle is detected fully. Different types of vehicles, such as buses and passenger cars, often have different safety requirements. Therefore, the coefficient k is used to adjust the braking distance to meet the safety requirements. The total stopping distance $h_{totoal}(.)$ function is defined as follows:

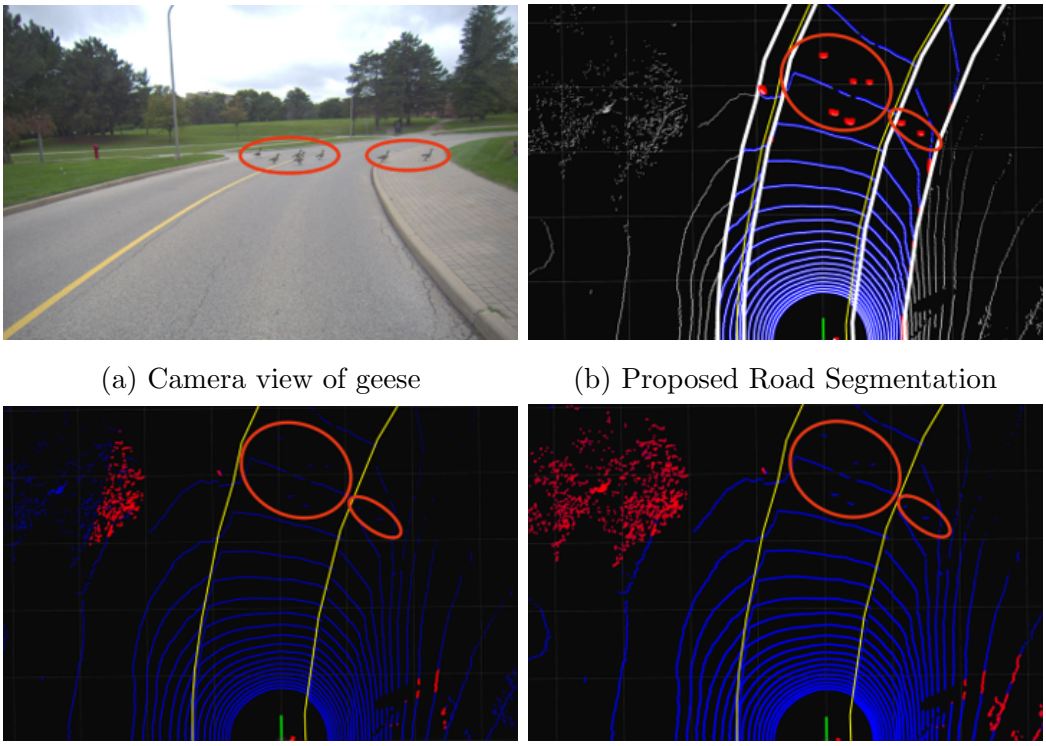
$$h_{total}\left(V_{\text{ego}}\right) = h_{S-R} + h_{brake} = v_{ego}t_{S-R} + \frac{v_{ego}^2}{2\mu g} \tag{3.4}$$

where h_{S-R} represents the system reaction distance function, and h_{brake} represents the ego-vehicle breaking distance function. The system reaction distance h_{S-R} is calculated using the autonomous driving system reaction time t_{S-R} multiplied with the current ego-vehicle speed. The autonomous driving system reaction time t_{S-R} is the system time delay between the time when the sensor first detected the obstacle and the time the system reacts. This system time delay is determined by both autonomous driving software and hardware. This system time delay can be measured with offline testing using simulation or recorded data. The breaking distance h_{brake} is a function of ego vehicle's speed v_{ego}, the coefficient of friction μ, and the gravity of Earth g.

Semantic-based ground segmentation

After removing points outside the region-of-interest, ground segmentation is applied to remove ground points. This ground segmentation is vital for detecting small, unknown obstacles using clustering based algorithm. It also provides movable plane estimation for the planning algorithm.

As discussed in the previous section, existing region-wise ground segmentation approaches partition ground region using cartesian-grids or polar-grids/sectors [172] [84] [139] [49] [7]. That is, these regions do not have semantic meanings. It is worth noting that



(a) Camera view of geese

(b) Proposed Road Segmentation

(c) Unified ROI Road Segmentation

(d) No ROI Road Segmentation

Figure 3.5: (a) Camera view of geese crossing the road. (b) The proposed Semantic-based Road Segmentation. (c) Road Segmentation on Unified ROI. (d) Road Segmentation on Raw Point Cloud. White points represents the raw point cloud inputs. Blue points represents the detected ground points. Red points represents the non-ground points. Please note traditional ground segmentation algorithms cannot distinguish the geese from the ground. Small animals, such as goose and squirrels, entering or crossing happens frequently in the WATonoBus operation environment.

different semantic region often has different ground surfaces geometry and topology. For instance, road, sidewalk, green space and parking lots, all have drastically different ground surface geometry and topology. However, the previous approach does not consider the semantic differences in region partition, which leads to unnecessary complexity in ground plane estimation. To address this issue, this thesis proposes to combine semantic information provided by HD map for ground segmentation tasks.

More specifically, regions with different semantic meanings are partitioned for independent ground plane estimation. This allows different semantic regions to have different ground plane models. This semantic region partition scheme is defined as follows:

$$\mathcal{R} = \bigcup_{i \in \langle Map \rangle} Region_i \tag{3.5}$$

where $\langle Map \rangle$ denotes different semantic regions provided by the HD map, $Region_i$ is a area that has identical semantic type, R is the ROI.

To further address complex road geometry that might exist within each semantic region, this thesis proposes combining semantic-wise region division with traditional grid-wise ground segmentation. More specifically, the ground surface is further partitioned into small grids for ground extraction within each semantic region. For this thesis, a polar grids partition method [139] is utilized for ground segmentation within each semantic region. This process is defined as follows:

$$Region_i = \bigcup_{i \in \langle N \rangle} Grid_i \tag{3.6}$$

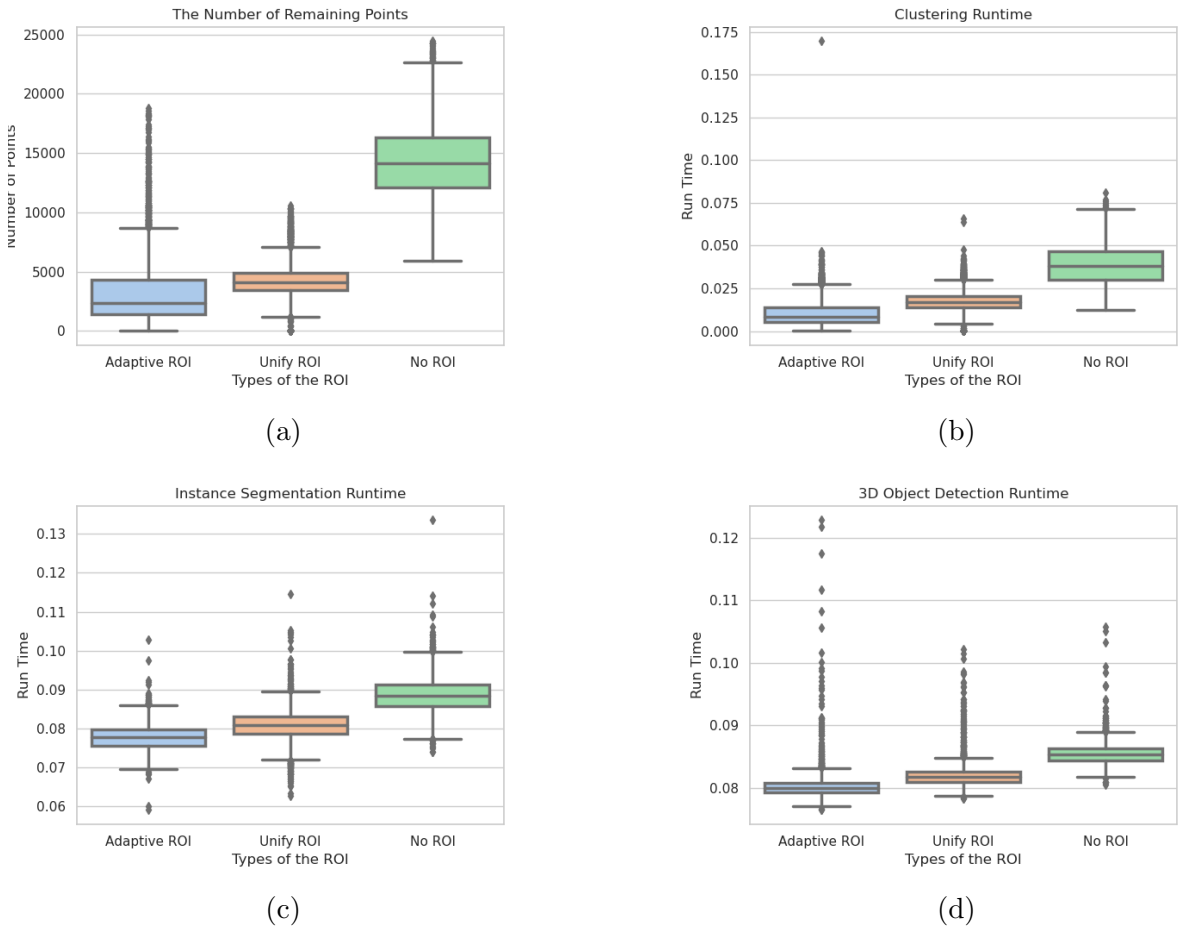


Figure 3.6: (a) The number of remaining points, (b) The run time of clustering algorithm, (c) The run time of ApolloSeg (d) The run time of PointPillars

3.3 Experiments

3.3.1 Testing Platform

The proposed algorithm is tested on the data collected from the Waterloo all-Weather autonomous shuttle (WATonoBus) project, shown in Figure 3.4. The WATonoBus project from the University of Waterloo consists of three autonomous shuttle buses. Each shuttle bus is equipped with a network of advanced sensors, including LiDARs, cameras, radars, a Global Navigation Satellite System (GNSS), an Inertial Measurement Unit (IMU) and 5G communication. The main LiDAR on WATonoBus is a mechanical LiDAR of 32 leaser beams which operates on 10 Hz. This main LiDAR is installed on the front of the shuttle bus, which provides frontal 180-degree FOV. The real-time localization module of the shuttle bus consists of high-precision IMU and GNSS system with Real-Time Kinematic (RTK). This localization system is capable of providing a centimetre-level global position. The computation platform for the perception task on WATonoBus is an NVIDIA Jetson Xavier computer.

3.3.2 Dataset

The dataset acquired from WATonoBus contains recorded sensor measurements from LiDAR, camera, GNSS, IMU, odometry and HD map. The WATonoBus operates on the university's ring road, a 2.7-kilometre route around the campus. The ring road is the main route within the University of Waterloo, which has heavy traffic throughout the day and many vulnerable road users. For instance, pedestrians and geese regularly cross the road without any warnings on many road parts. The ring road also has complex road topologies. More specifically, the road has steep curves and slopes, challenging for road surface estimation. The areas adjacent to the road are a mixture of green space, sidewalk, parking lot, bus stops and building entrances. The dataset contains 11960 frames of LiDAR point cloud, frames images.

Table 3.1: The number of remaining points after applying different types of ROI filter.

ROI types	The number of points	
	μ	σ
No ROI	14339	3128
Fixed Unify ROI	4154	1390
Adaptive ROI	3208	2603

3.4 Results and Discussion

3.4.1 The impacts on perception run-time

One of the ROI filter goals is to reduce the computation and memory complexity of the LiDAR perception task by removing unrelated points. In this section, the impacts of different ROI filters on common LiDAR-based perception tasks is analyzed. This thesis choose three of the most popular algorithms for LiDAR-based perception tasks. For 3D object detection, the euclidean clustering algorithm and PointPillars network [124] are tested. For 3D instance segmentation, ApolloSeg instance segmentation network [1] is tested. The euclidean clustering is a class-agnostic object detection algorithm that groups points that are located closely. It is used to detect both known and unknown objects from the LiDAR point cloud. The PointPillars and ApolloSeg are lightweight deep learning based models designed for real-time applications. All algorithms are tested on NVIDIA Jetson Xavier computer.

As shown in Fig. 3.6a, compared to not applying the ROI filter, both the proposed adaptive ROI and the traditional unified ROI significantly reduced the total number of points for later perception tasks. As shown in Table 3.1, the proposed adaptive ROI filter on average has 5.6 times fewer data points compared to the raw point cloud. Compared to fixed unified ROI, the proposed adaptive ROI removes slightly more points per scan on average.

Table 3.2: The run time of different perception tasks after applying different types of ROI filter.

ROI Types	Method	Frequency	Run-time (S)	
			μ	σ
No ROI	Eucliean Cluster	26.0 Hz	0.0385	0.0113
	PointPillar	11.7 Hz	0.0855	0.0017
	ApolloSeg	11.3 Hz	0.0886	0.0040
Fixed Unify ROI	Eucliean Cluster	57.8 Hz	0.0173	0.0058
	PointPillar	12.2 Hz	0.0821	0.0021
	ApolloSeg	12.4 Hz	0.0807	0.0034
Adaptive ROI	Eucliean Cluster	99.0 Hz	0.0101	0.0073
	PointPillar	12.4 Hz	0.0804	0.0026
	ApolloSeg	12.9 Hz	0.0777	0.0029

The reduction of unrelated points has a significant impact on the run-time of the euclidean clustering algorithm. Its run-time is reduced by 3.8 times after adaptive ROI filtering. This means the clustering algorithm could operate at 99 Hz, a drastic increase compared to its original 26 Hz operation frequency. Deep learning based perception models, such as PointPillar and ApolloSeg, also witness some reduction in run-time. However, the reduction is less significant than the euclidean clustering algorithm. This is because PointPillar and ApolloSeg transform point cloud into a fix size feature space, which cannot benefit from the sparsity of points. Therefore, the run-time reduction mainly comes from the pre-processing steps in these models.

3.4.2 The impact on ground segmentation

Effectiveness on small objects

The WATonoBus operates in the university campus, which has large areas of green space. The WATonoBus frequently encounters small animals, such as geese, squirrels, dogs, raccoons, entering the road in random parts of the route. Apart from animals, portions of the operating route are sometimes partially closed with small traffic cones for construction. These scenarios present unique challenges for the operation safety of the vehicle. It is vital for the perception system to detect these small obstacles from as far as possible to avoid sudden breaks or traffic accidents. However, these small unknown objects are very challenging for deep learning based systems, which are class-aware detectors. Deep learning models are trained to detect a predefined set of object classes. Furthermore, it is impossible to build an annotated dataset that includes all the potential object classes the shuttle bus might encounter [216]. This dilemma is often referred to as the open-set detection problem.

This thesis employ a euclidean clustering algorithm to solve these edges cases, which is a class-agnostic object detector. The euclidean clustering algorithm detects obstacles on the point cloud by grouping points that are located closely. The euclidean clustering algorithm's prerequisite for detecting small objects is accurate ground point removal. This is because the euclidean clustering algorithm cannot distinguish ground points and non-ground points. If ground points are not fully removed, these remaining ground points will be treated as obstacles in the clustering algorithm. This would cause the shuttle bus to stop or maneuver incorrectly. On the other hand, if non-ground points are incorrectly removed, the shuttle bus might run into these obstacles. More importantly, small obstacles are often closely located to the ground plane. Therefore, accurate ground plane estimation is vital for the vehicle's safe operation.

Fig.3.5a illustrates a comparison of different road segmentation results on geese. In this scenario, geese are on the road and the sidewalk. It is worth noting that the road

and sidewalk are of different heights, and a steep slope is next to the sidewalk. As shown in Fig.3.5c and Fig. 3.5d, traditional road segmentation approach can identify most of the ground point correctly. However, due to the need to fit complex ground geometry of different regions, small objects, such as geese, are treated as uneven ground points. This problem is solved by the proposed semantic-based ground segmentation, which is shown in Fig. 3.5b.

3.5 Summary

In this chapter, a task-attentive LiDAR filter is proposed. The proposed adaptive ROI filter significantly reduces computation and memory cost for later perception tasks while ensuring driving safety. The proposed semantic-based ground segmentation can accurately distinguish between ground and non-ground points in complex environments. This enables the perception modules to detect small objects near the ground surface, which is challenging otherwise.

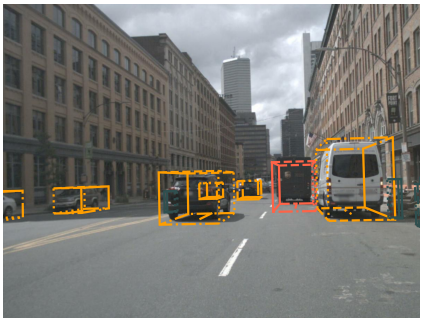
Chapter 4

Domains-3D: A Novel Domain Adaptation Dataset

This chapter introduces Domains-3D, a novel domain adaptation dataset developed for autonomous driving applications. Domains-3D has 18 domains and 54 potential cross-domain splits. It addresses the limitations of existing datasets by providing a comprehensive, representative, and challenging benchmark for assessing the generalization capabilities of 3D perception models for autonomous driving. Section 4.1 begins with an overview of the background and challenges of synthetic data generation and domain adaptation for autonomous driving. Section 4.2 then presents the methodology and mathematical theorem used to calculate domain shifts between two domains with respect to the task at hand. This information is then used to guide the synthetic data generation process and reduce domain shifts between the synthetic and target domains. Section 4.3 describes the dataset's construction, which begins with an in-depth analysis of the target domains built on the nuScenes dataset.



(a) Simulated Urban Domain



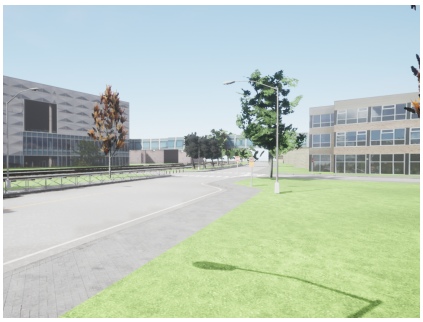
(b) Real Urban Domain



(c) Simulated suburban/rural Domain



(d) Real suburban/rural Domain



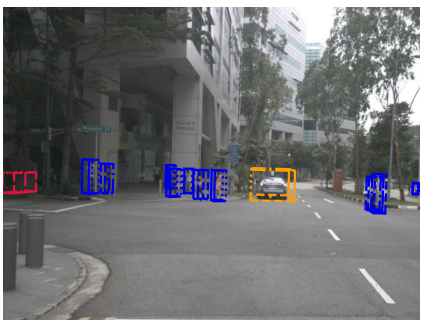
(e) Simulated Campus Domain



(f) Real Campus Domain



(g) Simulated Public Domain



(h) Real Public Domain

4.1 Background

Collecting and labeling a large-scale autonomous driving dataset is extremely expensive, time-consuming and labor intensive. As a result, only a limited number of large-scale open-source datasets are available. These challenges also prevents many researchers from creating their own dataset that best suited their needs and applications. By comparison, generating synthetic data using a autonomous driving simulator offers a variety of benefits. It's cost-efficient, provides better annotation accuracy, and allows the creation of costumed and repeatable scenarios. While synthetic data has the potential to overcome many of the challenges associated with creating an autonomous driving dataset in the real-world, there are still difficulties to overcome. One of the these lays in the precise simulation of real-world sensor characteristic and behavior. Additionally, the simulation of complex real-world scenarios, including both fore-ground and background objects presents another major challenge [275] [159] [30]. This is especially relevant in the context of domain adaptation studies, which often neglects the discrepancy between synthetic-to-real scenarios. For instance, the number and spatial locations of foreground objects are often generated at random.

Domain adaptation is of paramount importance for autonomous driving research, as self-driving cars need to adapt to a wide range of environments and conditions in order to drive safely and reliably. For instance, models trained on nuScenes [17] which is collected in Boston and Singapore, could perform poorly in Canadian cities and rural area [145] [144] [148] [149] [107]. However, the absence of a comprehensive domain adaptation dataset for autonomous driving and the limit diversity of available autonomous driving datasets has become the bottleneck for studies in this field.

This thesis addresses the above challenges by introducing a novel comprehensive domain adaptation dataset for autonomous driving. A detailed comparison between the proposed Domains-3D and other autonomous driving datasets is presented in Table.4.1. In addition, this thesis proposes a novel approach for synthetic data generation using real-world prior that minimize the divergence between the synthetic domain and their real-world

Table 4.1: Comparison of existing autonomous driving datasets that provide point cloud data. Domains-3D stands out as a large-scale synthetic-real dataset with the most broad range of domains of any extant dataset. It is a one-of-a-kind dataset that allows for domain adaptation research on cross-sensor, cross-scene, and sim-to-real domain shifts.

Type	Dataset	Cross-sensor	Cross-scene	Cross-Scene Types	Simulate to Real	Num of seq	Annotated frames		
							Seg.	2D Det.	3D Det.
Real-world	KITTI [11]	N/A	N/A	N/A	N/A	22	200	15k	15k
	H3D [180]	N/A	4	city2city	N/A	160	-	-	27k
	A*3D [181]	N/A	N/A	N/A	N/A	-	-		39k
	nuScenes [17]	N/A	2	city2city	N/A	1,000	40k	40k	40k
	Waymo [217]	N/A	3	city2city	N/A	1,150	200k	200k	230k
Synthetic	PreSIL [97]	N/A	N/A	N/A	N/A	N/A	50k	50k	50k
	AIODrive [262]	N/A	8	city2city	N/A	100	100k	100k	100k
	SHIFT [219]	N/A	8	city2city	N/A	4,850	2.5M	2.5M	2.5M
Synthetic+Real	Domains-3D	**4**	**18**	city2city, **urban2rural**, **public2campus**, **Sim2Real**, **Sim2Sim**	**Yes**	7,000	2.1M	2.1M	2.1M

counterparts. This, in turn, leads to better knowledge transfer between the synthetic and real domains and improved performance of autonomous driving systems. The proposed Domains-3D contains both real-world data and synthetic data. The real-world data was extracted from the nuScenes dataset and carefully annotated to distinguish domains between cities, urban, and suburban/rural environments, as well as between public roads and a campus. Based on this real-world prior, synthetic data of respective domains are reproduced to mimic their real-world counterparts. Additionally, this thesis also include simulated domains without access to real-world data to provide a more comprehensive dataset. To sum up, the proposed dataset contains a total of 18 domains, and 54 possible cross-domain splits to provide a diverse set of scenarios for autonomous driving systems to adapt to. In addition to the cross-scene domain adaptation, cross-sensor domain adaptation is also added to further enhance its diversity and usefulness. This comprises a total of 4 cross-sensor domains and 12 possible cross-sensor domain splits.

4.2 Methodology

4.2.1 Problem Formulation

In this research, the overriding research question is: how to generate challenging synthetic data for autonomous driving perception tasks that closely resembles the target domain. To formally define this problem, this thesis first introduce the definitions of key terms and essential components. This is followed by the break down of this research problem into two sub-tasks: synthetic domain generation and scene-related noise generation. Finally, the methodologies used to tackle these tasks are explained in detail.

To start, this thesis refer to the synthetic dataset generated by a simulator as S, which consists of labeled data points $S = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$. The target domain ($T$) is also labeled and consists of $T = \{(x_i^t, y_i^t)\}_{i=1}^{n_t}$. The target data xi^t and synthetic data x_i^s are sampled independently and identically distributed (i.i.d.) from data distributions P_t and P_s over the input space $\mathcal{X}$, respectively. As is commonly the case in autonomous driving simulators, this thesis assumes the computer graphics simulator can automatically produce labels for the task of interest. And the labeling function $f : \mathcal{X} \to \mathcal{Y}$ represents the mapping from the input space $\mathcal{X}$ to the output space $\mathcal{Y}$, and is denoted as f_t for the target domain and f_s for the synthetic domain.

In the context of autonomous driving, the input space $\mathcal{X}$ can be represented by various sensor measurements such as lidar, radar, and camera data. By utilizing accurate sensor models in the computer graphics simulation, the synthetic data generated can closely resemble the real-world sensor measurements. Nevertheless, this research focuses on cross-scene domain adaption problem and assumes the correct sensor model is available in the simulator. The label y_i in $S = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ represents the output or prediction of the task of interest, which can vary depending on the specific task being addressed. For example, label y_i of 3D object detection task might represent the bounding box coordinates and class label of the detected objects in the scene.

Problem 4.2.1 (Domain Adaptation) *Given a hypothesis h, and its loss function*

$\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}+$*under distribution* $\mathcal{D}$*. The risk of a hypothesis* h *w.r.t. the labeling function* f *is:*

$$R^{\ell}_{\mathcal{D}}(h, f) = \mathbb{E}x \sim \mathcal{D}[\ell(h(x), f(x))] \tag{4.1}$$

To simplify, this thesis further define $R^{\ell}_{S}(h) = R^{\ell}_{P_s}(h, f_s)$ and $R^{\ell}_{T}(h) = R^{\ell}_{P_t}(h, f_t)$. For domain adaptation, the intend is to learn a function $h : \mathbb{R}^d \rightarrow \mathcal{Y}$ that can generalize well on both the source domain S and target domain T, when trained on the source domain S.

Theorem 4.2.1 (Zhao et al. [290]). *Suppose that* $D_{\text{JS}}\left(P_s(y) \| P_t(y)\right) \geq D_{\text{JS}}\left(P_s(z) \| P_t(z)\right)$ *These is:* $R^{\ell}_{T}(h) + R^{\ell}_{S}(h) \geq \frac{1}{2}\left(\sqrt{D_{\text{JS}}\left(P_s(y) \| P_t(y)\right)} - \sqrt{D_{\text{JS}}\left(P_s(z) \| P_t(z)\right)}\right)^2$ *where* D_{JS} *corresponds to the Jensen-Shanon divergence.*

For this research, Theorem 4.2.1 holds great significance, as it emphasizes the importance of proper data generation in the simulator to avoid mismatch between real and virtual worlds. Specifically, theorem 1 states that if the sampled simulated foreground objects' distribution is different from the target domain (e.g. real world), SOTA adversarial learning algorithms might not work. In other words, the synthetic data needs to resemble its real-world counterparts to a certain extend in order to facilitate knowledge transfer between the two domains. This emphasizes the importance of manipulating the data generation process to reduce the mismatch between the synthetic and the real-world domains.

4.2.2 Synthetic Domain Generation Using Real-wold Prior

In a given domain, it exists a set of ground truth agents of interest $\mathcal{A}_f = \{a_1, a_2, a_3, ..., a_N\}$ determined by the labeling function of interest f. These ground truth agents can be described by a set of states $\mathcal{S} = \{s_1, s_2, s_3, ..., s_N\}$ indicating their sizes, spatial positions and locations. Similarly, the labeling function f predicts this state description $\mathcal{S}$ given sensor observation $\mathcal{X}$.

In this research, it is assumed the labels of the target domain is generated by human experts, which can be viewed as a ground truth labeling function f_{gt}^{t}. Similarly, the labeling function f_{gt}^{s} on the synthetic domain is obtained directly from the simulator's ground truths. Therefore, the dissimilarity between f_t and f_s could be viewed as negligible.

The divergence between the data distributions $P_s(x)$ and $P_t(x)$ of the source (i.e. simulator) and target (i.e. real-world) domain can be measured by the KL divergence:

$$D_{KL}(P_s(x)||P_t(x)) = \sum_x P_s(x) \log \frac{P_s(x)}{P_t(x)} \tag{4.2}$$

Similarly, the divergence between the marginal distributions $P_s(y)$ and $P_t(y)$ of the source (i.e. simulator) and target (i.e. real-world) domain can be measured by the KL divergence:

$$D_{KL}(P_s(y)||P_t(y)) = \sum_y P_s(y) \log \frac{P_s(y)}{P_t(y)} \tag{4.3}$$

Based on these assumptions, the correlation between the KL divergence of marginal distributions $P_s(y)$ and $P_t(y)$ and the KL divergence of data distributions $P_s(x)$ and $P_t(x)$ is:

$$D_{KL}(P_s(y)||P_t(y)) \propto D_{KL}(P_s(x)||P_t(x)) \text{ if } \lambda\left(f_{gt}^{t}, f_{gt}^{s}\right) \approx 0 \tag{4.4}$$

where D_{KL} represents the KL divergence between the two distributions and λ represents the dissimilarity between the ground truth labeling functions f_{gt}^{t} and f_{gt}^{s}. This equation suggests that if the dissimilarity between the ground truth labeling functions is negligible, then the divergence between the marginal distributions $P_s(y)$ and $P_t(y)$ is positively correlated with the divergence between the data distributions $P_s(x)$ and $P_t(x)$.

To put it more simply, if the dissimilarity between the ground truth labeling functions is small, then a small divergence between the labeling distributions $P_s(y)$ and $P_t(y)$ indicates a small divergence between the data distributions $P_s(x)$ and $P_t(x)$.

Problem 4.2.2 (Synthetic Domain Generation) *Given a set of sample labels $P_t(y)$ from the target domain T, and the ground truth labeling function f^s_{gt} of simulator S, the task of interest f, the synthetic domain generation solves*

$$\min_{\mathcal{S}}[D_{KL}(P_s(y)||P_t(\mathcal{S})] \tag{4.5}$$

The synthetic domain generation problem can be expressed as a optimization problem:

$$\begin{aligned} \mathcal{S}'^* \in \arg\min_{\mathcal{S}} & [D_{KL}(P_s(y)||P_t(\mathcal{S})] \\ \text{s.t. } & \mathcal{S} \in \mathcal{A}_f \end{aligned} \tag{4.6}$$

In a simulator, various aspects ground truth objects can be precisely manipulated with a given configuration π, including the classes that appear, their locations, frequency and timing of appearance. Therefore, using the ground truth labeling function f^s_{gt} to generate a particular marginal distributions $P_i(y)$ on a given simulator S can be expressed as:

$$\mathcal{S} = f^s_{gt}(\mathcal{A}_f; \pi), \pi \in P_i(y) \tag{4.7}$$

4.2.3 Algorithms for CARLA

To implement the above proposed methodologies, two algorithms for CARLA simulator [48] has been developed. The first algorithm, *rand_spawn_point*, generates random spawn points for foreground objects (i.e. non-player character (NPC)s) in the CARLA autonomous driving simulator based on a given prior distribution. Using the first algorithm, the second algorithm generates different scenarios for data collecting, each scenario lasts a set amount of time. For the first algorithm, the function takes in a given spatial distribution of a certain real-world domain. It then generates a random heading for each NPCs. To ensure that the generated spawn points are realistic, the algorithm examines if the location is on the road. If it is on the road, it looks for the nearest waypoint and uses

Algorithm 1 Generating Spawn Points for NPC in CARLA

Input:

world: the current world in the CARLA simulator

ego_vehicle: the ego vehicle

prior_spatial_distribution: the prior spatial distribution of the target domain

n_npc: the number of NPCs to spawn

Output:

rand_spawn_points: a list of spawn points

```
procedure GEN_SPAWN_POINTS
    repeat
        XY_ ← prior_spatial_distribution.resample(size=1)
        XYlocal ← ego_vehicle.get_transform().transform(carla.Location(x=XY_[0],y=XY_[1], z=2.0))
        head ← np.random.uniform(0, 360.0)
        location ← carla.Transform(carla.Location(x=XYlocal[0], y=XYlocal[1], z=XYlocal[2]),
carla.Rotation(yaw=head, pitch=0.0, roll=0.0))
        waypoint ← world.get_map().get_waypoint(location.location,project_to_road=True,
lane_type=carla.LaneType.Driving)
        if waypoint then
            location ← waypoint.transform
        end if
        if check_spawn_point_collision(ego_vehicle, location,6) then
            rand_spawn_points.append(location)
        end if
    until len(rand_spawn_points) == n_npc
    return rand_spawn_points
end procedure
```

that location instead. Otherwise, the randomly generated location is used. Additionally, the algorithm checks if the location is too close to the ego vehicle before adding it to the list of spawn points.

The second algorithm is designed to generate a variety of different scenarios in the CARLA autonomous driving simulator for data collection. The algorithm starts by initializing the scene creation loop, and for each iteration a new scene will be created. Specifically, in a new scene, foreground objects are spawned by the algorithm 1 according to the target domain distribution. During each data collection step, the algorithm checks if the threshold has been met. This threshold includes the number of data frame collected, miles driven, ego-vehicle's state. Once the threshold has been met, the current loop is terminated and the scene creation loop restarts from the beginning.

Algorithm 2 Rich Scenes Generation for Data Collection

Input:

world: the current world in the CARLA simulator

ego vehicle: the ego vehicle

actors: The spawn actors(NPCs)

threshold: threshold for restarting a new scene, this can be the number of data collected, miles driven, is stuck, etcs.

```
procedure SCENE CREATION LOOP
    Restart World(world, ego vehicle)
    Spawn ego-vehicle and sensors(world, ego vehicle)
    Spawn foreground objects(world, foreground objects distribution)
    repeat
        Data Collection Process()
    until check threshold(data collected, miles driven, stuck) == False
end procedure
```

4.3 Dataset Designs

In this section, an overview of the design of the proposed Domains-3D dataset is first presented. This is followed by a detailed description of domain shifts types it encompasses. After that, explanations of the sensor setup and annotations were discussed. Next, real-world domain meta analysis is presented, offering a statistical overview of the foreground object distributions across each domain. Finally, this thesis presents and analyzes the synthetic data generated using the proposed real-world guided synthetic data generation method and compare it to the real-world domains.

4.3.1 Dataset Overview

The proposed Domains-3D consists of a real-world and a synthetic data component, both of which are annotated and organized based on their respective domain labels. The real-world component of the dataset is sourced from the nuScenes dataset[17]. The nuScenes dataset [17] includes a diverse range of domains from urban, suburban/rural, and university campus areas. However, the nuScenes dataset does not offer official annotations that distinguish between these different domains. To address this issue, all 1000 scenes in the nuScenes dataset was carefully annotated to create six different domains. Specifically, three types of domain shifts was considered during this annotation process: city to city, urban to suburban/rural, and public road to university campus. To acquire and simulate the real-world ground truth object distribution, this thesis leverages the 3D object annotations of each real-world domain to obtain a statistical distribution of foreground objects.

To provide a more comprehensive representation of different domains, the synthetic component of the Domains-3D was generated through two complementary approaches. The first approach leverages the real-world prior and the proposed novel real-world guided synthetic data generation method to generate counterparts for six real-world domains. Another six synthetic domains, on the other hand, was constructed without this prior information. These two complementing methodologies allow the simultaneous study of

domain adaptation in both sim-to-sim and sim-to-real scenarios, providing a comprehensive evaluation of domain adaptation algorithms.

Domain Shift Types: The Domains-3D considers three distinct types of discrete domain shifts, namely cross-scene, sim-to-real, and cross-sensor. In terms of cross-scene domain shifts, the dataset includes urban to suburban/rural, city to city, and public road to university campus. For sim-to-real domain shifts, each of these six domains contains two synthetic and one real-world components. In the case of cross-sensor domain shifts, four different sensors were set up on the ego-vehicle for each recorded scene. Additionally, the dataset also encompasses continuous domain shifts such as varying weather and lighting conditions, providing a more comprehensive representation of the real-world environment.

4.3.2 Map Design

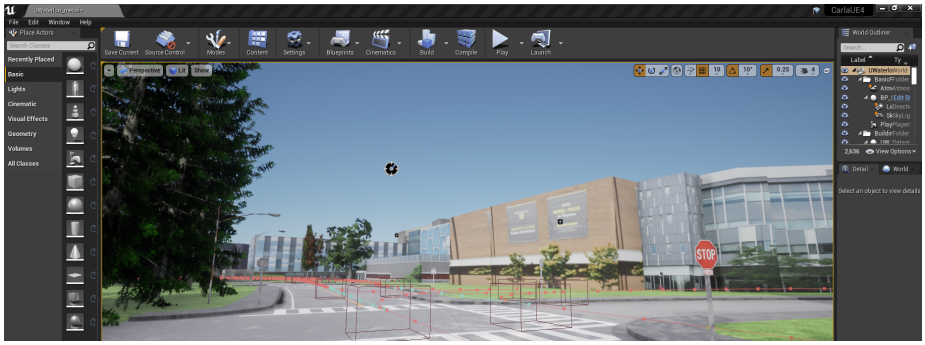
The CARLA simulator offers a variety of maps, including multiple cities, urban and sub-urban/rural landscapes. However, as the simulator lacks a university campus map, a dedicated CARLA map was built by the MVSL Lab to closely resemble the University of Waterloo campus. An illustration of this campus map is shown in Fig. 4.2. This map was meticulously constructed in Unreal Engine to capture the distinctive features and characteristics of the university campus setting.

4.3.3 Sensor Suite

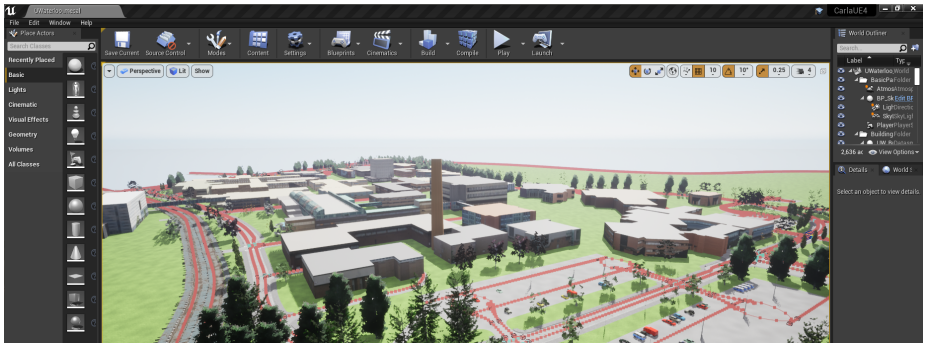
The CARLA simulator only simulates traditional mechanical LiDARs, not the newer solid-state LiDARs. However, the latter has a particular scanning pattern that differs from that of traditional LiDARs, which is shown in Fig.4.3. For instance, LiVOX LiDAR uses a unique non-repeating scanning pattern [140], which is not yet supported in the CARLA simulator. To address this, a new sensor model has been developed specifically for CARLA version 0.9.14. This new model accurately mimics the non-repeating scanning pattern of



(a) The Engineering 7 building



(b) The QNC building.

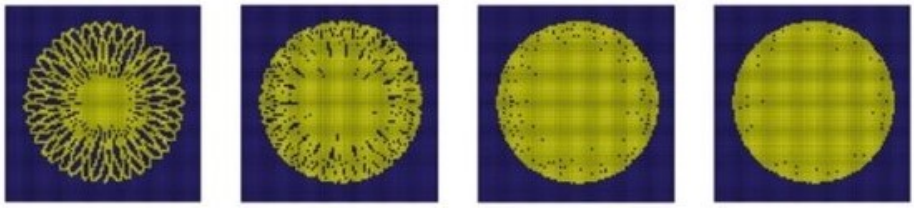


(c) An overview of the whole campus map.

Figure 4.2: The University campus map based on University of Waterloo in Unreal Engine Editor.



(a) Scanning patterns of mechanical LiDAR accumulated for 4 frames.



(b) Scanning patterns of solid-state LiVOX LiDAR accumulated for 4 frames.

Figure 4.3: A comparison between the accumulated scanning patterns of traditional mechanical LiDAR and recent solid-state LiVOX LiDAR [140].

LiVOX LiDAR. This is an important addition since it allows for the first time to compare domain adaption methods between solid-state and classic mechanical LiDARs.

Using this solid-state LiDAR model, four different LiDAR configurations are simulated in each simulated scene: a 32-beam LiDAR at 20 Hz (the configuration used in nuScenes), a 64-beam LiDAR at 10 Hz (the configuration used in KITTI), and two solid state LiDARs, the LIVOX Horizon and a generic 128-beam model. These configurations enable cross-sensor domain shifts. Importantly, Domains-3D is the first large-scale multi-task autonomous driving dataset with annotated solid state LiDAR data, filling a significant gap in the field as solid state LiDARs are expected to replace mechanical LiDARs in the near future.

Table 4.2: The per-frame foreground objects distribution on different domains. This thesis denotes pedestrian as 'Pedes', motorcycle as 'MotorC' and Traffic Cone as 'Cone'.

Domains	Pedes	Bicycle	MotorC	Car	Bus	Truck	Cone
Urban	9.23	0.50	0.41	14.12	0.47	2.15	4.41
suburban/rural	2.84	0.18	0.26	17.06	0.35	3.91	1.67
Boston	5.83	0.24	0.19	16.72	0.36	3.30	3.29
Singapore	4.97	0.32	0.43	6.46	0.41	0.71	1.34
Public	9.23	0.50	0.41	14.12	0.47	2.15	4.41
Campus	5.48	0.22	0.53	7.54	0.78	0.43	0.54

4.3.4 Annotations

Annotations are provided for a variety of critical perception tasks in autonomous driving, including 2D and 3D bounding box detection, instance and semantic segmentation. Due to the high expense of manual labelling, real-world datasets generally limit annotations to keyframes. Contrary to these real-world datasets, Domains-3D provides full annotations for every frame in the sequences, providing a more complete description of the data.

4.4 Experiments and Meta Analysis

One important thing to note is that datasets developed for autonomous driving research are typically collected using non-probability sampling methods. This means that the scenes included in the dataset were not randomly selected, but were chosen to include more challenging traffic, under-represented classes, and diverse weather/lighting conditions. As a result, the distribution of foreground and background objects in these datasets might not accurately reflect the true underlying data distribution in the environments where the

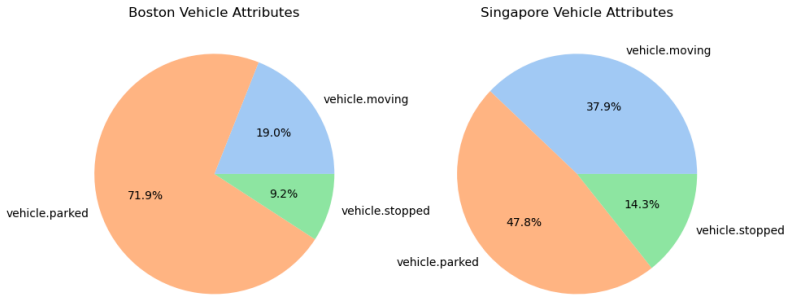
data was collected. Despite this limitation, studying the effects of background stuff in the proposed three domain datasets can still provide valuable insights into the effects of domain shifts on autonomous driving systems.

Boston to Singapore:

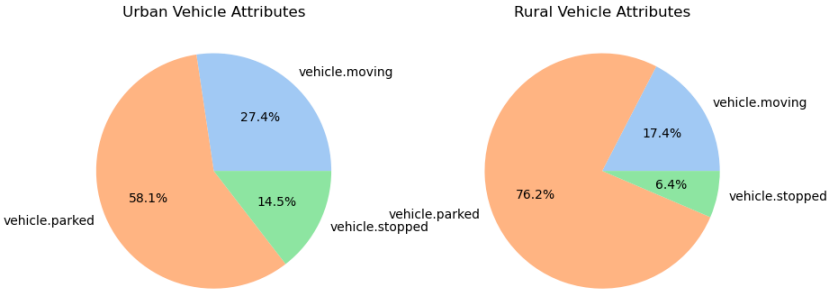
Domain Shift Analysis: On average, Boston has more foreground objects per frame than Singapore, indicating that the Boston domain is more densely populated with objects. Upon further analysis, this thesis found that Boston has 2.7 times more cars and 4 times more trucks per frame than Singapore. Additionally, it is notable that a large proportion of cars in Boston are parked (71.9% compared to 47.8% in Singapore, as shown in Fig. 4.4a), which might be due to the prevalence of parked cars on the sides of streets in urban scenes in Boston. In contrast, a larger portion of Singapore's cars are moving (37.9% compared to 19% in Boston), with fewer stopped cars in Singapore (14.3% compared to 9.2% in Boston). This might suggest that the collected scenes in Singapore might have better traffic flow than those in Boston. While both cities have a similar number of pedestrians per frame, classes such as bicycles, buses, and motorcycles are all very underrepresented in both cities.

One potential reason for the higher number of trucks recorded in Boston might be due to the fact that the data was collected around the seaports area of Boston, where large semi-trailer trucks are commonly found due to the port operations in the region. However, it is also possible that the data selected from Boston was specifically focused on scenes where truck traffic is more prevalent, contributing to a higher number of trucks in the Boston dataset. It is important to consider the impacts of sampling methods when interpreting the differences in object distributions between the source and target domains, as they might have influenced the observed results due to biased sampling. Non-probability sampling, such as the method used to collect the nuScenes dataset, can lead to data that is not representative of the true underlying population.

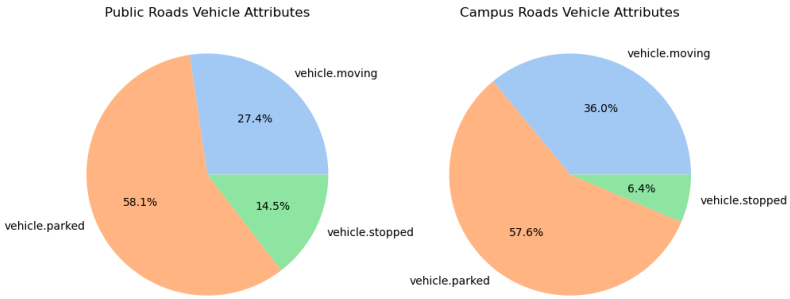
Synthetic Data Generation: For the proposed dataset, two synthetic Boston domains and two Singapore domains are generated with and without real-world priors. The



(a) City to City Splits

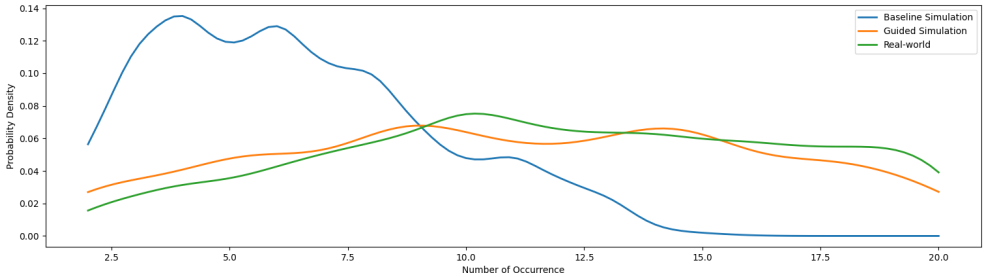


(b) Urban to suburban/rural Splits

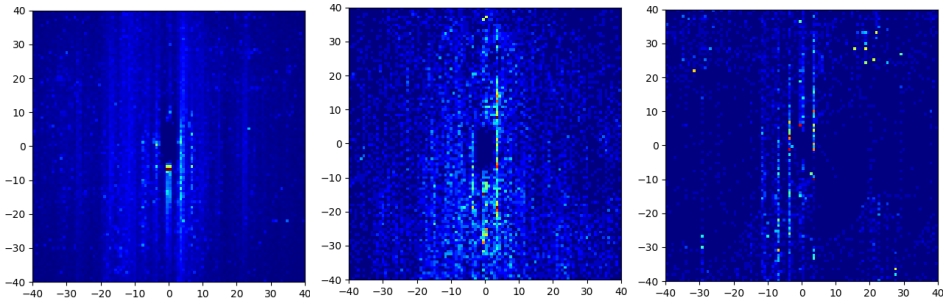


(c) Public to Campus Splits

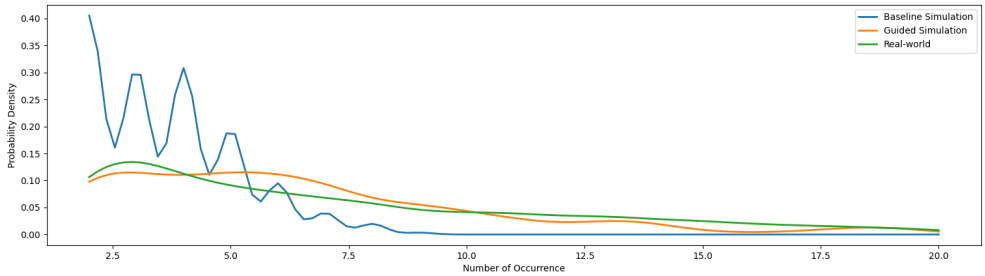
Figure 4.4: The intra-class attributes of Vehicles, where orange represents parked vehicle, blue represents moving vehicle and green represents stopped vehicle.



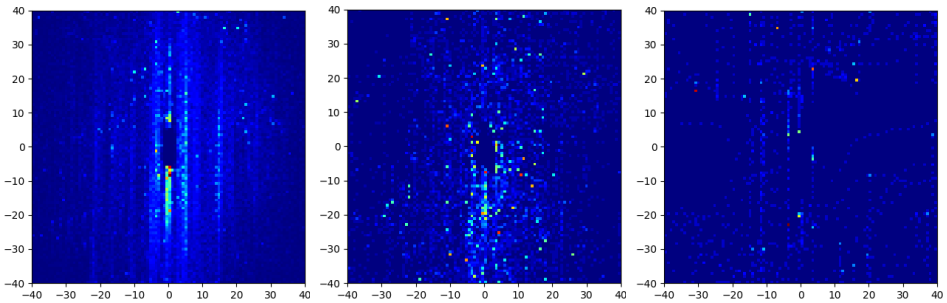
(a) Vehicle per frame in Simulated and Real Boston (Bos) Domain



(b) Real Boston. (c) Guided Sim Bos. (d) Baseline Sim Bos.

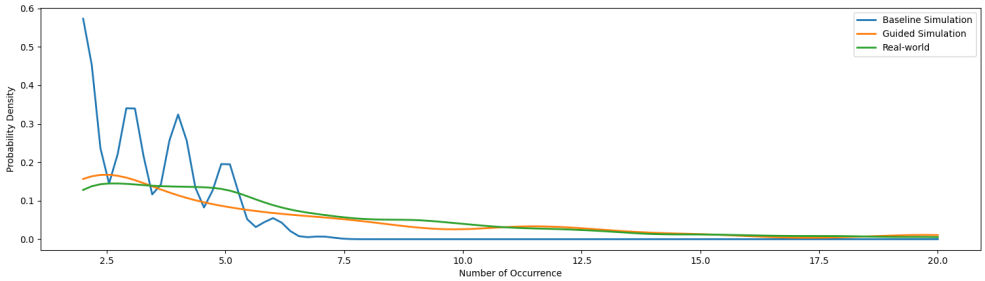


(e) Vehicle per frame in Simulated and Real Singapore (Sgp) Domain

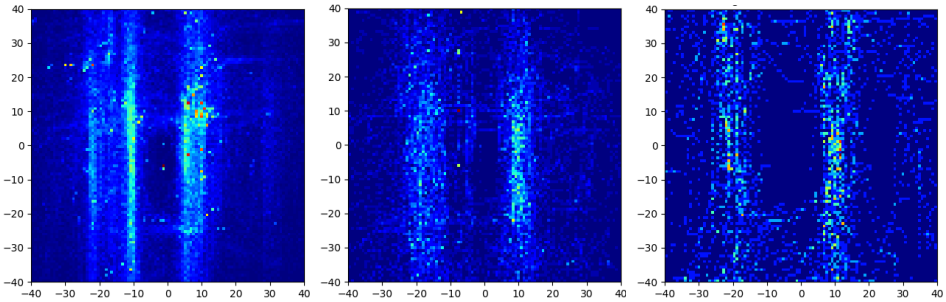


(f) Real Sgp. (g) Guided Sim Sgp. (h) Baseline Sim Sgp.

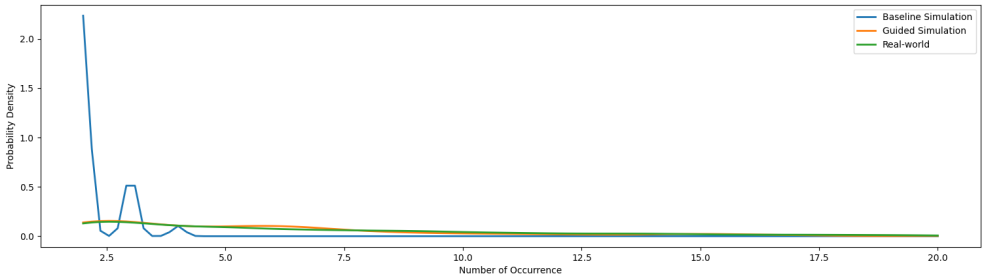
Figure 4.5: The vehicle's temporal and spatial distributions of cars in real and simulated Boston and Singapore domains



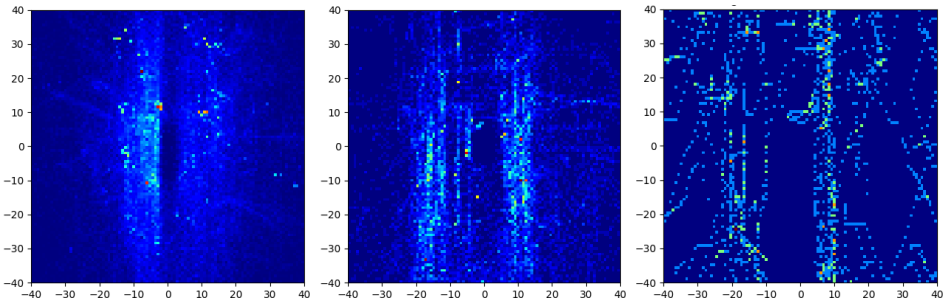
(a) Pedestrian per frame in Simulated and Real Boston (Bos) Domain



(b) Real Boston. (c) Guided Sim Bos. (d) Baseline Sim Bos.

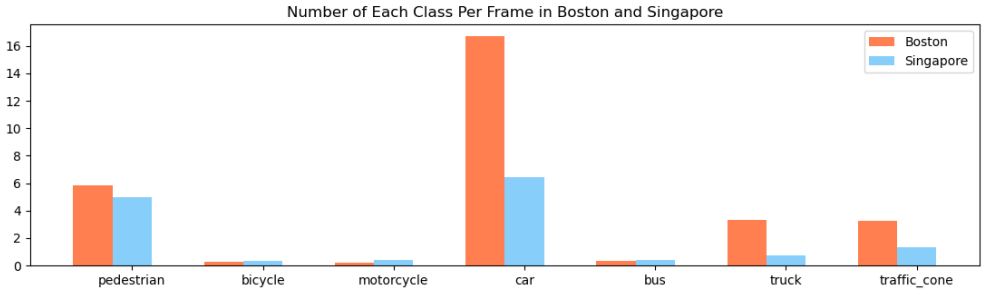


(e) Pedestrian per frame in Simulated and Real Singapore (Sgp) Domain

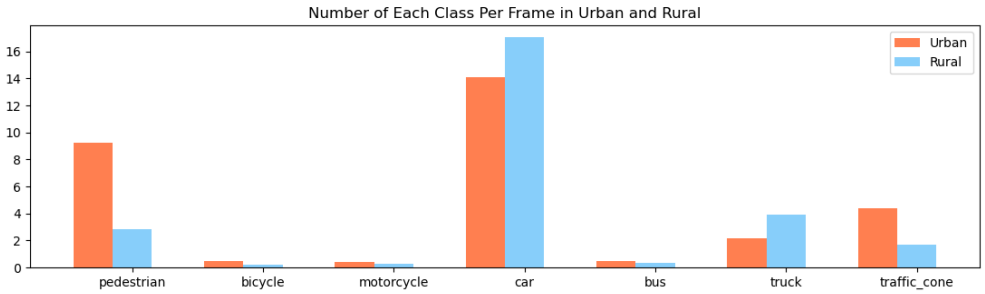


(f) Real Sgp. (g) Guided Sim Sgp. (h) Baseline Sim Sgp.

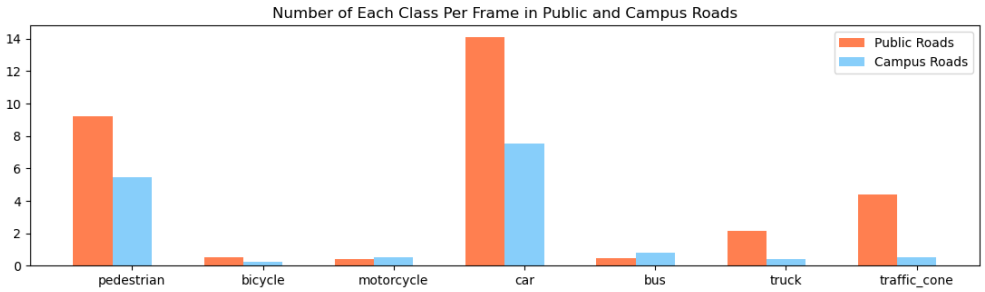
Figure 4.6: The pedestrian's temporal and spatial distributions of pedestrian in real and simulated Boston and Singapore domains



(a) City to City Splits



(b) Urban to suburban/rural Splits



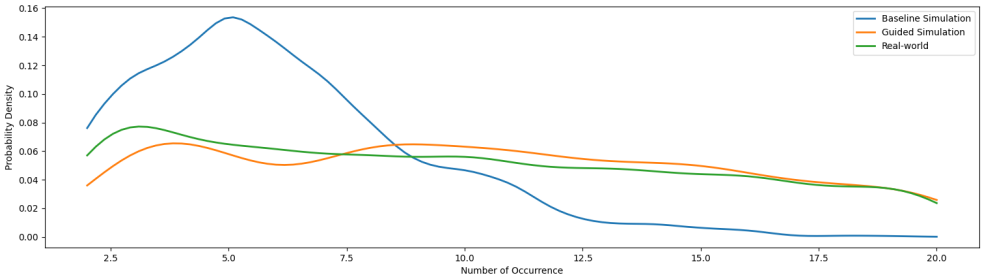
(c) Public to Campus Splits

Figure 4.7: The inter-class distribution, where orange and blue represents different domains.

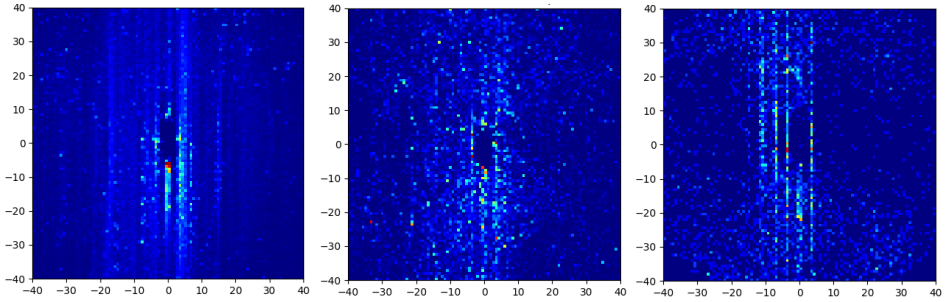
synthetic Boston domains and Singapore domains uses two different CARLA Town maps. For the baseline method, this thesis generates a random number of actors across the entire CARLA maps without any real-world priors. This baseline dataset is then compared with the proposed real-world guided synthetic data generated dataset. To validate the proposed method, this thesis calculates and compares the temporal and spatial distributions between the real-world domains and these two synthetically generated domains. The temporal distribution refers to the number of objects across each data frame in the collected data sequences, while the spatial distribution is represented in the form of heatmaps.

As shown in Fig.4.5, the vehicle simulation results indicate that the real-world guided simulation closely matches the real-world temporal distribution, which exhibits a more uniform (long-tailed) distribution. On the other hand, the baseline method displays a normal distribution, with most frames having means of six vehicles per frame in Boston and two in Singapore. Similarly, the proposed method's heatmap largely resembles the real-world geographical distribution, but with a more uniform distribution throughout the entire heatmap (40 by 40 meters). This mostly caused by vehicles traveling after spawning. Specifically, the spawned vehicle must operate in accordance with the traffic patterns (e.g., on the roads) present in the simulation environment. As a consequence, the resulting spatial distribution is heavily influenced by the map designs of the CARLA simulator. These results are further supported by KL-divergence, which shows that the divergence between the actual world and simulation is 1.65 and 3.78 for the proposed and baseline methods, respectively.

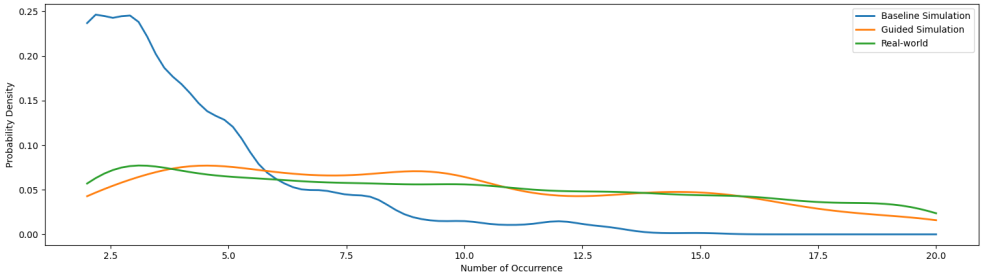
The pedestrian simulation, depicted in Fig.4.6, shows the real-world guided simulation accurately reflects the real-world temporal distribution. However, the pedestrian's spatial distribution in the real-world guided simulation deviates from the actual real-world distribution. This discrepancy stems from the pedestrian navigation rules imposed by the CARLA simulator. These rules dictate the behavior and movement patterns of simulated pedestrians within the virtual environment, which could be different than the real-world ones. For instance, simulated pedestrians are only allowed to navigate within predefined sidewalk lanes and will only cross the road through predefined cross-walks.



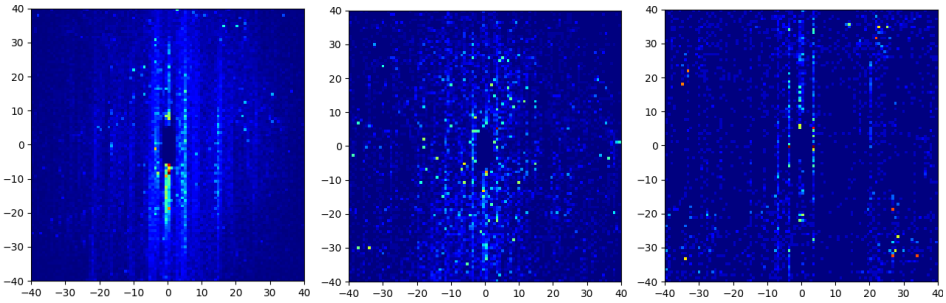
(a) Vehicle per frame in Simulated and Real Urban Domain



(b) Real Urban. (c) Guided Simulation. (d) Baseline Sim.



(e) Vehicle per frame in Simulated and Real suburban/rural Domain



(f) Real Rural. (g) Guided Simulation. (h) Baseline Sim.

Figure 4.8: The vehicle's temporal and spatial distributions of real and simulated urban and suburban/rural domains

Urban to suburban/rural:

The nuScenes dataset does not provide any pre-defined splits between urban and suburban/rural data. In order to create this split, this thesis carefully reviewed and annotated all the scenes based on geo-location and images. The final urban split includes 433 scenes, with 218 from Boston and 215 from Singapore, while the suburban/rural split consists of 321 validation scenes in total, with 249 from Boston and 72 from Singapore. This thesis further splits the urban scenes into 80% training set and 20% validation set.

The suburban/rural areas in Boston mostly consist of large parking lots, seaports, warehouses, and suburban districts. In Singapore, the data was collected from three districts: one-north, Queenstown, and Holland Village. The one-north scenes are predominantly urban streets, while the data from Holland Village is mostly residential suburban areas. It's important to note that the data collected in Queenstown is mostly from the campus of the National University of Singapore and the Singapore Institute of Technology (SIT). These campus scenes are sufficiently different from the public roads of both urban and suburban/rural scenes, warranting their own separate splits.

Domain Shift Analysis: It is interesting to observe that urban domain have slightly fewer cars per scene than suburban/rural domain, which is against the intuition. One potential reason for this might came from the presence of large parking lots in suburban/rural Boston, which is also reflected in a larger proportion of cars being parked (76.2% in suburban/rural compared to only 58.1% in urban environments). Additionally, many of the scenes recorded in One-North Singapore are off the main roads, and inside business districts where cars are less likely to be present. Among on-road cars, It is also notable that a higher percentage of cars are moving in suburban/rural areas compared to urban areas, at 73.1% and 65.4%, respectively. This might suggest that there is better traffic flow in suburban/rural areas compared to urban areas.

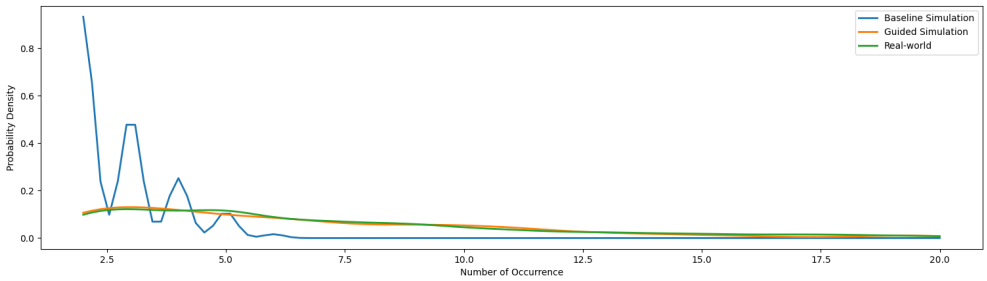
On the one hands, urban scenes have 3 times more pedestrians and 2 times more traffic cones per data frame than suburban/rural scenes, indicating that these objects might be more prevalent in urban environments. On the other hand, suburban/rural scenes have

significantly more trucks than urban scenes, which might because the suburban/rural scenes collected in Boston are located around seaport areas where truck traffic is more common. These findings suggest that there are notable differences in the distribution of objects between urban and suburban/rural environments, which might impact the performance of autonomous driving systems when adapting to these different domains.

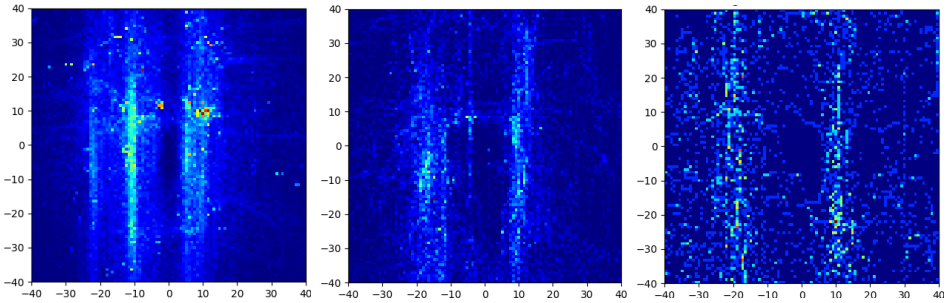
Synthetic Data Generation: Two synthetic urban and two synthetic suburban/rural domains are generated using the same experiment settings introduced above in Boston and Singapore experiment section. For urban domains, this thesis uses the Town10HD and Town03 maps from CARLA, which resembles typical city environments. For suburban/rural domains, Town02 and Town07 maps are used, which contains both suburban and rural environments. The results are illustrated in Fig.4.8, which further confirms the proposed approach by producing synthetic data that closely reflects real-world distributions in both temporal and spatial aspects. The proposed approach, in particular, has a more uniform temporal distribution, whereas the baseline method has a skewed normal distribution. The causes of spatial distribution discrepancies between the proposed and the real-world is discussed have been discussed in previous section. As per KL-divergence analysis, the proposed method has a lower divergence from the actual world than the baseline method, with divergence values of 1.21 and 3.12 for urban and suburban/rural domains, respectively. These findings suggest that the proposed approach is effective in generating realistic synthetic data for different urban and suburban/rural scenarios.

Public to Campus:

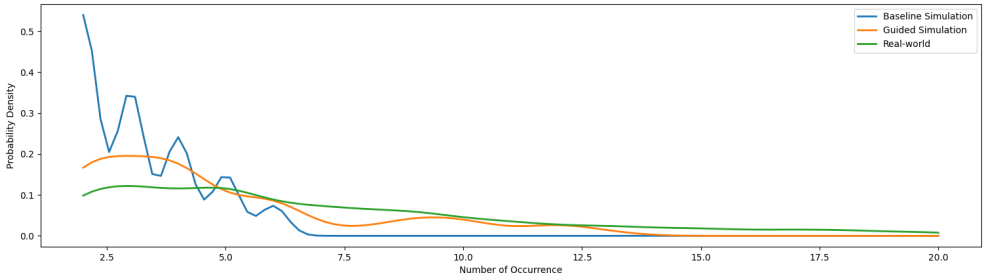
From a domain adaptation perspective, campus roads might be considered distinct from other roads due to their unique road topology and lacks of traffic elements (traffic lights, traffic signs, etcs.) and traffic characteristics (foreground objects). Campus roads often have narrower lanes and roads, sometimes even consisting of a single lane in one direction. They might also include unstructured roads, larger car parks and more green space. Additionally, campus roads typically have lower traffic density and a higher proportion



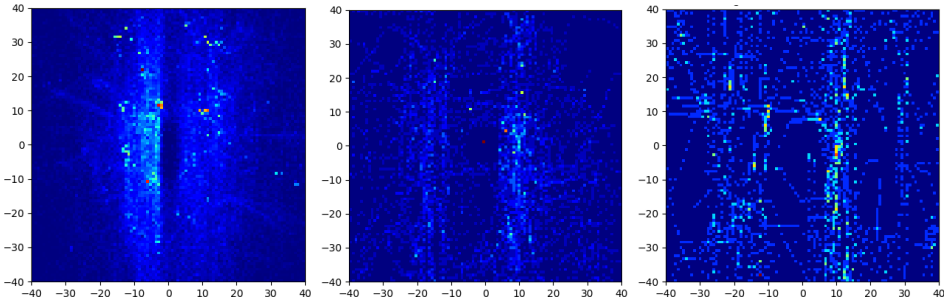
(a) Pedestrian per frame in Simulated and Real Urban Domain



(b) Real Urban. (c) Guided Simulation. (d) Baseline Sim.

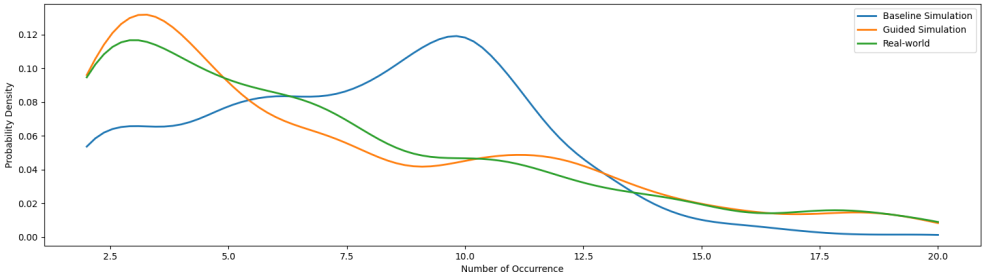


(e) Pedestrian per frame in Simulated and Real suburban/rural Domain

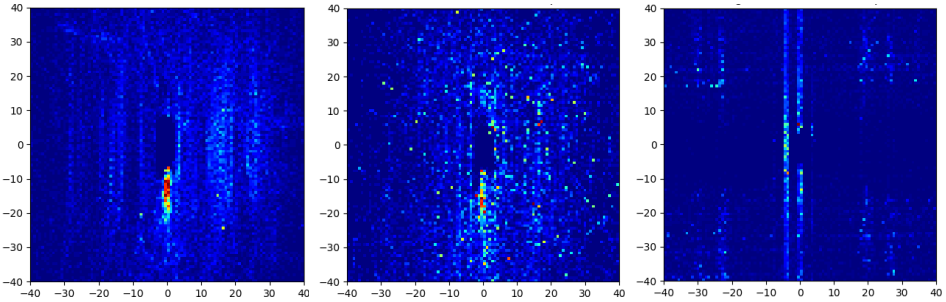


(f) Real Rural. (g) Guided Simulation. (h) Baseline Sim.

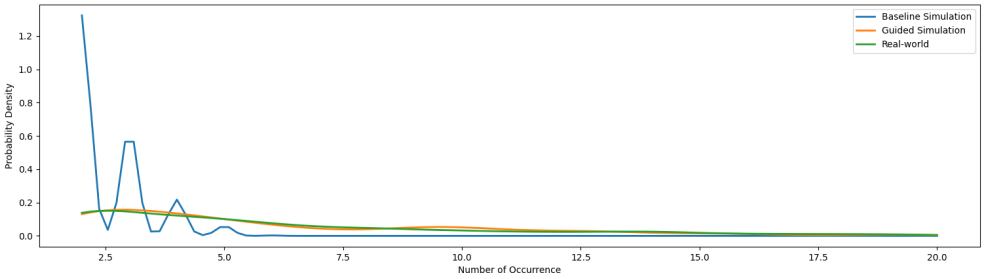
Figure 4.9: The pedestrian's temporal and spatial distributions of real and simulated urban and suburban/rural domains



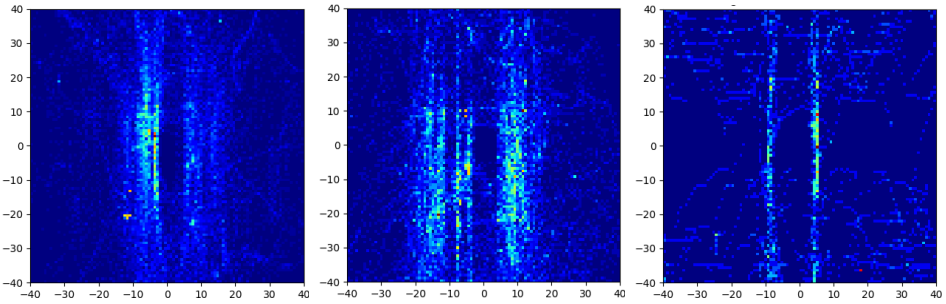
(a) Cars per frame in Simulated and Real Campus Domain



(b) Real Campus. (c) Guided Simulation. (d) Baseline Sim.



(e) Pedestrian per frame in Simulated and Real Campus Domain



(f) Real Campus. (g) Guided Simulation. (h) Baseline Sim.

Figure 4.10: The vehicle's and pedestrian's temporal and spatial distributions of real and simulated campus domains

of pedestrian and bicycle traffic compared to other types of roads. These differences can make it challenging for models trained on other types of roads to generalize well to campus roads, and vice versa. As such, it would be beneficial to create a separate splits for campus roads for domain adaptation studies.

The data collected from Queenstown for the nuScenes dataset was carefully reviewed and selected to create the public to campus splits. The scenes chosen were primarily from or near the National University of Singapore and the Singapore Institute of Technology (SIT) campus. This manual selection process ensured that the dataset accurately reflects the unique characteristics of the university environment.

Foreground Domain Shifts: Fig. 4.7c has revealed that public roads have significantly more objects across all classes, except for buses, compared to campus roads. This reflects a denser traffic on public roads and potentially more emphasis on public transportation in campus environments. Specifically, public roads have 2 times more cars, 5 times more trucks, and 8.2 times more traffic cones than campus roads. This research also suggests that the proportion of cars parked on public roads and on campus roads is relatively similar, at 58.1% and 57.6%, respectively. However, a significantly higher proportion of on-road cars are moving on campus compared to public roads, at 84.9% and 65.4%, respectively. This might suggest that traffic flow on campus roads is generally smoother.

Synthetic Data Generation: Two synthetic campus domains are generated with the same experiment settings introduced in the previous sections. Notably, these two domains are generated on a purposely built university campus map. The results, given in Fig.4.10, provide further evidence that the proposed approach produces synthetic data that more closely reflects real-world domains. Specifically, KL-divergence analysis reveals that the suggested technique is less divergent from reality than the baseline method, with divergence values of 2.98 and 4.63 for urban and suburban/rural domains, respectively. These findings support the approach's ability in creating realistic synthetic data for a variety of circumstances, including those particular to campus contexts.

4.5 Summary

To summarize, a novel domain adaptation dataset (Domains-3D) for autonomous driving and a novel real-world guided synthetic data generation approach are developed. The proposed dataset consists of both real-world and synthetically generated scenarios of various scenes and domains. The dataset features diverse traffic, weather, and lighting conditions. Most notably, it also covers a wide range of domains, including cross-sensor, cross-scene, and sim-to-real domains. This provision of multiple domain transitions is a unique aspect of Domains-3D. As it gives researchers unique opportunities to investigate and address domain adaption challenges in autonomous driving. Moreover, this is the first large-scale dataset to incorporate solid-state LiDAR which are not available in existing datasets. These charecteristics set Domains-3D apart from other datasets and make it an attractive dataset for developing domain-invariant perception models. In addition to the dataset, the proposed synthetic data generation procedure enables synthetic generated domains to closely matches their real-world counterparts. The experiment results proves that the proposed approach is effective in generating realistic synthetic data for various domains.

Chapter 5

Background-Aware Point Cloud Domain Adaptation

This chapter introduces a novel plug-and-play method for improving the generalization ability of a point-cloud-based 3D object detection model across scene domains. This chapter begin with a background overview, highlighting the key contributions, and discussing recent related works. Followed by section 5.2, which shows mathematically that removing background points from a point cloud reduces the total amount of domain shifts. The experimental results of the proposed method on the new Domains-3D dataset are presented in Section 5.3. Experiments show that this algorithm improves the performance of cross-scene domain adaptation models on foreground objects. Furthermore, the experiments uncovers that the performance of intra-domain 3D object detection improves, most likely due to the removal of superfluous information from the point cloud.

5.1 Background

The domain shifts problem refer to a scenario where a deep learning model is trained on a source data distribution and deployed on a different target data distribution. For 3D object

detection, domain shifts could leads to significant performance degradation [145] [144]. Domain adaptation are techniques to mitigate the domain shifts problem. Its purpose is to train a model on a source domain (a set of data from one environment) and then generalize well on a target domain (i.e. a different set of data from a different environment).

In the context of autonomous driving, most current studies on domain adaptation are about 2D image-based perception tasks, such as 2D object detection [80] [88] [111] [114] [273] and 2D semantic segmentation [29] [86] [92] [153]. These works mainly focus on transferring models trained on one labelled source domain to an unlabeled domain. For instance, adapting a model trained on synthetic images to real-world images. The domain shifts are often caused by difference in image styles, which could have drastically different texture distributions.Unlike dense structured image data, point cloud is irregular and lacks dense texture information. This makes mitigation of domain shifts more challenging on point cloud data. Recent works [273] [254] [284] on domain adaptation on point cloud models focuses on mitigation of the scale variation of bounding boxes in different domains. [254] proposed an online data augmentation method that utilize target domain object statistics to normalizes the bounding box and their respective point cloud scale during training. [284] [187] explores domain-invariant representations for 3D object detection. [273] proposed an unsupervised training pipeline for fine tunning a pre-trained model to an unlabelled target domain.

However, existing approaches fails to account for the significant variations in background objects. These variations in background objects, such as road verges, dividers, and buildings, that can occur across different locations can have a significant impact on the performance of 3D object detection models. For example, the street backgrounds of Boston and Singapore are vastly different (illustrated in Fig. 4.1a), and this can affect the ability of a model trained on one location to generalize to the other. In addition to differences between geographic locations, the background objects can also vary depending on the type of road. For example, the background objects on a highway may be different from those on an urban street or a rural road. These domain shifts can have a significant impact on the performance of deep learning models. By studying the effects of background points, bet-

ter understanding of how these variations affect model performance can be gained. And strategies to mitigate their impact can be developed. Furthermore, various existing 3D perception software stacks, such as Autoware and Baidu Apollo, already filter point clouds to remove ground points and points outside of the region of interest. Therefore, this study could improve the performance of these systems in real-world applications, resulting in a more robust and reliable system in various conditions.

In this thesis, the impact of background points on domain adaptation for 3D point clouds is examined. To the best of author's knowledge, this is the first work of its kind. To provide a thorough analysis of the effects of background points on domain adaptation, experiments on three distinct domain shifts are conducted. On the basis of these findings, this thesis propose a novel plug-and-play domain adaptation algorithm that reduces the domain shift between the source and target domains. Experiments indicate that this algorithm improves the performance of models for cross-scene domain adaptation on foreground objects. In addition, this thesis find that the performance of intra-domain 3D object detection is enhanced, most likely as a result of the removal of superfluous information from the point cloud.

5.2 Methodology

5.2.1 Problem Statement

Let $S = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ be the source domain dataset, where $x_i \in \mathbb{R}^d$ is a point cloud sample with d being the dimensionality. And $y_i \in \mathcal{Y}$ is its corresponding label with $\mathcal{Y}$ being the label space. Similarly, let $T = \{(x_i^t, y_i^t)\}_{i=1}^{n_t}$ be the target domain dataset. For the task of interest, the labeling function that maps the data inputs to labels is denoted as $f : \mathcal{X} \to \mathcal{Y}$. And thesis use f_s and f_t to indicate the labeling functions for the source and target domains, respectively. In the context of domain adaptation, the goal is to address the insufficient information on the target domain, by utilizing knowledge extracted from a

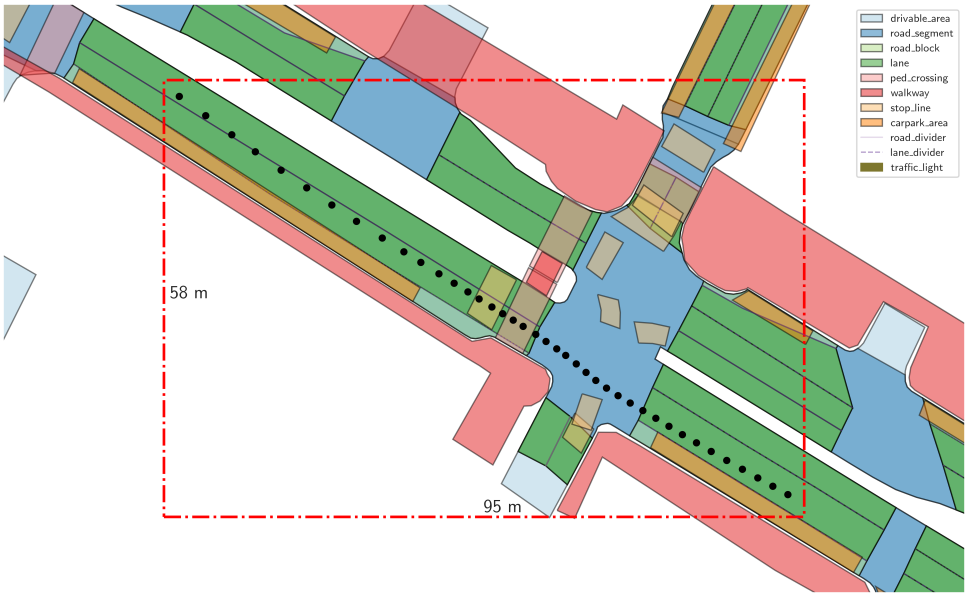


Figure 5.1: An instance of *scenes-0247* visualized on the HD-map, where ego-poses are illustrated in black dots and map layers are represented in different colors.

source domain. Specifically, a function $h : \mathbb{R}^d \to \mathcal{Y}$ that can generalize well on both the source domain S and target domain T is required, when trained on the source domain S.

Theorem 1 (Acuna et al.[3]). *Suppose* $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, 1] \subset \operatorname{dom} \phi^*$ *and denote* $\lambda^* := R_S^\ell(h^*) + R_T^\ell(h^*)$. *let* h^* *be the ideal joint hypothesis.*

$$R_T^\ell(h) \leq R_S^\ell(h) + D_{h,\mathcal{H}}^\phi(P_s \| P_t) + \lambda^*. \tag{5.1}$$

where $R_{\mathcal{D}}^\ell(h, f)$ is the risk of a hypothesis h w.r.t. the labeling function f, using a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}+$under distribution $\mathcal{D}$. And $R_{\mathcal{D}}^\ell(h, f)$ can be formally defined as: $R_{\mathcal{D}}^\ell(h, f) = \mathbb{E}x \sim \mathcal{D}[\ell(h(x), f(x))]$. To further simplify: $R_S^\ell(h) = R_{P_s}^\ell(h, f_s)$ and $R_T^\ell(h) = R_{P_t}^\ell(h, f_t)$. The first two terms of theorem 1 states that for a model to successfully generalize from source to target domains, it must perform well on the source domain, and the target and source domain should have minimal discrepancy. Finally, the model should have an optimal adaptable hypothesis that performs well in both domains, which is measured by the third term λ^*.

The challenge of achieving the goal lies in adapting the domain shifts between the source and target domains. This domain shifts can be defined as the difference in the data distribution between the two domains:

$$d_{S,T} = |P_S(X) - P_T(X)| \tag{5.2}$$

where $P_S(X)$ and $P_T(X)$ are the probability distribution of the source and target domains. In the context of a point cloud dataset, $P_S(X)$ and $P_T(X)$ is the distribution of points in the two domains. Given that, a point cloud sample can be further categorized into foreground points x_i^f and background points x_i^b, where the foreground (*resp.* background) points are defined as the points that belong to the pre-defined foreground (*resp.* background) objects.

Now, let $P_S(X^f)$ and $P_T(X^f)$ be the data distribution of the foreground points in the source and target domains, respectively. Similarly, let $P_S(X^b)$ and $P_T(X^b)$ be the data distribution of the background points in the source and target domains, respectively. Therefore, the total amount of domain shifts between the two domains is:

$$d_{S,T} = d_{S,T}^f + d_{S,T}^b \tag{5.3}$$

By removing the background points, the domain shift between the background points is effectively eliminated, that is $d_{S,T}^b$. As such, the no-background domain shifts $d'_{S,T}$ is less than the total domain shift between the two domains:

$$d'_{S,T} := d_{S,T}^f \leq d_{S,T} \tag{5.4}$$

Intuitively, in cross-scenes domain shifts, such as urban to rural, the majority of the foreground objects remain similar in terms of appearance and behavior. However, the background environments, such as buildings, road types, and vegetation, can vary significantly between the source and target domains. Based on the above observations, this thesis proposes to mitigate domain shift in point-cloud-based 3D object detection by removing background points. Specifically, if a model is trained and achieves optimal performance on the source domain, its generalization ability on the target domain can be further improved by reducing domain shifts between the two domains.

Table 5.1: The per-scene road structures and traffic elements distribution on different domains.The 'Road-B' denotes road blocks,'Ped-C' denotes pedestrian crossing,'Road-D' denotes Road Dividers, 'Lane-D' denotes lane dividers

Domains	Road-B	Lane	Ped-C	Walkway	Stop-L	Carpark	Road-D	Lane-D	Traffic-L	Inters
Urban	1.91	3.40	1.09	1.53	2.97	0.53	0.71	2.00	1.76	0.97
Rural	1.77	3.50	0.73	1.94	3.68	0.24	1.44	1.61	0.64	0.99
Boston	1.98	2.79	1.15	0.46	3.32	0.52	1.01	1.85	1.48	0.78
Singapore	1.16	4.38	0.50	3.29	3.33	0.26	1.10	1.52	0.72	1.32
Public	1.91	3.40	1.09	1.53	2.97	0.53	0.71	2.00	1.76	0.97
Campus	0.00	3.92	0.35	2.11	3.97	0.43	1.28	0.62	0.06	1.30

5.2.2 A Meta Analysis of Background Stuff

Dataset: This research uses Domains-3D dataset, developed in the previous chapter, for training and validation. To better understand the role of background stuff on domain shifts, the underlying data distribution needs to be investigated and analyzed . One key factor that could contribute to domain shifts is road layout and design. Different domains tend to have unique road layouts and designs, leading to different traffic patterns (of foreground objects). To investigate road layout and design across domains, this thesis use the HD-map and geo-location information provided by the dataset to gather information on the road and traffic elements in each scene, which is illustrated in Fig. 5.1. this thesis then average these results across each domain to obtain a statistical overview of the data distribution. More specifically, for each domain, this thesis study road elements such as road blocks, lanes, intersections, pedestrian crossings, walkways, carpark areas, road dividers, and lane dividers. this thesis also examine traffic elements such as stop lines and traffic lights.

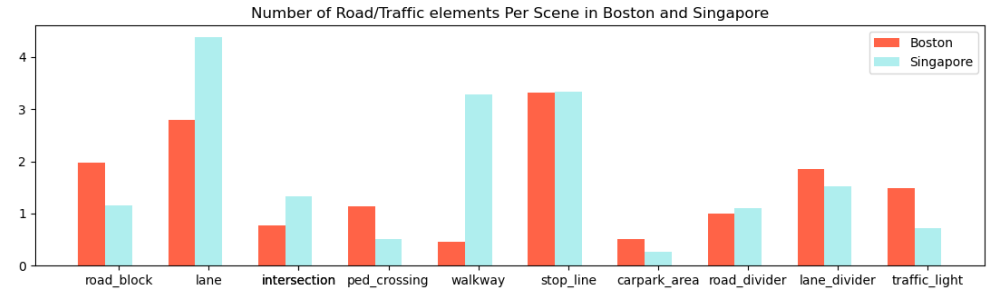
City-City Domains:

For city to city domain shifts, this thesis consider Boston as the source domain and Singapore as the target domain. This means that the proposed model will be trained on data collected in Boston and evaluated on data collected in Singapore. Specifically,

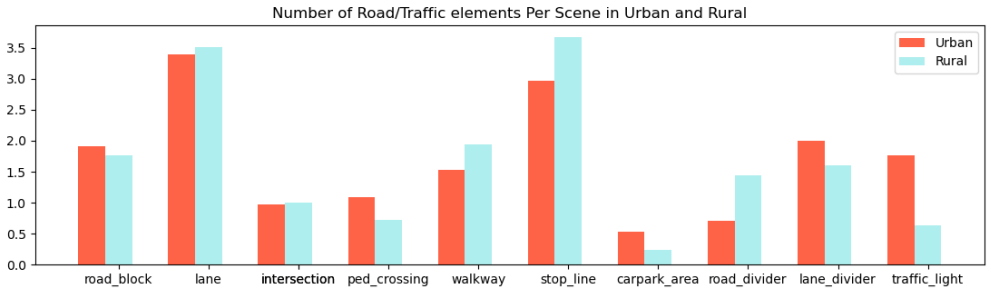
models are trained on a portion of the official training split that was collected in Boston, and validate its performance on both the Boston validation split and and the full dataset from Singapore (combined training and validation splits). The Boston domain contains 467 driving scenes with a total of 18,785 annotated frames, while the Singapore domain contains 383 driving scenes with 15,364 annotated frames.

The comparison and analysis of the background stuff distributions between the Boston and Singapore domains reveals some interesting insights. As illustrated in Fig. 5.2, Boston has twice as many road blocks per scene as Singapore, while Singapore has more lanes per scene than Boston. This suggests that in Singapore where the data was collected, roads tend to be multi-lane. Furthermore, Singapore has significantly more walkways per scene than Boston, at 3.29 and 0.46, respectively. This might suggest that the infrastructure in Singapore is more pedestrian-friendly. Interestingly, Boston has fewer intersections than Singapore (0.78 vs. 1.32), but more pedestrian crossings (1.15 and 0.5, respectively). This differences in the number of intersections and pedestrian crossings between Boston and Singapore could be traced to the non-probability sampling methods employed to acquire the data. As previously mentioned, the nuScenes dataset was collected mainly from two areas in Boston: the downtown and seaport. While the downtown area has a high number of pedestrian crossings and intersections, the seaport area has fewer intersections due to the lack of cross streets. This skewed distribution of intersections in Boston could be a result of the non-random selection of scenes in the nuScenes dataset. This underscores the importance of considering sampling methods when studying the effects of domain shifts on autonomous driving systems.

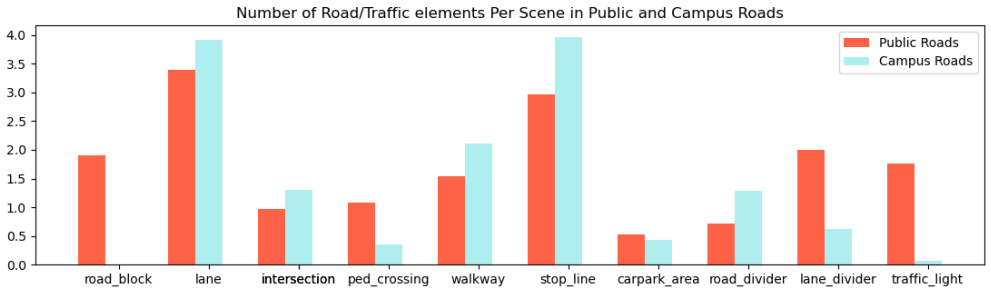
Urban-Rural Domains: Based on the statistical results from Table. 5.1, Urban sceneries feature slightly more road obstacles per scene, whilst rural situations have slightly more lanes per scene. This might indicate that rural environments have broader roads with more lanes. Urban and rural scenes have a similar number of intersections, but compared to rural scenes, urban scenes have 3 times more traffic lights and 1.5 times more pedestrian crossings. This difference suggests that the traffic patterns and infrastructure in urban and rural environments differs, with urban environments having more traffic control devices and



(a) City to City Splits



(b) Urban to Rural Splits



(c) Public to Campus Splits

Figure 5.2: The distribution of road and traffic elements in two different domains (indicated by red and blue bars).The 'ped_crossing' represents a designated area where pedestrians can legally cross the road. The 'road block' refers to a segment of the road with the same traffic direction. The 'road divider' and 'lane divider' serve to separate one road block and lanes from others.

pedestrian crossings. Rural scenes also have significantly more road dividers but slightly fewer lane dividers. Road dividers are typically used to separate opposing lines of traffic, whereas lane dividers are used to separate lanes of traffic travelling in the same direction. This might be attributed to the frequent presence of one-way roads in the narrow alleys and streets of urban environments, which suggests a more crowded and enclosed background space due to the presence of buildings.

Public-Campus Domains: As shown in Fig. 5.2c, campus environments has zero road blocks and very few traffic lights (0.06 per scene), but more lanes than public roads (at 3.92 and 3.40 respectively). This indicates unique road typologies, characterized by narrow lanes and a lack of traffic elements, are strong evidence to support the assumption that campus environments pose enough domain shifts to be considered their own domain.Furthermore, campus has 1.8 times more road divider but 3.3 times less lane dividers than public roads. This might be came from the unique traffic characteristics of campus roads, such as their one-way lanes and low traffic density, which might require fewer lane dividers for traffic control.

5.3 Experiments

3D object detection algorithms

this thesis studies the impacts of background points on popular 3D object detection networks. Specifically, the original CenterPoint [277], PointPillars [123] and Free-Anchor [286] networks are compared against the proposed PnP method in terms of mAP for selected object class in the Domains-3D dataset.

The CenterPoint [277] network proposed a bottom-up approach, first detecting the centers of objects using a keypoint detector and then using additional point features on the object to refine the estimates of attributes such as 3D size, orientation, and velocity. PointPillars [123] is a framework for detecting objects in 3D point cloud data using a CNN. It use PointNets [186] to encode the point features of 3D point cloud data organized

Table 5.2: 3D object detection results for cross-city domain adaptation. This table reports mean average precision (mAP) averaged over match thresholds of 0.5, 1, 2, and 4 meters. The models's performance is evaluated on objects of interest within the range of -52 meters to 52 meters. This table denotes pedestrian as 'Pedes', motorcycle as 'MotorC' and Traffic Cone as 'Cone'.

Boston → Singapore	Networks	Car	Bus	Truck	Pedes	Bicycle	MotorC	Cone
Original	CenterPoint	0.684	0.528	0.284	**0.554**	0.000	0.044	0.140
	PointPillar	0.475	0.240	0.190	0.036	0.000	0.000	0.000
	FreeAnchor	0.313	0.154	0.145	0.153	0.000	0.000	0.000
Ours	CenterPoint	**0.735**	**0.600**	**0.382**	0.522	0.000	**0.145**	**0.394**
	PointPillar	**0.526**	**0.251**	**0.216**	**0.264**	0.000	**0.083**	**0.070**
	FreeAnchor	**0.518**	**0.338**	**0.160**	**0.238**	0.000	0.000	**0.040**

into pillars-like structure. This allows the algorithm to effectively process the point cloud data and use a 2D CNN for object detection. By leveraging the strengths of both 3D and 2D methods, PointPillars is able to perform object detection in real-time. Free-Anchor [286]improve upon existing anchor-based methods by using free anchor matching instead of hand-crafted anchor assignment.This allows the algorithm to learn features that are optimal for both classification and localization of objects The use of MLE also allows FreeAnchor to be easily integrated with other CNN-based detectors, making it a flexible and effective approach for 3D object detection.

Datasets Pre-Processing:

To remove the background points, the provided per-point labels are leveraged in the nuScenes dataset to filter certain classes of background points while keeping others. Specifically, this thesis remove background objects such as ground surfaces and static man-made objects, while retaining other background classes such as noise, ego-vehicles, and other dynamic objects. This distinction allows us to accurately simulate the outputs of common filters used in real-time autonomous vehicle perception systems, such as region of interest (ROI) and ground points removal filters. More importantly, by only removing background

Table 5.3: 3D object detection results average on all classes for cross-city domain adaptation. This table reports mATE (Meters) mASE(1-IoU) mAOE(Rad) mAVE(m/s) mAAE(1-acc) NDS averaged over all classes. The models's performance is evaluated on objects of interest within the range of -52 meters to 52 meters. This table denotes pedestrian as 'Pedes', motorcycle as 'MotorC' and Traffic Cone as 'Cone'.

Boston → Singapore	Networks	mATE	mASE	mAOE	mAVE	mAAE	NDS
Original	CenterPoint	0.5223	0.4004	0.6551	1.6219	0.4929	0.3112
	PointPillar	0.7334	0.5359	1.0701	1.3773	0.6844	0.1517
	FreeAnchor	0.6166	0.4492	1.1579	2.0044	0.6658	0.1724
Ours	CenterPoint	0.3979	0.7198	1.5355	1.4978	0.3975	0.3284
	PointPillar	0.5667	0.3720	1.1542	1.5167	0.4795	0.2234
	FreeAnchor	0.4894	0.3952	1.1384	1.6599	0.5154	0.2304

points that can be filtered in real-time on actual autonomous vehicle perception stacks, this method can be easily deployed on these existing vehicles.

Metrics: In this study, the performance of 3D object detection models is evaluated using the mAP metric defined in [17]. This metric is computed by matching predicted objects with ground truth objects based on the 2D center distance on the ground plane with averaged AP over match thresholds of 0.5, 1, 2, and 4 meters. This metric allows us to accurately compare the performance of the proposed model with other 3D object detection algorithms on the domain-adaptation tasks.

5.3.1 Cross-City Domain Adaptation Results

The results of the proposed method on the city to city adaptation has shown a significant improvement over the original methods, which are reported in Table. 5.2. Specifically, the proposed method outperformed the original networks (i.e., CenterPoint, PointPillar, and FreeAnchor) in terms of mean average precision (mAP) for most object classes. For cars, for example, the proposed technique produced a mAP of 0.735, whereas the original

CenterPoint network produced a mAP of 0.65. Similarly, for pedestrians, the proposed method achieved a mAP of 0.659, compared to 0.554 for the original CenterPoint network. In addition, for bus, truck, and pedestrian classes, the proposed approach outperformed the original PointPillar and FreeAnchor networks, further confirming its efficacy. However, for classes such as bicycle and motorcycle, the PointPillar and FreeAnchor networks failed on both the source and target domains. This failure, however, should not be attributed to the ineffectiveness of the proposed method, but further confirms the conclusion from Theorem 1. Theorem 1 states that before a model can be successful in the target domain, it must first generalise well in the source domain.

In addition to the improvements observed in mAP values for various object classes, the proposed method also demonstrates improved performance in other evaluation metrics, as illustrated in Table 5.3. It is worth mentioning that the NDS score[17] is a weighted sum of the mATE, mASE, mAOE, mAVE, and mAAE measures. This score provides a comprehensive assessment of the algorithm's performance, taking into account not only the precision but also various error types related to translation, scale, orientation, velocity, and attributes. Therefore, when comparing the proposed methods with the original networks in terms of NDS, it is evident that the proposed approaches consistently achieves superior overall performance. To summarize, the proposed approach demonstrates the ability to improve the performance of 3D object detection networks in cross-city domain shifts.

5.3.2 Urban-to-Rural Domain Adaptation Results

As shown in Table 5.4, the proposed method has demonstrated significant improvements over the original networks. In addition to these improvements, we also observe enhanced performance in average evaluation metrics, as shown in Table 5.5. Most notable, the proposed methods outperforms the original methods in detecting pedestrian objects, with an average improvement of 15%. For example, the CenterPoint network improves from 0.561 to 0.668, PointPillar from 0.018 to 0.180, and FreeAnchor from 0.130 to 0.301. In addition to pedestrian objects, the proposed method improves mAP across many other object

Table 5.4: 3D object detection results on rural domain for urban to rual domain adaptation. This table report mean average precision (mAP) averaged over match thresholds of 0.5, 1, 2, and 4 meters. The models's performance is evaluated on objects of interest within the range of -52 meters to 52 meters. This table denotes pedestrian as 'Pedes', motorcycle as 'MotorC' and Traffic Cone as 'Cone'.

Urban → Rural	Networks	Car	Bus	Truck	Pedes	Bicycle	MotorC	Cone
Original	CenterPoint	0.706	0.378	0.275	0.561	0.036	0.144	0.320
	PointPillar	0.488	**0.034**	0.090	0.018	0.000	0.000	0.000
	FreeAnchor	0.461	0.068	0.124	0.130	0.000	0.001	0.025
Ours	CenterPoint	**0.736**	**0.409**	**0.317**	**0.668**	**0.076**	**0.200**	**0.535**
	PointPillar	**0.502**	0.022	**0.108**	**0.180**	0.000	0.000	**0.071**
	FreeAnchor	**0.528**	**0.086**	**0.160**	**0.301**	**0.001**	**0.003**	**0.104**

classes. For instance, the proposed method improved the mAP for cars from 0.706 to 0.736, buses from 0.378 to 0.409, and trucks from 0.275 to 0.317, among others. Surprisingly, the original PointPillar method outperformed the proposed method for buses, which could be attributed to the fact that bus samples are severely under-represented in the dataset.

In terms of performance averaged across all classes, it is evident that the proposed approach consistently achieves superior performance in terms of NDS. For instance, the adapted CenterPoint network attains an NDS of 0.4254, outperforming the original CenterPoint network's NDS of 0.3073. Likewise, the adapted PointPillar and FreeAnchor networks surpass their original counterparts, with NDS values of 0.2202 and 0.2468, respectively, compared to the original networks' NDS values of 0.1484 and 0.2049. In summary, these findings demonstrate the effectiveness of the proposed method in adapting to the rural domain and improving the accuracy of 3D object detection in this difficult environment.

Table 5.5: 3D object detection results average on all classes for urban to rual domain adaptation. This table reports mATE (Meters) mASE(1-IoU) mAOE(Rad) mAVE(m/s) mAAE(1-acc) NDS averaged over all classes. The models's performance is evaluated on objects of interest within the range of -52 meters to 52 meters. This table denotes pedestrian as 'Pedes', motorcycle as 'MotorC' and Traffic Cone as 'Cone'.

Urban → Rural	Networks	mATE	mASE	mAOE	mAVE	mAAE	NDS
Original	CenterPoint	0.3568	0.7162	1.5531	2.1255	0.3458	0.3073
	PointPillar	0.7283	0.5454	1.1286	1.2729	0.5813	0.1484
	FreeAnchor	0.5729	0.3528	1.0319	1.8685	0.4515	0.2049
Ours	CenterPoint	0.3326	0.2723	0.6591	1.9451	0.3586	0.4254
	PointPillar	0.5085	0.3327	1.1005	1.7025	0.4339	0.2202
	FreeAnchor	0.5119	0.3418	0.9673	1.6959	0.4373	0.2468

5.3.3 Public-to-Campus Domain Adaptation Results

The table in 5.6 presents the results of the 3D object detection task for the public to campus domain adaptation. For several object classes, such as cars, buses, and trucks, the proposed technique outperformed the original networks, with mAP increases ranging from 5% to 10%. In particular, the proposed method showed exceptional performance in detecting pedestrians, with a significant improvement of 23.7%, 19.5%, and 13.5% in mAP over the original CenterPoint, Pointillar, and FreeAnchor networks, respectively. Similarly, the improvement for trucks was also substantial, with increases of 17.1%, 3.5%, and 5.1% over the original CenterPoint, Pointillar, and FreeAnchor networks, respectively. The mAP for the car class was also higher, with a 0.738, 0.449 and 0.496 mAP compared to 0.708, 0.447 and 0.409 for the original CenterPoint, Pointillar, and FreeAnchor networks. However, similar to the results from the urban-to-rural experiment, the PointPillar and FreeAnchor networks failed to generalize well for certain object classes such as bicycle, motorcycle, and traffic cone. This could be attributed to the fact that these objects are sparsely represented in the dataset and require additional training data to perform accurately.

The proposed approach also demonstrates improvements in the NDS metric across

Table 5.6: 3D object detection results on nuScenes for public to campus adaptation. This table report mean average precision (mAP) averaged over match thresholds of 0.5, 1, 2, and 4 meters. The models's performance is evaluated on objects of interest within the range of -52 meters to 52 meters. This table denotes pedestrian as 'Pedes', motorcycle as 'MotorC' and Traffic Cone as 'Cone'.

Public → Campus	Networks	Car	Bus	Truck	Pedes	Bicycle	MotorC	Cone
Original	CenterPoint	0.708	0.384	0.284	0.551	0.005	0.192	**0.349**
	PointPillar	0.447	**0.275**	0.051	0.143	0.000	0.000	0.000
	FreeAnchor	0.409	0.380	0.146	0.356	0.000	0.049	0.000
Ours	CenterPoint	**0.738**	**0.707**	**0.455**	**0.788**	**0.005**	**0.371**	0.284
	PointPillar	**0.449**	0.266	**0.086**	**0.338**	0.000	**0.039**	**0.002**
	FreeAnchor	**0.496**	**0.524**	**0.197**	**0.491**	0.000	**0.084**	0.000

all networks, as shown in Table 5.7. Specifically, the adapted CenterPoint, PointPillar, and FreeAnchor networks achieve NDS values of 0.3979, 0.2202, and 0.2328, respectively, outperforming their original counterparts, which have NDS values of 0.3356, 0.1652, and 0.2263. This highlights the effectiveness of our method in enhancing the overall performance of the networks in the public-to-campus domain adaptation scenario.

Overall, the results from the public to campus experiment demonstrate that the proposed method effectively adapts to different domains and enhances the performance of 3D object detection networks in various object classes. The results are consistent with the findings from the city to city and urban to rural experiments, further confirming the effectiveness and generalizability of the proposed method.

To summarize, the proposed PnP method demonstrated significant improvement in 3D object detection for cross-scene adaptation over their original models. Specifically, the proposed method enhanced mAP across multiple object classes, including cars, pedestrians, buses, and trucks. However, for certain object classes, the original networks (e.g. PointPillar and FreeAnchor) failed in both the source and target domains. This demonstrates how important generalization in the source domain is for successful adaptation in the target

Table 5.7: 3D object detection results average on all classes for urban to rual domain adaptation. This table reports mATE (Meters) mASE(1-IoU) mAOE(Rad) mAVE(m/s) mAAE(1-acc) NDS averaged over all classes. The models's performance is evaluated on objects of interest within the range of -52 meters to 52 meters. This table denotes pedestrian as 'Pedes', motorcycle as 'MotorC' and Traffic Cone as 'Cone'.

Public → Campus	Networks	mATE	mASE	mAOE	mAVE	mAAE	NDS
Original	CenterPoint	0.3394	0.7035	1.5542	1.5040	0.3236	0.3356
	PointPillar	0.6949	0.5097	1.1206	1.4187	0.5635	0.1652
	FreeAnchor	0.5950	0.3443	1.0187	1.5384	0.4783	0.2263
Ours	CenterPoint	0.3837	0.3986	0.6279	1.5043	0.4740	0.3979
	PointPillar	0.5085	0.3327	1.1005	1.7025	0.4339	0.2202
	FreeAnchor	0.4935	0.4685	0.9693	1.7897	0.6371	0.2328

domain. Overall, the experiments results support the conclusion that the proposed method is a promising approach for addressing the challenges of cross-scene domain adaptation in 3D object detection. These findings have important implications for autonomous vehicle and robotics applications, where robust object detection is crucial for safe and efficient navigation.

Ablation study

The ablation study on the same-scene validation set shows that the proposed proposed method outperforms the original methods. This is especially evident for small-size objects such as pedestrians and traffic cones, where the proposed method achieved higher mAP scores compared to the original methods. The improved performance is likely attribute to the reduction of irrelevant information in the point cloud with background points removed. This enables the model to focus on important objects for the task. To conclude, the proposed approach demonstrates its effectiveness in enhancing the performance of 3D object detection algorithms in same-location scenarios.

Table 5.8: 3D object detection results on nuScenes for same-location validation. This table report mean average precision (mAP) averaged over match thresholds of 0.5, 1, 2, and 4 meters. The models's performance is evaluated on objects of interest within the range of -52 meters to 52 meters. This table denotes pedestrian as 'Pedes', motorcycle as 'MotorC' and Traffic Cone as 'Cone'.

Boston → Boston	Networks	Car	Bus	Truck	Pedes	Bicycle	MotorC	Cone
Original	CenterPoint	0.779	0.563	0.425	0.757	0.340	0.150	0.670
	PointPillar	0.609	0.169	0.176	0.093	0.000	0.000	0.018
	FreeAnchor	0.501	0.095	0.141	0.335	0.000	0.000	0.110
Ours	CenterPoint	**0.795**	**0.580**	**0.440**	**0.839**	**0.359**	**0.195**	**0.755**
	PointPillar	**0.617**	**0.208**	**0.222**	**0.356**	**0.057**	**0.002**	**0.248**
	FreeAnchor	**0.573**	**0.188**	**0.200**	**0.459**	0.000	0.000	**0.274**

5.3.4 Real-time performance

To evaluate the proposed methods on mobile computing platform Jetson AGX Xavier, optimization has been performed on CenterPoint, PointPillars and FreeAnchor for real-time performace. More specifically, all models are first converted from their original Pytorch format to the ONNX (Open Neural Network Exchange) format. The ONNX format enables them to be deployed across a variety of platforms and devices. Subsequently, the ONNX models are converted into TensorRT engine files. TensorRT is a high-performance deep learning inference optimizer and runtime. It provides capabilities such as layer fusion, precision calibration, kernel auto-tuning, and dynamic tensor caching to enhance the performance of deep learning models. By leveraging the TensorRT runtime, models can be optimized to accelerate inference times with minimal accuracy losses.

These model optimization process has resulted in significant improvements in real-time performance for all three models compared to their original PyTorch versions. Most notably, the CenterPoint model's performance improved from 4Hz in pytorch to 20Hz for the proposed background-aware CenterPoint using TensorRT runtime. This also surpassed the performance of the original CenterPoint-TensorRT model, operating at 18Hz. Similarly,

the PointPillars model, which initially operated at the threshold of real-time performance at 10Hz, improved its performance to 12Hz with the proposed background-aware method.

5.4 Summary

To summarize, this chapter presents a novel approach for PnP background-aware domain adaptation for 3D point cloud-based deep learning models. The proposed method removes background points to reduce the amount of irrelevant information in the point cloud. This results in improved performance on both inter-domain and intra-domain tasks. The experiments conducted on the Domains-3D dataset demonstrate the effectiveness of this approach in reducing domain shift and improving the model's generalization capability.

Chapter 6

Conclusions and Outlooks

This chapter presents conclusions and future research directions. The main findings of this thesis are summarised in Section 6.1. Finally, recommendations for future study are presented in section 6.2.

6.1 Conclusions

In this thesis, several contributions to the field of 3D perception have been presented, with focuses on domain adaptation and computation efficiency. Firstly, to tackle the real-time performance issue of point-cloud-based 3D perception systems, a task-attentive 3D object detection framework is proposed. It reduces computation and memory complexity while enhancing small obstacle identification. This paradigm also includes the concept of PIA, which are adaptive regions adjacent to the drivable area that might contain dynamic objects that could interact with the ego vehicle.

Furthermore, a brand new dataset called Domains-3D, was developed to fill the gap of large-scale domain adaptation dataset for autonomous driving application. Domains-3D is a comprehensive domain adaption dataset that includes both real-world and synthetic

data. It contains 18 cross-scene domains that may be partitioned into 54 different domain splits. The proposed synthetic data generation method allows real-world domains to be closely re-created in virtual simulations. As a result, improved strategies for incorporating simulated data to enhance the performance of perception systems in real-world scenarios were provided.

Finally, to improve the generalization capability of existing point-cloud-based deep learning models, a novel PnP background-aware domain adaptation was proposed. These experiments showed that removing background points reduces the amount of domain shift between the source and target domains and the amount of irrelevant information in the point cloud. This leads to improved performance on both target and source domains.

To conclude, this thesis has demonstrated that reducing computation and memory complexity, as well as using domain adaption techniques, can greatly improve the performance of 3D object detection models for autonomous driving. The proposed methods have significant implications for advancing open-world autonomous driving by enhancing the efficiency and robustness of SOTA point cloud based perception systems.

6.2 Outlook And Future Works

For future directions, there are multiple avenues that the proposed perception algorithm could be further improved. One of the under addressed issue of this thesis is open-set detection, where model trained on a close-set dataset encounters unknown objects outside training set. These previously unknown objects often cause performance degradation of a deep learning model. This open-set challenge is especially critical in the context of autonomous driving systems designed for open-world driving, where the system needs to safely and efficiently respond to a vast array of potential unseen scenarios.

Although this thesis has touched on this problem to a degree in chapter 3 from a system design perspective using class-agnostic detector such as clustering model. However, this subject is not explicitly explored in this thesis. An interesting line of future research

could be integrating SOTA techniques such as few-shot learning, zero-shot learning, or meta-learning into the current framework. Furthermore, the recent advances in multi-modal large language models (LLMs) provide a compelling case for their potential utility in this context. LLMs, with their capacity for one-shot and zero-shot learning, could be leveraged to improve the system's ability to identify and handle novel objects or situations. This combined with the LLM's ability to reason and understanding commonsense could potentially lead to a breakthrough in open-set detection.

The proposed task-attentive 3D object detection framework could be supplemented with additional sensor modalities such as radar and camera data. These additional information might facilitate a more robust understanding of the environment. This could further improve the accuracy of PIA estimation and object detection. The noval Domains-3D dataset developed by this thesis, which was utilized to train and test the proposed domain adaption strategies, also presents opportunities for further refinement. The impact of the sim-to-real gap generated by the dataset was not explored within the scope of this thesis. Investigation in this direction could provide valuable insights into the generalization capabilities of the proposed PnP Background-Aware Point Cloud Domain Adaptation method. Another promising direction is to incorporate adversarial scene generation into the process of synthetic data generation. This involves testing generated scenes against a suite of representative models and incorporating only the most challenging scenes into the updated dataset. This updated dataset, along with new edge cases identified during real-world testing, were utilized to fine-tune the target model. This procedure is then repeated for the next iteration, with the target model constantly improving. Finally, integrating all the proposed methods into one perception system and evaluating its performance in real-world scenarios on autonomous vehicles might further demonstrate its practicality and effectiveness.

Letter of Copyright Permission

IEEE, as the publisher of the two manuscripts fully or partly adopted in Chapter 1, Chapter 2, Chapter 3, Chapter 5, and Chapter 6 allow the reuse of published papers in the thesis without formal permissions. Thus, the waivers of copyright from IEEE are achieved by the following statement:

Policy Regarding Thesis/Dissertation Reuse, from IEEE Copyright Clearance Centre

"The IEEE does not require individuals working on a thesis to obtain a formal reuse license;however, you may print out this statement to be used as a permission grant".

References

[1] Baidu Apollo team (2017), Apollo: Open Source Autonomous Driving, howpublished = https://github.com/apolloauto/apollo, note = Accessed: 2019-02-11.

[2] David Acuna, Jonah Philion, and Sanja Fidler. Towards optimal strategies for training self-driving perception models in simulation. 2022.

[3] David Acuna, Guojun Zhang, Marc T Law, and Sanja Fidler. f-domain adversarial learning: Theory and algorithms. In *International Conference on Machine Learning*, pages 66–75. PMLR, 2021.

[4] Iñigo Alonso, Luis Montesano, Luis Riazuelo, and Ana Cristina Murillo. Domain adaptation in lidar semantic segmentation by aligning class distributions. In *Proc.(Int. Asia Conf. Inform. Control, Autom., Robot.)*, number ART-2021-124899, 2021.

[5] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[6] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017.

[7] Alireza Asvadi, Paulo Peixoto, and Urbano Nunes. Detection and tracking of moving objects using 2.5d motion grids. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 788–793, 2015.

[8] Alireza Asvadi, Cristiano Premebida, Paulo Peixoto, and Urbano Nunes. 3d lidar-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes. *Robotics and Autonomous Systems*, 83:299–311, 2016.

[9] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *ACM Transactions on Graphics*, 37(4):1–12, Aug 2018.

[10] H. Badino, D. Huber, and T. Kanade. Integrating lidar into stereo for fast and improved disparity computation. In *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 405–412, May 2011.

[11] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019.

[12] Yochanan E Bigman and Kurt Gray. Life and death decisions of autonomous vehicles. *Nature*, 579(7797):E1–E2, 2020.

[13] D. Boscaini, J. Masci, S. Melzi, M. M. Bronstein, U. Castellani, and P. Vandergheynst. Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. *Computer Graphics Forum*, 34(5):13–23, 2015.

[14] Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV (1)*, pages 44–57, 2008.

[15] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*, 2014.

[16] Lucas Caccia, Herke Van Hoof, Aaron Courville, and Joelle Pineau. Deep generative modeling of lidar data. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5034–5040. IEEE, 2019.

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

[18] Luca Caltagirone, Mauro Bellone, Lennart Svensson, and Mattias Wahde. Lidar–camera fusion for road detection using fully convolutional neural networks. *Robotics and Autonomous Systems*, 111:125–131, Jan 2019.

[19] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 135–150, 2018.

[20] Juan Castorena, Gint Puskorius, and Gaurav Pandey. Motion guided lidar-camera self-calibration and accelerated depth upsampling, 2018.

[21] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[22] L. Chen, Q. Zou, Z. Pan, D. Lai, L. Zhu, Z. Hou, J. Wang, and D. Cao. Surrounding vehicle detection using an fpga panoramic camera and deep cnns. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–13, 2019.

[23] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[24] S. Chen, C. Duan, Y. Yang, D. Li, C. Feng, and D. Tian. Deep unsupervised learning of 3d point clouds via graph topology inference and filtering. *IEEE Transactions on Image Processing*, 29:3183–3198, 2020.

[25] Tongtong Chen, Bin Dai, Ruili Wang, and Daxue Liu. Gaussian-process-based real-time ground segmentation for autonomous land vehicles. *Journal of Intelligent & Robotic Systems*, 76(3):563–582, Dec 2014.

[26] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 424–432. Curran Associates, Inc., 2015.

[27] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals using stereo imagery for accurate object class detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1259–1272, May 2018.

[28] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[29] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7892–7901, 2018.

[30] Yun Chen, Frieda Rong, Shivam Duggal, Shenlong Wang, Xinchen Yan, Sivabalan Manivasagam, Shangjie Xue, Ersin Yumer, and Raquel Urtasun. Geosim: Realistic video simulation via geometry-aware composition for self-driving. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2021.

[31] Xinjing Cheng, Peng Wang, Guan Chenye, and Ruigang Yang. Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. *Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)*, 2020.

[32] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. *Lecture Notes in Computer Science*, page 108–125, 2018.

[33] Xuelian Cheng, Yiran Zhong, Yuchao Dai, Pan Ji, and Hongdong Li. Noise-aware unsupervised deep lidar-stereo fusion. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.

[34] Hung-Yueh Chiang, Yen-Liang Lin, Yueh-Cheng Liu, and Winston H. Hsu. A unified point-based framework for 3d segmentation. *2019 International Conference on 3D Vision (3DV)*, Sep 2019.

[35] Nathaniel Chodosh, Chaoyang Wang, and Simon Lucey. Deep convolutional compressed sensing for lidar depth completion. *Lecture Notes in Computer Science*, page 499–513, 2019.

[36] Sunglok Choi, Taemin Kim, and Wonpil Yu. Performance evaluation of ransac family. volume 24, 01 2009.

[37] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

[38] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[39] Imre Csiszár. Information-type measures of difference of probability distributions and indirect observation. *studia scientiarum Mathematicarum Hungarica*, 2:229–318, 1967.

[40] Yaodong Cui, Ren Chen, Wenbo Chu, Long Chen, Daxin Tian, Ying Li, and Dongpu Cao. Deep learning for image and point cloud fusion in autonomous driving: A review. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):722–739, 2021.

[41] Yaodong Cui, Ren Chen, Wenbo Chu, Long Chen, Daxin Tian, Ying Li, and Dongpu Cao. Deep learning for image and point cloud fusion in autonomous driving: A review. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):722–739, 2022.

[42] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias NieBner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[43] Angela Dai and Matthias Nießner. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. *Lecture Notes in Computer Science*, page 458–474, 2018.

[44] M. Darms, P. Rybski, and C. Urmson. Classification and tracking of dynamic objects with multiple sensors for autonomous driving in urban environments. In *2008 IEEE Intelligent Vehicles Symposium*, pages 1197–1202, June 2008.

[45] Belur V Dasarathy. Sensor fusion potential exploitation-innovative architectures and illustrative applications. *Proceedings of the IEEE*, 85(1):24–38, 1997.

[46] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[47] Dominic Zeng Wang, I. Posner, and P. Newman. What could move? finding cars, pedestrians and bicyclists in 3d laser data. In *2012 IEEE International Conference on Robotics and Automation*, pages 4038–4044, May 2012.

[48] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[49] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel. On the segmentation of 3d lidar point clouds. In *2011 IEEE International Conference on Robotics and Automation*, pages 2798–2805, May 2011.

[50] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel. On the segmentation of 3d lidar point clouds. In *2011 IEEE International Conference on Robotics and Automation*, pages 2798–2805, May 2011.

[51] S. Dreyfus. The computational solution of optimal control problems with time lag. *IEEE Transactions on Automatic Control*, 18(4):383–385, August 1973.

[52] X. Du and K. K. Tan. Comprehensive and practical vision system for self-driving vehicle lane-level localization. *IEEE Transactions on Image Processing*, 25(5):2075–2088, 2016.

[53] Xinxin Du, Marcelo H. Ang, Sertac Karaman, and Daniela Rus. A general pipeline for 3d detection of vehicles. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018.

[54] Fabio Duarte. Self-driving cars: A city perspective. *Science Robotics*, 4(28), 2019.

[55] Andreas Eitel, Jost Tobias Springenberg, Luciano Spinello, Martin Riedmiller, and Wolfram Burgard. Multimodal deep learning for robust rgb-d object recognition. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep 2015.

[56] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Propagating confidences through cnns for sparse data regression. *arXiv preprint arXiv:1805.11913*, 2018.

[57] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Confidence propagation through cnns for guided sparse depth regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1, 2019.

[58] Asmaa Elhadidy, Mohamed Afifi, Mohammed Hassoubah, Yara Ali, and Mohamed ElHelw. Improved semantic segmentation of low-resolution 3d point clouds using supervised domain adaptation. In *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pages 588–593, 2020.

[59] Cathrin Elich, Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. 3d bird's-eye-view instance segmentation. *Pattern Recognition*, page 48–61, 2019.

[60] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Simoudis et al. [60], pages 226–231.

[61] Di Feng, Christian Haase-Schutz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, page 1–20, 2020.

[62] Davi Frossard and Raquel Urtasun. End-to-end learning of multi-sensor 3d tracking by detection. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018.

[63] V. Gandhi, J. Čech, and R. Horaud. High-resolution depth maps based on tof-stereo fusion. In *2012 IEEE International Conference on Robotics and Automation*, pages 4742–4749, May 2012.

[64] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[65] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[66] D. C. Garcia, T. A. Fonseca, R. U. Ferreira, and R. L. de Queiroz. Geometry coding for dynamic voxelized point clouds using octrees and multiple contexts. *IEEE Transactions on Image Processing*, 29:313–322, 2020.

[67] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[68] Chuanxing Geng, Sheng-Jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3614–3631, 2021.

[69] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun 2014.

[70] Fabian Groh, Patrick Wieschollek, and Hendrik P. A. Lensch. Flex-convolution. *Lecture Notes in Computer Science*, page 105–122, 2019.

[71] S. Gu, T. Lu, Y. Zhang, J. M. Alvarez, J. Yang, and H. Kong. 3-d lidar + monocular camera: An inverse-depth-induced fusion framework for urban road detection. *IEEE Transactions on Intelligent Vehicles*, 3(3):351–360, 2018.

[72] S. Gu, Y. Zhang, J. Tang, J. Yang, J. M. Alvarez, and H. Kong. Integrating dense lidar-camera road detection maps by a multi-modal crf model. *IEEE Transactions on Vehicular Technology*, 68(12):11635–11645, Dec 2019.

[73] J. Guo, U. Kurup, and M. Shah. Is it safe to drive? an overview of factors, metrics, and datasets for driveability assessment in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–17, 2019.

[74] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(12):4338–4364, 2020.

[75] S. Gupta, P. Arbeláez, R. Girshick, and J. Malik. Aligning 3d models to rgb-d images of cluttered scenes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4731–4740, June 2015.

[76] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. *Lecture Notes in Computer Science*, page 345–360, 2014.

[77] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

[78] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[79] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

[80] Zhenwei He and Lei Zhang. Multi-adversarial faster-rcnn for unrestricted object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6668–6677, 2019.

[81] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.

[82] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data, 2015.

[83] P. Hermosilla, T. Ritschel, P-P Vazquez, A. Vinacua, and T. Ropinski. Monte carlo convolution for learning on non-uniformly sampled point clouds. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2018)*, 37(6), 2018.

[84] M. Himmelsbach, Felix v. Hundelshausen, and H.-J. Wuensche. Fast segmentation of 3d point clouds for ground vehicles. In *2010 IEEE Intelligent Vehicles Symposium*, pages 560–565, 2010.

[85] Michael Himmelsbach, Felix von Hundelshausen, and Hans-Joachim Wünsche. Fast segmentation of 3d point clouds for ground vehicles. *2010 IEEE Intelligent Vehicles Symposium*, pages 560–565, 2010.

[86] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. Pmlr, 2018.

[87] Ji Hou, Angela Dai, and Matthias NieBner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.

[88] Han-Kai Hsu, Chun-Han Yao, Yi-Hsuan Tsai, Wei-Chih Hung, Hung-Yu Tseng, Maneesh Singh, and Ming-Hsuan Yang. Progressive domain adaptation for object detec-

tion. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 749–757, 2020.

[89] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11108–11117, 2020.

[90] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[91] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 984–993, 2018.

[92] Haoshuo Huang, Qixing Huang, and Philipp Krahenbuhl. Domain transfer through deep activation matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 590–605, 2018.

[93] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and et al. Speed/accuracy trade-offs for modern convolutional object detectors. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[94] P. Huang, M. Cheng, Y. Chen, H. Luo, C. Wang, and J. Li. Traffic sign occlusion detection using mobile laser scanning point clouds. *IEEE Transactions on Intelligent Transportation Systems*, 18(9):2364–2376, 2017.

[95] Z. Huang, J. Fan, S. Cheng, S. Yi, X. Wang, and H. Li. Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion. *IEEE Transactions on Image Processing*, 29:3429–3441, 2020.

[96] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller. Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction. *IEEE Transactions on Intelligent Vehicles*, 3(1):5–17, 2018.

[97] Braden Hurl, Krzysztof Czarnecki, and Steven Waslander. Precise synthetic image and lidar (presil) dataset for autonomous vehicle perception. *2019 IEEE Intelligent Vehicles Symposium (IV)*, Jun 2019.

[98] E. Imre, J. Guillemaut, and A. Hilton. Through-the-lens multi-camera synchronisation and frame-drop detection for 3d reconstruction. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission*, pages 395–402, 2012.

[99] DJI Inc. DJI Drone product description, 2023.

[100] ZOOX inc. ZOOX shuttlebus product description, 2023.

[101] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.

[102] Neil Jacobstein. Autonomous vehicles: An imperfect path to saving millions of lives. *Science Robotics*, 4(28), 2019.

[103] Maximilian Jaritz, Raoul De Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. Sparse and dense data with cnns: Depth completion and semantic segmentation. *2018 International Conference on 3D Vision (3DV)*, Sep 2018.

[104] Maximilian Jaritz, Jiayuan Gu, and Hao Su. Multi-view pointnet for 3d scene understanding, 2019.

[105] Maximilian Jaritz, Tuan-Hung Vu, Raoul de Charette, Emilie Wirbel, and Patrick Pérez. xmuda: Cross-modal unsupervised domain adaptation for 3d semantic seg-

mentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12605–12614, 2020.

[106] Mingyang Jiang, Yiran Wu, Tianqi Zhao, Zelin Zhao, and Cewu Lu. Pointsift: A sift-like network module for 3d point cloud semantic segmentation, 2018.

[107] Guoliang Kang, Liang Zheng, Yan Yan, and Yi Yang. Deep adversarial attention alignment for unsupervised domain adaptation: the benefit of target expectation maximization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 401–416, 2018.

[108] H. Karunasekera, H. Wang, and H. Zhang. Energy minimization approach for negative obstacle region detection. *IEEE Transactions on Vehicular Technology*, 68(12):11668–11678, Dec 2019.

[109] Sabine Kastner and Mark A Pinsk. Visual attention as a multilevel selection process. *Cognitive, Affective, & Behavioral Neuroscience*, 4(4):483–500, 2004.

[110] Shinpei Kato, Shota Tokunaga, Yuya Maruyama, Seiya Maeda, Manato Hirabayashi, Yuki Kitsukawa, Abraham Monrroy, Tomohito Ando, Yusuke Fujii, and Takuya Azumi. Autoware on board: Enabling autonomous vehicles with embedded systems. In *Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems*, ICCPS '18, page 287–296. IEEE Press, 2018.

[111] Mehran Khodabandeh, Arash Vahdat, Mani Ranjbar, and William G Macready. A robust learning approach to domain adaptive object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 480–490, 2019.

[112] S. Kim, H. Kim, W. Yoo, and K. Huh. Sensor fusion algorithm design in detecting vehicles using laser scanner and stereo vision. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1072–1084, 2016.

[113] S. Kim, D. Min, S. Kim, and K. Sohn. Unified confidence estimation networks for robust stereo matching. *IEEE Transactions on Image Processing*, 28(3):1299–1313, 2019.

[114] Taekyung Kim, Minki Jeong, Seunghyeon Kim, Seokeon Choi, and Changick Kim. Diversify and match: A domain adaptive representation learning paradigm for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12456–12465, 2019.

[115] Stephen Cole Kleene. Representation of events in nerve nets and finite automata. Technical report, RAND PROJECT AIR FORCE SANTA MONICA CA, 1951.

[116] Roman Klokov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[117] Artem Komarichev, Zichun Zhong, and Jing Hua. A-cnn: Annularly convolutional neural networks on point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7421–7430, 2019.

[118] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25, 01 2012.

[119] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[120] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L. Waslander. Joint 3d proposal generation and object detection from view aggregation. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018.

[121] Harold W. Kuhn. The hungarian method for the assignment problem. In *50 Years of Integer Programming*, 2010.

[122] Shiyi Lan, Ruichi Yu, Gang Yu, and Larry S. Davis. Modeling local geometric structure of 3d point clouds using geo-cnn. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.

[123] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.

[124] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[125] William Larson and Weihua Zhao. Self-driving cars and the city: Effects on sprawl, energy consumption, and housing affordability. *Regional Science and Urban Economics*, page 103484, 2019.

[126] Nalpantidis Lazaros, Georgios C. Sirakoulis, and Antonios Gasteratos. Review of stereo vision algorithms: From software to hardware. 2008.

[127] Huan Lei, Naveed Akhtar, and Ajmal Mian. Octree guided cnn with spherical kernels for 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9631–9640, 2019.

[128] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J. Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, Michael Sokolsky, Ganymed Stanek, David Stavens, Alex Teichman, Moritz Werling, and Sebastian Thrun. Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 163–168, 2011.

[129] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3d lidar using fully convolutional network. *Robotics: Science and Systems XII*.

[130] Daiqing Li, Huan Ling, Seung Wook Kim, Karsten Kreis, Adela Barriuso, Sanja Fidler, and Antonio Torralba. Bigdatasetgan: Synthesizing imagenet with pixel-wise annotations, 2022.

[131] Peiliang Li, Xiaozhi Chen, and Shaojie Shen. Stereo r-cnn based 3d object detection for autonomous driving. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.

[132] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 820–830. Curran Associates, Inc., 2018.

[133] Ying Li, Lingfei Ma, Weikai Tan, Chen Sun, Dongpu Cao, and Jonathan Li. Grnet: Geometric relation network for 3d object detection from point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 165:43–53, 2020.

[134] Ying Li, Lingfei Ma, Zilong Zhong, Fei Liu, Michael A. Chapman, Dongpu Cao, and Jonathan Li. Deep learning for lidar point clouds in autonomous driving: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8):3412–3432, 2021.

[135] Yunpeng Li, Dominik Roblek, and Marco Tagliasacchi. From here to there: Video inbetweening using direct 3d convolutions. *arXiv preprint arXiv:1905.10240*, 2019.

[136] Jian Liang, Ran He, Zhenan Sun, and Tieniu Tan. Aggregating randomized clustering-promoting invariant projections for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 41(5):1027–1042, 2018.

[137] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[138] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[139] Hyungtae Lim, Minho Oh, and Hyun Myung. Patchwork: Concentric zone-based region-wise ground segmentation with ground likelihood estimation using a 3d lidar sensor. *IEEE Robotics and Automation Letters*, 6(4):6458–6465, Oct 2021.

[140] Jiarong Lin and Fu Zhang. Loam livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3126–3131, 2020.

[141] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[142] Seppo Linnainmaa. Taylor expansion of the accumulated rounding error. *BIT Numerical Mathematics*, 16(2):146–160, 1976.

[143] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, 2018.

[144] Dongnan Liu, Donghao Zhang, Yang Song, Fan Zhang, Lauren O'Donnell, Heng Huang, Mei Chen, and Weidong Cai. Unsupervised instance segmentation in microscopy images via panoptic domain adaptation and task re-weighting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4243–4252, 2020.

[145] Xiaofeng Liu, Chaehwa Yoo, Fangxu Xing, Hyejin Oh, Georges El Fakhri, Je-Won Kang, Jonghye Woo, et al. Deep unsupervised domain adaptation: A review of recent advances and perspectives. *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022.

[146] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. *CoRR*, abs/1904.07601, 2019.

[147] Ze Liu, Han Hu, Yue Cao, Zheng Zhang, and Xin Tong. A closer look at local aggregation operators in point cloud analysis. *arXiv preprint arXiv:2007.01294*, 2020.

[148] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

[149] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. *Advances in neural information processing systems*, 29, 2016.

[150] H. Lu, X. Chen, G. Zhang, Q. Zhou, Y. Ma, and Y. Zhao. Scanet: Spatial-channel attention network for 3d object detection. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1992–1996, May 2019.

[151] Jonathon Luiten, Tobias Fischer, and Bastian Leibe. Track to reconstruct and reconstruct to track. *IEEE Robotics and Automation Letters*, 5(2):1803–1810, Apr 2020.

[152] R. C. Luo, C. C. Chang, and C. C. Lai. Multisensor fusion and integration: Theories, applications, and its perspectives. *IEEE Sensors Journal*, 11(12):3122–3138, Dec 2011.

[153] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2507–2516, 2019.

[154] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. *2019 International Conference on Robotics and Automation (ICRA)*, May 2019.

[155] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018.

[156] W. Maddern and P. Newman. Real-time probabilistic fusion of sparse 3d lidar and dense stereo. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2181–2188, Oct 2016.

[157] William P. Maddern, Alexander D. Stewart, Colin McManus, Ben Upcroft, Winston Churchill, and Paul Newman. Illumination invariant imaging : Applications in robust vision-based localisation , mapping and classification for autonomous vehicles. 2014.

[158] Mariem Maiouak and Tarik Taleb. Dynamic maps for automated driving and uav geofencing. *IEEE Wireless Communications*, 26(4):54–59, 2019.

[159] Sivabalan Manivasagam, Shenlong Wang, Kelvin Wong, Wenyuan Zeng, Mikita Sazanovich, Shuhan Tan, Bin Yang, Wei-Chiu Ma, and Raquel Urtasun. Lidarsim: Realistic lidar simulation by leveraging the real world. *computer vision and pattern recognition*, 2020.

[160] Jiageng Mao, Xiaogang Wang, and Hongsheng Li. Interpolated convolutional networks for 3d point cloud understanding. *CoRR*, abs/1908.04512, 2019.

[161] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, Sep. 2015.

[162] Gregory P. Meyer, Jake Charland, Darshan Hegde, Ankit Laddha, and Carlos Vallespi-Gonzalez. Sensor fusion for joint 3d object detection and semantic segmentation, 2019.

[163] Gregory P. Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and Carl K. Wellington. Lasernet: An efficient probabilistic 3d object detector for autonomous driving. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.

[164] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015.

[165] F. Moosmann, O. Pink, and C. Stiller. Segmentation of 3d lidar data in non-flat urban environments using a local convexity criterion. In *2009 IEEE Intelligent Vehicles Symposium*, pages 215–220, June 2009.

[166] Pietro Morerio, Jacopo Cavazza, and Vittorio Murino. Minimal-entropy correlation alignment for unsupervised deep domain adaptation. *International Conference on Learning Representations*, 2018.

[167] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[168] M. G. Mozerov and J. van de Weijer. Accurate stereo matching by two-step energy minimization. *IEEE Transactions on Image Processing*, 24(3):1153–1163, 2015.

[169] B. Nagy, L. Kovács, and C. Benedek. Online targetless end-to-end camera-lidar self-calibration. In *2019 16th International Conference on Machine Vision Applications (MVA)*, pages 1–6, May 2019.

[170] Kazuto Nakashima and Ryo Kurazume. Learning to drop points for lidar scan synthesis. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 222–229. IEEE, 2021.

[171] Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2019.

[172] Patiphon Narksri, Eijiro Takeuchi, Yoshiki Ninomiya, Yoichi Morales, Naoki Akai, and Nobuo Kawaguchi. A slope-robust cascaded ground segmentation in 3d point cloud for autonomous vehicles. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 497–504, 2018.

[173] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017.

[174] Kevin Nickels, Andres Castano, and Christopher M. Cianci. Fusion of lidar and stereo range for mobile robots. 2003.

[175] NIO. NIO ET7 product description, 2023.

[176] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62, 2021.

[177] K. Park, S. Kim, and K. Sohn. High-precision depth estimation using uncalibrated lidar and stereo fusion. *IEEE Transactions on Intelligent Transportation Systems*, 21(1):321–335, Jan 2020.

[178] S. H. Park, B. Kim, C. M. Kang, C. C. Chung, and J. W. Choi. Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1672–1678, 2018.

[179] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation, 2016.

[180] Abhishek Patil, Srikanth Malla, Haiming Gang, and Yi-Ting Chen. The h3d dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes. *2019 International Conference on Robotics and Automation (ICRA)*, May 2019.

[181] Quang-Hieu Pham, Pierre Sevestre, Ramanpreet Singh Pahwa, Huijing Zhan, Chun Ho Pang, Yuda Chen, Armin Mustafa, Vijay Chandrasekhar, and Jie Lin. A*3d dataset: Towards autonomous driving in challenging environments. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020.

[182] Florian Piewak, Peter Pinggera, and Marius Zöllner. Analyzing the cross-sensor portability of neural network architectures for lidar-based semantic labeling. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3419–3426. IEEE, 2019.

[183] Cristiano Premebida, J. Carreira, Jorge Batista, and Urbano Nunes. Pedestrian detection combining rgb and dense lidar data. pages 4112–4117, 09 2014.

[184] Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018.

[185] Charles R. Qi, Hao Su, Matthias NieBner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

[186] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, 2017.

[187] Can Qin, Haoxuan You, Lichen Wang, C-C Jay Kuo, and Yun Fu. Pointdan: A multi-scale 3d domain adaption network for point cloud representation. *Advances in Neural Information Processing Systems*, 32, 2019.

[188] M. M. Rahman, Y. Tan, J. Xue, and K. Lu. Recent advances in 3d object detection in the era of deep neural networks: A survey. *IEEE Transactions on Image Processing*, 29:2947–2962, 2020.

[189] B. Ranft and C. Stiller. The role of machine vision for intelligent vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):8–19, 2016.

[190] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

[191] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3577–3586, 2017.

[192] Christoph B Rist, Markus Enzweiler, and Dariu M Gavrila. Cross-sensor deep domain adaptation for lidar detection and segmentation. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1535–1542. IEEE, 2019.

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

[194] Guodong Rong, Byung Hyun Shin, Hadi Tabatabaee, Qiang Lu, Steve Lemke, Mārtiņš Možeiko, Eric Boise, Geehoon Uhm, Mark Gerow, Shalin Mehta, et al. Lgsvl simulator: A high fidelity simulator for autonomous driving. *arXiv preprint arXiv:2005.03778*, 2020.

[195] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[196] Olga Russakovsky and Li Fei-Fei. Attribute learning in large-scale datasets. In *European Conference of Computer Vision (ECCV), International Workshop on Parts and Attributes*, Crete, Greece, September 2010.

[197] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[198] Khaled Saleh, Ahmed Abobakr, Mohammed Attia, Julie Iskander, Darius Nahavandi, Mohammed Hossny, and Saeid Nahvandi. Domain adaptation for vehicle detection from bird's eye view lidar point cloud data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[199] Ahmad El Sallab, Ibrahim Sobh, Mohamed Zahran, and Nader Essam. Lidar sensor modeling and data augmentation with gans for autonomous driving. *CoRR*, abs/1905.07290, 2019.

[200] J. Schlosser, C. K. Chow, and Z. Kira. Fusing lidar and images for pedestrian detection using convolutional neural networks. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2198–2205, May 2016.

[201] Tixiao Shan, Jinkun Wang, Fanfei Chen, Paul Szenher, and Brendan Englot. Simulation-based lidar super-resolution for ground vehicles. *Robotics and Autonomous Systems*, 134:103647, 2020.

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

[203] Kiwoo Shin, Youngwook Paul Kwon, and Masayoshi Tomizuka. Roarnet: A robust 3d object detection based on region approximation refinement. *2019 IEEE Intelligent Vehicles Symposium (IV)*, Jun 2019.

[204] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 746–760, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[205] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P. Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 2016.

[206] Martin Simon, Karl Amende, Andrea Kraus, Jens Honer, Timo Sämann, Hauke Kaulbersch, Stefan Milz, and Horst Michael Gross. Complexer-yolo: Real-time 3d object detection and tracking on semantic point clouds, 2019.

[207] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3693–3702, 2017.

[208] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[209] Vishwanath A. Sindagi, Yin Zhou, and Oncel Tuzel. Mvx-net: Multimodal voxelnet for 3d object detection. *2019 International Conference on Robotics and Automation (ICRA)*, May 2019.

[210] Paul Smith, Mubarak Shah, and Niels da Vitoria Lobo. Determining driver visual attention with one camera. *IEEE transactions on intelligent transportation systems*, 4(4):205–218, 2003.

[211] Shuran Song and Jianxiong Xiao. Sliding shapes for 3d object detection in depth images. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 634–651, Cham, 2014. Springer International Publishing.

[212] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 808–816, 2016.

[213] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice networks for point cloud processing. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018.

[214] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.

[215] Chen Sun, Zejian Deng, Wenbo Chu, Shen Li, and Dongpu Cao. Acclimatizing the operational design domain for autonomous driving systems. *IEEE Intelligent Transportation Systems Magazine*, 14(2):10–24, 2022.

[216] Chen Sun, Jean M. Uwabeza Vianney, Ying Li, Long Chen, Li Li, Fei-Yue Wang, Amir Khajepour, and Dongpu Cao. Proximity based automatic data annotation for autonomous driving. *IEEE/CAA Journal of Automatica Sinica*, 7(2):395–404, 2020.

[217] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Sheng Zhao, Shuyang Cheng, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset, 2020.

[218] Tao Sun, Mattia Segu, Janis Postels, Yuxuan Wang, Luc Van Gool, Bernt Schiele, Federico Tombari, and Fisher Yu. Shift: A synthetic driving dataset for continuous multi-task domain adaptation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21339–21350, 2022.

[219] Tao Sun, Mattia Segu, Janis Postels, Yuxuan Wang, Luc Van Gool, Bernt Schiele, Federico Tombari, and Fisher Yu. Shift: A synthetic driving dataset for continuous multi-task domain adaptation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2022.

[220] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning, 2016.

[221] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[222] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

[223] Araz Taeihagh and Hazel Si Min Lim. Governing autonomous vehicles: emerging responses for safety, liability, privacy, cybersecurity, and industry risks. *Transport Reviews*, 39(1):103–128, Jul 2018.

[224] Morteza Taiebat, Austin L. Brown, Hannah R. Safford, Shen Qu, and Ming Xu. A review on energy, environmental, and sustainability implications of connected and automated vehicles. *Environmental Science & Technology*, 52(20):11449–11465, 2018. PMID: 30192527.

[225] Jie Tang, Fei-Peng Tian, Wei Feng, Jian Li, and Ping Tan. Learning guided convolutional network for depth completion, 2019.

[226] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. *CoRR*, abs/1807.02443, 2018.

[227] Shangzhi Teng, Xiaobin Liu, Shiliang Zhang, and Qingming Huang. Scan: Spatial and channel attention network for vehicle re-identification. In Richang Hong, Wen-Huang Cheng, Toshihiko Yamasaki, Meng Wang, and Chong-Wah Ngo, editors, *Advances in Multimedia Information Processing – PCM 2018*, pages 350–361, Cham, 2018. Springer International Publishing.

[228] Hugues Thomas, François Goulette, Jean-Emmanuel Deschaud, Beatriz Marcotegui, and Yann LeGall. Semantic classification of 3d point clouds with multiscale spherical neighborhoods. In *2018 International conference on 3D vision (3DV)*, pages 390–398. IEEE, 2018.

[229] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, Francois Goulette, and Leonidas Guibas. Kpconv: Flexible and deformable convolution for point clouds. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019.

[230] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, Kenny Lau, Celia Oakley, Mark Palatucci, Vaughan Pratt, Pascal Stang, Sven Strohband, Cedric Dupont, Lars-Erik Jendrossek, Christian Koelen, Charles Markey,

Carlo Rummel, Joe van Niekerk, Eric Jensen, Philippe Alessandrini, Gary Bradski, Bob Davies, Scott Ettinger, Adrian Kaehler, Ara Nefian, and Pamela Mahoney. Stanley: The robot that won the darpa grand challenge. *Journal of Field Robotics*, 23(9):661–692, 2006.

[231] G. Toulminet, M. Bertozzi, S. Mousset, A. Bensrhair, and A. Broggi. Vehicle detection by means of stereo vision-based obstacles features extraction and monocular pattern analysis. *IEEE Transactions on Image Processing*, 15(8):2364–2375, 2006.

[232] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[233] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger. Sparsity invariant cnns. In *2017 International Conference on 3D Vision (3DV)*, pages 11–20, Oct 2017.

[234] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017.

[235] Geoffrey Underwood, Peter Chapman, Neil Brocklehurst, Jean Underwood, and David Crundall. Visual attention while driving: sequences of eye fixations made by experienced and novice drivers. *Ergonomics*, 46(6):629–646, 2003. PMID: 12745692.

[236] Geoffrey Underwood, Peter Chapman, Neil Brocklehurst, Jean Underwood, and David Crundall. Visual attention while driving: sequences of eye fixations made by experienced and novice drivers. *Ergonomics*, 46(6):629–646, 2003.

[237] unitree Inc. unitree robot dog product description, 2023.

[238] Abhinav Valada, Rohit Mohan, and Wolfram Burgard. Self-supervised model adaptation for multimodal semantic segmentation. *International Journal of Computer Vision*, Jul 2019.

[239] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. Sparse and noisy lidar completion with rgb guidance and uncertainty. *2019 16th International Conference on Machine Vision Applications (MVA)*, May 2019.

[240] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[241] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[242] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[243] Sourabh Vora, Alex H. Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection, 2019.

[244] J. Wang and L. Zhou. Traffic light recognition with high dynamic range imaging and deep learning. *IEEE Transactions on Intelligent Transportation Systems*, 20(4):1341–1352, 2019.

[245] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics (TOG)*, 36(4):1–11, 2017.

[246] Q. Wang, L. Chen, B. Tian, W. Tian, L. Li, and D. Cao. End-to-end autonomous driving: An angle branched network approach. *IEEE Transactions on Vehicular Technology*, 68(12):11599–11610, Dec 2019.

[247] Qian Wang and Toby Breckon. Unsupervised domain adaptation via structured prediction based selective pseudo-labeling. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 6243–6250, 2020.

[248] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2589–2597, 2018.

[249] T. Wang, F. Wang, J. Lin, Y. Tsai, W. Chiu, and M. Sun. Plug-and-play: Improve depth prediction via sparse data propagation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5880–5886, May 2019.

[250] Tsun-Hsuan Wang, Hou-Ning Hu, Chieh Hubert Lin, Yi-Hsuan Tsai, Wei-Chen Chiu, and Min Sun. 3d lidar and stereo fusion using stereo matching network with conditional cost volume normalization. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2019.

[251] Tsun-Hsuan Wang, Fu-En Wang, Juan-Ting Lin, Yi-Hsuan Tsai, Wei-Chen Chiu, and Min Sun. Plug-and-play: Improve depth estimation via sparse data propagation, 2018.

[252] Y. Wang, H. Liu, and Z. Fu. Low-light image enhancement via the absorption light scattering model. *IEEE Transactions on Image Processing*, 28(11):5679–5690, 2019.

[253] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q. Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.

[254] Yan Wang, Xiangyu Chen, Yurong You, Li Erran Li, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. Train in germany, test in

the usa: Making 3d object detectors generalize. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11713–11723, 2020.

[255] Z. Wang, W. Zhan, and M. Tomizuka. Fusing bird's eye view lidar point cloud and front view camera image for 3d object detection. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–6, June 2018.

[256] Ze Wang, Sihao Ding, Ying Li, Minming Zhao, Sohini Roychowdhury, Andreas Wallin, Guillermo Sapiro, and Qiang Qiu. Range adaptation for 3d object detection in lidar. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[257] Zhangjing Wang, Yu Wu, and Qingqing Niu. Multi-sensor fusion in automated driving: A survey. *IEEE Access*, 2019.

[258] Zhixin Wang and Kui Jia. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2019.

[259] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995.

[260] J. Weng, N. Ahuja, and T. S. Huang. Cresceptron: a self-organizing neural network which grows adaptively. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, volume 1, pages 576–581 vol.1, June 1992.

[261] Xinshuo Weng and Kris Kitani. Monocular 3d object detection with pseudo-lidar point cloud, 2019.

[262] Xinshuo Weng, Yunze Man, Dazhi Cheng, Jinhyung Park, Matthew O'Toole, and Kris Kitani. All-In-One Drive: A Large-Scale Comprehensive Perception Dataset with High-Density Long-Range Point Clouds. *arXiv*, 2020.

[263] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[264] Sascha Wirges, Bjorn Roxin, Eike Rehder, Tilman Kuhner, and Martin Lauer. Guided depth upsampling for precise mapping of urban environments. *2017 IEEE Intelligent Vehicles Symposium (IV)*, Jun 2017.

[265] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018.

[266] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4376–4382. IEEE, 2019.

[267] Kuo-Lung Wu and Miin-Shen Yang. Mean shift-based clustering. *Pattern Recogn.*, 40(11):3035–3052, November 2007.

[268] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.

[269] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[270] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015.

[271] Y. Xing, C. Lv, H. Wang, D. Cao, E. Velenis, and F. Wang. Driver activity recognition for intelligent vehicles: A deep learning approach. *IEEE Transactions on Vehicular Technology*, 68(6):5379–5390, June 2019.

[272] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018.

[273] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10368–10378, 2021.

[274] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Ipod: Intensive point-based object detector for point cloud, 2018.

[275] Zhenpei Yang, Yuning Chai, Dragomir Anguelov, Yin Zhou, Pei Sun, Dumitru Erhan, Sean Rafferty, and Henrik Kretzschmar. Surfelgan: Synthesizing realistic sensor data for autonomous driving. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020.

[276] Li Yi, Boqing Gong, and Thomas Funkhouser. Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15363–15373, 2021.

[277] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *CVPR*, 2021.

[278] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q. Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving, 2019.

[279] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020.

[280] Wei Zeng and Theo Gevers. 3dcontextnet: Kd tree guided hierarchical learning of point clouds using local and global contextual cues. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[281] Yiming Zeng, Yu Hu, Shice Liu, Jing Ye, Yinhe Han, Xiaowei Li, and Ninghui Sun. Rt3d: Real-time 3-d vehicle detection in lidar point cloud for autonomous driving. *IEEE Robotics and Automation Letters*, 3(4):3434–3440, 2018.

[282] D. Zermas, I. Izzat, and N. Papanikolopoulos. Fast segmentation of 3d point clouds: A paradigm on lidar data for autonomous vehicle applications. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5067–5073, May 2017.

[283] Junming Zhang, Manikandasriram Srinivasan Ramanagopalg, Ram Vasudevan, and Matthew Johnson-Roberson. Listereo: Generate dense depth maps from lidar and stereo imagery, 2019.

[284] Weichen Zhang, Wen Li, and Dong Xu. Srdan: Scale-aware and range-aware domain adaptation network for cross-dataset 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6769–6779, 2021.

[285] Wenwei Zhang, Hui Zhou, Shuyang Sun, Zhe Wang, Jianping Shi, and Chen Change Loy. Robust multi-modality multi-object tracking. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019.

[286] Xiaosong Zhang, Fang Wan, Chang Liu, Rongrong Ji, and Qixiang Ye. FreeAnchor: Learning to match anchors for visual object detection. In *Neural Information Processing Systems*, 2019.

[287] Y. Zhang, Y. Su, J. Yang, J. Ponce, and H. Kong. When dijkstra meets vanishing point: A stereo vision approach for road detection. *IEEE Transactions on Image Processing*, 27(5):2176–2188, 2018.

[288] Y. Zhang, H. Zhang, G. Wang, J. Yang, and J. Hwang. Bundle adjustment for monocular visual odometry based on detections of traffic signs. *IEEE Transactions on Vehicular Technology*, 69(1):151–162, Jan 2020.

[289] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *CVPR*, 2021.

[290] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, pages 7523–7532. PMLR, 2019.

[291] Hengyu Zhao, Yubo Zhang, Pingfan Meng, Hui Shi, Li Erran Li, Tiancheng Lou, and Jishen Zhao. Driving scenario perception-aware computing system design in autonomous vehicles. In *2020 IEEE 38th International Conference on Computer Design (ICCD)*, pages 88–95, 2020.

[292] Sicheng Zhao, Yezhen Wang, Bo Li, Bichen Wu, Yang Gao, Pengfei Xu, Trevor Darrell, and Kurt Keutzer. epointda: An end-to-end simulation-to-real domain adaptation framework for lidar point cloud segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3500–3509, 2021.

[293] Xin Zhao, Zhe Liu, Ruolan Hu, and Kaiqi Huang. 3d object detection using scale invariant and feature reweighting networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9267–9274, 2019.

[294] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018.

[295] J. Zhu, L. Fan, W. Tian, L. Chen, D. Cao, and F. Wang. Toward the ghosting phenomenon in a stereo-based map with a collaborative rgb-d repair. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–11, 2019.

[296] Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. 2017.

[297] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018.