LABEL EFFICIENT LEARNING OF 3D POINT CLOUD RECOGNITION

AORAN XIAO

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2023

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

01/Aug./2023

Date

XIAO Aoran

AORAN XIAO

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

06/08/2023

Date

Prof. Shijian Lu

Authorship Attribution Statement

This thesis includes material from six papers that have been published in or submitted to peer-reviewed journals/conferences, with myself listed as the first author.

Chapter 2 is related to 2 papers:

Aoran Xiao, Jiaxing Huang, Dayan Guan, Xiaoqin Zhang, Shijian Lu, Ling Shao. Unsupervised Representation Learning for Point Clouds with Deep Neural Networks: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2023. The contributions of the co-authors are as follows:

- Prof. Shijian Lu provided overall supervision throughout the research.
- I proposed the idea, reviewed relevant studies, and prepared the manuscript.
- Jiaxing Huang assisted in problem formulation and paper drafting.
- The manuscript was revised by Shijian Lu.
- Jiaxing Huang, Dayan Guan, Xiaoqin Zhang, Ling Shao provided comments for manuscript revision and possible future direction.

Aoran Xiao, Xiaoqin Zhang, Ling Shao, Shijian Lu. A Survey of Label-Efficient Deep Learning for 3D Point Clouds. Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (Under-review). The contributions of the co-authors are as follows:

- Prof. Shijian Lu provided overall supervision throughout the research.
- I proposed the idea, reviewed relevant studies, and prepared the manuscript.
- The manuscript was revised by Shijian Lu.
- Xiaoqin Zhang, Ling Shao provided comments for manuscript revision.

Chapter 3 is related to 1 publication:

Aoran Xiao, Jiaxing Huang, Dayan Guan, Kaiwen Cui, Shijian Lu, Ling Shao. PolarMix: A General Data Augmentation Technique for LiDAR Point Clouds. Advances in Neural Information Processing Systems (NeurIPS), 2022. The contributions of the co-authors are as follows:

- Prof. Shijian Lu provided overall supervision throughout the research.
- I proposed the idea, designed the experiments, and prepared the manuscript.
- Jiaxing Huang assisted in idea formulation.
- The manuscript was revised by Shijian Lu.

- Shijian Lu, Jiaxing Huang, Dayan Guan, and Ling Shao provided comments for problem formulation, experiment design, and manuscript revision. Dayan Guan and Jiaxing Huang also helped revise the storyline of this paper.

Chapter 4 is related to 2 papers:

Aoran Xiao, Jiaxing Huang, Dayan Guan, Fangneng Zhan, Shijian Lu. Transfer Learning from Synthetic to Real LiDAR Point Cloud for Semantic Segmentation. AAAI Conference on Artificial Intelligence (AAAI), 2022. The contributions of the co-authors are as follows:

- Prof. Shijian Lu provided overall supervision throughout the research.
- I proposed the idea, collected data, designed the experiments, conducted the experiments, and prepared the manuscript.
- Jiaxing Huang assisted in idea formulation.
- The manuscript was revised by Shijian Lu.
- Jiaxing Huang Dayan Guan, and Shijian Lu provided comments for problem formulation, data collection, experiment design, and manuscript revision. Fangneng Zhan provided comments for method design and the final manuscript revision.

Aoran Xiao, Dayan Guan, Xiaoqin Zhang, Shijian Lu. Domain Adaptive LiDAR Point Cloud Segmentation with 3D Spatial Consistency. IEEE Transactions on Multimedia, 2023. The contributions of the co-authors are as follows:

- Prof. Shijian Lu provided overall supervision throughout the research.
- I proposed the idea, designed the experiments, and prepared the manuscript.
- The manuscript was revised by Shijian Lu.
- Shijian Lu, Dayan Guan, and Xiaoqin Zhang provided comments for problem formulation, experiment design, and manuscript revision.

Chapter 5 is related to 1 publication:

Aoran Xiao, Jiaxing Huang, Weihao Xuan, Ruijie Ren, Kangcheng Liu, Dayan Guan, Abdulmotaleb El Saddik, Shijian Lu, Eric Xing. 3D Semantic Segmentation in the Wild: Learning Generalized Models for Adverse-Condition Point Clouds. Computer Vision and Pattern Recognition (CVPR), 2023. The contributions of the co-authors are as follows:

- Prof. Shijian Lu provided overall supervision throughout the research.
- I proposed the idea, led the dataset labeling project, designed the experiments, and prepared the manuscript.
- Jiaxing Huang assisted with the idea.
- The manuscript was revised by Shijian Lu.
- Weihao Xuan and Ruijie Ren provided the data annotation and inspection.

- Shijian Lu, Jiaxing Huang, and Dayan Guan provided comments for problem formulation, experiment design, and manuscript revision. Abdulmotaleb EI Saddik and Eric Xing provided comments for the final manuscript revision.

01/Aug./2023

Date

AORAN XIAO

Acknowledgements

On the last day of 2019, as I prepared to embark on my Ph.D. journey in Singapore, I came across a news piece about an unknown pneumonia outbreak in Wuhan, where I lived in the past seven years. Little did I know that this unforeseen event would not only keep me away from my family for several years but also introduce numerous challenges during my Ph.D. pursuit. Now, as I sit here, reflecting on this challenging yet immensely rewarding journey, I am filled with gratitude and appreciation for all the support and guidance I have received.

First and foremost, I extend my deepest gratitude to my Ph.D. supervisor, Professor Shijian Lu. His unwavering patience, enthusiastic encouragement, and invaluable guidance have transformed me from an inexperienced student into a passionate researcher with a deep understanding of point clouds, computer vision, and deep learning. His mentorship and dedicated teaching, from reading papers to conducting experiments and presenting academically, have been instrumental in shaping my research skills. I am truly grateful for the exemplary model he has set for me in becoming a proficient researcher.

My heartfelt thanks also go to my teammates, Dr. Jiaxing Huang and Dr. Dayan Guan, for their invaluable advice and support throughout my Ph.D. journey. I extend my appreciation to all the members of the SCALE and MICL Labs, as well as Prof. Lu's research group, for their enthusiastic discussions and unwavering encouragement over the past four years. Special memories were created with Dr. Pengdeng Li, Kaiwen Cui, and Yun Xing, enriching my academic and personal growth.

I am grateful to my friends who have been a constant source of support. Despite being apart for the entire four years, I want to thank Dr. Changjie Wu from Tsinghua University for maintaining frequent communication and sharing thoughts on both academic and daily life matters. My sincere appreciation also goes to Dr.

Yexian Ren from Fudan University and Kai Chen from CUHK for their patient guidance during moments of confusion and struggle.

To my thesis committee members, Prof. Lin Guosheng, Prof. Liu Ziwei, and Prof. Cham Tat Jen, I extend my thanks for dedicating time and providing valuable feedback during my qualification examination and thesis preparation. Your patience and kind support have significantly contributed to the refinement of my work.

Lastly, and most importantly, I want to express my deepest gratitude to my family. My parents have been instrumental in nurturing an equal parent-child relationship, fostering independence, and working tirelessly to support my dreams. Their sacrifices for my education have been immeasurable. To my beloved wife, Jixiang, I am eternally grateful for never giving up on our relationship, even during the two years of separation caused by the COVID pandemic, and for accepting my proposal with unwavering love and support. Your presence has been my pillar of strength, guiding me to chase my dreams and maintain faith even in the face of challenges.

As I reach the culmination of my Ph.D. journey, I am humbled by the experiences and memories that have shaped me into the researcher and person I am today. It is with heartfelt appreciation that I acknowledge the unwavering support of all those who have played a role in this transformative chapter of my life. I have gained more than I expected, and I am filled with passion and confidence for what I am about to experience.

Aoran, July 2023

Contents

List of Figures

List of Tables

Abstract

The ability to recognize the three-dimensional (3D) world profoundly impacts our comprehension, visualization, interaction, and re-creation of the physical environment. Point cloud data, renowned for its accurate representation of 3D geometric structures, has gained significant attention in both academia and industry. Meanwhile, deep neural networks (DNNs) have revolutionized various domains, including computer vision and natural language processing. Integrating point clouds with DNNs has given rise to powerful deep point cloud models, enabling enhanced recognition and understanding of the 3D world.

However, current DNN models for point cloud recognition heavily rely on large amounts of densely-labelled training data, which is extremely laborious and costly to obtain. This limitation hampers the scalability of existing point cloud datasets and hinders efficient exploration across tasks and applications.

This thesis explores Label-Efficient Learning for Point Cloud Recognition, aiming to minimize annotation efforts during deep network training while achieving effective results in point cloud recognition. The study focuses on three key label-efficient learning categories: *data augmentation*, *domain transfer learning from synthetic to real data*, and *domain transfer learning from normal to adverse weather conditions*. Through these representative approaches, we aim to enhance the efficiency and effectiveness of point cloud recognition methodologies.

Within the label-efficient learning paradigm, *data augmentation* plays a crucial role in expanding the diversity of limited labelled training data, requiring fewer annotated point clouds to train accurate recognition models. In this thesis, we introduced a novel LiDAR point cloud augmentation technique that generates new frames within the polar coordinate system, facilitating model training in various 3D perception tasks and scenarios.

Domain transfer learning from synthetic to real data leverages knowledge from synthetic point clouds with automatically generated labels to enhance the performance

of deep models in recognizing real-world point clouds. By using infinite synthetic labelled point clouds, human annotations in real point clouds can be reduced or eliminated, alleviating significant annotation efforts. In this thesis, we first created a large-scale synthetic LiDAR point cloud dataset with precise point-wise annotations. Building upon this dataset, we presented two novel methodologies, involving style translation and unsupervised domain adaptation, to address domain discrepancies between synthetic and real LiDAR point clouds and facilitate synthetic-to-real domain transfer learning.

Domain transfer learning from normal to adverse weather data aims to train robust recognition models using point clouds captured under normal weather conditions to perform well across diverse adverse weather conditions. This objective arises from considerable additional challenges in annotating point clouds of adverse weather since they share different geometric data characteristics compared to normal weather data. We explore transferring knowledge from normal to adverse weather point clouds to reduce the need for extensive manual annotations for adverse weather point clouds. To achieve this, we first constructed a large-scale adverse-weather point cloud dataset with point-wise annotations. Subsequently, we proposed a domain generalization and aggregation method, which enables the training of robust models exclusively using normal data, empowering them to effectively handle various adverse weather conditions.

Extensive experimentation conducted across diverse point cloud recognition benchmarks demonstrates the superior performance achieved by our proposed label-efficient learning approaches.

Chapter 1

Introduction

1.1 Scope and Overview

Point clouds are collections of three-dimensional (3D) points that accurately depict the shape and geometry of objects or environments. This quality makes them highly applicable to various 3D recognition tasks, including 3D shape analysis, 3D object detection, and 3D semantic segmentation. In recent years, there has been a rapid development in 3D acquisition technologies for capturing point cloud data, which is evident in the increasing popularity of various 3D sensors in both industrial and daily-life settings. Examples include LiDAR sensors in autonomous vehicles, RGB-D cameras in devices like Kinect and Apple products, and 3D scanners used in various reconstruction tasks. Meanwhile, the remarkable advancements in deep learning have significantly contributed to the field of point cloud recognition, resulting in the emergence of numerous deep point cloud structures and networks. The concurrence of the two has witnessed increasing demands in utilizing point clouds to capture 3D shape representations of objects and scenes, ranging from autonomous navigation and robotics to remote sensing applications and beyond.

This section begins with an overview of the advancements in point cloud recognition, highlighting the need for conducting label-efficient learning for related tasks. Subsequently, we review recent progress made in label-efficient point cloud learning by focusing on three key types of label-efficient learning approaches that significantly reduce the need for extensive human annotation efforts. These approaches include data augmentation, domain transfer learning from synthetic-to-real point

clouds, and domain transfer learning from normal-to-adverse weather point clouds. Each of these approaches is explored based on their specific data prerequisites and their ability to mitigate the burden of human annotation efforts in training robust point cloud recognition networks.

Due to the unstructured and disordered characteristics of point clouds, deep learning for 3D point cloud recognition presents unique challenges compared to image recognition in 2D vision and standard convolutional neural networks cannot be directly applied to process point clouds. The advent of PointNet [8], which utilizes multi-layer perceptrons (MLPs), has revolutionized point cloud recognition across a wide range of tasks, such as object classification, part segmentation, and scene semantic parsing. Since then, various deep neural architectures such as graph neural networks [9] and sparse convolution networks [10], have made significant advancements in diverse point cloud recognition tasks, including 3D shape classification, 3D object detection, 3D semantic segmentation, etc.

Despite the significant advancements in deep learning for point cloud recognition, most existing research heavily relies on large-scale, precisely annotated 3D data for network training. Although the collection of extensive training point clouds has become more acceptable, the annotation process remains notoriously laborious and time-consuming due to the high complexity of the data, significant variation in point sparsity, the presence of rich noises, occlusion, and frequent 3D view changes during annotation. The labor-intensive nature of point cloud annotation makes the construction of large-scale point cloud datasets extremely expensive and time-consuming. This directly leads to limited sizes and diversity in existing public point cloud datasets, posing a great challenge when developing generalizable point cloud learning algorithms across various applications.

To tackle the burden associated with point cloud annotation, a promising solution is label-efficient learning – a machine learning paradigm that prioritizes model training with minimal annotation while still achieving the desired accuracy. Label-efficient point cloud learning has recently emerged as a thriving research field due to its importance and high practical values. Various label-efficient learning approaches have been investigated, each with its own data requirements and application scenarios. In this thesis, we examine three representative forms of label-efficient learning, namely data augmentation, domain transfer learning from synthetic-to-real point clouds, and domain transfer learning from normal-to-adverse weather point clouds:

1) Data augmentation involves generating new training data from existing samples to enhance the training distribution and facilitate network training. This technique proves particularly advantageous when the available labelled training data is limited. 2) Domain transfer learning from synthetic-to-real point clouds involves utilizing synthetic point clouds to train recognition models. By exploiting automatically generated labels, this approach leverages the abundance of synthetic data as an alternative to annotating real point clouds. 3) Domain transfer learning from normal-to-adverse weather point clouds aims to develop robust point cloud recognition models by training on labelled data collected under normal weather conditions. This is crucial due to the substantial geometric distortions and ambiguity present in point clouds captured during adverse weather, which pose significant challenges for annotation. The extensive exploration of these three label-efficient learning techniques stems from their promising potential in various 3D vision tasks, encompassing shape classification, instance detection, semantic segmentation, and others.

1.2 Major Contributions

The primary focus of my Ph.D. research thesis investigates reliable perception techniques for 3D LiDAR point clouds obtained from mobile vehicles. This data plays a pivotal role in various applications, contributing to instantaneous recognition and understanding of surrounding environments. Specifically, in autonomous navigation, these point cloud perceptions serve as a fundamental module for fine-grained scene understanding, crucial for decision-making and navigation in robotics and vehicular applications. The development of an accurate and robust 3D perception system is imperative in these applications, yet its efficacy heavily relies on extensive annotated data for training. Acquiring such data proves to be labor-intensive and expensive, especially given the diverse and challenging nature of real-world scenarios.

We thus explore label-efficient learning from various perspectives, specifically through the utilization of data augmentation and domain transfer learning (specifically from synthetic-to-real point clouds and from normal-to-adverse weather point clouds). These endeavors aim to train more robust and generalized 3D perception models with reduced annotation efforts, addressing the crucial need for efficient learning

methodologies in this field. The contributions are depicted in Fig. 1.1. In the following sections, I will delve into more detailed explanations and provide additional information regarding these approaches.

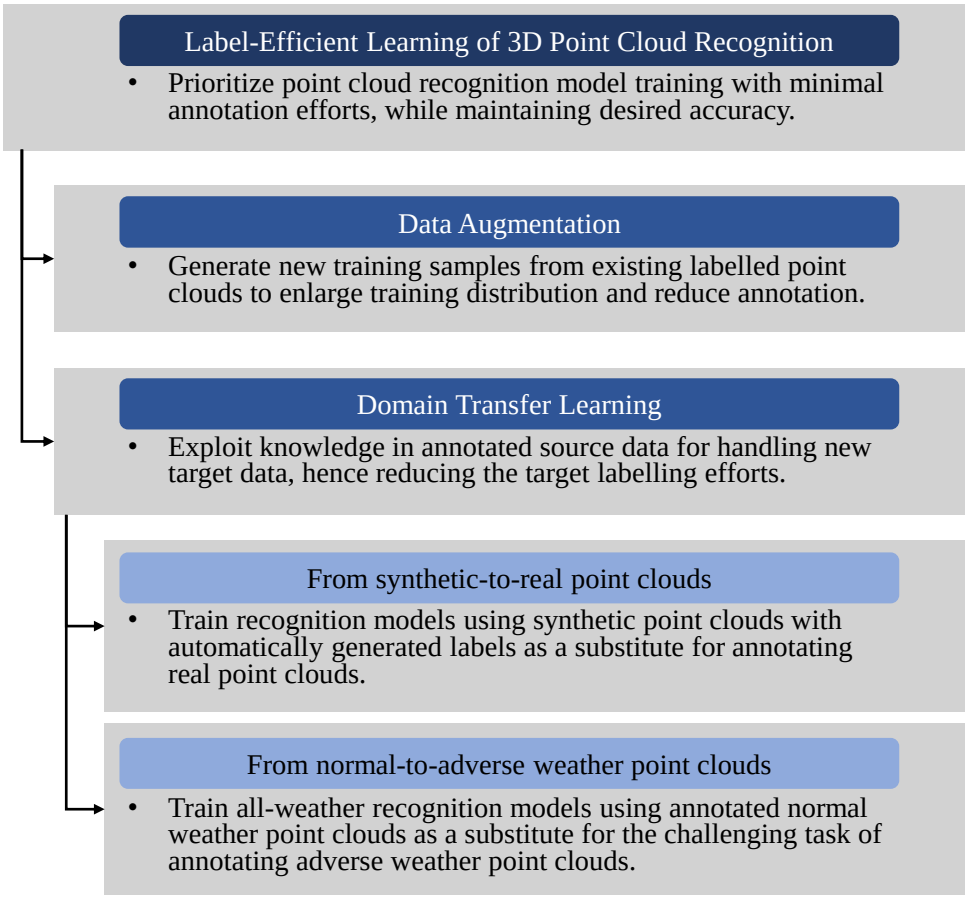


FIGURE 1.1: The overview of different types of label-efficient learning of 3D point cloud recognition.

1.2.1 Data Augmentation

The objective of data augmentation is to explore effective ways to generate more diverse training samples from existing ones. In other words, with data augmentation techniques, we can train robust recognition models with minimum labelled training point clouds, hence significantly reducing annotation efforts and learning point cloud recognition models in a label-efficient way.

We propose a novel and general data augmentation method for LiDAR point cloud recognition. Specifically, observed by the essential properties of LiDAR point

clouds, namely, partial visibility and density variation that are closely associated with the sweeping mechanism of LiDAR sensors, we present *PolarMix*, a point cloud augmentation technique that is simple and generic but can mitigate the data constraint effectively across different perception tasks and scenarios. PolarMix enriches point cloud distributions and preserves point cloud fidelity via two cross-scan augmentation strategies that cut, edit, and mix point clouds along the scanning direction. The first is scene-level swapping which exchanges point cloud sectors of two LiDAR scans cut along the azimuth axis. The second is instance-level rotation and paste which crops point instances from one LiDAR scan, rotates them by multiple angles (to create multiple copies), and paste the rotated point instances into other scans. Extensive experiments show that PolarMix achieves superior performance consistently across different perception tasks and scenarios. In addition, it can work as a plug-and-play for various 3D deep architectures and also performs well for unsupervised domain adaptation.

1.2.2 Domain Transfer Learning from Synthetic to Real Point Clouds

Knowledge transfer from synthetic to real data has been widely studied to mitigate data annotation constraints in various computer vision tasks such as semantic segmentation. A primary advantage lies in the ability to obtain an unlimited amount of synthetic training data with automatically generated labels, enabling the learning of rich semantic structures in recognition models. However, the study mainly focused on 2D images and its counterpart in 3D point cloud lags far behind due to the lack of large-scale synthetic datasets and effective transfer learning methods.

We address this issue by collecting SynLiDAR, a large-scale synthetic LiDAR dataset that contains point-wise annotated point clouds with accurate geometric shapes and comprehensive semantic classes. SynLiDAR was collected from multiple virtual but realistic environments with rich scenes and layouts which consist of over 19 billion points of 32 semantic classes.

In addition, we design PCT, a novel point cloud translator that translates synthetic point clouds to be similar to real data, aiming to mitigate the domain gap on the input space and facilitate domain transfer learning. Specifically, we decompose

the synthetic-to-real gap into an appearance component and a sparsity component and handle them separately which improves the point cloud translation effectively. We conducted extensive experiments over three transfer learning setups including data augmentation, semi-supervised domain adaptation, and unsupervised domain adaptation (UDA). Extensive experiments show that SynLiDAR provides a high-quality data source for studying 3D transfer and the proposed PCT achieves superior point cloud translation consistently across the three setups.

Furthermore, we study UDA from synthetic-to-real point clouds that work on the output space. Specifically, we designed a simple yet effective spatial consistency training (SCT) framework that can learn superior domain-adaptive feature representations from unlabelled target point clouds. The idea stems from the observation that the data distribution of LiDAR point clouds varies significantly and frequently due to diverse sensor configurations, environmental conditions, occlusions, and other factors, while the semantic meaning of the corresponding points should remain unchanged. SCT leverages three types of spatial consistency, namely, geometric-transform consistency, sparsity consistency, and mixing consistency. These strategies capture the semantic invariance of point clouds concerning changes in viewpoint, sparsity, and local context, respectively. With a concise mean teacher learning strategy, our experiments show that the proposed SCT outperforms the state-of-the-art significantly and consistently across multiple public benchmarks for synthetic-to-real adaptive LiDAR segmentation.

1.2.3 Domain Transfer Learning from Normal to Adverse Weather Point Clouds

Robust point cloud recognition under all-weather conditions is crucial to many applications such as autonomous driving. However, how to learn a universal recognition model is largely neglected as most existing benchmarks are dominated by point clouds captured under normal weather. The primary reason for this neglect is the considerable new challenges involved in annotating point clouds captured under adverse weather conditions. For example, adverse weather introduces geometry distortions and the presence of invalid regions (such as thick snow covers), making object recognition and semantic labeling extremely difficult. In this thesis, we study domain transfer learning from normal to adverse weather point clouds

to avoid such extensive manual annotation. Specifically, we first introduce SemanticSTF, an adverse-weather point cloud dataset that provides dense point-level annotations. We then study all-weather 3D recognition under two setups including domain adaptive learning that adapts from normal-weather data to adverse-weather data, and domain generalizable learning that learns all-weather recognition models from normal-weather data. Our studies reveal the challenge while existing recognition methods encounter adverse-weather data, showing the great value of SemanticSTF in steering the future endeavor along this very meaningful research direction. In addition, we design a domain randomization technique that alternatively randomizes the geometry styles of point clouds and aggregates their embeddings, ultimately leading to a generalizable model from normal weather point clouds that can improve 3D recognition under various adverse weather effectively.

1.3 Outline of the Thesis

Chapter 1 provides an outline of the scope of this thesis and offers an overview of point cloud recognition, label efficient learning, and three prominent types of label efficient learning for point clouds. Additionally, it summarizes and discusses the significant contributions made by this thesis.

Chapter 2 reviews the background of point cloud recognition and label efficient learning. It begins by exploring the recent advancements in deep learning-based point cloud recognition and emphasizes their significance, thereby establishing the need for label efficient learning of point clouds. The chapter then proceeds to introduce the general concept and taxonomy of label efficient learning, covering its three typical types.

Chapter 3 introduces the proposed data augmentation technique for LiDAR point cloud recognition. The chapter focuses on exploring a mixing strategy within the polar coordinate system and showcases the superior augmentation performance of the proposed method across various tasks, datasets, and different backbone structures.

Chapter 4 focuses on the studies for domain transfer learning from synthetic to real point clouds. It begins by constructing a large-scale dataset comprising synthetic LiDAR point clouds with dense point-wise annotations. The chapter then

explores two types of domain transfer learning objectives: point cloud translation and consistency learning. Extensive experiments are conducted to demonstrate the superior transfer learning performance of the proposed methods across various datasets.

Chapter 5 focuses on the proposed studies for domain transfer learning from adverse to normal weather conditions. The chapter begins by introducing a large-scale LiDAR dataset captured under adverse weather conditions, which includes dense point-wise annotations. Two types of domain transfer learning are then explored: unsupervised domain adaptation and domain generalization. Additionally, a domain randomization and aggregation method is designed to facilitate the training of generalizable models that effectively improve LiDAR segmentation under various adverse weather conditions.

Chapter 6 serves as the conclusion of this thesis and outlines potential directions for further research.

Chapter 2

Literature Review[1] [2]

This chapter presents a comprehensive review of label efficient learning for point cloud recognition. We begin by providing an introduction to the background of point cloud recognition in Section 2.1. This includes an overview of common 3D point cloud recognition tasks (Section 2.1.1), commonly used deep architectures (Section 2.1.2), and annotation challenges for point clouds (Section 2.1.3). Next, we delve into the concept of label efficient learning and its representative types in Section 2.2. These types include data augmentation (Section 2.2.1) and domain transfer learning (Section 2.2.2).

2.1 Point Cloud Recognition

A point cloud, denoted as P, is a collection of vectors given by $P = \{p_1, ..., p_N\}$, where each vector represents an individual point $p_i = [C_i, A_i]$. In this representation, $C_i \in \mathbf{R}^{1\times 3}$ denotes the 3D coordinates (x_i, y_i, z_i) of the point, and A_i refers to optional and variable feature attributes associated with the point, such as RGB values, LiDAR intensity value, normal values, and so on. These attributes

[1]Part of the work in this chapter has been published at "Aoran Xiao, Jiaxing Huang, Dayan Guan, Xiaoqin Zhang, Shijian Lu and Ling Shao, "Unsupervised Point Cloud Representation Learning with Deep Neural Networks: A Survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10.1109/TPAMI.2023.3262786.

[2]Part of the work has been submitted to "Aoran Xiao, Xiaoqin Zhang, Ling Shao, Shijian Lu. A Survey of Label-Efficient Deep Learning for 3D Point Clouds. IEEE Transactions on Pattern Analysis and Machine Intelligence (Under-review)."

are dependent on the 3D sensors and specific applications, thus exhibiting diverse characteristics.

These points accurately capture the spatial positions of object surfaces, providing a precise representation of the underlying 3D geometry. Unlike 2D images, 3D point clouds lack a predefined order or structure, making them a unique and challenging data format. Consequently, they cannot be directly processed or learned using standard convolutional networks, necessitating specialized techniques for their analysis and understanding.

2.1.1 Common 3D Recognition Tasks

Object classification aims to classify point cloud objects (e.g. *car*, *plane*) into a number of pre-defined categories. Two evaluation metrics are most frequently used: The *overall Accuracy* (OA) represents the averaged accuracy for all instances in the test set; The *mean class accuracy* (mAcc) represents the mean accuracy of all object classes for the test set.

Object part segmentation aims to assign a part category label (e.g., airplane wing, table leg, etc.) to each point. The mean Intersection over Union (mIoU) [8] is the most widely adopted evaluation metric. For each instance, IoU is computed for each part belonging to that object category. The mean of the part IoUs represents the IoU of that object instance. The overall IoU is computed as the average of IoUs over all test instances while category-wise IoU (or class IoU) is calculated as the mean over instances under that category.

3D semantic segmentation on point clouds is another critical task for 3D understanding. Different from the object part segmentation that segments point cloud objects, 3D semantic segmentation aims to assign a category label to each point in scene-level point clouds with much higher complexity. The widely adopted evaluation metrics include OA, mIoU over semantic categories, and mAcc.

3D object detection on point clouds is a crucial and indispensable task for many real-world applications, such as autonomous driving and domestic robots. The task aims to localize and recognize objects in the 3D space, *i.e.* 3D object bounding boxes. The average precision (AP) metric has been widely used for evaluations in 3D object detection [11, 12].

3D instance segmentation aims to detect and delineate each distinct object of interest in scene-level point clouds. On top of semantic segmentation that considers the semantic category only, instance segmentation assigns each object a unique identity. Mean Average Precision (mAP) has been widely adopted for the quantitative evaluation of this task.

2.1.2 Common Deep Architectures

Over the last decade, deep learning has been playing an overwhelming role in point-cloud recognition. This can be observed by the abundance of deep architectures that have been developed in recent years. Different from traditional 3D vision that transforms point clouds to structures like Octrees [13] or Hashed Voxel Lists [14], deep learning favors more amenable structures for differentiability and/or efficient neural processing which have achieved very impressive performance over various 3D tasks. At the other end, DNN-based point cloud processing and understanding lags far behind as compared with its counterparts in NLP and 2D computer vision, largely due to the lack of regular representations in point cloud data. Specifically, word embeddings and 2D images have regular and well-defined structures, but point clouds represented by unordered point sets have no such universal and structural data format.

In this subsection, we introduce deep architectures that have been widely explored for the recognition of point clouds. For clarity of description, we group them into four categories broadly, namely, point-based architectures, graph-based architectures, sparse voxel-based architectures, and spatial CNN-based architectures. It is noted that there are other deep architectures also exist for various 3D tasks, such as projection-based networks [15–20], recurrent neural networks [21–23], 3D capsule networks [24], and Transformer-based architectures [25], etc. A more detailed review of deep architectures for point cloud deep learning could be found in [26].

Point-based networks were designed to process raw point clouds directly without point data transformations beforehand. Independent point features are usually first extracted by stacking networks with Multi-Layer Perceptrons (MLPs), which are then aggregated into global features with symmetric aggregation functions. PointNet [8] is a pioneer point-based network, which stacks several MLP layers to learn point-wise features independently and forwards the learned features to

a max-pooling layer to extract global features for permutation invariance. To improve PointNet, Qi *et al.* proposed PointNet++ [27] to learn local geometry details from the neighborhood of points, where the set abstraction level includes sampling layer, grouping layer, and PointNet layer for learning local and hierarchical features. PointNet++ achieves great success in multiple 3D tasks including object classification and semantic segmentation. By taking PointNet++ as the backbone, Qi *et al.* designed VoteNet [11], the first point-based 3D object detection network. VoteNet adopts the Hough voting strategy, which generates new points around object centers and groups them with the surrounding points to produce 3D box proposals.

Graph-based networks treat point clouds as graphs in Euclidean space with vertexes being points and edges capturing neighboring point relations. It works with graph convolution where filter weights are conditioned on edge labels and dynamically generated for individual input samples. This allows for the reduction of the degrees of freedom in the learned models by enforcing weight sharing and extracting localized features that can capture dependencies among neighboring points. The Dynamic Graph Convolutional Neural Network (DGCNN) [9] is a typical graph-based network. It is stacked with a graph convolution module named EdgeConv that performs convolution on graph dynamically in the feature space. DGCNN integrates EdgeConv into the basic version of PointNet structures for learning global shape properties and semantic characteristics for point cloud understanding.

Sparse voxel-based architecture voxelizes point clouds into 3D grids before applying 3D CNN on the volumetric representations. Due to the sparseness of point cloud data, It often involves huge computation redundancy or sacrifices the representation accuracy while processing a large number of points. To overcome this constrain, [6, 10, 28, 29] adopt *sparse tensor* as the basic unit where point clouds are represented with a data list and an index list. Unlike standard convolution operation that employs sliding windows (*im2col* function in PyTorch and TensorFlow) to build the computational pipeline, *sparse convolution* [28] collects all atomic operations including convolution kernel elements and saves them in a *Rulebook* as computation instructions. Recently, Choy et al. proposed Minkowski Engine [10] that introduces generalized sparse convolution and an auto-differentiation library

for sparse tensors. They proposed MinkowskiNet which achieves competitive recognition performance with a good trade-off between accuracy and efficiency.

Spatial CNN-based networks have been developed to extend the capabilities of regular-grid CNNs to analyze irregularly spaced point clouds. They can be divided into continuous and discrete convolutional networks according to the convolutional kernels [26]. Continuous convolutional networks define the convolutional kernels in a continuous space, where the weights of neighboring points are determined by their spatial distribution relative to the center point. Differently, discrete convolutional networks operate on regular grids and define the convolutional kernels in a discrete space where neighboring points have fixed offsets relative to the center point. One typical example of continuous convolution models is RS-CNN [30] which extracts geometric topology relations among local centers with their surrounding points, and it learns dynamic weights for convolutions.

2.1.3 Annotation Efforts for Point clouds

Annotating point clouds is challenging which usually requires special training due to the unique characteristics of point-cloud data. It faces several new challenges compared with annotating data of other modalities such as images. First, the display of point clouds is often unaligned with human perceptions. Point clouds are often incomplete, sparse, and may not contain color information, leading to rich ambiguity in point semantics and point geometries. Second, 3D view changes complicate the annotation process greatly which even cause motion sickness for annotators. Hence, point cloud annotators require good expertise and experience to ensure annotation accuracy and consistency while labelling, e.g., 3D bounding boxes and point-wise categories for 3D detection and segmentation tasks.

Third, fully automatic point cloud annotation is still infeasible at the current stage. Although some tool such as semi-automatic labelling has been explored to streamline the process, the annotation accuracy remains low and plenty of manual efforts are required to inspect and correct the automatic annotations. Though different approaches have been proposed to simplify the manual annotation process, most of they do not generalize well with various extra requirements. For instance, Behley et al. [7] superimpose multiple LiDAR scans to formulate dense point representations,

TABLE 2.1: A summary of commonly used public datasets for point cloud recognition.

Dataset	Year	#Samples	#Classes	Type	Representation	Label
ModelNet40[31]	2015	12,311 objects	40	Synthetic object	Mesh	Object category
ShapeNet[32]	2015	51,190 objects	55	Synthetic object	Mesh	Object/part category
ScanObjectNN[33]	2019	2,902 objects	15	Real-world object	Points	Object category
SUN RGB-D[34]	2015	5K frames	37	Indoor scene	RGB-D	Bounding box
S3DIS[35]	2016	272 scans	13	Indoor scene	RGB-D	Point category
ScanNet[36]	2017	1,513 scans	20	Indoor scene	RGB-D & mesh	Point category & Bounding box
KITTI[37]	2013	15K frames	8	Outdoor driving	RGB & LiDAR	Bounding box
nuScenes[38]	2020	40K	32	Outdoor driving	RGB & LiDAR	Point category & Bounding box
Waymo[39]	2020	200K	23	Outdoor driving	RGB & LiDAR	Point category & Bounding box
STF[40]	2020	13.5K	4	Outdoor driving	RGB & LiDAR & Radar	Bounding box
ONCE[41]	2021	1M scenes	5	Outdoor driving	RGB & LiDAR	Bounding box
Semantic3D[42]	2017	15 dense scenes	8	Outdoor TLS	Points	Point category
SemanticKITTI[7]	2019	43,552 scans	28	Outdoor driving	LiDAR	Point category
SensatUrban[43]	2020	1.2 km^2	31	UAV Photogrammetry	Points	Point category
SynLiDAR[2]	2022	198,396 scans	32	Outdoor driving	Synthetic LiDAR	Point category
SemanticSTF[44]	2023	2,086 scans	21	Outdoor driving	RGB & LiDAR	Point category

allowing labelling multiple scans concurrently and consistently while collecting SemanticKITTI. However, the superimposing process requires accurate and instant localization and pose of LiDAR sensors, and the superimposed moving objects are often distorted and indistinguishable. In summary, manual approach remains the primary way of point cloud annotation which requires vast time and efforts as well as well-trained annotators.

The labour-intensive nature of point cloud annotation makes the construction of large-scale point-cloud datasets extremely time-consuming. This directly leads to limited sizes and diversity in public point-cloud datasets as shown in Table 2.1, and poses a great challenge while developing generalizable point cloud learning algorithms. Studying label-efficient point cloud learning has become an urgent need to mitigate the limitation of existing point-cloud data.

2.2 Label-efficient Learning of Point Clouds

Different from the classic supervised training process used in most 3D recognition models (as discussed in Section 2.1), label-efficient learning optimizes the learning process with minimal annotation efforts while maintaining the desired accuracy. Unlike traditional supervised learning that relies on large-scale fully and precisely annotated samples, label-efficient learning aims to enable computers or systems to learn like humans do. This means intelligently utilizing less supervision, i.e., labels, to effectively tackle new tasks and data.

A typical example of label-efficient learning is that human beings can recognize new objects with only a few labelled samples. For instance, a baby can easily identify

unseen dogs with just a few provided images of dogs. Another typical example involves humans easily transferring knowledge from one scenario to another, such as people living in Asia can recognize daily objects with different appearances in Europe. In the context of label-efficient learning, the former setup is referred to as *data augmentation*, which aims to train accurate recognition models with limited labelled training data. On the other hand, the latter setup is formulated as *domain transfer learning*, aiming to leverage knowledge from labelled source domains to train networks that perform well on unlabelled target domains without requiring any additional (target) annotations. We provided a literature review for these two representative types of label-efficient learning in the ensuing subsections. A more comprehensive review could be found in [44].

2.2.1 Data Augmentation

Data augmentation (DA) is a widely adopted strategy in the training of deep learning networks [45]. The primary aim of DA is to increase the size and diversity of a dataset by artificially generating new training data from the existing one, without the need for additional data collection and annotation. This approach is particularly beneficial in scenarios where the available annotated training data is limited, and the model needs to learn from diverse and representative examples to perform well on unseen data. Therefore, DA is considered an important label-efficient learning approach with widespread applications across various fields.

We review existing DA studies in point cloud network training, which can be broadly grouped into two categories: *intra-domain augmentation* and *inter-domain augmentation*. The former aims to enrich training data by generating new training data from the existing as detailed in Section 2.2.1.1. The latter leverages additional data to enlarge the existing training data distribution as detailed in Section 2.2.1.2.

2.2.1.1 Intra-domain Augmentation

Intra-domain DA aims to maximize the training knowledge by only utilizing the limited annotated training data available. We first provide an introduction to

conventional DA that is generic and applicable in various point cloud tasks. Subsequently, we review DA methods that are designed for specific 3D tasks, including 3D shape classification, 3D object detection, and 3D semantic segmentation.

Conventional augmentation techniques. Conventional DA has been extensively explored as a pre-processing operation in various 3D tasks [8, 20, 27, 46–48]. It adopts different spatial transformations to generate diverse views of point clouds that are crucial for learning transformation-invariant and generalizable representations.

- *Scaling* changes the scale of the point cloud by multiplying the coordinates with a ratio s, where a value of $s < 1$ indicates shrinkage and $s > 1$ indicates enlargement.
- *Flipping* randomly flips points along the x-axis or y-axis.
- *Rotation* rotates the points around the z-axis with a random angle.
- *Jittering* adds random perturbations to point clouds with Gaussian noise with zero mean and a standard deviation of β[8].
- *Translation* involves shifting all points in the same direction and distance.

Note conventional DA can be applied to both global point clouds and local point patches [49–51].

Conventional DA has been widely adopted in various point cloud learning tasks due to its simplicity and efficiency. However, it often leads to insufficient training due to two major factors. First, the DA process and network training are independent with little interaction, where the training outcome provides little feedback for DA optimization. Second, the new training samples are augmented from individuals instead of a combination of multiple existing samples, leading to limited training data distribution. Many DA strategies have been designed to address the two limitations.

DA for 3D shape classification. Several studies [52, 53] explored adaptive DA for 3D shape classification. For example, Li et al. [52] designed Pointaugment that generates training samples with shape-wise transformation and point-wise displacement. The Pointaugment and object classifier are jointly optimized via adversarial learning. Kim et al. [53] exploited local deformations of objects, aiming to generate realistic object samples with more variation, e.g., a person of varying

poses. It introduces AugTune which allows adaptively controlling the strength of local augmentation while preserving the shape identity.

Another line of research generates more diverse training objects by mixing existing ones. Inspired by MixUp [54, 55] in 2D image classification, Chen et al. [56] proposed PointMixup that generates object samples via shortest path linear interpolation between two objects of different classes. However, the interpolated samples may lose structural information of the original objects due to geometrical distortion. Several studies attempt to preserve the local object structures during mixing. For example, RSMix [57] mixes and generates new training samples by extracting rigid subsets from different point cloud objects. PointCutMix [58] cuts and replaces local object parts with the optimally assigned pair from other objects. SageMix [59] leverages saliency guidance to preserve local object structures in mixing. Point-MixSwap [60] mixes objects of the same categories to enrich the geometric variation.

DA for 3D object detection. 3D object detection works with scene-level point clouds that are very different from object-level point clouds. Specifically, scene-level point clouds have much more points, more diverse surroundings, larger density variation, and more noises or outliers, which pose both chances and challenges for DA. For example, Cheng et al. [61] proposed Progressive Population-Based Augmentation that searches for optimal DA strategies across point cloud datasets. Chen et al. [62] proposed Azimuth-Normalization to address the significant variation of LiDAR point clouds along the azimuth direction. Leng et al. [63] exploited pseudo labels of unlabelled data for DA in point cloud learning.

The mixing idea has also been explored for 3D object detection. For instance, Yan et al.[64] proposed GT-Aug to enrich foreground instances by copying objects from other LiDAR frames and randomly pasting them into the current frame. However, GT-Aug does not consider the relationships between objects in real-world scenarios during the pasting. To address this limitation, Sun et al.[65] performed object pasting by utilizing a correlation energy field to represent the functional relationship between objects. In addition, Wu et al. [66] fused multiple LiDAR frames to generate denser point clouds and then use them as references to enhance object detection in single-frame scenarios.

DA for 3D semantic segmentation. Mixing-based DA has shown impressive performance gains in point cloud segmentation. For example, Nekrasov et al.[67] proposed Mix3D that directly concatenates two point clouds and their labels for out-of-context augmentation. We proposed PolarMix[68] (which will be introduced in Chapter 3) that mixes LiDAR frames in the polar coordinate system to preserve unique properties of LiDAR point clouds such as partial visibility and density variation.

2.2.1.2 Inter-domain Augmentation

Inter-domain DA utilizes extra data to enhance network training. It can be broadly grouped into two categories depending on the types of data used: synthetic data and cross-modality data.

Synthetic data. Several studies explored synthetic point clouds to augment real ones to improve point cloud network training [2, 69]. For example, Fang et al.[69] designed LiDAR-Aug that inserts CAD objects such as pedestrians into point clouds of road scenes for generating training LiDAR scans with richer objects and training better 3D detectors. We collected self-annotated LiDAR point clouds from game engines[2] and combined them with real point clouds to train 3D segmentation networks, as to be introduced in Chapter 4. Though synthetic data provide a promising solution to mitigate the data constraint, they have clear domain gaps [2] with real point clouds which often limits their effectiveness.

Cross-modality data. Several studies fuse point clouds with data from other modalities for alleviating the inherent limitations of 3D sensors. For example, RGB images are widely adopted to improve network training for 3D object detection [70–75] and 3D semantic segmentation [76]. Recently, several studies [77, 78] fused radar point clouds and LiDAR point clouds for learning more robust and generalizable point cloud models.

2.2.2 Domain Transfer Learning

Domain transfer learning aims to exploit knowledge in previously collected and annotated data for handling various new data, hence reducing the labelling efforts of the new data significantly. However, transferring knowledge across data of different

domains often faces *domain discrepancy* [79, 80], the distributional bias/shift across data of different domains. Consequently, models trained with source-domain data often experience clear performance drops when tested on data of target domains. The domain discrepancy problem has greatly hindered the deployment of point cloud models in various tasks. It has been studied in two typical approaches: *domain adaptation* and *domain generalization*. While both approaches aim to learn robust models from source data that can perform well on target data, domain adaptation permits access to target data in training while domain generalization does not.

2.2.2.1 Domain Adaptation

Domain adaptation aims to adapt a model trained on a source domain to a specific target domain. It provides an economical solution for utilizing existing annotated training data with the same label space for fine-tuning models from a source domain to a target domain. For point clouds, domain adaptation studies have different setups depending on data prerequisites and application scenarios. Specifically, most existing studies focus on unsupervised domain adaptation (UDA) that learns from labelled source point clouds and unlabelled target point clouds. This subsection begins by presenting the problem setup of UDA. Subsequently, it reviews the progress of UDA in three key areas: 3D shape classification, 3D object detection, and 3D semantic segmentation. Finally, it provides an overview of other types of domain adaptation techniques applied to point clouds.

Problem setup. Given source-domain point clouds X^S with the corresponding labels Y^S and target-domain point clouds X^T without labels, the goal of point cloud adaptation is to learn a model F that can produce accurate predictions $\hat{Y}^T$ for unseen target data. The network training in UDA consists of two typical learning tasks, i.e., supervised learning from the labelled source data and unsupervised adaptation toward unlabelled target data. Adaptation is usually achieved via four learning approaches: adversarial training, self-training, self-supervised learning, and style transfer.

- *Adversarial training* [81, 82] aims to learn domain-invariant features. It is achieved by training the model to extract features (from source and target samples) that are indistinguishable by a domain discriminator.

- *Self-training* [83, 84] employs a source-trained model to pseudo-label target data and adopts confident target predictions to retrain the model iteratively. It assumes that the confident target predictions have correct labels.
- *Self-supervised learning* (SSL) [85] aims to learn useful representations from unlabelled target data without any explicit supervision. It is domain-agnostic and exploits the inherent data structure or patterns to define a task that can be solved without human annotations. With SSL over target data, the network can learn features that are tolerant to domain shifts, hence improving the model generalization on target data.
- *Style transfer* [2, 86] aims to translate source data to be similar to the target data for training. It works by learning a mapping function that transforms the source data to have similar styles as the target data. Models trained with the transferred data usually perform better on target data due to the reduced domain discrepancy.

The following subsections review domain adaptive point cloud learning for various 3D tasks.

Domain adaptation for 3D shape classification. Object-level point clouds are often collected from various sources such as synthetic CAD models [31, 32] and real 3D scans [36, 87]. Due to differences in acquisition techniques and object characteristics, the collected point clouds may exhibit clear geometric discrepancies. Several studies have recently explored UDA for 3D shape classification across different 3D object datasets.

Specifically, Qin et al. [88] explored adversarial training and designed PointDAN that utilizes the Maximum Classifier Discrepancy[81] to align features across domains. Several subsequent work [89–91] explored self-paced self-training for domain adaptive 3D shape classification, where the confidence threshold gradually lowers while selecting pseudo labels. Fan et al.[92] designed a voting strategy that pseudo-labels target samples by searching for the nearest source neighbours in a shared feature space. Chen et al.[93] proposed quasi-balanced self-training to address the class imbalance in pseudo-labelling. Cardace et al. [94] proposed to refine noisy pseudo-labels by matching shape descriptors that are learned by the unsupervised task of shape reconstruction on both domains.

Several studies designed SSL tasks to encourage networks to learn domain-invariant features from unlabelled point cloud objects. For example, Zou et al. [89] introduced a joint task that predicts rotation angles and distortion locations. Fan et al.[92] reconstructed the squeezed 2D projections of objects back to 3D space. Shen et al. [90] learned unsupervised features by computing approximations of unsigned distance fields.

Domain adaptation for 3D object detection. Due to differences in physical environments, sensor configurations, weather conditions, etc., scene-level point clouds are subject to more geometry shifts than object-level point clouds in term of point density and occlusion ratios. Domain adaptation across scene-level point clouds is thus even more challenging and it has recently attracted increasing attention thanks to the great values of scene-level 3D tasks such as 3D object detection and 3D semantic segmentation.

Domain adaptive 3D object detection has been studied extensively over the past few years. For example, Wang et al. [95] noticed that car size plays a crucial role in 3D object detection while adapting across data of different countries. They designed a simple normalization strategy for car size, which achieves superb adaptation performance. Later, adversarial training was explored for domain adaptive 3D object detection. For example, Su et al. [96] observed that semantic features contain both domain-specific attributes and other features that may mislead the discriminator. They thus disentangle the domain-specific attributes from the semantic features of LiDAR for better adversarial learning. Zhang et al. [97] recognized the distinctive geometric properties of LiDAR point clouds, i.e., larger and closer objects have more points, and designed scale-aware and range-aware domain alignment strategies for better adversarial training of 3D detectors.

Several methods [98–101] explored self-training for domain adaptive 3D detection. For instance, ST3D [99] updates pseudo labels with a quality-aware triplet memory bank and trains networks with curriculum data augmentation. Luo et al. [100] designed a multi-level consistency network that learns with consistency at the levels of points, instances, and neural statistics. Some work [102–105] instead explored style transfer. For example, Hahner et al.[103, 105] simulated fog and snowfall over authentic point clouds to alleviate domain discrepancy across weather. Xu et al. [104] generated semantic points at foreground regions with missing object parts and combine the generated points with the original to enhance detection across

domains. Further, Yihan et al.[106] proposed a 3D contrastive co-training approach to improve the transferability of learned point features. Wei et al.[107] introduced a teacher-student framework that distills knowledge from high-beam LiDAR data to low-beam data, aiming to reduce the domain gap caused by different LiDAR beam configurations.

Domain adaptation for 3D semantic segmentation. Studies on domain adaptive point cloud segmentation can be broadly classified into two categories namely, uni-modal UDA that works with point clouds alone [2, 4, 108–110] and cross-modal UDA that employs both point clouds and image data in training [111–114].

For *uni-modal UDA*, a line of studies [115–119] projected point clouds to depth images and adopted 2D UDA methods to mitigate domain shifts. For example, Li et al. [119] proposed an adversarial training framework to learn to generate source masks to mimic the pattern of irregular target noise, thereby narrowing the domain gap from synthetic point clouds to real ones. However, the 3D-to-2D projection loses geometric information, and most 2D UDA methods cannot handle the unique geometry of point clouds. Moreover, most 2D UDA methods adopt CNN architectures and cannot be generalized to point cloud architectures.

Another line of methods [2, 4, 108] performed domain adaptive point cloud segmentation over point clouds directly. For example, [108] tackled domain adaptation by transforming it into a 3D surface completion task. [2] employed GANs to translate synthetic point clouds to match the sparsity and appearance of real ones. [4, 68] mix point clouds of source and target domains to generate intermediate representations with less domain discrepancy. While most studies focus on outdoor LiDAR point clouds, [110] recently explored synthetic-to-real adaptation of indoor point clouds.

For *cross-modal UDA*, each training sample typically comprises a 2D image and a 3D point cloud that are synchronized across LiDAR and camera sensors. Point-wise 3D annotations are provided for source data. The goal is to learn a robust 3D segmentor that can work independently and requires no images for testing. Though the paired images can enrich the learned representation, cross-modal UDA is more challenging due to the heterogeneity of the input spaces for images and point clouds as well as additional domain shifts between source and target images. Jaritz et al. [111] developed xMUDA, the first cross-modal UDA framework that adopts

a two-stream architecture to address the domain gap of each modality individually. Peng et al. [112] achieved cross-modal UDA with two modules, the first employing intra-domain cross-modal learning for cross-modal interaction while the second adopting adversarial learning for cross-domain feature alignment via inter-domain cross-modal learning.

Extension. *Source-free UDA* [120] is a variant of UDA that aims to adapt source-trained models to target distributions without accessing the source data in training. It is useful when data privacy and data portability are critical. Recently, Saltori et al. [109] proposed a pioneering study for source-free UDA of point clouds. They designed adaptive self-training with a geometric-feature propagation for semantic segmentation of LiDAR point cloud of road scenes.

Test-time domain adaptation (TTA) is a setup where a source-pretrained model is adapted using only the unlabelled test data, usually with a *single epoch* of training. Unlike typical UDA, the goal of TTA is to avoid collecting target data in advance, where the model is adapted with the test data flow. Though TTA is practical in real-world scenarios, it is challenging as the target data is available in test-stage only. Recently, Inkyu Shin et al. [113] proposed the first TTA attempt on multi-modal 3D semantic segmentation. They designed a multi-modal fusion module to combine multi-modal input data for more accurate segmentation.

2.2.2.2 Domain Generalization

Another research direction in the field of domain transfer learning is *domain generalization* (DG) [121], which aims to train a model using labelled source data that can generalize to any target domains without accessing target data in training. DG removes the dependency on target training data, making it very useful in many real-world tasks where target data is difficult or expensive to obtain before deploying the model. It is also a critical research area for point cloud learning, as many point cloud tasks require 3D deep models to be robust and generalizable to unseen domains. For instance, autonomous vehicles require generalizable 3D perception to operate safely in various unseen places and scenarios.

Problem setup. Given labelled point clouds of K similar but distinct source domains $\mathcal{S} = \{S_k = \{(x^{(k)}, y^{(k)})\}\}_K^{k=1}$, where x denotes a point cloud and y is its labels, DG aims to learn a deep model F with the source data only that can

perform well in unseen target domain $\mathcal{T}$. Similar to 2D DG studies [121], we review two DG settings for 3D point clouds. The first is multi-source DG which assumes the availability of more than one source domain in training, i.e., $K > 1$. The motivation is to learn domain-invariant features (from multiple similar but distinct source domains) that can generalize well to any unseen domains. The second is single-source DG which is more challenging as it allows training data from a single source domain only. At the other end, single-source DG methods are more generic and can be applied to multi-source DG problems by ignoring the domain label.

Domain generalization for 3D shape classification. The pioneer work [122] first explored geometry shifts from simulated point clouds of CAD objects (e.g. ModelNet dataset[31]) to real object point clouds (e.g. ScanObjectNN [87]). It presents a meta-learning framework to train generalizable 3D classification models across domains. Later, Huang et al.[123] designed a manifold adversarial training scheme that exploits multiple geometric transformations to generate adversarial training samples of intermediate domains. Both studies fall under the single-source DG setting.

Domain generalization for 3D object detection. Improving the generalizability of 3D detectors is essential in 3D vision tasks such as autonomous driving where perception algorithms must maintain stable performance over unseen domains. However, DG for 3D object detection remains a relatively under-explored area. The pioneer study in [124] presented the first attempt at single-source DG for 3D object detection. It presents an adversarial augmentation method that learns to deform point clouds in training to enhance the generalization of 3D detectors. Recently, [125] introduced a single-source DG approach for multi-view 3D object detection in Bird-Eye-View (BEV). It decouples depth estimation from camera parameters, employs dynamic perspective augmentation, and adopts multiple pseudo-domains for better generalization toward various unseen new domains.

Domain generalization for 3D semantic segmentation. Several studies on domain-generalizable point cloud segmentation have been reported recently. Xiao et al.[44] study outdoor point cloud segmentation under adverse weather, where a domain randomization and aggregation learning pipeline was designed to enhance the model generalization performance. [126] augments the source domain and introduces constraints in sparsity invariance consistency and semantic correlation consistency for learning more generalized 3D LiDAR representations.

Chapter 3

Data Augmentation for LiDAR Point Cloud Recognition[1]

In this chapter, we present PolarMix, a novel data augmentation approach for LiDAR point clouds. PolarMix is tailored for 3D LiDAR perceptions of road scenes, which is crucial to various applications such as robotics and autonomous driving. It comprises two cross-scan augmentations operating in a polar coordinate system, namely scene-level swapping and instance-level rotate-pasting. These techniques effectively enhance the distribution of training point clouds while maintaining their fidelity. Consequently, we are able to train accurate and robust point cloud recognition models using fewer labelled training point clouds, improving the performance of 3D recognition models in a label-efficient manner.

3.1 Introduction

In the past decade, LiDAR sensors have been increasingly employed in various perception-related applications such as autonomous driving. They provide accurate and robust depth sensing of the surrounding environments which is crucial for scene understanding for autonomous navigation indoors and outdoors. With the recent advance of deep neural networks (DNNs), point cloud understanding has

[1]The work in this chapter has been published at "Aoran Xiao, Jiaxing Huang, Dayan Guan, Kaiwen Cui, Shijian Lu, and Ling Shao. Polarmix: A general data augmentation technique for lidar point clouds[J]. Advances in Neural Information Processing Systems, 2022, 35: 11035-11048."

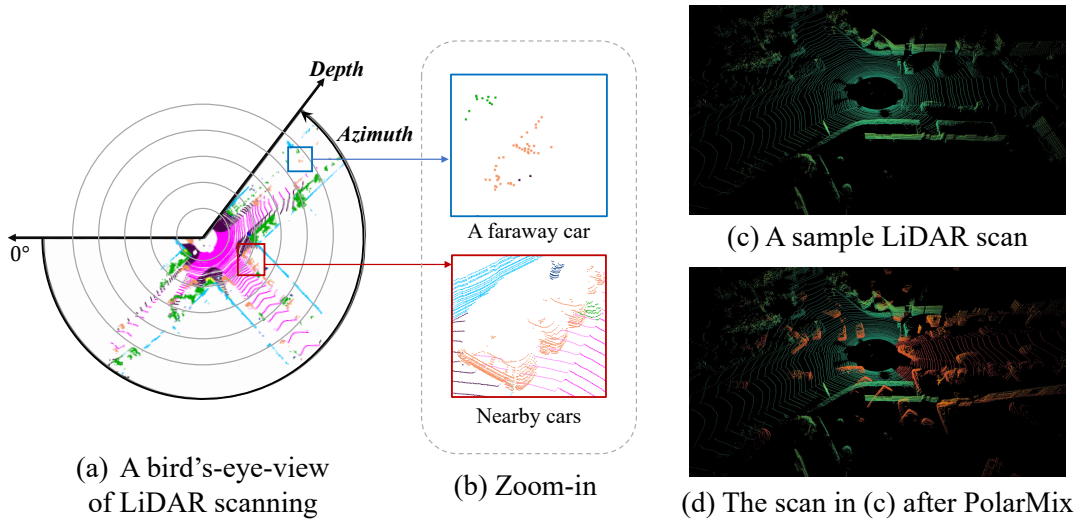


FIGURE 3.1: A LiDAR sensor rotates and scans environments by the azimuth in XY plane, and the captured points (as shown in a bird's-eye view in (a)) bear LiDAR-specific properties including partial visibility (i.e., only object sides facing the LiDAR sensor have points captured) and density variation along the depth as illustrated in close-up views in (b). PolarMix mixes points across LiDAR scans along the scanning direction which enriches point distribution while preserving data fidelity effectively. For a sample LiDAR scan in (c), (d) shows one of its augmentations with PolarMix where points in orange color are cropped and copied from another LiDAR scan.

achieved significant progress in various perception tasks such as semantic segmentation [6, 20, 47, 127–129] and object detection [64, 130, 131]. On the other hand, training reliable DNN models requires a large amount of well-annotated training data, whereas collecting and annotating large amounts of point clouds is often laborious, time-consuming, and has poor scalability across tasks and domains. This has become one major constraint in LiDAR point cloud analytics and understanding.

Data augmentation (DA) [132, 133], which aims to expand the distribution of the training data by modifying and creating new training samples, has been widely studied for 2D images and demonstrated great potential in training robust DNN models with limited training images. However, most existing DA methods do not work well for LiDAR point clouds, a very meaningful but largely neglected task. Specifically, most existing DA methods perform *global augmentation* such as random scaling, flipping, and rotation which cannot augment local structures or model relationships across neighbouring point cloud scans. Recently, several studies [54, 134, 135] explore *local augmentation* that creates new training samples

via cut and mix of 2D images. However, the *local augmentation* does not work well for point clouds as it does not consider the unique scanning mechanism of LiDAR sensors (e.g., via continuous 360-degree sweeping) and specific properties of the captured point data.

This work focuses on effective and efficient data augmentation for better learning from limited LiDAR point cloud data. To this end, we design *PolarMix*, a simple yet generic DA technique that can effectively work across different perception tasks and datasets. PolarMix achieves these unique features by capturing the essential properties of LiDAR point clouds, namely, partial visibility and density variation that are closely associated with the sweeping mechanism of LiDAR sensors. Specifically, objects in LiDAR scans are incomplete where only object sides facing the LiDAR sensor are scanned with points as illustrated in Fig. 3.1(a). In addition, point density varies with point depth as illustrated in Fig. 3.1(b). Effective data augmentation needs to cater for these LiDAR-specific features to ensure the fidelity and usefulness of the augmented point clouds in network training.

Inspired by the above observations, PolarMix crops, edits, and mixes points along the LiDAR scanning direction (*i.e.*, the *azimuth* in the 3D polar system in Fig. 3.1(a)) to enrich point cloud distributions while maintaining its fidelity. It consists of two cross-scan augmentation approaches. The first is scene-level swapping which exchanges point cloud sectors of two circular LiDAR scans that are cut along the azimuth axis as illustrated in Fig. 3.2(a). The second is instance-level rotation and paste which cuts point cloud instances from one LiDAR scan, rotates them along the scanning direction multiple times (to create multiple copies), and pastes the rotated instances into another LiDAR scan as illustrated in Fig. 3.2(b). For the sample LiDAR scan in Fig. 3.1(c), Fig. 3.1 (d) shows one of its augmentations where point cloud sectors and instances are mixed in with high fidelity. Extensive evaluations show that the PolarMix augmented point clouds improve the network training consistently across various tasks and benchmarks, more details to be described in the experiment part.

The contribution of this paper can be summarized in three aspects. *First*, we introduce PolarMix, a simple yet effective point cloud augmentation technique that can enrich point cloud distributions while maintaining point cloud fidelity concurrently. *Second*, PolarMix is generally applicable and can work for different network architectures, perception tasks (e.g., object detection, semantic segmentation, etc.), and

datasets/domains with consistent performance gains. *Third*, PolarMix is easy to use and can be incorporated as a plug and play by most existing point cloud networks. It also works well for unsupervised domain adaptation with state-of-the-art performance.

3.2 Related Works

Data augmentation for 2D images. Data augmentation has been widely studied across different 2D computer vision tasks such as image classification [136, 137], object detection [138], and semantic segmentation [139]. It plays an important role in effective and efficient deep network training since collecting and annotating training images is often laborious and time-consuming. One typical DA approach is global augmentation that aims to learn certain transformation invariance in image recognition tasks [135, 140], *e.g.*, random cropping [141–143], random scaling[142, 143], random erasing [144], color jittering [143], etc. Another typical DA approach is local augmentation which performs various mix operations to generate new training data. For example, mixup [54, 55] generates new data via convex combinations of the input pixels/feature embeddings and the output labels. CutMix [134] pastes rectangular crops from other images instead of mixing the whole images. Recently, several studies [70, 135, 145, 146] introduce the object concept into the cut and mix operations: [145] extends mixup and cutmix into object detection; [70, 135, 146] cut instances from one image and paste them into another for training better instance segmentation networks.

Data augmentation for 3D point clouds. Data augmentation of point clouds has also attracted increasing attention in recent years. Similar to 2D computer vision tasks, one direct approach is to adopt global augmentation in 3D space such as random scaling, rotation, and translation, which can be directly incorporated for expanding 3D objects [31, 32], indoor point clouds [36], and outdoor point clouds [1, 7, 147]. Several studies also explored to augment local structures of point clouds: PointAugment [52] introduces an auto-augmentation network for shape-wise transformation and point-wise displacement; PatchAugment [50] exploits data augmentation in local areas; PointWOLF [53] applies weighted transformations in local neighbourhoods for enhancing the diversity of 3D objects. However, the

aforementioned works focus on object-level augmentation which are not suitable for scene-level point clouds such as those in autonomous driving.

Recently, several studies explored the idea of mixing for augmenting point clouds. For example, PointMixUp [56] interpolates 3D objects to create new samples for training. PointCutMix [57] replaces subsets of point objects with that of other objects to enrich training data. However, both work focuses on object-level augmentation only. Several studies [2, 64, 67, 69] also explore scene-level mix but they are constrained with specific vision tasks. For example, GT-Aug [64, 69] cuts instances and pastes them into other LiDAR scans for the object detection task (requiring 3D bounding boxes for object cutting); Mix3D [67] concatenates points of two scenes as an out-of-context augmentation for the specific task of semantic segmentation. As a comparison, the proposed PolarMix can perform both object-level and scene-level augmentation. More importantly, its designs are perfectly aligned with LiDAR-specific data properties including partial visibility and density variation with depth which guarantees superior fidelity and effectiveness of the augmented point clouds. Furthermore, PolarMix is generic and applicable to various computer vision tasks such as semantic segmentation and object detection, more details to be discussed in the experiment part.

3.3 PolarMix for Augmenting LiDAR Point Cloud Recognition

Problem statement. Let $s \in \mathbb{R}^{N\times 4}$ and y denote a LiDAR scan with N points and its labels, respectively. Each point p_i in s is a 1×4 vector with a 3D Cartesian coordinate relative to the scanner (x_i, y_i, z_i) and an intensity value of returning laser beam. The goal of PolarMix is to generate new training samples $(\tilde{s}, \tilde{y})$ by cutting and mixing across two training samples (s^A, y^A) and (s^B, y^B). The generated training samples $(\tilde{s}, \tilde{y})$ is used for network training with the original loss function.

The polar coordinates. We adopt a 3D polar coordinate system where the position of a point is defined by three numbers (θ, r, ϕ): θ is the *azimuth* angle from x-axis to y-axis which defines the rotation scanning angle of LiDAR sweeping; r

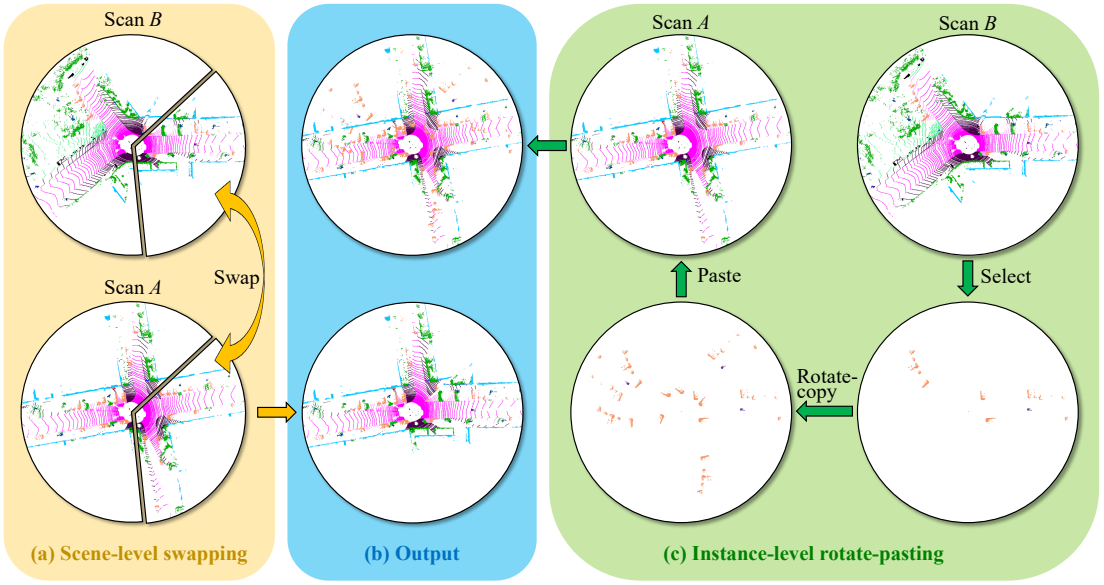


FIGURE 3.2: The proposed *PolarMix* consists of two data augmentation designs: (a) The scene-level swapping exchanges sectors of LiDAR scans A and B that are cut with certain azimuth angles; (c) The instance-level augmentation cuts point instances from scan B, rotates them about the z-axis by multiple azimuth angles (for creating multiple copies of the cut point instances), and pastes the cut and rotated instances into scan A; The augmentations of scan A by the two proposed augmentation approaches are shown in (b).

denotes the *depth* which is the distance of the point to the LiDAR sensor; ϕ is the *inclination* angle between the z-axis and the point vector (x_i, y_i, z_i).

PolarMix for LiDAR data augmentation. We designed two point cloud augmentation approaches in PolarMix including a scene-level swapping approach $Sw()$ and an instance-level rotate-paste approach $Rp()$ for mixing LiDAR scans s^A, s^B and their labels y^A, y^B. The combination of the two augmentation approaches in PolarMix can be defined as

$$\begin{aligned} \tilde{s} &= Sw(s^A, s^B|\alpha, \beta) \oplus Rp(s^A, s^B|C, \Omega) \\ \tilde{y} &= Sw(y^A, y^B|\alpha, \beta) \oplus Rp(y^A, y^B|C, \Omega) \end{aligned} \tag{3.1}$$

where $\oplus$ denotes a concatenation operation. C and Ω represent class list and angle list for instance-level rotation and paste, respectively.

The scene-level swapping $Sw(s^A, s^B|\alpha, \beta)$ aims to cut a point cloud sector from azimuth angle α to azimuth angle β from a LiDAR scan s^A, and switch it with the similarly cut point cloud sector from another LiDAR scan s^B. The swapping

operation can be defined as follows

$$\begin{aligned} Sw(s^A, s^B|\alpha, \beta) &= ((1 - \mathbf{M}_A^{\alpha,\beta}) \odot s^A) \oplus (\mathbf{M}_B^{\alpha,\beta} \odot s^B) \\ Sw(y^A, y^B|\alpha, \beta) &= ((1 - \mathbf{M}_A^{\alpha,\beta}) \odot y^A) \oplus (\mathbf{M}_B^{\alpha,\beta} \odot y^B) \end{aligned} \tag{3.2}$$

where $\mathbf{M}_A^{\alpha,\beta}$ denotes a binary mask indicating the azimuth range $[\alpha, \beta]$ that is cut out from LiDAR scan s^A, and $\odot$ is element-wise multiplication. The scene-level swapping thus exchanges two point cloud *sectors* of the same size from two LiDAR scans in a bird's-eye-view as illustrated in Fig. 3.2 (a) since the maximum scanning area in the horizontal plane is a circle. Note the angles α, β should be within the horizontal field-of-view of LiDAR sensor (*e.g.* within 360°).

The instance-level augmentation $Rp(s^A, s^B|C, \Omega)$, as illustrated in Fig. 3.2 (c), crops point instances of semantic classes C from LiDAR scan s^B, rotates them about z-axis by multiple azimuth angles $\Omega = \{\omega_1, \ldots, \omega_J\}$ to create multiple copies, and pastes all cropped and rotated point instances into another scan s^A. This augmentation operation can be defined by

$$\begin{aligned} Rp(s^A, s^B|C, \Omega) &= s^A \oplus \sum_{\omega \in \Omega} \mathcal{R}_\omega(\mathbf{M}_B^C \odot s^B) \\ Rp(y^A, y^B|C, \Omega) &= y^A \oplus \sum_{\omega \in \Omega} (\mathbf{M}_B^C \odot y^B) \end{aligned} \tag{3.3}$$

where $\mathbf{M}_B^C$ is a binary mask indicating which semantic classes of points instances to crop from LiDAR scan B, and $\mathcal{R}_\omega$ represents the rotation matrix around the z-axis for an azimuth angle ω.

Using a polar coordinate system to augment LiDAR points has two desirable features. First, it allows point cutting, rotating, and mixing to be perfectly aligned with the LiDAR scanning mechanism which greatly helps to preserve LiDAR-specific data properties such as partial visibility and density variation along the depth. Second, it simplifies the augmentation process with negligible computational overhead: The scene-level swapping involves point slicing and concatenating only while the instance-level augmentation can be achieved by simple dot products followed by point concatenation. Beyond that, PolarMix works in the input space which is naturally compatible with different network architectures. Algorithm 1 summarizes the pipeline of the proposed PolarMix.

Algorithm 1 PolarMix.

Input: Points and labels of two LiDAR scans: $\{s^A, y^A\}, \{s^B, y^B\}$; Class list and angle list for instance-level rotate and paste: C, Ω; Azimuth range for scene-level swapping: α, β.

Output: A new LiDAR scan for training: $\{\tilde{s}, \tilde{y}\}$.

$\tilde{s}, \tilde{y} = s^A, y^A$ *# Initialization*
if rand() $\leq \delta_1$ **then** *# Scene-level swapping*
    Calculate azimuth θ for points in $\tilde{s}, s^B$ as $\tilde{\theta}, \theta^B$
    Delete points with labels in $\tilde{s}, \tilde{y}$ if $\alpha \leq \tilde{\theta} \leq \beta$
    Cut points with labels in s^B, y^B if $\alpha \leq \theta^B \leq \beta$
    Update $\tilde{s}, \tilde{y}$ by concatenating with cut points and labels from Scan B
end if
if rand() $\leq \delta_2$ **then** *# Instance-level rotate-pasting*
    Copy points with labels from s^B, y^B according to C
    for ω_j in Ω **do**
        Rotate copied points with $\mathcal{R}_{\omega_j}$, duplicate their labels
        Update $\tilde{s}, \tilde{y}$ by concatenating rotated points and labels
    end for
end if

PolarMix for unsupervised domain adaptation. The proposed PolarMix can be directly applied for unsupervised domain adaptation via self-training [148]. With LiDAR data from a labeled *source domain*, a supervised network model can be trained and applied to predict pseudo labels for LiDAR data from an unlabeled *target domain*. PolarMix can then cut and mix LiDAR scans between the source domain (with ground-truth labels) and the target domain (with pseudo labels). Such augmented LiDAR data mitigates the inter-domain discrepancy which facilitates unsupervised domain adaptation effectively, more details to be described in the ensuing experiments.

3.4 Experiments

We evaluate how PolarMix benefits deep neural network training for LiDAR point cloud understanding. In Section 3.4.1, we evaluate it over the semantic segmentation task across different deep architectures and benchmarking datasets. In Section 3.4.2, we evaluate it over object detection, mainly to examine its generalization capability across different computer vision tasks. In Section 3.4.3, we evaluate how it facilitates unsupervised domain adaptation over multiple synthetic-to-real domain

adaptation benchmarks of LiDAR data. Finally, we provide an in-depth analysis of different components in PolarMix in Section 3.4.4.

3.4.1 PolarMix for Semantic Segmentation

We first study how PolarMix helps learn better representation for semantic segmentation. The experiments were conducted over multiple deep architectures and public datasets.

3.4.1.1 Experimental Settings

Dataset. We evaluate PolarMix over three LiDAR datasets of driving scenes that have been widely adopted for benchmarking in semantic segmentation. The first is **SemanticKITTI** [7] which is a large-scale dataset collected in a city of Germany. It has 43,551 LiDAR scans with 64 beams with point-wise annotations of 19 semantic classes. We follow the widely-adopted split and use sequences 00-07, 09-10 as the training set and sequence 08 for validation. The second is **nuScenes-lidarseg** [147] dataset which has 40,000 scans captured in 1000 scenes of 20s duration. It is collected with a 32 beams LiDAR sensor at 20Hz frequency with point-wise annotations of 16 semantic classes. We follow the officially split of training data and validation data. The third is **SemanticPOSS** [1] which consists of 2,988 annotated point cloud scans of 14 semantic classes. We follow the official benchmark setting, *i.e.* sequence 03 for validation and the rest for training. For all semantic segmentation experiments, we adopt mean intersection-over-union (mIoU) as the evaluation metric.

Architectures and implementation details. We evaluate PolarMix over four widely adopted semantic segmentation networks: 1) MinkNet [10] which is a typical voxel-based sparse CNN; 2) SPVCNN [6] which is a hybrid network with a sparse convolutional and a point-based sub-network; 3) RandLA-Net [47] which is a standard point-based network; and 4) Cylinder3D [127] which a state-of-the-art cylindrical and asymmetrical 3D CNN. We adopt the default training hyper-parameters in the open-source repositories[2][3][4] for all four networks, and the only modification

[2]MinkNet and SPVCNN: `https://github.com/mit-han-lab/spvnas`
[3]RandLA-Net: `https://github.com/QingyongHu/RandLA-Net`
[4]Cylinder3D: `https://github.com/xinge008/Cylinder3D`

TABLE 3.1: Semantic segmentation over the validation set of the dataset SemanticKITTI. The baseline with either MinkNet or SPVCNN does not involve any data augmentation. CGA means conventional global augmentation which includes random scaling and random rotation. The symbol † mean that the related local data augmentation is on top of CGA, e.g., *+CutMix*† means that the network training involves both CGA and CutMix. PolarMix achieves clearly the best semantic segmentation across both deep networks.

Methods	car	bi.cle	mt.cle	truck	oth-v.	pers.	bi.clst	mt.clst	road	parki.	sidew.	oth-g.	build.	fence	veget.	trunk	terra.	pole	traf.	mIoU
MinkNet[10]	95.9	3.7	44.9	53.2	42.1	53.7	68.9	0.0	92.8	43.0	80.0	1.8	90.5	60.0	87.4	64.5	73.3	62.1	43.7	55.9
+CGA	96.3	8.7	52.3	63.2	51.6	63.5	74.4	0.1	93.3	46.6	80.4	0.8	90.3	60.0	88.0	65.1	74.5	62.8	46.8	58.9(+3.0)
+CutMix†[134]	96.0	10.2	59.3	**78.7**	52.1	63.4	79.4	0.0	93.5	47.8	80.7	**1.6**	90.3	61.0	87.5	66.2	73.3	64.0	46.8	60.6(+5.7)
+CopyPaste†[135]	**96.6**	18.4	62.8	76.3	**64.6**	68.9	82.8	1.0	93.1	45.3	80.2	1.4	90.5	60.7	88.1	67.8	74.6	63.7	49.1	62.4(+6.5)
+Mix3D†[67]	96.3	29.6	61.8	68.5	55.4	**72.7**	77.7	1.0	**94.3**	**52.9**	**81.7**	0.9	89.1	55.5	88.3	**69.3**	74.6	**65.2**	**50.3**	62.4(+6.5)
+PolarMix†(ours)	96.3	**51.2**	**75.6**	63.4	63.9	71.9	**85.6**	**4.9**	93.6	45.8	81.4	1.4	**91.0**	**62.8**	**88.4**	68.5	**75.0**	64.6	49.9	**65.0(+9.1)**
SPVCNN[6]	94.9	9.1	55.8	66.5	33.7	61.8	75.9	0.2	93.1	45.3	79.6	0.4	91.4	62.7	87.5	66.2	72.9	62.8	42.7	58.0
+CGA	96.1	21.8	57.8	**69.2**	49.8	66.7	80.8	0.0	93.4	44.8	80.1	0.2	90.9	62.9	88.5	64.8	75.7	63.6	46.2	60.7(+2.7)
+CutMix†[134]	96.1	21.4	59.6	71.2	54.2	66.8	81.8	0.0	93.5	**49.6**	81.1	2.2	90.9	63.1	87.9	66.9	74.1	63.8	49.8	61.7(+3.7)
+CopyPaste†[135]	96.0	32.4	66.4	67.1	52.9	74.8	84.3	3.6	93.3	46.9	80.2	**2.5**	91.1	**64.1**	88.1	67.0	73.9	64.0	**51.6**	63.2(+5.2)
+Mix3D†[67]	**96.5**	35.9	65.0	66.6	60.2	75.3	83.3	0.0	**93.8**	49.0	81.1	1.4	90.6	60.0	**89.2**	**70.2**	**76.4**	**64.8**	50.5	63.7(+5.7)
+PolarMix†(ours)	**96.5**	**53.9**	**79.7**	68.5	**64.9**	**75.6**	**87.8**	**7.5**	93.5	47.3	**81.2**	1.1	**91.2**	63.8	88.2	68.2	74.2	64.5	49.4	**66.2(+8.5)**

is the batch size for SPVCNN and MinkNet (we change it to 8). We conducted experiments with a single Tesla 2080Ti GPU for MinkNet and SPVCNN and a Tesla V100 GPU for RandLA-Net and Cylinder3D. Note training RandLA-Net and Cylinder3D takes a relatively longer time, we therefore uniformly sub-sampled the same 10% of SemanticKITTI for faster experiments with these two networks.

For augmentation with scene-level swapping, we randomly crop 180° sectors from 360° for $[\alpha, \beta]$ for point swapping. For augmentation with instance-level rotate and paste, we take three rotation angles for dataset SemanticKITTI (0°, and another two rotation angles randomly picked from $(0°, 120°]$ and $(120°, 240°]$), and two rotation angles for datasets SemanticPOSS and nuScenes-lidarseg (0° and another rotation angle randomly chosen from either $+90°$ or $-90°$). We set δ_1, δ_2 as 0.5, 1, respectively. We also examine hyper-parameters of PolarMix with details provided in the appendix.

3.4.1.2 Results

PolarMix improves semantic segmentation by large margins. Since data augmentation for LiDAR semantic segmentation is a relatively under-explored task with few existing works, we selected the highly-related mixing-based methods including Cut-Mix [134] and Copy-paste [135] in 2D vision and the pioneering work

TABLE 3.2: Semantic segmentation over the validation set of the datasets nuScenes-lidarseg and SemanticPOSS. The baseline with either MinkNet or SPVCNN does not involve any data augmentation. CGA means conventional global augmentation which includes random scaling and random rotation. The symbol † mean that the related local data augmentation is on top of CGA, e.g., *+CutMix*† means that the network training involves both CGA and CutMix. PolarMix achieves clearly the best semantic segmentation across both deep networks.

DA methods	MinkNet [10]		SPVCNN [6]	
	nuScenes-lidarseg	SemanticPOSS	nuScenes-lidarseg	SemanticPOSS
None	67.1	52.1	68.4	50.7
+CGA	70.2(+3.1)	55.1(+3.0)	69.1(+0.7)	55.3(+4.6)
+CutMix† [134]	70.4(+3.3)	56.0(+3.9)	71.7(+3.3)	54.7(+4.0)
+Copy-Paste† [135]	70.8(+3.7)	55.9(+2.8)	71.3(+2.9)	56.2(+5.5)
+Mix3D† [67]	70.1(+3.0)	55.3(+3.2)	70.5(+2.1)	54.4(+3.7)
+PolarMix†(Ours)	**72.0(+4.9)**	**57.4(+5.3)**	**72.1(+3.7)**	**58.6(+7.9)**

Mix3D [67] in 3D vision as baseline augmentation methods, and compared the proposed PolarMix with them. Tables 3.1 and 3.2 show experimental results across two networks *MinkNet* and *SPVCNN* and three datasets SemanticKITTI, nuScenes-lidarseg, and SemanticPOSS. It can be observed that the baseline with either MinkNet or SPVCNN (without involving any data augmentation in network training) produces fair semantic segmentation for all three datasets. However, including global augmentation with random scaling and rotation (i.e., *+CGA* in the two tables) improves semantic segmentation consistently across the two networks and the three datasets. On top of the global augmentation, further including local augmentation (i.e., *+CutMix*†, *+CopyPaste*†, and *+Mix3D*†) further improves the semantic segmentation in most cases. As a comparison, PolarMix introduces the best performance gains consistently across the two baseline networks and the three evaluated benchmarking datasets, demonstrating its great robustness and generalization capabilities across different network architectures and datasets.

PolarMix demonstrates noteworthy achievements: 1) It delivers substantial segmentation enhancements, particularly for long-tailed classes within the *foreground* category that possess fewer examples in the original training set, such as *bicycle, motorcycle, truck, other-vehicle, person, bicyclist*, and *motorcyclist*. This improvement is attributed to the instance-level rotate pasting process, which significantly augments the samples of these classes in the training set, thereby enhancing recognition. 2) It showcases evident advancements in the segmentation of *background*

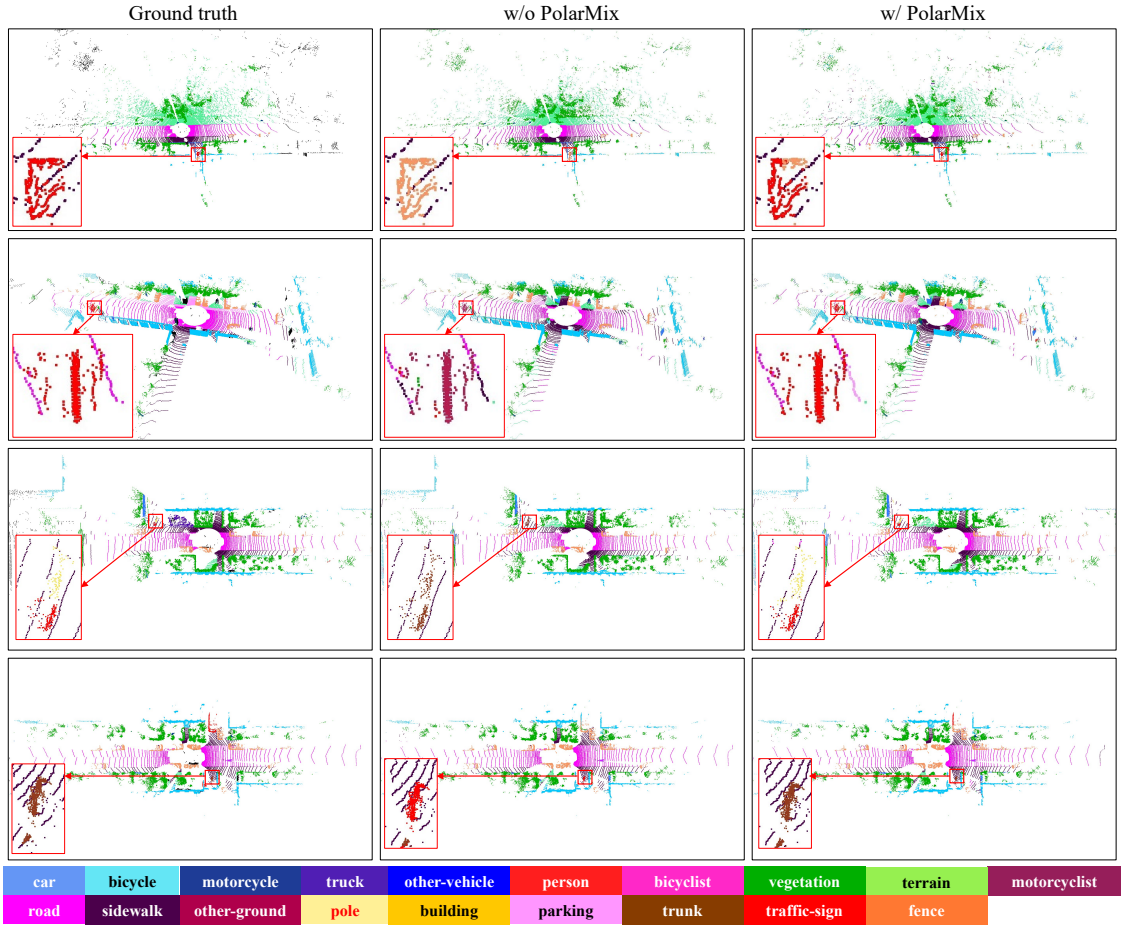


FIGURE 3.3: Illustration of semantic segmentation of SemanticKITTI point cloud by SPVCNN. The left column shows examples with ground-truth segmentation; The middle column shows predictions of SPVCNN; The right column shows predictions of SPVCNN trained with our PolarMix. We zoom in on areas in red boxes for better illustration. PolarMix can achieve better segmentation results.

classes like *parking*, *building*, and *fence*. This progress arises from scene-level swapping, which generates new training samples featuring novel layouts. Segmentation models demonstrate increased robustness when exposed to these diverse samples, leading to clearer improvements in their performance.

We provide visual illustrations of PolarMix in semantic segmentation over SemanticKITTI. Fig. 3.3 and Fig. 3.4 show predictions of SPVCNN trained with or without PolarMix. We can see that PolarMix helps to achieve better segmentation aligned with the results in Table 3.1.

PolarMix improves data-efficiency. The proposed PolarMix works reliably

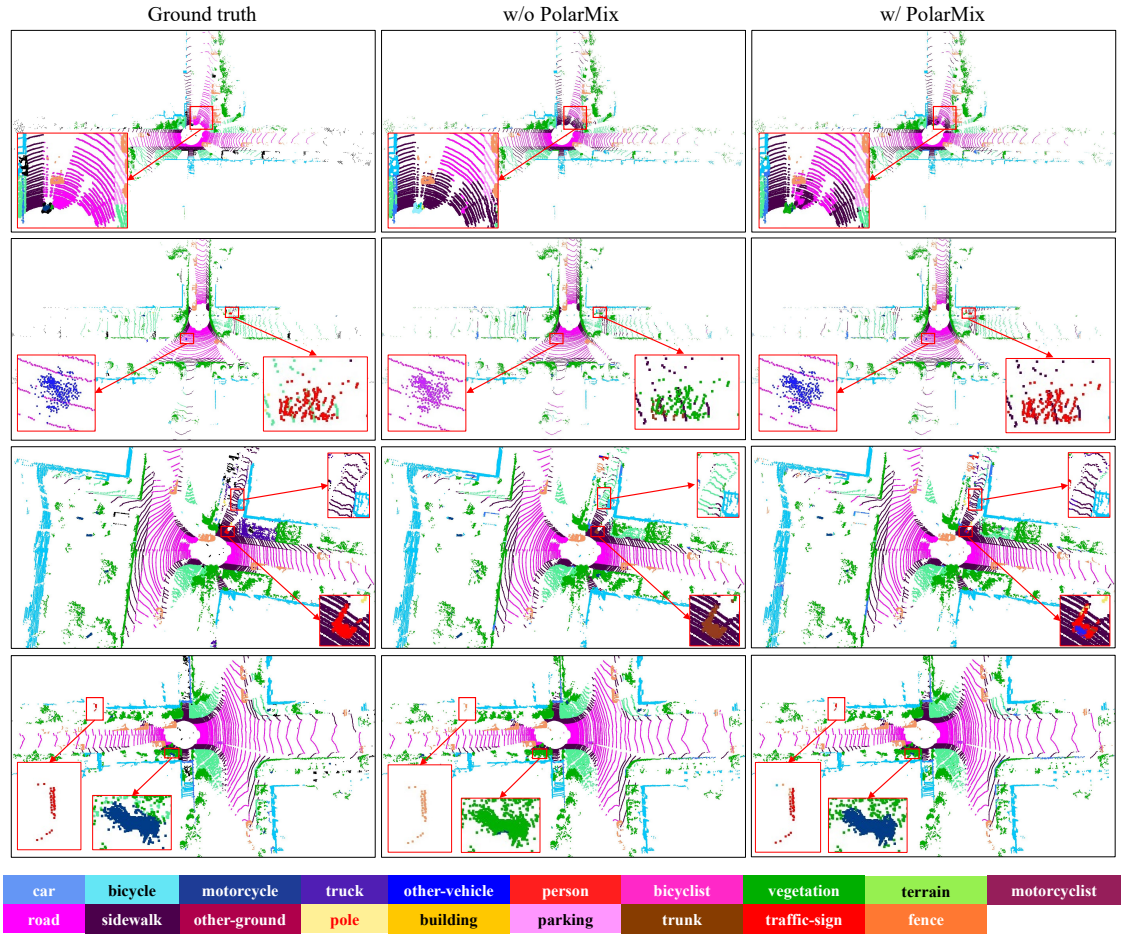


FIGURE 3.4: Illustration of semantic segmentation of SemanticKITTI point cloud by SPVCNN. The left column shows examples with ground-truth segmentation; The middle column shows predictions of SPVCNN; The right column shows predictions of SPVCNN trained with our PolarMix. We zoom in on areas in red boxes for better illustration. PolarMix can achieve better segmentation results.

with different amounts of training data, and it also improves data efficiency by reducing training data and annotations effectively. As shown in Fig. 3.5, the data augmentation with PolarMix consistently helps across different proportions of training data of SemanticKITTI as well as two different segmentation networks MinkNet and SPVCNN. In addition, including PolarMix can achieve similar segmentation mIoU while using 75% of SemanticKITTI in network training (as compared with training with 100% SemanticKITTI without using PolarMix) for both networks, hence saves 25% efforts in training data collection and annotations.

PolarMix works across deep architectures. The proposed PolarMix can work across different deep architectures beyond the voxel-based MinkNet and the sparse

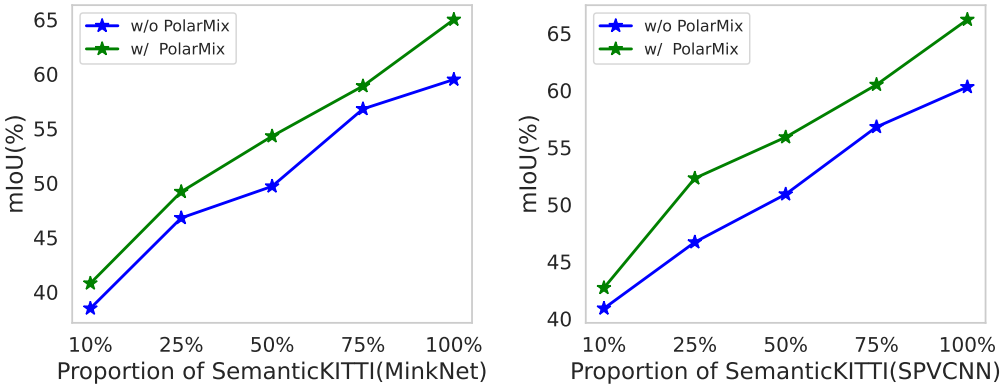


FIGURE 3.5: PolarMix helps reduce annotated training data effectively. For both MinkNet and SPVCNN, including PolarMix achieves similar segmentation accuracy by using around 75% annotated training data only, hence helps save around 25% efforts in training data collection and annotation.

TABLE 3.3: Semantic segmentation results over the validation set of the SemanticKITTI dataset. We subsample the same 10% of the dataset for training. PolarMix consistently works across different 3D deep architectures.

Methods	mIoU
RandLA-Net [47]	45.4
RandLA-Net [47]+PolarMix	50.5(+5.2)
Cylinder3D [127]	60.6
Cylinder3D [127]+PolarMix	62.5(+1.9)

convolution network SPVCNN. We evaluate this feature by experimenting with two new deep architectures including the point-based network RandLA-Net [47] and the more recent 3D cylindrical convolutional architecture of Cylinder3D [127]. We train the two networks with default settings as in the officially released repositories. As shown in Table 3.3, incorporating PolarMix improves the segmentation performance consistently by large margins for both strong baselines (trained with 10% of SemanticKITTI data). This further verifies that PolarMix has superior generalization capability across different 3D deep architectures.

3.4.2 PolarMix for Object Detection

Setup. The proposed PolarMix works in the input space with independence of specific tasks. We verify this property by evaluating it over object detection, another classical 3D understanding task that aims to predict 3D bounding box and label for each interested object instance. We perform experiments with dataset

TABLE 3.4: Object detection results on the validation set of nuScenes dataset. Incorporating PolarMix into the network training consistently improves the object detection across three different deep frameworks including PointPillar, Second, and CenterNet.

Methods	Car	Truck	Bus	Trailer	CV	Ped	Motor	Bicycle	TC	Barrier	mAP	NDS
PointPillar [130]	80.4	45.0	54.0	25.4	10.5	71.0	36.0	9.4	44.8	42.8	41.8	54.9
+PolarMix	80.9	50.1	59.2	33.7	13.6	69.3	37.0	6.4	44.7	42.0	43.7(+1.9)	55.7(+0.8)
Second [64]	80.8	49.8	60.5	27.3	14.4	78.0	41.8	20.8	61.0	53.4	48.8	58.6
+PolarMix	81.3	53.6	68.3	34.4	20.0	76.5	38.2	14.7	59.6	56.8	50.3(+1.5)	60.0(+1.2)
CenterNet[131]	81.0	51.6	62.3	27.9	14.9	79.5	56.0	41.2	59.0	54.9	52.8	59.6
+PolarMix	80.4	53.4	68.8	32.5	17.5	79.1	58.3	44.1	57.7	62.2	55.4(+2.6)	61.1(+1.5)

nuScenes [147] and three classical deep networks including PointPillar [130], Second [64], and CenterNet[131]. For implementation, we adopted default training hyper-parameters and optimizer in the OpenPCDet repository and training with two Tesla 2080Ti GPUs (11GB). We used random flip along the X and Y axis, random rotation, and random scaling for basic data augmentation. For a fair comparison, PolarMix is directly implemented on top of the baseline with the same configurations. For evaluation metrics, we adopted the widely used mean Average Precision (mAP) and nuScenes detection score (NDS).

Results. Table 3.4 shows experimental results. It can be observed that incorporating PolarMix improves both mAP and NDS consistently across the three tested deep networks. This experiment shows that the proposed PolarMix has superior generalization capability across different computer vision tasks, largely due to its cut-edit-mix strategy which enriches the distribution of the training data in the input space without changing LiDAR-specific data properties.

3.4.3 PolarMix for Reducing Domain Discrepancy

Unsupervised domain adaptation (UDA) is an important research topic, aiming to solve the noticeable performance drops of deep neural networks while training and testing across different domains, as a result of the distribution bias (domain shift). UDA has been widely studied in both 2D vision [83, 84, 150–153] and 3D vision [2, 4, 88, 100, 109, 111]. The proposed PolarMix can be easily extended for UDA by mixing labelled source point data and unlabeled target point

TABLE 3.5: Experiments on unsupervised domain adaptation with SynLiDAR (as source) and SemanticKITTI and SemanticPOSS (as target). PolarMix achieves clearly the best semantic segmentation across both unsupervised domain adaptation setups.

Methods	SynLiDAR → SemanticKITTI	SynLiDAR → SemanticPOSS
Source Only	20.4	20.1
ADDA [82]	22.8	24.9
Ent-Min [149]	25.5	25.5
Self-training [84]	26.5	27.1
PCT [2]	28.9	29.6
PolarMix(Ours)	**31.0**	**30.4**

data. We evaluate this nice feature by conducting experiments over two challenging synthetic-to-real point cloud segmentation benchmarks including SynLiDAR → SemanticKITTI and SynLiDAR → SemanticPOSS. SynLiDAR [2] is a synthetic LiDAR point cloud dataset that consists of 198k scans as collected from virtual scenes. It shares 19 common point classes with the SemanticKITTI and 13 common point classes with the SemanticPOSS. In the experiments, we train networks with the labelled SynLiDAR data (as the source data) and the unlabeled SemanticKITTI and SemanticPOSS data (as the target data), and perform evaluations over the validation set of SemanticKITTI and SemanticPOSS. We follow the existing benchmarks [2] and adopt MinkNet as the segmentation model.

We adopt the self-training approach as described in section 3.3 for unsupervised domain adaptation. Specifically, we first train a supervised model with the labelled source data and apply the supervised model to predict pseudo labels for the unlabelled target data. We then apply PolarMix to cut and mix between the labelled source data and the pseudo-labelled target data, and further train the model with all augmented point data. As shown in Table 3.5, incorporating PolarMix achieves state-of-the-art mIoUs for both SemanticKITTI and SemanticPOSS. The superior segmentation performance is largely attributed to the cut-and-mix strategy in PolarMix which effectively mitigates the distribution discrepancy across LiDAR scans of two different domains.

3.4.4 Ablation Study

We perform several ablation studies to examine the contribution of the two data augmentation components in the proposed PolarMix. In the ablation studies, we

TABLE 3.6: Ablation study of PolarMix for semantic segmentation over SemanticKITTI dataset. SPVCNN is trained on sequence 00 and tested on the validation set.

Methods	mIoU
SPVCNN [6] (baseline)	48.9
w/ Scene-level swapping	50.8(+1.9)
w/ Intance-level pasting (simple-pasting)	50.9(+2.0)
w/ Intance-level pasting (rotate-pasting)	53.2(+4.3)
w/ PolarMix (complete)	54.8(+5.9)

train SPVCNN with the sequence 00 of SemanticKITTI and evaluate the trained models over the validation set of SemanticKITTI. We adopt the same training configurations as described in Section 3.4.1 and Table 3.6 shows experimental results. With the conventional global augmentation including random rotation and random scaling, the trained SPVCNN model achieves a mIoU of 48.9%. On top of that, including the proposed scene-level swapping alone improves the mIoU by 1.4%. In addition, including the basic version of the proposed instance-level cut-and-mix (i.e., without multiple rotations to create multiple copies of the cropped object instances) alone improves the mIoU by 2.0% while incorporating the full instance-level cut-and-mix improves the mIoU by 4.3%. Finally, incorporating both augmentation components (i.e., the full PolarMix) improves the mIoU by 5.9%, demonstrating the complementary property of the two approaches in point data augmentation.

3.4.5 Discussion

Mixing varies samples. We conducted experiments to examine whether mixing more than two LiDAR scans further improves the segmentation performances. Specifically, we increased the mixed LiDAR scans and benchmarking them without using PolarMix. The experiments were conducted with SPVCNN that is trained with sequence 00 of SemanticKITTI. As Table 3.7 shows, mixing two scans produces clearly the best performance. We examined the mixed data and found that mixing more scans introduces more hardly distinguishable objects. The experimental results are consistent with other mixing-based augmentation works [54, 56, 67, 134].

Limitations. As depicted in Table 3.1, the instance-level rotate pasting demonstrates its ability to enhance *foreground* classes exclusively, whereas scene-level

TABLE 3.7: Varying number of mixed scans. 'no mixing' represents the vanilla training without augmentation of PolarMix.

#Scans	no mixing (baseline)	2	3	4
mIoU	48.9	**54.8**	52.2	51.3

swapping generates novel layouts but falls short in significantly enriching the long-tailed classes within the *background*. Consequently, classes like *other ground* persist with notably low segmentation results.

3.5 Conclusion

This chapter presents PolarMix which is a data augmentation method for LiDAR point cloud learning. It produces new point cloud data by mixing LiDAR scans for training networks. There are two approaches that are designed based on LiDAR data properties in PolarMix. Specifically, the scene-level swapping exchanges points within the same range of azimuth angles while the instance-level rotating-pasting selects points of certain classes from one LiDAR scan and rotates for multiple copies before pasting into another scan. Extensive experiments show the superiority of our method in data augmentation for both semantic segmentation and object detection across a variety of deep frameworks and public datasets. We also extended PolarMix into unsupervised domain adaptation and achieved state-of-the-art performances in multiple synthetic-to-real LiDAR data segmentation benchmarks.

Chapter 4

Domain Transfer Learning from Synthetic to Real Point Clouds [1] [2]

In this chapter, we explore a representative scenario of domain transfer learning, specifically focusing on the transfer from synthetic point clouds to real point clouds. Our study encompasses one synthetic LiDAR point cloud dataset and two technical approaches.

More specifically, we first present a large-scale synthetic LiDAR point cloud dataset with point-wise annotations, bridging the gap in available datasets for benchmarking domain transfer learning in synthetic-to-real point cloud semantic segmentation.

Secondly, we explore a translation-based method to transfer synthetic point clouds to be more similar to real point clouds. Specifically, we decompose the synthetic-to-real gap into an appearance component and a sparsity component and handle them separately which improves the point cloud translation greatly. This approach effectively bridges the gap between synthetic and real domains, facilitating better knowledge transfer across domains.

[1] Part of the work in this chapter has been published at "Aoran Xiao, Jiaxing Huang, Dayan Guan, Fangneng Zhan, Shijian Lu. Transfer Learning from Synthetic to Real LiDAR Point Cloud for Semantic Segmentation. Proceedings of the AAAI Conference on Artificial Intelligence. 36, 3 (Jun. 2022), 2795-2803. DOI:https://doi.org/10.1609/aaai.v36i3.20183."

[2] Part of the work in this chapter has been published at "Aoran Xiao, Dayan Guan, Shijian Lu. Domain Adaptive LiDAR Point Cloud Segmentation with 3D Spatial Consistency. IEEE Transactions on Multimedia. DOI:10.1109/TMM.2023.3335879"

Thirdly, we investigate a consistency learning-based unsupervised learning objective for domain adaptation. Notably, point clouds from different domains often exhibit distinct distribution discrepancies due to variations in LiDAR sensor configurations, environmental conditions, occlusions, and other factors. To address this, we design a spatial consistency training framework that leverages three types of spatial consistency: geometric-transform consistency, sparsity consistency, and mixing consistency. These consistency measures capture the semantic invariance of point clouds with respect to viewpoint changes, sparsity changes, and local context changes, respectively. Ultimately, this framework enables the generation of domain-invariant feature representations from unlabelled target point clouds.

Further details of each technique will be presented in the subsequent sections.

4.1 Transfer Learning from Synthetic to Real LiDAR Point Clouds

4.1.1 Introduction

Semantic segmentation of LiDAR sequential point cloud is critical in various scene perception tasks and it has attracted increasing attention from both industry and academia [1, 6, 7, 47, 154] in recent years. However, training effective segmentation models requires large amounts of annotated point cloud, which are prohibitively time-consuming to collect due to the view change of 3D data and visual inconsistency between LiDAR point cloud and physical world. This can be observed by the small size of existing *real* LiDAR sequential datasets as listed in Table 4.1.

Inspired by the great success of transfer learning from synthetic to real data in two-dimensional (2D) field [155–157], one possible way to mitigate the data annotation constraint is to leverage synthetic point cloud data that can be collected and annotated automatically by computer engines. However, collecting large-scale synthetic LiDAR sequential point cloud is a nontrivial task which involves a large number of virtual scenes and objects as well as complicated point generation processes. In addition, most existing transfer learning methods [120, 150, 151, 158–160] focus on 2D images which do not work for 3D point cloud. To the best of our knowledge, few researches tackle the challenge of synthetic-to-real transfer of point cloud of

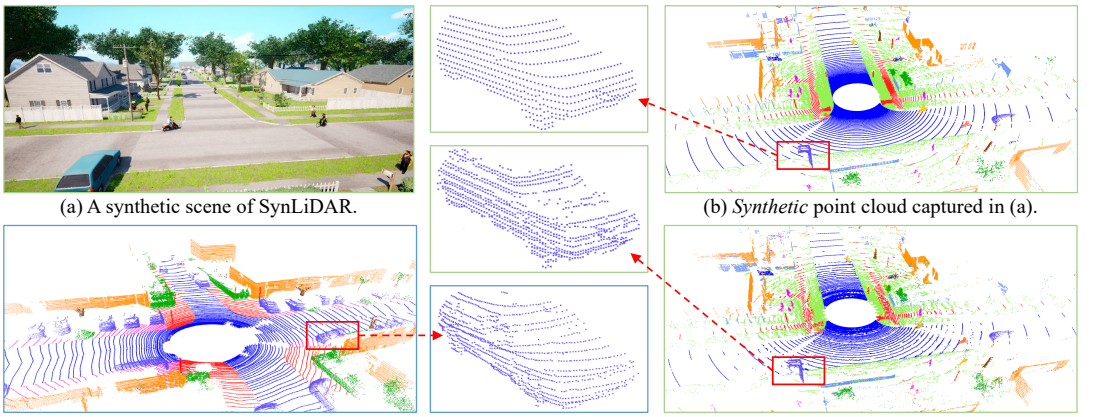


FIGURE 4.1: We create SynLiDAR, a large-scale multiple-class synthetic LiDAR point cloud dataset as illustrated in (b). SynLiDAR contains over 19 billion annotated points of 32 semantic classes which was collected by constructing multiple virtual environments and 3D object models as shown in (a). To make synthetic point cloud more useful for handling real-world LiDAR point cloud as shown in (c), we design a point cloud translator (PCT) that translates synthetic point cloud by decomposing the domain gap into an appearance component and a sparsity component. The translated data in (e) has a closer distribution as real point cloud and is more effective in processing real point cloud. The close-up views in (d) show the translation effects.

nature scenes, largely due to the lack of large-scale synthetic data with accurate geometries and rich semantic annotations.

We address the said issues by creating SynLiDAR, a large-scale LiDAR sequential point cloud dataset for facilitating the research of synthetic-to-real transfer of 3D point cloud data. We collected SynLiDAR from multiple virtual environments that were constructed by professional 3D generalists with advanced graphic tools. Each virtual environment contains configurable object models that are similar to real-world data in both geometry and layout. The dataset is ideal for the study of synthetic-to-real transfer as it consists of comprehensive and diverse point semantics (32 semantic classes with over 19 billion point-wise annotated points) and its collected points are also highly accurate in geometry.

In addition, we designed PCT, a point-cloud translator for mitigating domain gaps between synthetic and real point cloud as illustrated in Fig. 4.1. The design of PCT is inspired by the observations that point clouds can be viewed as discrete samplings of continuous 3D geometric environments where the domain gaps between synthetic and real point clouds come from either appearance differences (due to environments variations) or sparsity differences (due to sampling variation

by sensors). We hence disentangle the domain gap into an appearance component and a sparsity component, and design an appearance translation module (ATM) and a sparsity translation module (STM) to handle the two gap components separately. Specifically, ATM first up-samples synthetic point cloud and translates it to have similar appearance as real point cloud. STM then extracts sparsity features from real point cloud and fuses it with the ATM output to translate synthetic point cloud to have real sparsity. To the best of our knowledge, PCT is the first translation method for LiDAR point clouds in natural scenes.

The contribution of this work can be summarized in three aspects as listed:

- We create SynLiDAR, a large-scale synthetic LiDAR sequential point cloud dataset that has rich semantic classes and a large number of points with accurate point-wise annotations. SynLiDAR will lay a strong foundation for the study of the under-explored synthetic-to-real transfer in LiDAR point cloud segmentation.
- We examine the major underlying factors of the domain gap between synthetic and real point clouds, and design PCT, a pioneer LiDAR point cloud translator that can transform synthetic point clouds to have similar features and distributions as real point clouds and accordingly mitigate the domain gap effectively.
- We design three experimental setups for the study of synthetic-to-real point cloud transfer: data augmentation (DA), semi-supervised domain adaptation (SSDA) and unsupervised domain adaptations (UDA). We conducted extensive experiments under the three setups which will form valuable bases for the future investigation of synthetic-to-real point cloud transfer.

4.1.2 Related Works

4.1.2.1 Semantic Segmentation of LiDAR Point Clouds

LiDAR point clouds have been widely exploited in various autonomous navigation tasks for 3D scene understanding. This triggers several large-scale LiDAR point-cloud datasets [1, 2, 7, 147, 161] which greatly promote the research in 3D point

TABLE 4.1: Overview of outdoor LiDAR sequential point cloud datasets with semantic annotations: #scans: Number of scans for the datasets; #points: Number of points in millions (M); #classes: Number of semantic classes.

dataset	format	#scans	#points	#classes	annotation	type
SemanticKITTI [7]	point	43,552	4,549M	25	point-wise	real
SemanticPOSS [1]	point	2,988	216M	14	point-wise	real
nuScenes-Lidarseg [38]	point	40,000	1,400M	32	point-wise	real
GTA-LiDAR [162]	image	121,087	-	2	pixel-wise	synthetic
PreSIL [169]	point	51,074	3,135M	2	point-wise	synthetic
SynLiDAR (ours)	point	198,396	19,482M	32	point-wise	synthetic

cloud segmentation. Meanwhile, different deep architectures and learning algorithms have been proposed. One typical approach is to project 3D point clouds into 2D depth images and then adopt standard 2D convolution neural networks for segmentation [20, 154, 162–165]. This approach is efficient for processing large-scale point clouds but tends to lose geometric information in its 3D-to-2D mapping process. Another approach employs multilayer perceptron for point cloud representation learning [8, 27, 47] but is computationally intensive for large-scale point clouds. Beyond that, several studies [6, 10, 127, 166] quantize point clouds into discrete 3D grids and leverage 3D convolutions [167] or sparse convolutions [10, 29, 168] for learning and segmenting voxelized points. We follow [2, 4] and adopt the state-of-the-art MinkowskiNet [10] which is a sparse convolutional 3D point cloud segmentation network with a fine balance between accuracy and efficiency.

4.1.2.2 LiDAR Sequential Point Cloud Datasets

LiDAR *sequential* point clouds provide point cloud scans, each containing sparse and incomplete points collected in a sweep by LiDAR sensors. Several real world datasets, including SemanticKITTI [7], SemanticPOSS [1] and nuScenes-Lidarseg [38], have been proposed recently and promote the developments of LiDAR point cloud segmentation researches. However, labeling point-wise semantic annotations is prohibitively time-consuming for LiDAR sequential point clouds. Therefore, existing real point cloud datasets have very limited data sizes as listed in Table 4.1.

Inspired by the success of 2D synthetic image datasets [170], a few pioneer studies [162, 169] have explored to collect synthetic point cloud data from GTA-V games. However, 3D meshes in GTA-V games are not accurate and GTA-V games provide

only two object classes *Car* and *Pedestrian* [162]. Its collected synthetic data are thus insufficient for studying fine-grained LiDAR point cloud segmentation. We instead construct a wide range of realistic virtual environments and object models by leveraging graphic tools and professionals. The synthetic point clouds within the SynLiDAR thus capture much more accurate geometries and the rich diversity of semantic labels as in natural scenes.

4.1.2.3 Transfer Learning of Point Cloud

Transfer learning aims to transfer knowledge from the source domain to the target domain. It is an important tool to solve the inefficient training data problem [171]. This paper discusses three important transfer learning tasks of point cloud: DA combines multiple labeled datasets for training to reach better performances than training on each single one [52, 56]; SSDA exploits the knowledge from the source data with annotations and uses a certain number of unlabeled examples and a few labeled ones from the target domain to learn a target model; UDA instead uses annotated source data and target data without annotations to learn the target model [88, 99, 102]. Several pioneer works [108, 115, 172] have been proposed for the research of UDA problem in the LiDAR segmentation task. Instead, PCT mitigates domain gap problem in the input space and is effective for different kinds of transfer learning setups.

4.1.2.4 Domain Translation of Point Cloud

Domain Translation aims to learn meaningful mapping across domains. It is well developed for 2D images between paired domains [173], unpaired domains [174], multiple modalities [175], etc. For 3D data, some attempt has been reported for translation from images to depth [176], from point cloud to depth [177], from point cloud to images [178], etc. However, existing generative methods [179, 180] mainly focus on 3D objects while the translation between LiDAR point clouds in scenes is largely neglected. We address this challenge for mitigating the gap between synthetic and real point clouds.

4.1.3 The SynLiDAR Dataset

SynLiDAR is collected from multiple realistic virtual scenes constructed by professional 3D generalists using the Unreal Engine 4 platform [181] (as shown in Fig. 4.2. These virtual scenes include different types of outdoor environments such as cities, towns, harbour, etc, covering large areas of virtual areas. They are constituted by a large number of physically accurate object models that are produced by expert modelers with 3D-Max software, to ensure the high quality of synthetic data. Specifically, accurate coordinates and precise point-wise annotations of point cloud are collected automatically from these virtual environments. Note that SynLiDAR can be easily expanded by including new virtual scenes, 3D objects of new classes, etc.

Another important attribute of LiDAR point cloud is *intensity*, which is challenging to simulate due to the complicated signal transmission, propagation and reception processes in real environments [115]. In SynLiDAR, we address this issue by training a rendering model that learns from real LiDAR point cloud and predicts intensity values for SynLiDAR. Specifically, we train an intensity prediction model by using MinkowskiNet [182], where coordinates and semantic labels of cloud points in SemanticKITTI [7] are used as inputs and the point intensity (SemanticKITTI provided) is used as references.

SynLiDAR presents several advantages that position it as an optimal data source for 3D LiDAR segmentation, as compared with existing datasets: 1) **Extensive data size**: SynLiDAR encompasses 13 LiDAR point cloud sequences accompanied by meticulous point-wise annotations. It comprises a total of 198,396 scans of point clouds containing approximately 19 billion points. On average, each scan consists of around 98,000 points. 2) **Broad diversity**: This dataset offers precise point-wise annotations for 32 semantic classes, facilitating fine-grained scene comprehension. 3) **Accurate labels**: In contrast to human annotations, which may involve inherent errors, SynLiDAR generates labels automatically, ensuring absolute accuracy. These inherent properties establish SynLiDAR as the largest LiDAR point cloud dataset for road scenes dedicated to semantic segmentation. Table 4.1 illustrates how SynLiDAR surpasses existing LiDAR point cloud datasets in terms of both point count and semantic classes. Additionally, Figure 4.3 compares the point numbers for categories of 'thing' (countable foreground classes) and 'stuff' (uncountable background classes) in SynLiDAR and SemanticKITTI (the largest

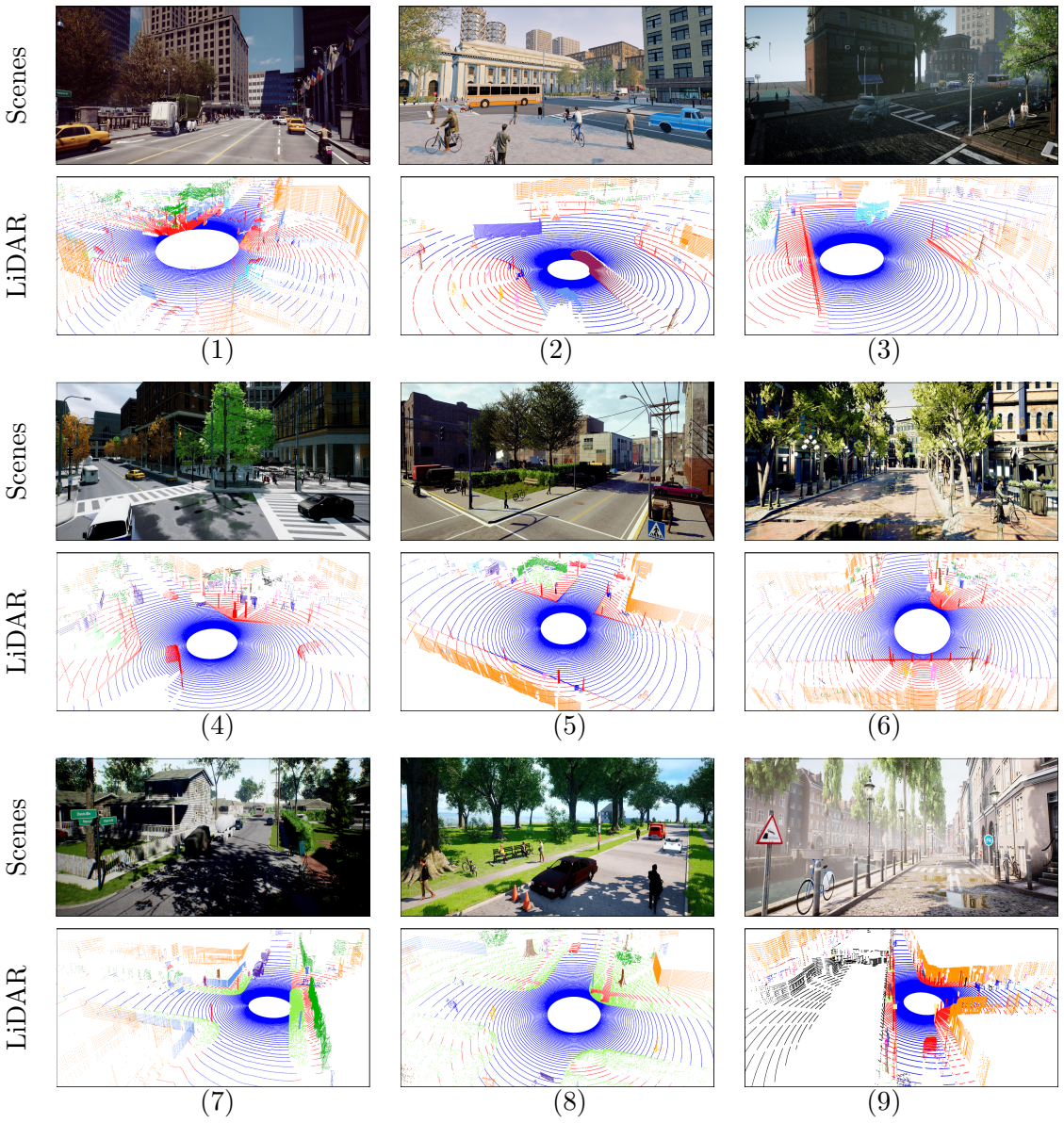


FIGURE 4.2: SynLiDAR is collected from nine virtual scenes: (1), (2), (3), (4) are virtual cities; (5) and (6) are virtual suburban towns; (7) and (8) are virtual neighborhood environments; and (9) is a virtual harbour. The *Scenes* show example images of constructed scenes and the *LiDAR* shows the corresponding LiDAR point cloud scans colored by semantic annotations.

known real point cloud dataset). This comparison underscores that SynLiDAR stands as a truly 'large-scale' point cloud dataset, making it an optimal choice for research into transfer learning involving synthetic-to-real point clouds.

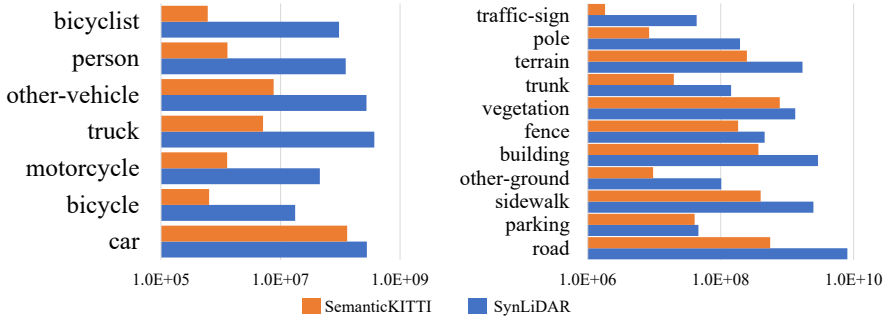


FIGURE 4.3: The numbers of annotated points (x-axis) per class (y-axis) for SemanticKITTI and SynLiDAR. Left: *Thing* classes; Right:*Stuff* classes.

4.1.4 Point Cloud Translation

Similar to most synthetic data, point clouds in SynLiDAR have a clear domain gap with real LiDAR data, and the model trained using SynLiDAR usually experiences clear performance drops while applied to real point clouds. This paper proposes PCT, the first translator of LiDAR data to mitigate the domain gap for the challenging scene semantic segmentation task, as illustrated in this section.

An intuitive idea to mitigate the domain gap is to employ existing 3D generative models [179, 183, 184] to translate synthetic data to have real data distributions. However, these models are designed for 3D objects with uniformly distributed points, and standard supervisions like Chamfer loss or Earth Mover's Distance (EMD) [185] fail to regularize LiDAR data distribution of 3D scenes directly. As a result, few studies have attempted to address the problem of LiDAR point cloud translation.

We observed that point clouds are discrete samplings of the continuous geometric world, hence the synthetic-to-real gaps could be disentangled into two components: The appearance component reflects the differences between synthetic and real continuous environments, and the sparsity component shows differences in sampling patterns introduced by different LiDAR sensors. The proposed *PCT* mitigates these two components separately and its translated synthetic point clouds have both similar appearance and sparsity as targeting real-world data.

The pipeline of PCT: Given synthetic point clouds $X_s \in \mathbb{R}^{N_s \times 3}$ and real point clouds $X_r \in \mathbb{R}^{N_r \times 3}$, we aim to translate X_s into $\hat{X}_{s \to r} \in \mathbb{R}^{\hat{N}_s \times 3}$ that have similar appearance and sparsity as X_r. Firstly, the ATM up-samples X_s into $U(X_s)$,

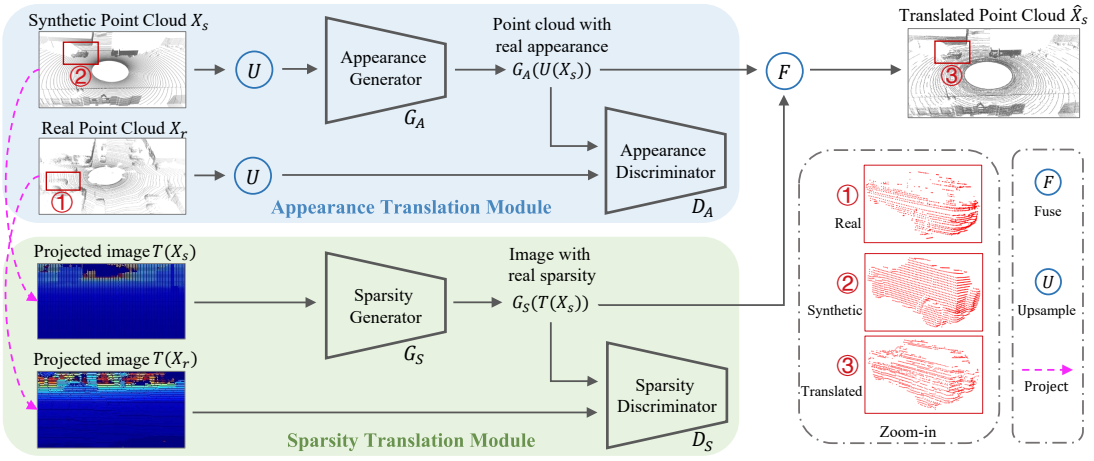


FIGURE 4.4: The proposed PCT disentangles point-cloud translation into appearance translation and sparsity translation tasks. Given synthetic point cloud, the appearance translation first learns to reconstruct dense point clouds that have similar appearance as real point clouds. The sparsity translation then learns real sparsity distribution in 2D space and fuses it with the reconstructed point cloud in 3D space. The final translation has similar appearance and sparsity as real point cloud as illustrated.

and translates it to have certain real appearance by a generator G_A as $X'_s = G_A(U(X_s))$. Then the STM projects (T) point cloud into images and extracts sparsity features of real point cloud into synthetic data by another generator G_S, *i.e.* $X''_s = G_S(T(X_s))$. Finally, translated point cloud is generated by fusing X'_s and X''_s as $\hat{X}_{s\to r} = F(X'_s, X''_s)$. More details of the two translation modules are provided in the ensuing two subsections.

ATM aims to translate synthetic point clouds to have real appearance and we realize it through a 3D generative adversarial network that consists of a generator G_A and a discriminator D_A, and train them in an adversarial manner. In this stage, we first up-sample both synthetic and real point cloud so as to eliminate the effect of domain-specific sparsity features. The generator aims to produce intermediate representation X'_s (from the up-sampled synthetic data) to have real appearance to fool the discriminator, while the discriminator learns to distinguish X'_s from $U(X_r)$. Specifically, The adversarial learning loss can be formulated as follows:

$$\begin{aligned} L_A^{adv} = &\log(D_A(-G_A(U(X_s))\log(G_A(U(X_s)))))+ \\ &\log(1 - D_A(U(X_r)\log(U(X_r)))) \end{aligned} \tag{4.1}$$

We introduce EMD for generator to keep geometries of X_s and X'_s

$$L_A^{emd}(X_s, X'_s) = \sum_{x \in X_s} ||x - \phi(x)||_2 \tag{4.2}$$

where $\phi : X_s \rightarrow X'_s$ is a bijection. The objective function of the ATM can be formulated as:

$$L_A(G_A, D_A) = \arg\min_{G_A} \max_{D_A} (\lambda_A^{adv} L_A^{adv} + \lambda_A^{emd} L_A^{emd}) \tag{4.3}$$

STM aims to transfer sparsity information from real point cloud to synthetic point cloud. However, existing supervisions [185] such as Chamfer loss and EMD loss cannot capture sparsity information well as they lack sparsity regulation. To address this problem, we propose to first learn sparsity information in 2D space and then fuse it back into 3D space. Specifically, we first project X_s and X_r into depth images (as $T(X_s)$ and $T(X_r)$ respectively) through spherical projection [20], and then employ an image-to-image translation model to translate $T(X_s)$ to have similar sparsity as $T(X_r)$. The widely-used spherical projection [20, 115, 154, 162] transfers 3D point cloud into 2D images through following equation. The pixel coordinate (u, v) of a point with spatial coordinate (x, y, z) in a single scan of point cloud could be calculated as:

$$\begin{cases} u = \frac{1}{2}[1 - \arctan(y, x)\pi^{-1}]W \\ v = [1 - (\arcsin(z, r^{-1}) + f_{up})f^{-1}]H \end{cases} \tag{4.4}$$

where r is depth ($r = \sqrt{x^2 + y^2 + z^2}$); f_{down} and f_{up} represent low and up bound of vertical field-of-view of LiDAR sensor ($f = f_{up} + f_{down}$); (H, W) are height and width of projected images. The translation network is also GAN-based with a generator G_S and a discriminator D_S. The adversarial learning loss can be formulated as follows

$$\begin{aligned} L_S^{adv} = &\log(D_S(-G_S(T(X_s))\log(G_S(T(X_s))))) \\ &+ \log(1 - D_S(T(X_r)\log(T(X_r)))) \end{aligned} \tag{4.5}$$

To preserve the geometry information during the translation, we further include a geometry consistency loss to ensure that the translated depth image preserves

similar geometry as the original image:

$$L_S^{geo} = ||\overline{T(X_s)} - \overline{G_S(T(X_s))}||_2 + ||\overline{T(X_r)} - \overline{G_S(T(X_r))}||_2 \tag{4.6}$$

where $\overline{A} - \overline{B}$ means that the distance is computed for pixels existing in both A and B only. The overall objective of the sparsity translation module can be formulated by:

$$L_S(G_S, D_S) = \arg\min_{G_S} \max_{D_S} (\lambda_S^{adv} L_S^{adv} + \lambda_S^{geo} L_S^{geo}) \tag{4.7}$$

Finally, the translated image $G_S(T(X_s))$ with real sparsity information are projected back into 3D space for fusion. Specifically, $G_S(T(X_s))$ are used for guidance to drop out points in X'_s. The semantic labels of the translated point cloud are assigned according to labels of neighboring points in the original point cloud.

When training PCT, we first pre-trained ATM following the same procedures as described in [179]. This pre-trained model is also used for up-sampling real point cloud. Then we fine-tuned this model by inputting synthetic point cloud for the generator and modified the discriminator to distinguish the generated point cloud and up-sampled real point cloud. The appearance translation is conducted by semantic classes. We trained the network for 100k iterations by using the Adam algorithm. The learning rates of the appearance generator and appearance discriminator are set as 1e-4, respectively. In STM, we spherically projected both SynLiDAR and real point clouds into depth images to learn sparsity information in 2D spaces. The projected image size is 2048×128. We adopted the cycle-consistency [174] for the translation. We trained the network for 200 epochs by using the Adam algorithm. The learning rate is 0.0002 for both the sparsity generator and sparsity discriminator.

4.1.5 Experiments

We evaluate how SynLiDAR benefits semantic segmentation over multiple real point cloud datasets and how PCT mitigates domain gaps between SynLiDAR and real point cloud data. We conducted experiments on three transfer setups including DA, SSDA, and UDA as introduced in section 2.3. All experiments were conducted by using state-of-the-art 3D semantic segmentation network MinkowskiNet [182].

4.1.5.1 Datasets and Implementation Details

We conduct experiments over two real-world point cloud datasets. The first is *SemanticKITTI* [7] that consists of 43,552 scans of annotated sequential LiDAR point cloud with 19 semantic classes. It is the largest real-world sequential LiDAR point cloud dataset for semantic segmentation to the best of our knowledge. We follow the commonly-used protocol that splits sequences 00-07, 09-10 for training and sequence 08 for validation. The second is *SemanticPOSS* [1] which is collected in a university campus. It consists of 2,988 annotated point cloud scans of 14 semantic classes. We follow the benchmark setting by using sequence 03 for validation and the rest for training. We *ignore* extra classes of SynLiDAR by mapping them as 'unlabeled' for each real dataset. Mean Intersection of Union (mIoU) is used as the evaluation metric.

For point cloud translation, parameters λ_A^{adv}, λ_A^{emd}, λ_S^{adv}, λ_S^{geo} are set at 0.01, 1, 5, 1, respectively. For semantic segmentation by MinkowskiNet, the maximum point number of each scan is 80,000 for SemanticKITTI and 50,000 for SemanticPOSS; The voxel size is 0.05 and we use coordinates and intensity of point cloud as input features.

4.1.5.2 Experiments on Data Augmentation

We first evaluate how SynLiDAR augments real point cloud data as compared with state-of-the-art data augmentation methods as shown in Tables 4.2 and 4.3. We can observe that incorporating SynLiDAR helps train better models for both SemanticKITTI (+2.2%) and SemanticPOSS (+2.6%). It also shows that SynLiDAR shares good similarity with real-world dataset and provides a high-quality data source for transfer learning of LiDAR point cloud.

We also evaluate how PCT mitigates the domain gap and improves data augmentation. Experiments show that including PCT-translated SynLiDAR in training improves the mIoU by 2.2% and 2.6%, respectively, for SemanticKITTI and SemanticPOSS. The mIoU improvements clearly demonstrate the effectiveness of PCT in mitigating domain gap between SynLiDAR and the two real datasets.

TABLE 4.2: Data Augmentation experiments on SemanticKITTI: Combining the training data of SynLiDAR and SemanticKITTI trains more accurate semantic segmentation models. PCT mitigates the domain gap effectively and combining the PCT-translated SynLiDAR with SemanticKITTI further improves the segmentation.

method	car	bi.cle	mt.cle	truck	oth-v.	pers.	bi.clst	mt.clst	road	parki.	sidew.	oth-g.	build.	fence	veget.	trunk	terra.	pole	traf.	mIoU
baseline	95.7	25.0	57.0	62.1	46.4	63.4	77.3	0.0	93.0	47.9	80.5	2.2	89.7	58.6	89.5	66.5	78.0	64.6	50.1	60.3
Jittering [8]	95.7	27.8	56.2	66.0	45.8	65.3	82.8	0.0	93.0	48.2	79.9	2.5	89.7	62.9	88.9	64.0	77.0	64.8	51.0	61.2
Dropout [186]	96.0	28.5	57.1	65.1	46.4	64.1	83.6	0.1	93.5	47.6	80.1	2.3	89.3	61.9	90.1	66.9	78.8	65.8	49.1	61.4
PointAug [52]	95.9	29.2	70.0	76.3	50.0	67.0	84.4	2.4	93.8	48.1	81.2	4.6	89.8	58.4	87.5	65.4	72.7	62.4	50.5	62.6
+SynLiDAR	95.9	33.0	62.8	78.9	50.2	71.4	83.5	0.7	92.3	52.8	79.9	0.1	89.8	59.5	86.3	65.4	72.8	63.6	48.9	62.5
PCT	96.3	38.7	73.4	82.9	56.1	71.1	85.3	1.6	94.1	54.3	81.6	1.3	89.5	59.6	87.8	66.9	73.6	65.4	50.5	64.7

TABLE 4.3: Data Augmentation experiments on SemanticPOSS: Combining the training data of SynLiDAR and SemanticPOSS trains more accurate semantic segmentation models. PCT mitigates the domain gap effectively and combining the PCT-translated SynLiDAR with SemanticPOSS further improves the segmentation.

method	pers.	rider	car	trunk	plants	traf.	pole	garb.	buil.	cone.	fence	bike	grou.	mIoU
baseline	55.6	45.1	66.9	44.4	73.9	45.4	41.6	14.5	76.1	7.9	57.0	54.1	75.3	50.6
Jittering [8]	55.2	48.7	65.1	45.5	75.2	45.9	40.9	18.1	76.4	15.1	57.1	56.4	75.0	51.9
Dropout [186]	56.9	56.7	67.8	43.3	75.6	40.3	30.7	26.8	75.7	17.7	57.3	55.6	78.6	52.5
PointAug [52]	62.3	60.7	69.6	39.3	76.0	41.4	33.8	24.1	78.0	13.7	62.2	56.5	79.2	53.6
+SynLiDAR	57.6	59.3	61.1	37.1	76.1	32.7	40.9	34.7	72.7	37.7	57.6	43.3	81.2	53.2
PCT	57.3	61.4	65.8	36.2	77.4	42.5	42.1	49.2	74.5	32.4	55.8	48.9	81.8	55.8

We further evaluate by combining SynLiDAR and PCT-translated SynLiDAR with different portions of SemanticPOSS, aiming to examine how SynLiDAR could alleviate data collection and annotation efforts. Fig. 4.5 shows experiment results. We can see that incorporating SynLiDAR consistently improves the segmentation under different portions of SemanticPOSS, and it can save up to 40% SemanticPOSS without sacrificing segmentation performance. In addition, including the PCT-translated SynLiDAR further improves the segmentation consistently under similar setups.

We next examine how the appearance component and sparsity component compose the domain gap between synthetic and real point clouds and investigate the effectiveness of each module in PCT. We conducted experiments in the data augmentation (DA) setup, *i.e.* incorporating translated SynLiDAR by different modules of PCT into SemanticPOSS for training segmentation models. We followed the same settings and hyper-parameters as DA experiments in Table 4.3. The result is shown in Table 4.4. Joint training with SemanticPOSS and raw SynLiDAR

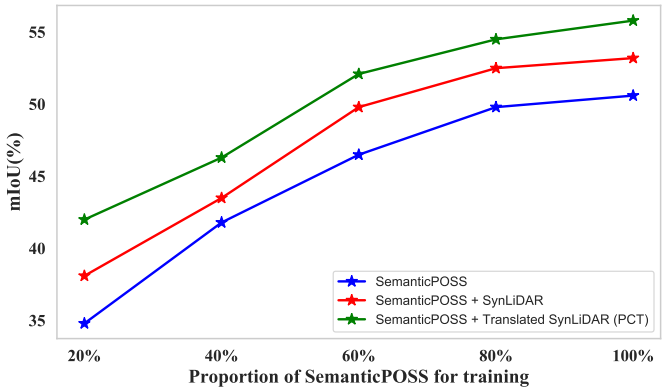


FIGURE 4.5: SynLiDAR can effectively augment real-world LiDAR point cloud (SemanticPOSS) in a point cloud segmentation task. The PCT-translated SynLiDAR further improves the augmentation consistently by large margins.

reached 53.2% mIoU as the baseline performance. STM directly samples points in the up-sampled synthetic data without appearance translation, and training its translated SynLiDAR with SemanticPOSS improved mIoU by 1.7% as compared with baseline. On top of it, we further incorporate ATM for a complete PCT that translates SynLiDAR to have both similar appearance and sparsity as real data. Incorporating its generated point cloud in training improved another 0.9% mIoU for segmentation by reaching 55.8% mIoU. The experiment results indicate the effectiveness of PCT to mitigate the domain gap by handling two components separately.

TABLE 4.4: Ablation study of two translation modules in PCT: Baseline denotes joint training with SemanticPOSS and raw SynLiDAR. STM replaces raw SynLiDAR data by its generation. PCT uses both STM and ATM in translation.

Model	mIoU
Baseline	53.2
Baseline+STM	54.9
Baseline+PCT	55.8

4.1.5.3 Experiments on SSDA

In this subsection we evaluate the effectiveness of PCT in another setup of semi-supervised domain adaptation (SSDA) with SynLiDAR (as source) and real datasets (as target). We follow the setting of [159] with three parts of training data, *i.e.* labeled source samples, unlabeled target samples and 1 labeled target sample that are randomly selected for 1-shot SSDA-based semantic segmentation.

TABLE 4.5: Experiments on semi-supervised domain adaptation with SynLiDAR (as source) and SemanticKITTI (as target): PCT translates SynLiDAR and mitigates domain gaps in the input space effectively. It is complementary to APE and combining them outperforms the baseline SynLiDAR + SemanticKITTI (*i.e.*, S+T) by large margins.

Method	car	bi.cle	mt.cle	truck	oth-v.	pers.	bi.clst	mt.clst	road	parki.	sidew.	oth-g.	build.	fence	veget.	trunk	terra.	pole	traf.	mIoU
S+T	56.2	3.0	15.1	1.0	5.0	20.2	42.1	2.8	52.1	0.7	19.8	0.0	41.3	5.8	62.1	34.0	42.0	24.6	1.4	22.6
MMD [187]	56.4	3.3	13.3	1.5	6.1	21.4	34.6	1.6	54.3	0.4	21.4	0.0	50.2	5.8	61.2	37.0	44.9	31.6	2.2	23.5
MME [188]	51.0	5.6	13.1	1.3	7.3	15.1	54.4	4.4	43.1	0.2	28.3	0.0	60.7	13.3	66.1	30.1	39.9	24.8	6.6	24.5
APE [189]	58.6	6.2	16.6	3.1	11.3	14.2	35.8	3.7	61.5	1.7	30.3	0.0	54.7	15.4	64.6	20.0	45.5	23.9	9.1	25.1
PCT	56.0	7.0	17.1	2.8	9.9	23.7	43.7	5.6	55.3	0.8	22.9	0.0	50.1	8.4	65.3	23.1	43.5	28.8	7.5	24.8
APE + PCT	58.1	7.3	17.8	2.6	13.9	24.7	46.5	5.1	60.5	1.9	31.3	0.0	56.8	14.6	67.9	23.7	44.3	26.1	9.3	27.0

TABLE 4.6: Experiments on semi-supervised domain adaptation with SynLiDAR (as source) and SemanticPOSS (as target): PCT translates SynLiDAR and mitigates domain gaps in the input space effectively. It is complementary to APE and combining them outperforms the baseline SynLiDAR + SemanticPOSS (*i.e.*, S+T) by large margins.

Method	pers.	rider	car	trunk	plants	traf.	pole	garb.	buil.	cone.	fence	bike	grou.	mIoU
S+T	25.2	36.1	18.2	12.8	58.6	1.7	30.5	5.6	25.7	3.0	12.0	10.6	75.6	24.3
MMD [187]	25.5	35.7	28.9	6.7	64.3	1.7	23.2	5.6	53.3	3.3	30.2	13.9	70.4	27.9
MME [188]	33.2	40.2	25.0	11.0	61.9	0.4	31.2	7.3	56.1	5.7	37.1	6.7	71.2	29.8
APE [189]	34.3	40.1	21.5	16.3	62.6	0.9	31.1	2.3	55.9	13.3	34.3	9.6	71.6	30.3
PCT	25.8	36.8	27.8	11.3	62.2	1.9	31.2	5.2	58.7	2.6	34.3	8.5	68.7	28.8
APE + PCT	34.7	36.3	27.2	15.8	62.9	0.8	31.6	8.7	62.3	9.8	35.1	9.3	70.9	31.2

Tables 4.5 and 4.6 show experimental results of PCT and other state-of-the-art SSDA methods. As we can see, training labeled SynLiDAR and one-shot real sample with supervised loss (S+T) does not perform well for both two real datasets due to the domain gap. Including PCT-translated SynLiDAR in training improved the mIoU by 2.2% and 4.5% for SemanticKITTI and SemanticPOSS, respectively. Since PCT mitigates the domain gap in the input space while the state-of-the-art method APE [189] does in the feature space, these two methods are complementary and combining them reached new state-of-the-art performances at 27.0% and 31.2%, respectively.

4.1.5.4 Experiments on UDA

In this subsection, we focus on unsupervised domain adaptation (UDA) for point cloud segmentation. Different from SSDA, in this setup we only use labeled source

TABLE 4.7: Experiments on unsupervised domain adaptation with SynLiDAR (as source) and SemanticKITTI (as target): PCT translates SynLiDAR and mitigates domain gaps in the input space effectively. It complements ST and combining them outperforms the baseline (*i.e.*, source-only) by large margins.

Method	car	bi.cle	mt.cle	truck	oth-v.	pers.	bi.clst	mt.clst	road	parki.	sidew.	oth-g.	build.	fence	veget.	trunk	terra.	pole	traf.	mIoU
Source-Only	42.0	5.0	4.8	0.4	2.5	12.4	43.3	1.8	48.7	4.5	31.0	0.0	18.6	11.5	60.2	30.0	48.3	19.3	3.0	20.4
ADDA [190]	52.5	4.5	11.9	0.3	3.9	9.4	27.9	0.5	52.8	4.9	27.4	0.0	61.0	17.0	57.4	34.5	42.9	23.2	4.5	22.8
Ent-Min [149]	58.3	5.1	14.3	0.6	1.8	14.3	44.5	0.5	50.4	4.3	34.8	0.0	48.3	19.7	67.5	34.8	52.0	33.0	6.1	25.5
ST [191]	62.0	5.0	12.4	1.3	9.2	16.7	44.2	0.4	53.0	2.5	28.4	0.0	57.1	18.7	69.8	35.0	48.7	32.5	6.9	26.5
PCT	53.4	5.4	7.4	0.8	10.9	12.0	43.2	0.3	50.8	3.7	29.4	0.0	48.0	10.4	68.2	33.1	40.0	29.5	6.9	23.9
ST+PCT	70.8	7.3	13.1	1.9	8.4	12.6	44.0	0.6	56.4	4.5	31.8	0.0	66.7	23.7	73.3	34.6	48.4	39.4	11.7	28.9

TABLE 4.8: Experiments on unsupervised domain adaptation with SynLiDAR (as source) and SemanticPOSS (as target): PCT translates SynLiDAR and mitigates domain gaps in the input space effectively. It complements ST and combining them outperforms the baseline (*i.e.*, source-only) by large margins.

Method	pers.	rider	car	trunk	plants	traf.	pole	garb.	buil.	cone.	fence	bike	grou.	mIoU
Source-Only	3.7	25.1	12.0	10.8	53.4	0.0	19.4	12.9	49.1	3.1	20.3	0.0	59.6	20.1
ADDA [190]	27.5	35.1	18.8	12.4	53.4	2.8	27.0	12.2	64.7	1.3	6.3	6.8	55.3	24.9
Ent-Min [149]	24.2	32.2	21.4	18.9	61.0	2.5	36.3	8.3	56.7	3.1	5.3	4.8	57.1	25.5
ST [191]	23.5	31.8	22.0	18.9	63.2	1.9	41.6	13.5	58.2	1.0	9.1	6.8	60.3	27.1
PCT	13.0	35.4	13.7	10.2	53.1	1.4	23.8	12.7	52.9	0.8	13.7	1.1	66.2	22.9
ST + PCT	28.9	34.8	27.8	18.6	63.7	4.9	41.0	16.6	64.1	1.6	12.1	6.6	63.9	29.6

data (SynLiDAR) and unlabeled target data (real datasets) for training.

As we can see from Tables 4.7 and 4.8, training on labeled source data as source-only failed to learn satisfactory segmentation models due to presence of the domain gap. State-of-the-art UDA methods mitigate the domain gap in either feature space (ADDA) or output space (Ent-Min, ST) and improved the segmentation performance effectively. On the other hand, PCT mitigates the domain gap in input space and including its translated SynLiDAR improved mIoU by 3.5% and 2.8% for SemanticKITTI and SemanticPOSS, respectively. It also complements ST and their combination achieved new state-of-the-art performances at 28.9% and 29.6%, respectively. The experiment results further indicate that PCT effectively reduced the domain gap between SynLiDAR and two real datasets.

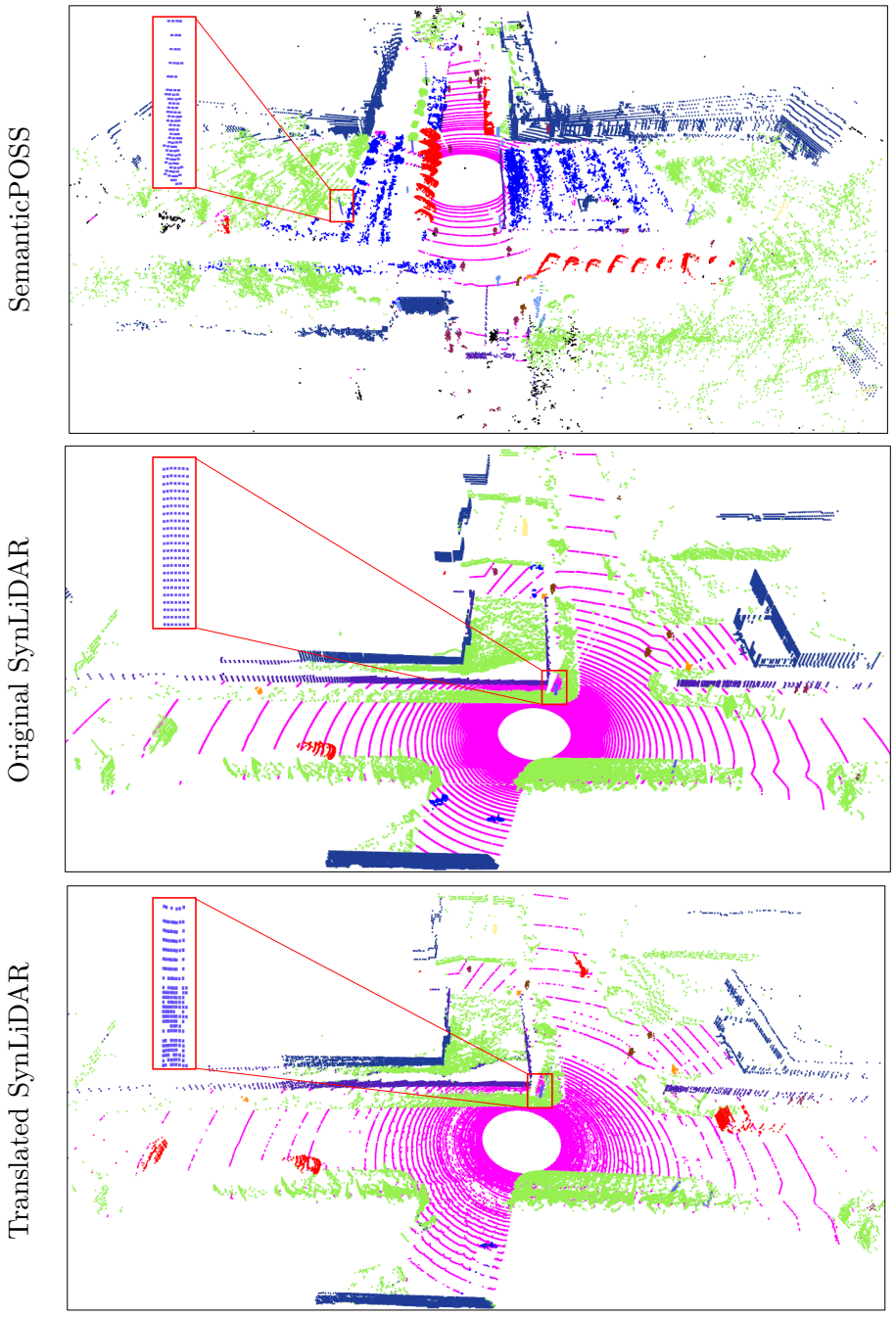


FIGURE 4.6: An example of PCT translating point cloud of SynLiDAR to SemanticPOSS. The close-up views show examples of *pole*, where SemanticPOSS points are sparse in the upper part and dense in the lower part whereas SynLiDAR points are evenly distributed. PCT translates SynLiDAR points to have similar real-world geometry and sparsity properly.

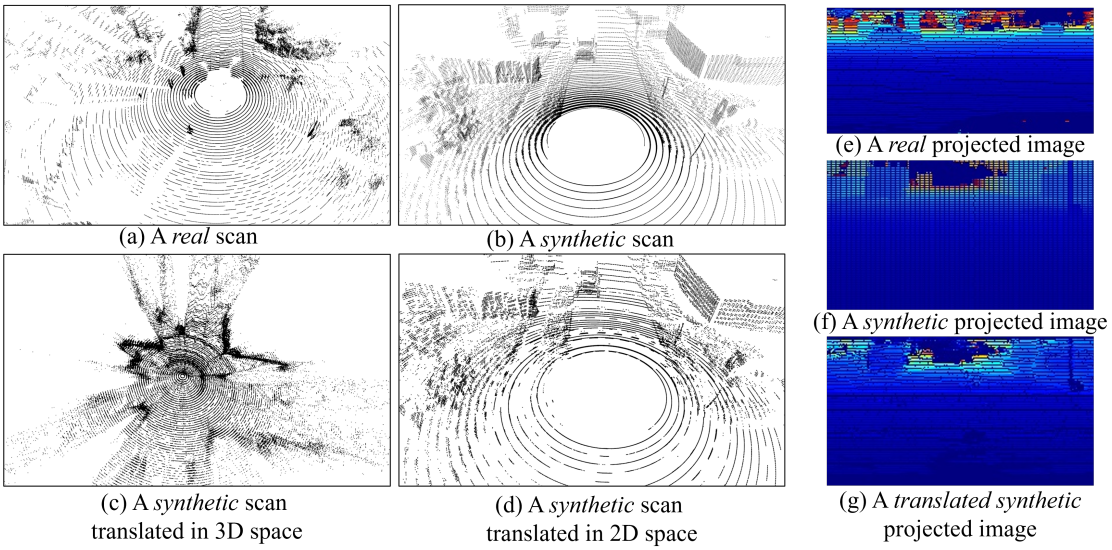


FIGURE 4.7: Translation results of different sparsity generator.

4.1.5.5 Discussion

Qualitative illustration of the proposed PCT. We provide more illustrations for translating point cloud from SynLiDAR to SemanticPOSS [1] in Fig. 4.6. It can be seen that the point appearance and sparsity are properly translated from SynLiDAR's very regular patterns to SemanticPOSS's noisy and real-world patterns. This can also be verified by the close-up view of a *pole* where SemanticPOSS points are sparse in the upper part and dense in the lower part whereas SynLiDAR points are evenly distributed. PCT translates SynLiDAR points to have similar real-world geometry and sparsity properly.

Sparsity translation in 2D space vs. 3D space, Our empirical experiments, conducted with a direct translation in the 3D space, proved unsuccessful in producing point clouds exhibiting true sparsity characteristics. This is because, as we state in Section 4.1.4, existing supervisions such as Chamfer loss and EMD loss cannot capture sparsity information well as they lack sparsity regulation. Consequently, we shifted our approach to a 2D solution, which yielded notably superior translation effects in our empirical evaluations. Figure 4.7 showcases visual illustrations of 2D and 3D translation solutions. Fig. (a) and (b) depict real and synthetic LiDAR point cloud scans, respectively. Meanwhile, Fig. (c) and (d) display the translation results of (b) using 3D and 2D translation methods, respectively. Figs. (e), (f), and (g) showcase translation results of projected images by the sparsity generator.

4.1.6 Conclusion

In this section, we present SynLiDAR and a point cloud translation method PCT for synthetic-to-real transfer learning. SynLiDAR is a large-scale synthetic LiDAR sequential point cloud dataset that contains 19 billion points with point-wise annotations of 32 semantic classes. PCT translates synthetic point clouds to have similar appearance and sparsity as real point clouds. Extensive experiments showed that SynLiDAR shares high geometrical similarities with real-world data, which effectively boosts semantic segmentation while combining with different proportions of real data. PCT mitigates synthetic-to-real gaps effectively and its translated data further improves point cloud segmentation consistently in three transfer learning setups.

4.2 Domain Adaptive 3D LiDAR Segmentation with Spatial Consistency

4.2.1 Introduction

Semantic segmentation of 3D LiDAR point clouds is critical in different computer vision tasks such as autonomous driving, remote sensing, robotics, etc. It has achieved great progress thanks to the recent advances in deep neural networks (DNNs). Nevertheless, effective DNN training usually requires large-scale densely annotated point clouds which are extremely laborious to collect. One approach that could alleviate the annotation constraint is to leverage synthetic point clouds that often come with automatically generated labels [2]. However, synthetic point clouds exhibit clear distribution discrepancies as compared with real point clouds [2, 4], and DNN models trained by using synthetic point clouds often experience clear performance drops while applied to real point clouds.

Unsupervised domain adaptation (UDA) can mitigate the distribution discrepancy between a labelled source domain and an unlabelled target domain, which has recently attracted increasing attention for the task of 3D LiDAR point cloud segmentation. Domain adaptive point cloud segmentation has been investigated in three major approaches: 1) self-training that selects confident target predictions

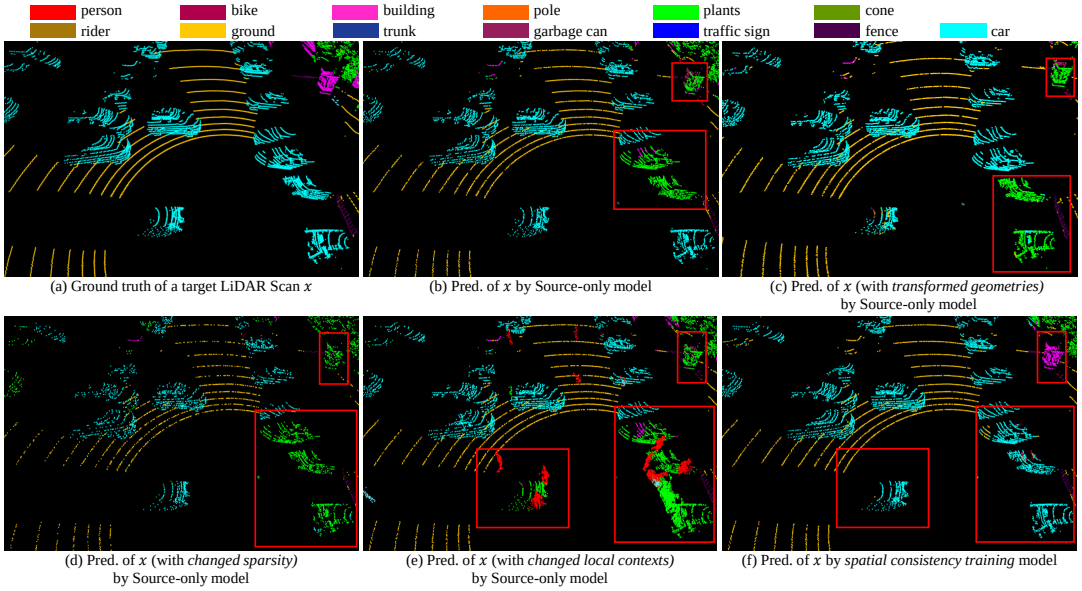


FIGURE 4.8: Spatial consistency training helps in domain adaptive LiDAR segmentation: (a) shows the ground-truth segmentation of one target LiDAR scan from SemanticPOSS [1] and the rest shows its segmentation by different models. Specifically, (b) shows the segmentation by a "Source-only" model trained with the source data (i.e., synthetic point clouds in SynLiDAR [2]) whose performance degrades clearly while applied to the target scan from a different domain. The performance degradation exacerbates when the target scan suffers from spatial perturbations such as geometric transformation (rotation, scaling, and flipping) in (c), sparsity changes (point down-sampling) in (d), and local context changes (i.e., the presence of *persons* (in red) around *cars*) in (e). Our *Spatial Consistency Training* exploits the inherent nature of LiDAR point clouds and semantic invariance for spatial perturbations to regularize the domain adaptation process, leading to clearly improved segmentation of the target scan as in (f). The red boxes highlight areas with substantial performance disparities, and LiDAR views in all subfigures are aligned for easy comparison. Best viewed in color.

as pseudo-labels for network training [4, 68]; 2) cross-domain point cloud translation [2, 108]; and 3) projecting point clouds into 2D space for adaptation [115, 117]. Meanwhile, consistency training [192, 193] has recently emerged as an effective UDA technique in various 2D image recognition tasks. It learns robust and generalizable target representations effectively by enforcing a model's output to remain consistent under the presence of input perturbations.

We investigate the efficacy of consistency training for domain adaptive semantic segmentation of 3D LiDAR point clouds. The primary challenge lies in designing effective consistency strategies that can facilitate the learning of domain adaptive representations for segmenting target point clouds in an unsupervised manner.

Intuitively, the semantics of point clouds should remain invariant under the presence of variations in sensor viewpoints, point sampling density, and local context. However, we observe that the predictions of deep models are highly susceptible to the above variations. This can be observed in Figs.4.8 (c), (d), and (e), where the inter-domain prediction errors (while applying a source-trained model to target point clouds as illustrated in Fig.4.8 (b)) are exacerbated in different manners when the input point clouds undergo changes in sensor viewpoints, sampling sparsity, and local context, respectively.

Inspired by the above observations, we design SCT, a simple yet effective spatial consistency training framework that can learn effective domain adaptive point cloud representations. SCT introduces spatial perturbations to mimic the aforementioned variation factors and learns by enforcing the prediction of spatially perturbed point clouds to be consistent with that of the original point clouds. We design three types of spatial perturbations: 1) *Geometric transform* that simulates the viewpoint change of LiDAR sensors; 2) *Sparsity variation* that down-samples input point clouds; and 3) *Mixing* that modifies the local context of input point clouds. With the three types of spatial perturbations, we formulate three types of spatial consistency including *geometric-transform consistency*, *sparsity consistency*, and *mixing consistency* which are well tailored to the spatial characteristics of point clouds. With a mean teacher learning strategy [3], a simple implementation of the three types of spatial consistency outperforms the state-of-the-art with significant margins as illustrated in Fig. 4.8 (f).

In summary, the contributions of this work are threefold. *First*, we identify that spatial perturbations including geometric transformation, sparsity changes, and local context changes can clearly degrade the cross-domain LiDAR point cloud segmentation. To this end, we design three types of spatial consistency learning strategies tailored for LiDAR point clouds, which help learn domain adaptive representations and enhance unsupervised cross-domain transfer of LiDAR point clouds greatly. *Second*, our proposed spatial consistency training framework, characterized by its elegant simplicity, exceptional effectiveness, and computational efficiency (can train with a single NVIDIA 2080Ti of 11 GB), can serve as a strong baseline and foundation for future studies. *Third*, extensive experiments over two challenging synthetic-to-real benchmarks (i.e., SynLiDAR [2] $\rightarrow$ SemanticKITTI [7] and

SynLiDAR → SemanticPOSS [1]) show that the proposed framework outperforms the state-of-the-art consistently by large margins.

4.2.2 Related Work

4.2.2.1 Domain Adaptive LiDAR Segmentation

Domain adaptive point cloud segmentation [2, 4, 44, 68, 108, 111, 115, 194] aims for optimal exploitation of previously annotated 'source' point clouds while handling unannotated 'target' point clouds collected in various new domains. It has attracted increasing attention recently due to the challenge in point cloud annotation [195]. Earlier studies [115, 116, 194, 196] project point clouds into depth images and then adopt 2D UDA methods for point cloud segmentation. However, these studies are model-specific and not applicable across deep architectures [26]. Recently, several model-agnostic studies [2, 4, 68] handle the domain discrepancy in the input space directly. For example, [2] translates synthetic point clouds to have similar appearances and sparsity as real point clouds. [108] formulates domain adaptation as a point cloud completion task to minimize density variation across domains. [4, 68] mitigate domain discrepancy by mixing source and target data and creating an intermediate domain with a smaller domain gap. Differently, we design a novel spatial consistency training framework that explores consistency training for domain adaptive 3D LiDAR segmentation.

4.2.2.2 Consistency Training

Consistency training has been widely explored for semi-supervised learning of 2D images, aiming to enhance the robustness of the learnt models while facing various input perturbations. Under this context, different ways of perturbations have been investigated, e.g., by including perturbation noises [197–200], image augmentation [201–203], etc. Recently, one line of research [192, 193, 204, 205] extends the concept of consistency training to the unsupervised domain adaptation of 2D images, largely by designing effective image augmentations for reducing domain gaps across datasets. In addition, another line of research extends consistency training to point cloud tasks. For instance, [206] presents a point-level consistency loss for 3D semi-supervised semantic segmentation, while [207] introduced a multi-level

consistency framework for domain adaptive 3D object detection. Beyond that, several self-supervised networks [48, 208, 209] adopt contrastive loss for learning consistent predictions over augmented point cloud views. As a comparison, we design three types of spatial consistency for learning domain invariant point cloud representations for the task of LiDAR point cloud semantic segmentation.

4.2.3 Mean-Teacher Structure

The "mean-teacher" architecture is a classical architecture that has been widely adopted in various 2D computer vision tasks such as semi-supervised learning [3, 210] and unsupervised domain adaptation [193]. It involves two networks: a student and a teacher. The student is the main network being trained, while the teacher is a copy of the student with a slower update. During the training, the teacher regularizes the student by ensuring that their predictions on unlabelled data are consistent. Recently, it has been extended into 3D point cloud recognition, including semi-supervised 3D segmentation [211], domain adaptive 3D detection [100] and segmentation [4], etc.

4.2.4 Methods

This section presents the proposed method, which consists of four subsection that describe the problem definition for UDA in LiDAR point cloud segmentation, the proposed SCT framework, a fast point-wise matching strategy, and the spatial consistency strategy, respectively.

4.2.4.1 Problem Definition

Under the setting of UDA, we have access to LiDAR point cloud data from a labeled source domain $\mathcal{D}_s = \{x_s^i, y_s^i\}_{i=1}^{N_s}$ and an unlabeled target domain $\mathcal{D}_t = \{x_t^i\}_{i=1}^{N_t}$, where N_s and N_t represent scan numbers of LiDAR point clouds from the source and target domains, respectively. Each LiDAR point cloud $x^i \in \mathbb{R}^{n^i \times 3}$ consists of n^i points with their 3D coordinates while $y_s^i \in \mathbb{R}^{n^i}$ denotes the point-wise labels of the corresponding training sample from the source domain. The goal of domain

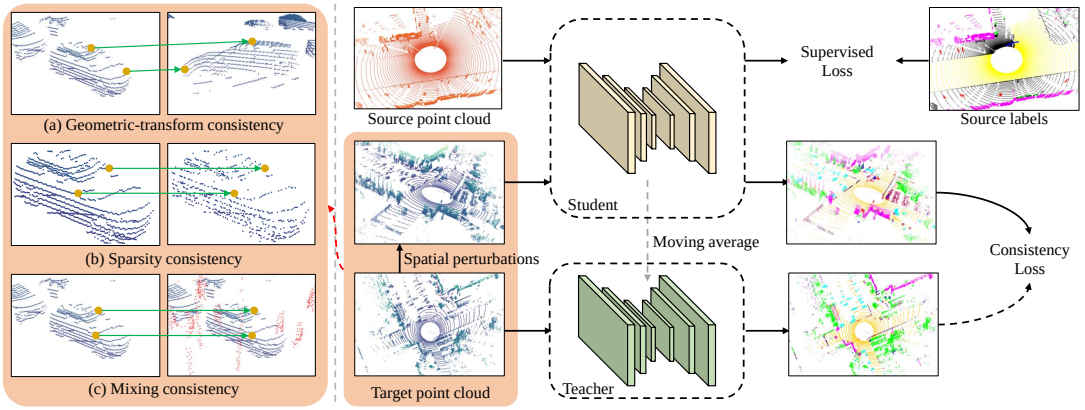


FIGURE 4.9: The pipeline of spatial consistency training (SCT) framework. Leveraging the mean-teacher scheme [3], the student network updates at each iteration by the exponential moving average of itself as the teacher network. The learning enforces the student predictions on spatially perturbed point clouds to be consistent with the teacher predictions on the corresponding raw point clouds under a *Consistency Loss*. We design three types of spatial consistency, namely, *geometric-transform consistency*, *sparsity consistency*, and *mixing consistency*, which are tailored to the spatial characteristics of LiDAR point clouds for enhancing the cross-domain segmentation performance.

adaptive point cloud segmentation is to learn a model F based on $\mathcal{D}_s$ and $\mathcal{D}_t$ that can produce accurate predictions $\hat{y}_t$ for new target data from $\mathcal{D}_t$.

4.2.4.2 Overall Framework

The proposed SCT integrates supervised knowledge from the source domain and self-supervised knowledge from the target domain for learning domain-adaptive representations for segmenting target LiDAR point clouds. Fig. 4.9 shows the overall network framework and Algorithm 1 provides the pseudo-code of the proposed SCT. In the following subsection, we present the *Network Architecture* of SCT, as well as the *Training* and *Inference* of SCT on the task of 3D point cloud semantic segmentation.

Network Architecture We adopted the mean-teacher architecture [3] in the implementation of the proposed SCT. Specifically, the network F consists of a *teacher* network F_T with parameters θ_T, and a *student* network F_S with parameters θ_S. Both are 3D segmentation networks and they share the same network structures.

Training For the labelled source domain, we adopt standard supervised learning to learn semantic structures. Specifically, for a source point cloud scan with corresponding labels $\{x_s^i, y_s^i\}$, we adopt standard cross entropy loss as supervised loss $\mathcal{L}_s$ to optimize the *student* network F_S. The loss is defined as:

$$\mathcal{L}_s = \frac{1}{N_s}\sum_{i=1}^{N_s}\frac{1}{n_s^i}\sum_{j=1}^{n_s^i}\mathcal{H}(y_s^{i,j}, p_s^{i,j}(y|x_s^{i,j})) \tag{4.8}$$

where $p_s^{i,j} \in \mathbb{R}^{1\times C}$ is the output probability distribution of source point j of x_s^i over C classes, i.e., $p_s^{i,j} = softmax(F_S(x_s^{i,j}))$, and $\mathcal{H}$ denotes the entropy.

For an unlabelled target scan x_t, we first generate a spatially perturbed view $\Omega(x_t)$ by randomly applying one of three types of spatial perturbations as to be described in Section 4.2.4.4. We then feed $\Omega(x_t)$ to the student network F_S to obtain prediction logits $F_S(\Omega(x_t))$. Similarly, we feed x_t to the teacher network F_T to obtain prediction logits $F_T(x_t)$. The learning from unlabelled target point clouds can thus be achieved by a cross-entropy loss that enforces the student's predictions to be consistent with the teacher's predictions as follows:

$$\mathcal{L}_t = \frac{1}{N_t}\sum_{i=1}^{N_t}\frac{1}{n_t^i}\sum_{j=1}^{n_t^j}\mathcal{H}(\hat{y}_t^{i,j}, p_t^{i,j}(y|\Omega(x_t^{i,j}))), \tag{4.9}$$

where $\hat{y}_t^{i,j}$ is the pseudo label generated by the teacher model, which is defined as the class with the maximum prediction probability, i.e., $\hat{y}_t^{i,j} = \arg\max(F_T(x_t^{i,j}))$.

Note the teacher network does not back-propagate gradients in training. Instead, it is updated iteratively through exponential moving average of the momentum of the student network as follows:

$$\theta_T = \beta\theta_T + (1-\beta)\theta_S \tag{4.10}$$

where β is the momentum update rate.

The overall objective is a weighted combination of the supervised and unsupervised losses as follows:

$$\mathcal{L} = \mathcal{L}_s + \lambda_t\mathcal{L}_t \tag{4.11}$$

where λ_t is a balancing weight.

Successful training with the spatial consistency pipeline in Fig. 4.9 has two prerequisites. *First*, it requires an efficient matching algorithm to match unlabelled target points of two different views, which is a nontrivial task as point clouds are disordered and unstructured data. *Second*, it requires effective spatial consistency strategies, i.e., the design of Ω for learning from unlabelled target data. For the first prerequisite, we design a fast online matching strategy as to be described in Section 4.2.4.3. For the second prerequisite, we design three types of spatial consistency as to be described in Section 4.2.4.4.

Inference After training, we employ the *student* network directly for inference. Hence, SCT introduces no additional computational overhead during the inference stage.

4.2.4.3 Fast Point-wise Matching

3D semantic segmentation of LiDAR point clouds is computationally intensive as each point-cloud scan consists of thousands of points. Existing networks adopt either random sampling [47, 212] or voxelization [6, 10, 127] for reducing the input points. However, both strategies cause point misalignment across two point-cloud views (i.e., x_t and $\Omega(x_t)$). In addition, many point-cloud augmentation strategies such as rotation and scaling alter the 3D coordinates of points, ruling out the possibility of nearest distance search across two point-cloud views. The concatenation of these operations makes efficient point-wise matching complicated and challenging. One solution is to build point-wise correspondence offline [48], but it constricts the variation of training data and also incurs great overhead in computation and storage space.

We develop a simple yet effective approach that can perform efficient point-wise matching across two point-cloud views. Specifically, after loading a LiDAR point-cloud scan as an array, we assign a unique digital identity to each point which is encoded based on the point position in the array and the position of the LiDAR scan in the dataset. While matching points in two views, we search for the intersections of point identities. The resultant indexes enable a direct retrieval of corresponding point-wise logits from both views, leading to point pairs that can be exploited to compute the spatial consistency loss efficiently. Algorithm 4.2.1

Algorithm 4.2.1 Pseudocode of SCT in a Pytorch-like style.

```
# i: index of the current target LiDAR scan
# x_t: point cloud of current target scan i
# F_T, F_S: teacher, student segmentation network

import numpy as np
F_T.params = F_S.params # initialize
F_T.params.detach() # no back-propagate gradients for teacher

# Data Preprocessing
pt_num = x_t.shape[0] # point number
ids = np.arange(pt_num)
ids = ids + (i << 32) # assign a unique id for each point
x_t2, ids2 = Omega(x_t,ids) # spatial perturbation
x_t1, ids1 = aug(x_t, ids) # randomly augmentation
x_t2, ids2 = aug(x_t2, ids2) # randomly augmentation

# Fast Point-wise Matching
co_ids = np.intersect1d(ids1, ids2) # ids of co-existed points in two views
sorter = np.argsort(ids1)
m_ids1 = sorter[np.searchsorted(ids1, co_ids, sorter=sorter)]
sorter = np.argsort(ids2)
m_ids2 = sorter[np.searchsorted(ids2, co_ids, sorter=sorter)]

# Forward to Model
pred_t1 = F_T.forward(x_t1)
pred_t2 = F_S.forward(x_t2)
# supervised loss and consistency loss
loss_s = CrossEntropyLoss(pred_t1, labels_t1)
loss_t = CrossEntropyLoss(pred_t2[m_ids2], pred_t1[m_ids1].detach())
loss = loss_s + lambda_t*loss_t

# Update Network
loss.backward()
update(F_S.params) # update student
F_T.param = beta*F_T.param + (1-beta)*F_S.param # ema update teacher
```

provides pseudocode for Fast Point-wise Matching in a Pytorch and Numpy-like style.

4.2.4.4 Spatial Consistency Strategies

We design three types of spatial consistency in SCT for learning domain-invariant point cloud representations that are tolerant to spatial perturbations Ω on target point clouds. More details about the three types of spatial consistency and the corresponding spatial perturbations are to be described in the ensuing three subsections.

1) ***Geometric-Transform Consistency.*** The spatial distribution of points in 3D LiDAR point clouds of different domains can vary greatly due to different configurations and viewpoints of LiDAR sensors which can lead to significant differences in geometric structures of point clouds. We introduce geometric perturbations by randomly applying a set of geometric transformations to the target point clouds, such as rotation, scaling, and flipping. The geometric consistency training can thus be achieved by enforcing the model to produce consistent predictions on the spatially transformed and original point clouds as illustrated in Fig. 4.9 (a). This consistency strategy guides the model to learn domain-invariant geometric features and enhances its ability to adapt to different domains.

2) ***Sparsity Consistency.*** The sparsity/density of 3D LiDAR point clouds also varies across domains due to different sensor settings (e.g., laser beam number, field of view, etc.) or environments, and such variation can greatly degrade the model's inter-domain recognition performance. We introduce sparsity perturbations by randomly masking a certain portion σ of input points to generate a sparse view of point clouds. Sparsity consistency training can thus be achieved by enforcing the model to produce consistent predictions across the original and the sparsified point clouds as illustrated in Fig. 4.9 (b). This consistency strategy encourages the model to learn sparsity-tolerant features, which helps the trained model better adapt across domains with different point-cloud sparsity.

3) ***Mixing Consistency.*** Semantic segmentation models often rely on various local contexts in recognition tasks. However, the reliance on such spatial priors in the source domain often misleads the model's recognition in the target domain due to spatial distribution variance across domains. To address this, we introduce spatial context perturbations by randomly mixing points from other LiDAR point-cloud scans which directly modifies the local context of the current LiDAR scan. The consistency with local contexts can thus be achieved by enforcing the prediction

consistency between the original and mixed views as illustrated in Fig. 4.9 (c). This consistency induces 3D segmentation models to learn local context-invariant representations, resulting in enhanced recognition ability in the target domain. In the implementation, we adopt the recent PolarMix [68] for context perturbation as PolarMix enriches the local distribution of the mixing data while preserving LiDAR data fidelity.

4.2.5 Experiments

4.2.5.1 Experimental Setup

1) ***Datasets.*** We conducted comprehensive experiments to validate the effectiveness of our SCT. The experiments were performed over two challenging synthetic-to-real benchmarks on domain adaptive 3D LiDAR semantic segmentation tasks: SynLiDAR [2] → SemanticKITTI [7] and SynLiDAR → SemanticPOSS [1]. The two benchmarks involve three LiDAR point cloud datasets of road scenes as listed:

- **SynLiDAR** is a large-scale synthetic LiDAR point cloud dataset with 198,396 LiDAR scans and point-level annotations of 32 semantic classes. This large-scale dataset was meticulously collected from nine realistic virtual environments constructed using Unreal Engine 4, including cities, towns, harbors, etc. The data acquisition process involved utilizing a cutting-edge LiDAR simulator capable of generating scans with 64 beam numbers. Following prior studies in [2, 4], we use the officially provided subset with 19,840 scans in our experiments.
- **SemanticKITTI** is a large-scale real LiDAR point cloud dataset collected in Germany. It was collected using a Velodyne HDL-64E LiDAR with 64 laser beams. The dataset consists of 43,552 LiDAR scans with point-wise annotations of 22 semantic classes. We use sequences 00-10 for training, except sequence 08 for validation, following the official split.
- **SemanticPOSS** consists of 2,988 real-world scans with point-level annotations over 14 semantic classes. It was collected on campus using a Pandora LiDAR sensor equipped with 40 laser channels, leading to distinctive spatial distributions that set it apart from the SemanticKITTI dataset. We use sequence 03 for validation and the remaining sequences for training, as per the official benchmark guidelines.

TABLE 4.9: Ablation study of different spatial consistency strategies over domain adaptive 3D LiDAR segmentation task SynLiDAR → SemanticPOSS. Geometric-transform consistency training (GT-CT) significantly increases domain generalized 3D segmentation performance. Incorporating sparsity consistency training (S-CT) or mixing consistency training (M-CT) with GT-CT further improves the target segmentation performance clearly. In addition, the combination of all three types of spatial consistency training achieves the best performance, demonstrating the synergic relation among our three designs.

Model	**GT-CT**	**S-CT**	**M-CT**	**mIoU**
Source-only				20.7
(a)	✓			41.7
(b)	✓	✓		44.6
(c)	✓		✓	43.7
(d)	✓	✓	✓	**46.3**

We evaluate our models using per-class Intersection-over-Union (IoU) and mean IoU (mIoU) metrics. Following prior studies [2, 4], we measure IoU and mIoU over 19 classes for SynLiDAR→SemanticKITTI, and 13 shared classes for SynLiDAR → SemanticPOSS.

2) ***Implementation Details.*** We follow [2, 4, 68] and adopt MinkowskiNet [10] as the backbone model for fair comparisons. We first pre-train the network with cross-entropy loss over source data for 15 epochs by using SGD optimizer with a learning rate of 0.01, momentum of 0.9, and weight decay of 1.e-4, with a batch size of 4. When adapting the model to the target domain, we first initialize the student and teacher models with the pre-trained weights and then train another 5 epochs with SGD optimizer. We set the learning rate to 0.001, momentum to 0.9, and weight decay to 1.e-4, with a batch size of 2 for both source and target data. The hyperparameters λ_t and β are set at 0.1 and 0.99, respectively. As for Ω in different spatial consistency strategies: For geometric-transform consistency, we rotate point clouds along the z-axis between $[-\pi/2, \pi/2]$, scale between $[0.95, 1.05]$, and flip along the x- or y-axis with 50% chance; For sparsity consistency, we randomly mask 50% of points, i.e. $\sigma = 0.5$; For mixing consistency, we implement random 90° scene-level swapping and three times instance-level rotate-pasting for PolarMix [68]. We use TorchSparse library [6] for implementation. All experiments are conducted on one NVIDIA RTX2080Ti with 11GB GPU memory.

TABLE 4.10: Experiments on unsupervised domain adaptation with SynLiDAR (as source) and SemanticKITTI (as target). SCT outperforms all typical and state-of-the-art methods consistently by large margins.

Method	car	bi.cle	mt.cle	truck	oth-v.	pers.	bi.clst	mt.clst	road	parki.	sidew.	oth-g.	build.	fence	veget.	trunk	terra.	pole	traf.	mIoU
Source-only [2]	42.0	5.0	4.8	0.4	2.5	12.4	43.3	1.8	48.7	4.5	31.0	0.0	18.6	11.5	60.2	30.0	48.3	19.3	3.0	20.4
ADDA [190]	52.5	4.5	11.9	0.3	3.9	9.4	27.9	0.5	52.8	4.9	27.4	0.0	61.0	17.0	57.4	34.5	42.9	23.2	4.5	23.0
Ent-Min [149]	58.3	5.1	14.3	0.6	1.8	14.3	44.5	0.5	50.4	4.3	34.8	0.0	48.3	19.7	67.5	34.8	52.0	33.0	6.1	25.8
ST [83]	62.0	5.0	12.4	1.3	9.2	16.7	44.2	0.4	53.0	2.5	28.4	0.0	57.1	18.7	69.8	35.0	48.7	32.5	6.9	26.5
PCT [2]	53.4	5.4	7.4	0.8	10.9	12.0	43.2	0.3	50.8	3.7	29.4	0.0	48.0	10.4	68.2	33.1	40.0	29.5	6.9	23.9
ST-PCT [2]	70.8	7.3	13.1	1.9	8.4	12.6	44.0	0.6	56.4	4.5	31.8	0.0	66.7	23.7	73.3	34.6	48.4	39.4	11.7	28.9
PolarMix [68]	76.3	**8.4**	17.8	3.9	6.0	26.6	40.8	15.9	70.3	0.0	44.4	0.0	**68.4**	14.7	69.6	38.1	37.1	40.6	10.6	31.0
CoSMix [4]	75.1	6.8	29.4	**27.1**	**11.1**	22.1	25.0	**24.7**	**79.3**	**14.9**	46.7	**0.1**	53.4	13.0	67.7	31.4	32.1	37.9	**13.4**	32.2
SCT (Ours)	**81.9**	3.7	**31.2**	1.6	7.4	**44.6**	**61.1**	3.4	78.2	3.5	**51.2**	0.0	68.0	**31.7**	**74.3**	**45.8**	**51.0**	**41.7**	4.1	**36.0**

4.2.5.2 Ablation Studies

We conduct comprehensive ablation studies to evaluate the effectiveness of the proposed SCT framework. We report five models over SynLiDAR → SemanticPOSS including: 1) *Source-only* that is trained using supervised loss $\mathcal{L}_s$ in Eq. 4.8 only, without involving target data in the training process; 2) *Model (a)* that performs geometric-transform consistency training over target data and supervised learning over source data; 3) *Model (b)* that further incorporates sparsity consistency training on top of the model (a); 4) *Model (c)* that incorporates mixing consistency training on top of the model (a); and 5) the full SCT *Model (d)* that combines geometric-transform consistency, sparsity consistency, and mixing consistency in training with target data, as well as supervised learning for source data.

The experimental results are summarized in Table 4.9. As expected, the *Source-only* model trained with SynLiDAR performs poorly due to the clear domain discrepancy. However, we observe a significant improvement in performance with the *Geometric-transform consistency training*, which outperforms the *Source-only* model by a large margin. In addition, incorporating *Sparsity consistency training* and *mixing consistency training* leads to further improvement in the adaptation, demonstrating the effectiveness of our designed spatial consistency strategies. Notably, the full SCT model achieves the best segmentation performance, indicating that the three types of spatial consistency strategies are complementary and synergistic in domain adaptive point cloud segmentation.

TABLE 4.11: Experiments on domain adaptive semantic segmentation from SynLiDAR (as source) to SemanticPOSS (as target). SCT outperforms all typical and state-of-the-art methods consistently by large margins.

Method	pers.	rider	car	trunk	plants	traf.	pole	garb.	buil.	cone.	fence	bike	grou.	mIoU
Source-Only	3.7	25.1	12.0	10.8	53.4	0.0	19.4	12.9	49.1	3.1	20.3	0.0	59.6	20.7
ADDA [190]	27.5	35.1	18.8	12.4	53.4	2.8	27.0	12.2	64.7	1.3	6.3	6.8	55.3	24.9
Ent-Min [149]	24.2	32.2	21.4	18.9	61.0	2.5	36.3	8.3	56.7	3.1	5.3	4.8	57.1	25.5
ST [83]	23.5	31.8	22.0	18.9	63.2	1.9	**41.6**	13.5	58.2	1.0	9.1	6.8	60.3	27.1
PCT [2]	13.0	35.4	13.7	10.2	53.1	1.4	23.8	12.7	52.9	0.8	13.7	1.1	66.2	22.9
ST-PCT [2]	28.9	34.8	27.8	18.6	63.7	4.9	41.0	16.6	64.1	1.6	12.1	6.6	63.9	29.6
PolarMix [68]	32.6	39.1	25.0	11.9	64.2	5.8	29.6	15.3	44.8	13.3	23.8	10.7	79.0	30.4
CoSMix [4]	55.8	51.4	**36.2**	23.5	**71.3**	22.5	34.2	**28.9**	66.2	20.4	24.9	10.6	78.7	40.4
SCT (Ours)	**57.1**	**54.3**	24.5	**52.3**	62.1	**40.3**	37.6	2.5	**69.7**	**31.7**	**42.7**	**47.6**	**79.5**	**46.3**

4.2.5.3 Comparison with State-of-the-Arts

We compared our spatial consistency training method with a number of state-of-the-art UDA methods. Tables 4.10 and 4.11 show experimental results over the tasks SynLiDAR → SemanticKITTI and SynLiDAR → SemanticPOSS, respectively. As the two tables show, our method outperforms all state-of-the-art UDA methods clearly and consistently across both tasks, achieving improvements of +3.8 and +5.9 percent points over the state-of-the-art [4], respectively. The superior segmentation performance demonstrates that the proposed spatial consistency training is indeed an advanced method of domain adaptive semantic segmentation for 3D LiDAR point clouds.

We also qualitatively compare our spatial consistency training with the *Source-only* and the state-of-the-art CoSMix [4] over SynLiDAR → SemanticKITTI. As Fig. 4.10 shows, the *Source-only* produces lots of false predictions due to domain bias. For CoSMix, many confident yet false predictions are selected as pseudo labels which accumulate in the iterative self-training process and finally impair the trained model. Differently, our SCT minimizes the divergence of predictions across point views with respect to different spatial perturbations and learns robust feature representation of the target domain, achieving superior segmentation performance over point clouds in the target domain.

Despite its superior adaptive segmentation performance, SCT still struggles under certain circumstances. One typical scenario happens when a large portion of segmentation failures belongs to long-tail classes that have very limited training samples. Such lack of training data often degrades representation learning and

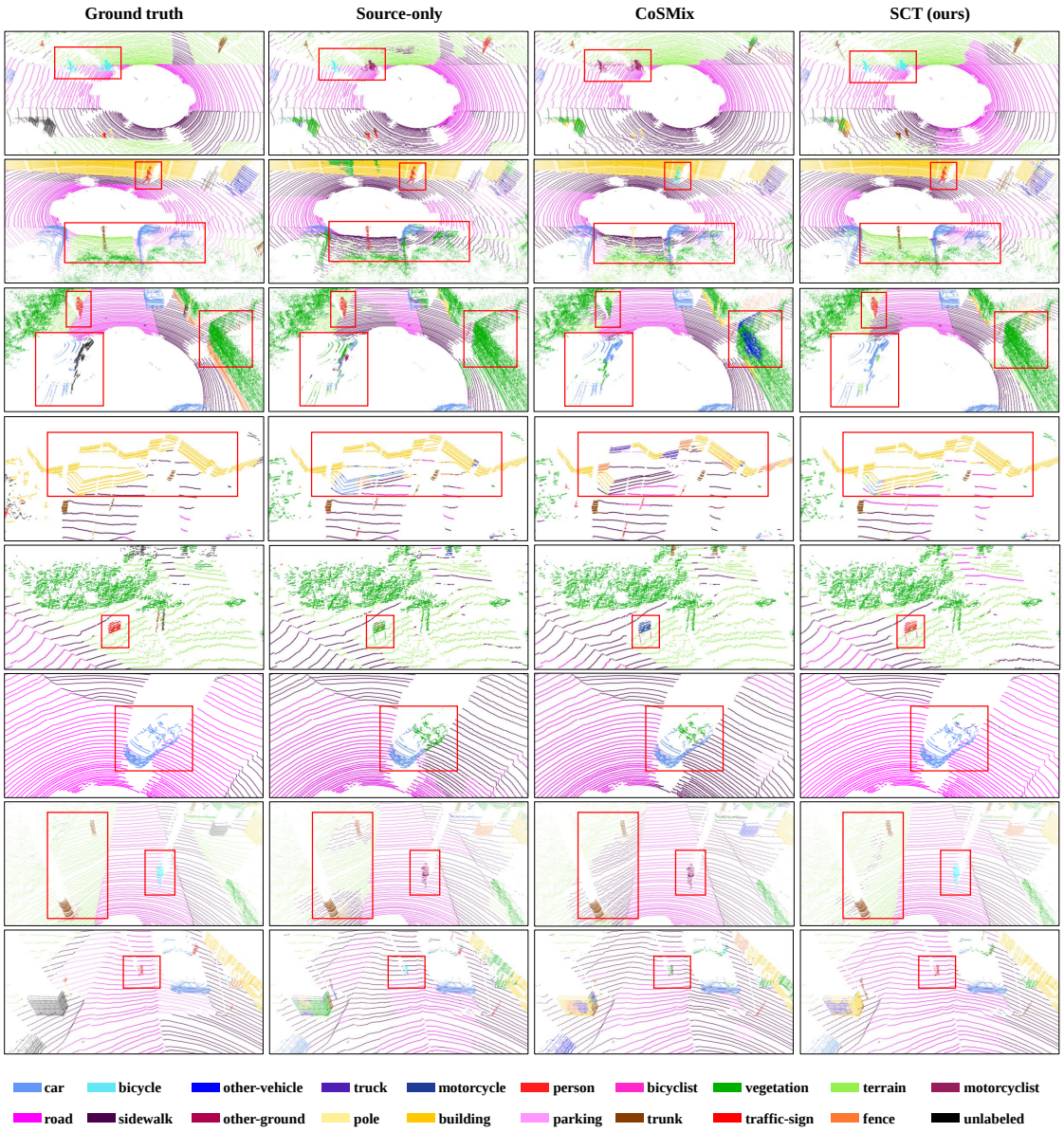


FIGURE 4.10: Qualitative comparison of SCT with the *Source-only* (with no adaptation) and the state-of-the-art CoSMix [4] in domain adaptive 3D LiDAR semantic segmentation. The comparison was conducted over the task "SynLiDAR → SemanticKITTI". The 'Ground truth' denotes the ground-truth annotations. The red rectangles highlight regions of interest. Best viewed in color.

compromises the adaptability of the learnt model. Besides, the checkpoint selection aiming to optimize mIoU potentially underplays the performance of long-tail classes as well.

It's worth noting that our SCT framework requires minimal computational resources, utilizing only one NVIDIA GTX2080Ti with 11GB of GPU memory. In

TABLE 4.12: Adaptation results on SemanticPOSS→SemanticKITTI.

Method	mIoU
Source-only	22.5
CoSMix [4]	26.4
SCT (Ours)	**29.4**

contrast, the state-of-the-art CosMix [4] requires much more powerful hardware, utilizing 4×NVIDIA A100 GPUs (each with 40GB SXM4). We will release our code as a strong baseline repository, lowering the research barrier and facilitating future research in domain adaptive 3D LiDAR segmentation.

Real-to-real adaptation. The proposed SCT can also handle real-to-real adaptation across LiDAR datasets with different numbers of LiDAR beam lines. Following CoSMix (detailed in its appendix), we performed evaluations on SemanticPOSS (40-line)→SemanticKITTI (64-line) where point clouds are captured by LiDAR sensors of different beam line numbers. As Table 4.12 shows, SCT outperforms CoSMix clearly, indicating its robustness and generalization ability in domain-adaptive LiDAR point cloud segmentation.

4.2.5.4 Analysis

We conducted comprehensive experiments to analyse the proposed SCT. The experimental results and findings are detailed in the subsequent subsections.

1) ***Varying*** λ_t**.** We examined the effect of parameter λ_t in Eq. 4.11, which balances the supervised loss in the source domain and the unsupervised spatial consistency loss in the target domain. Table 4.13 shows experimental results over the task SynLiDAR→SemanticPOSS. We can see that different λ_t produce only moderate variations in mIoU, and all of them outperform the source-only model (i.e., $\lambda_t = 0.0$) significantly. The best mIoU is achieved when $\lambda_t = 0.10$. The experiments show that our proposed SCT is tolerant to the variation in balance weight λ_t.

2) ***Varying*** β**.** We employ the momentum parameter β to update the teacher model. When β is set to 0, the teacher model is equivalent to the student model with no temporal momentum update. We examine the impact of different values of β in Table 4.14. We can see that the model performs much better with the

TABLE 4.13: Performance of spatial consistency training under different λ_t (the balance weight of spatial consistency loss as defined in Eq. 4.11) on the SynLiDAR→SemanticPOSS UDA task.

λ_t	mIoU
0.00	20.4
0.05	44.5
0.10	**46.3**
0.15	46.1
0.20	45.6
1.00	44.7

TABLE 4.14: Evaluation of the performance of spatial consistency training models with varying momentum update weight β on the SynLiDAR → SemanticPOSS task.

β	mIoU
0.0	32.3
0.9	45.5
0.99	**46.3**
0.999	45.8

TABLE 4.15: Segmentation performance of spatial consistency training on SynLiDAR → SemanticPOSS with combination of different geometric transformations.

Method	rotation	scaling	flipping	mIoU
(a)	✓			45.0
(b)	✓	✓		45.9
(c)	✓	✓	✓	46.3

exponential moving average and it performs the best when β is set to 0.99, indicating that a slowly progressing teacher model is beneficial. At the other end, the teacher model updates too slowly to capture the latest representative network parameters with a very high β, and it updates too fast and leads to less robust and unstable temporal ensembles with a very low β. Both scenarios impair the trained cross-domain segmentation models.

3) ***Geometric Transformations***. We study how different geometric transformations affect adaptation performance. Table 4.15 presents experimental results under several typical geometric transformations including *rotation*, *scaling*, and *flipping*. It can be observed that learning under different geometric perturbations improves the adaptation process and more complex geometric perturbations are helpful in enhancing the target performance.

TABLE 4.16: Results of our sparsity consistency with different proportions of sparsity σ over SynLiDAR→SemanticPOSS.

σ	mIoU
0.0	43.7
0.3	45.3
0.5	**46.3**
0.7	46.0

TABLE 4.17: Performance of spatial consistency training with two unsupervised losses: cross-entropy loss ($\mathcal{L}_{ce}$) as defined in Eq. 4.9, and mean squared error loss ($\mathcal{L}_{mse}$) as defined in Eq. 4.12. Results are shown over SynLiDAR→SemanticPOSS.

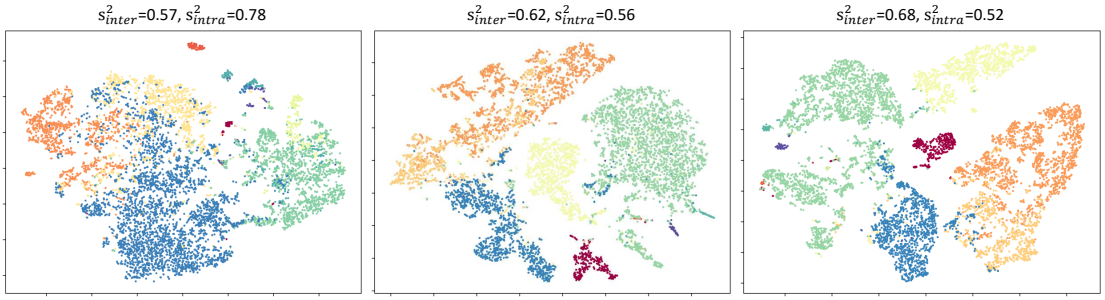
Consistency Loss	**mIoU**
N.A.	20.7
$\mathcal{L}_{mse}$	38.2
$\mathcal{L}_{ce}$	**46.3**

4) ***Sparsity Ratio***. We investigated the impact of sparsity ratios σ in sparsity consistency training. As Table 4.16 shows, incorporating sparsity consistency consistently leads to clear improvements in target segmentation compared to the baseline (i.e., $\sigma = 0$) while the best segmentation performance is achieved when $\sigma = 0.5$. The experiments reveal that setting an appropriate sparsity ratio is important as a large σ provides limited sparsity perturbations while a small σ tends to lose necessary geometric information for point recognition.

5) ***Different Consistency Losses.*** We also evaluated the mean squared error (MSE) loss between the teacher's and student's predictions for consistency regularization:

$$\mathcal{L}_{mse} = \frac{1}{N_t} \sum_{i=1}^{N_t} \frac{1}{n_t^i} \sum_{j=1}^{n_t^i} \left(F_T(x_t^{i,j}) - F_S(\Omega(x_t^{i,j}))\right)^2 \tag{4.12}$$

Table 4.17 shows experimental results on SynLiDAR → SemanticPOSS, where L_{ce} denotes the cross-entropy loss defined in Eq. 4.9. We can observe that optimizing both unsupervised losses outperforms the Source-only (N.A.) significantly, validating the effectiveness of spatial consistency training. In addition, optimizing the cross-entropy loss leads to significantly better performance, largely because the re-trained student model is supervised with *hard* pseudo-labels, which help to minimize prediction entropy.



(a) Source-only (b) CoSMix (c) SCT (ours)

FIGURE 4.11: Feature space visualization with t-SNE [5] on SynLiDAR → SemanticKITTI UDA task. The proposed *SCT* learns more compact feature space for target domain with smaller intra-class variance and larger inter-class variance as compared with the *Source only* and the state-of-the-art *CoSMix* [4]. Different colors denote different classes and best viewed in color.

6) ***Features Visualization***. To better assess the proposed SCT, we employ t-SNE [5] to visualize point cloud representations of the target domain. Fig.4.11 shows the feature visualizations for the source-only model, the state-of-the-art CoSMix [4], and our proposed SCT, respectively. We can observe that the SCT-produced features have clearly better discriminability than those produced by the source-only model, highlighting the outstanding adaptation performance of the proposed SCT. Additionally, SCT also produces more discriminative target features than CoSMix, achieving the largest inter-class variance while maintaining the smallest intra-class variance. This suggests that the upstream class-wise representations from SCT are more discernible, making it a reliable indicator of its effectiveness.

7) ***Pseudolabel Threshold.*** We employ all pseudo labels in the spatial consistency training. At the other end, it is possible to adopt thresholding to select confident pseudo labels only in the spatial consistency training. Table 4.18 show relevant experiments, where applying different thresholds δ degrades cross-domain segmentation consistently. We conjecture that the thresholding could produce many confident but false predictions which lead to a deviated solution with error propagation in training. Differently, employing all pseudo labels enables more comprehensive and robust adaptive learning in the target domain.

8) ***Training Time Comparsion.*** We compare SCT with CoSMix [4] to validate its superior computational efficiency. For CosMix, we use its official code with default configurations and train with four NVIDIA V100 GPUs over the benchmark

TABLE 4.18: Selecting pseudo labels by thresholding their prediction probabilities. With different thresholds $\delta \in [0, 1)$, the proposed spatial consistency training learns from different pseudo labels with different segmentation over SynLiDAR → SemanticPOSS.

δ	mIoU
0.0	**46.3**
0.5	44.8
0.9	44.9
0.95	44.3
0.99	42.2

TABLE 4.19: Training resource usage for CoSMix and our method SCT over SynLiDAR→SemanticKITTI.

Method	CoSMix [27]	SCT (Ours)
training time	4.6 hours	**2.2 hours**
GPU usage	4×V100 (4x32GB)	**1x2080Ti(1x11GB)**
mIoU	32.3	**36.0**

SynLiDAR→SemanticKITTI. As Table 4.19 shows, SCT (using a single NVIDIA 2080Ti) can be trained much faster than CosMix Ti. Besides, it achieves clearly better mIoU. The experiments highlight the potential of SCT which as a powerful tool could greatly reduce the research barrier and facilitate the future research in domain adaptive 3D LiDAR segmentation.

9) ***More Analysis for Sampling Strategy.*** Targeting a simple, efficient, and effective base technique in point cloud learning, we adopted random sampling (RS) in sparsity consistency design. We tested more sophisticated sampling techniques including grid sampling (GS) and distance-based sampling (DS). Table 4.20 shows experiments on SynLiDAR→SemanticPOSS, where DS(f)/DS(c) means assigning higher sampling weights to farther/closer points. We can see that GS performs similarly to RS while DS(c) performs clearly worse, suggesting that sampling should prioritize nearer and denser points. In addition, all sampling strategies outperform N.A. without using sparsity consistency, validating our finding on maintaining sparsity invariance in cross-domain LiDAR segmentation.

10) ***SCT with Different Backbone Models***. The proposed SCT is model-agnostic and can work with different backbone models. We verify this by implementing it with another widely adopted 3D segmentation model SPVCNN [6]. As shown in Table 4.21, SCT outperforms the Source-only clearly on SynLiDAR →

TABLE 4.20: Different sampling strategies for sparsity consistency in SCT, including random sampling ("RS"), grid sampling ("GS"), and distance-based sampling ("DS", DS(f)/DS(c) denoting higher sampling weights assigned to farther/closer points). Results are shown over SynLiDAR → SemanticPOSS.

Method	mIoU
N.A.	43.7
RS	**46.3**
GS	**46.3**
DS(f)	46.0
DS(c)	45.2

SemanticKITTI, demonstrating its superior robustness and generalization across different backbone models.

TABLE 4.21: Unsupervised domain adaptative point cloud segmentation with the backbone SPVCNN [6] (on SynLiDAR → SemanticKITTI). SCT improves UDA consistently with different backbone models.

Method	mIoU
Source-Only	23.7
SCT (Ours)	**31.3**

11) ***Failure Analysis.*** Most segmentation failures with SCT are associated with long-tail classes (such as "mt.clst." in Table 4.7 and "garb." in Table 4.8) that have very limited training samples and thereby often suffer from clear overfitting. In addition, our checkpoint selection prioritizes optimizing mIoU which inadvertently sacrifices the performance of long-tail classes. This issue could be alleviated by developing some class-balanced UDA approach that mitigates the long-tail distribution by balancing the data distribution across classes.

12) ***Analysis of PCT, PolarMix, and SCT.*** PCT and PolarMix operate within the input space, while SCT functions within the output space. As detailed in Section 4.2.3.4 under "3) Mixing Consistency," we integrate PolarMix into SCT, resulting in a notable performance improvement, as highlighted in Table 4.9. This outcome underscores their complementary nature. Furthermore, when we substitute SynLiDAR with PCT-translated SynLiDAR and train SCT on the benchmark SynLiDAR → SemanticKITTI, the outcomes displayed in Table 4.22 indicate that

Method	mIoU
PCT	22.9
SCT	46.3
SCT+PCT	46.5

TABLE 4.22: Segmentation result over SynLiDAR→SemanticKITTI.

the combination of PCT and SCT yields only marginal performance gains. This result arises since both PCT and SCT address variations in sparsity, thereby producing comparable effects, leading to limited additional improvements when combined.

4.2.6 Conclusion

This section proposes a novel spatial consistency training framework for addressing the domain shift problem in 3D LiDAR point cloud segmentation. The approach enforces prediction consistency between raw point clouds and their spatially perturbed views, guiding the segmentation network to learn domain-invariant feature representations across domains. Three novel spatial consistency strategies tailored to the data properties of LiDAR point clouds are introduced to facilitate effective consistency training in 3D space, namely geometric-transform consistency, sparsity consistency, and mixing consistency. Comprehensive experimental results demonstrate that the proposed spatial consistency training significantly improves the performance of 3D UDA tasks as compared with the state-of-the-art.

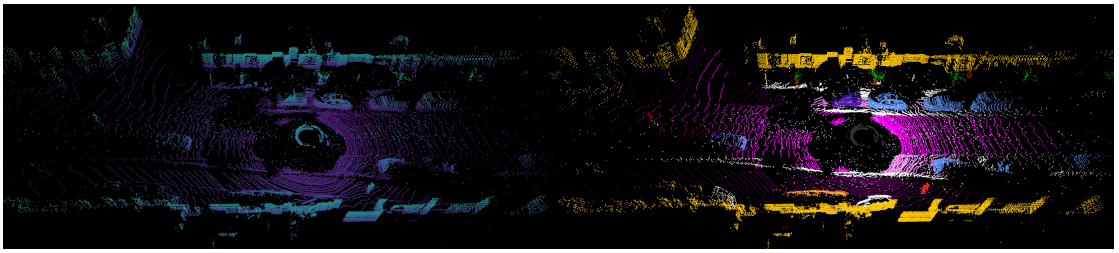
Chapter 5

Domain Transfer Learning from Normal to Adverse Weather Point Clouds[1]

In this chapter, we explore another significant type of domain transfer learning, focusing on the transfer from normal-weather point clouds to adverse weather point clouds. This transfer is crucial for mitigating the challenges associated with annotating adverse weather point clouds and training robust point cloud recognition models in a label-efficient manner.

To address this issue, we first introduce SemanticSTF, the first large-scale point cloud dataset captured under adverse weather conditions, with densely annotated labels for semantic segmentation. Our investigation examines the modeling of all-weather 3DSS (3D semantic segmentation) under two setups: 1) Domain adaptive 3DSS, which involves adapting from normal-weather data to adverse-weather data. 2) Domain generalizable 3DSS, which aims to learn all-weather 3DSS models from normal-weather data. In addition, we design a domain randomization technique that alternatively randomizes the geometry styles of point clouds and aggregates their embeddings, ultimately leading to a generalizable model that can improve 3DSS under various adverse weather effectively.

[1]The work in this chapter has been published at "Aoran Xiao, Jiaxing Huang, Weihao Xuan, Ruijie Ren, Kangcheng Liu, Dayan Guan, Abdulmotaleb El Saddik, Shijian Lu, Eric P. Xing; Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 9382-9392



(a) A LiDAR scan captured in a snowy day (b) Point-level annotations

FIGURE 5.1: We introduce SemanticSTF, an adverse-weather LiDAR point cloud dataset with dense point-level annotations that can be exploited for the study of point cloud semantic segmentation under all-weather conditions (including fog, snow, and rain). The graph on the left shows one scan sample captured on a snowy day, and the one on the right shows the corresponding point-level annotations.

Further details will be presented in the subsequent sections.

5.1 Introduction

3D LiDAR point clouds play an essential role in semantic scene understanding in various applications such as self-driving vehicles and autonomous drones. With the recent advance of LiDAR sensors, several LiDAR point cloud datasets [2, 7, 147] such as SemanticKITTI [7] have been proposed which greatly advanced the research in 3D semantic segmentation (3DSS) [6, 47, 127] for the task of point cloud parsing. As of today, most existing point cloud datasets for outdoor scenes are dominated by point clouds captured under normal weather. However, 3D vision applications such as autonomous driving require reliable 3D perception under all-weather conditions including various adverse weather such as fog, snow, and rain. How to learn a weather-tolerant 3DSS model is largely neglected due to the absence of related benchmark datasets.

Although several studies [40, 213] attempt to include adverse weather conditions in point cloud datasets, such as the STF dataset [40] that consists of LiDAR point clouds captured under various adverse weather, these efforts focus on object detection benchmarks and do not provide any point-wise annotations which are critical in various tasks such as 3D semantic and instance segmentation. To address this gap, we introduce *SemanticSTF*, an adverse-weather point cloud dataset that extends the STF Detection Benchmark by providing point-wise annotations of

21 semantic categories, as illustrated in Fig. 5.1. Similar to STF, SemanticSTF captures four typical adverse weather conditions that are frequently encountered in autonomous driving including dense fog, light fog, snow, and rain.

SemanticSTF provides a great benchmark for the study of 3DSS and robust point cloud parsing under adverse weather conditions. Beyond serving as a well-suited test bed for examining existing fully-supervised 3DSS methods that handle adverse-weather point cloud data, SemanticSTF can be further exploited to study two valuable weather-tolerant 3DSS scenarios: 1) domain adaptive 3DSS that adapts from normal-weather data to adverse-weather data, and 2) domain generalizable 3DSS that learns all-weather 3DSS models from normal-weather data. Our studies reveal the challenges faced by existing 3DSS methods while processing adverse-weather point cloud data, highlighting the significant value of SemanticSTF in guiding future research efforts along this meaningful research direction.

In addition, we design PointDR, a new baseline framework for the future study and benchmarking of all-weather 3DSS. Our objective is to learn robust 3D representations that can reliably represent points of the same category across different weather conditions while remaining discriminative across categories. However, robust all-weather 3DSS poses two major challenges: 1) LiDAR point clouds are typically sparse, incomplete, and subject to substantial geometric variations and semantic ambiguity. These challenges are further exacerbated under adverse weather conditions, with many missing points and geometric distortions due to fog, snow cover, etc. 2) More noises are introduced under adverse weather due to snow flicks, rain droplets, etc. PointDR addresses the challenges with two iterative operations: 1) *Geometry style randomization* that expands the geometry distribution of point clouds under various spatial augmentations; 2) *Embedding aggregation* that introduces contrastive learning to aggregate the encoded embeddings of the randomly augmented point clouds. Despite its simplicity, extensive experiments over point clouds of different adverse weather conditions show that PointDR achieves superior 3DSS generalization performance.

The contribution of this work can be summarized in three major aspects. *First*, we introduce SemanticSTF, a large-scale adverse-weather point cloud benchmark that provides high-quality point-wise annotations of 21 semantic categories. *Second*, we design PointDR, a point cloud domain randomization baseline that can

be exploited for future study and benchmarking of 3DSS under all-weather conditions. *Third*, leveraging SemanticSTF, we benchmark existing 3DSS methods over two challenging tasks on domain adaptive 3DSS and domain generalized 3DSS. The benchmarking efforts lay a solid foundation for future research on this highly meaningful problem.

5.2 Related Works

3D semantic segmentation aims to assign point-wise semantic labels for point clouds. It has been developed rapidly over the past few years, largely through the development of various deep neural networks (DNNs) such as standard convolutional network for projection-based methods [20, 115, 154, 164, 165], multi-layer perceptron (MLP)-based networks [8, 8, 47], 3D voxel convolution-based networks [10, 127], or hybrid networks [6, 214–217]. While existing 3DSS networks are mainly evaluated over normal weather point clouds, their performance for adverse weather point clouds is far under-investigated. The proposed SemanticSTF closes the gap and provides a solid ground for the study and evaluation of all-weather 3DSS. By enabling investigations into various new research directions, SemanticSTF represents a valuable tool for advancing the field.

Vision recognition under adverse conditions. Scene understanding under adverse conditions has recently attracted increasing attention due to the strict safety demand in various outdoor navigation and perception tasks. In 2D vision, several large-scale datasets have been proposed to investigate perceptions tasks in adverse visual conditions including localization [218], detection [219], and segmentation [220]. On the other hand, learning 3D point clouds of adverse conditions is far under-explored due to the absence of comprehensive dataset benchmarks. The recently proposed datasets such as STF [40] and CADC [213] contain LiDAR point clouds captured under adverse weather conditions. However, these studies focus on the object detection task [103, 105] with bounding-box annotations, without providing any point-wise annotations. Our introduced SemanticSTF is the first large-scale dataset that consists of LiDAR point clouds in adverse weather conditions with high-quality dense annotations, to the best of our knowledge.

Domain generalization [221, 222] aims to learn a generalizable model from single or multiple related but distinct source domains where target data is inaccessible during model learning. It has been widely studied in 2D computer vision tasks [121, 223–225] while few studies explore it in point cloud learning. Recently, [124] studies domain generalization for 3D object detection by deforming point clouds via vector fields. Differently, this work is the first attempt that explores domain generalization for 3DSS.

Unsupervised domain adaptation is a method of transferring knowledge learned from a labeled source domain to a target domain by leveraging the unlabeled target data. It has been widely studied in 2D image learning [120, 150, 152, 153, 226, 227] and 3D point clouds [96, 97, 99, 100, 103–105]. Recently, domain adaptive 3D LiDAR segmentation has drawn increasing attention due to the challenge in point-wise annotation. Different UDA approaches have been designed to mitigate discrepancies across LiDAR point clouds of different domains. For example, [115, 117] project point clouds into depth images and leverage 2D UDA techniques while [2, 4, 68, 108] directly work in the 3D space. However, these methods either work for *synthetic-to-real* UDA scenarios [2, 115] or *normal-to-normal* point cloud adaptation [108], ignoring *normal-to-adverse* adaptation which is highly practical in real applications. Our SemanticSTF dataset fills up this blank and will inspire more development of new algorithms for normal-to-adverse adaptation.

5.3 The SemanticSTF Dataset

5.3.1 Background

LiDAR sensors send out laser pulses and measure their flight time based on the echoes it receives from targets. The travel distance as derived from the time-of-flight and the registered angular information (between the LiDAR sensors and the targets) can be combined to compute the 3D coordinates of target surface which form point clouds that capture the 3D shape of the targets. However, the active LiDAR pulse system can be easily affected by the scattering media such as particles of rain droplets and snow [228–231], leading to shifts of measured distances, variation of echo intensity, point missing, etc. Hence, point clouds captured under adverse weather usually have clear distribution discrepancy as compared with those

collected under normal weather as illustrated in Fig. 5.1. However, existing 3DSS benchmarks are dominated by normal-weather point clouds which are insufficient for the study of universal 3DSS under all-weather conditions. To this end, we propose SemanticSTF, a point-wise annotated large-scale adverse-weather dataset that can be explored for the study of 3DSS and point cloud parsing under various adverse weather conditions.

5.3.2 Data Selection and Split

We collect SemanticSTF by leveraging the STF benchmark [40], a multi-modal adverse-weather dataset that was jointly collected in Germany, Sweden, Denmark, and Finland. The data in STF have multiple modalities including LiDAR point clouds and they are collected under various adverse weather conditions such as snow and fog. However, STF provides bounding-box annotations only for the study of 3D detection tasks. In SemanticSTF, we manually selected 2,076 scans captured by a Velodyne HDL64 S3D LiDAR sensor from STF that cover various adverse weather conditions including 694 snowy, 637 dense-foggy, 631 light-foggy, and 114 rainy (all rainy LiDAR scans in STF). During the selection, we pay special attention to the geographical diversity of the point clouds aiming for minimizing data redundancy. We ignore the factor of daytime/nighttime since LiDAR sensors are robust to lighting conditions. We split SemanticSTF into three parts including 1,326 full 3D scans for training, 250 for validating, and 500 for testing. All three splits have approximately the same proportion of LiDAR scans of different adverse weathers.

5.3.3 Data Annotation

Point-wise annotation of LiDAR point clouds is an extremely laborious task due to several factors, such as 3D view changes, inconsistency between point cloud display and human visual perception, sweeping occlusion, point sparsity, etc. However, point-wise annotating of adverse-weather point clouds is even more challenging due to two new factors. *First*, the perceived distance shifts under adverse weather often lead to various geometry distortions in the collected points which make them different from those collected under normal weather. This presents significant

challenges for annotators who must recognize various objects and assign a semantic label to each point. *Second*, LiDAR point clouds collected under adverse weather often contain a significant portion of invalid regions that consist of indiscernible semantic contents (e.g., thick snow cover) that make it difficult to identify the ground type. The existence of such invalid regions makes point-wise annotation even more challenging.

We designed a customized labeling pipeline to handle the annotation challenges while performing point-wise annotation of point clouds in SemanticSTF. Specifically, we first provide labeling instructions and demo annotations and train a team of professional annotators to provide point-wise annotations of a set of selected STF LiDAR scans. To achieve reliable high-quality annotations, the annotators leverage the corresponding 2D camera images and Google Street views as extra references while identifying the category of each point in this initial annotation process. After that, the annotators cross-check their initial annotations for identifying and correcting labeling errors. At the final stage, we engaged professional third parties who provide another round of annotation inspection and correction.

Annotation of SemanticSTF is a highly laborious and time-consuming task. For instance, while labeling downtown areas with the most complex scenery, it took an annotator an average of 4.3 hours to label a single LiDAR scan. Labeling a scan captured in a relatively simpler scenery, such as a highway, also takes an average of 1.6 hours. In addition, an additional 30-60 minutes are required per scan for verification and correction by professional third parties. In total, annotating the entire SemanticSTF dataset takes over 6,600 man-hours.

While annotating SemanticSTF, we adopted the same set of semantic classes as in the widely-studied semantic segmentation benchmark, SemanticKITTI [7]. Specifically, we annotate the 19 evaluation classes of SemanticKITTI, which encompass most traffic-related objects in autonomous driving scenes. Additionally, following [220], we label points with indiscernible semantic contents caused by adverse weather (e.g. ground covered by snowdrifts) as *invalid*. Furthermore, we label points that do not belong to the 20 categories or are indistinguishable as *ignored*, which are not utilized in either training or evaluations. Detailed descriptions of each class can be found in the appendix.

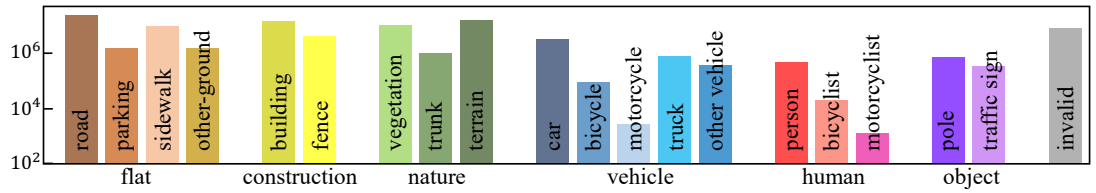


FIGURE 5.2: Number of annotated points per class in SemanticSTF.

5.3.4 Data Statistics

SemanticSTF consists of point-wise annotations of 21 semantic categories, and Fig. 5.2 shows the detailed statistics of the point-wise annotations. It can be seen that classes *road*, *sidewalk*, *building*, *vegetation*, and *terrain* appear most frequently whereas classes *motor*, *motorcyclist*, and *bicyclist* have clearly lower occurrence frequency. Such class imbalance is largely attributed to the various object sizes and unbalanced distribution of object categories in transportation scenes, and it is also very common in many existing benchmarks. Overall, the statistics and distribution of different object categories are similar to that of other 2D and 3D semantic segmentation benchmarks such as Cityscapes [232], ACDC [220], and SemanticKITTI [7].

To the best of our knowledge, SemanticSTF is the first large-scale adverse-weather 3DSS benchmark that provides high-quality point-wise annotations. Table 5.1 compares it with several existing point cloud datasets that have been widely adopted for the study of 3D detection and semantic segmentation. We can observe that existing datasets are either collected under normal weather conditions or collected for object detection studies with bounding-box annotations only. 3DSS benchmark under adverse weather is largely blank, mainly due to the great challenge in point-wise annotations of adverse-weather point clouds as described in previous subsections. From this sense, SemanticSTF fills up this blank by providing a large-scale benchmark and test bed which will be very useful to future research in universal 3DSS under all weather conditions.

5.3.5 Data Illustration

Fig. 5.3 provides examples of point cloud scans captured under adverse weather conditions in SemanticSTF (in row 1) as well as the corresponding annotations (in

TABLE 5.1: Comparison of SemanticSTF against existing outdoor LiDAR benchmarks. #Cls means the class number.

Dataset	#Cls	Type	Annotation	Fog	Rain	Snow
KITTI [37]	8	real	bounding box	✗	✗	✗
nuScenes [38]	23	real	bounding box	✗	✗	✗
Waymo [233]	4	real	bounding box	✗	✗	✗
STF [40]	5	real	bounding box	✓	✓	✓
SemanticKITTI [7]	25	real	point-wise	✗	✗	✗
nuScenes-LiDARSeg [147]	32	real	point-wise	✗	✗	✗
Waymo-LiDARSeg [233]	21	real	point-wise	✗	✗	✗
SynLiDAR [2]	32	synth.	point-wise	✗	✗	✗
SemanticSTF (ours)	21	real	point-wise	✓	✓	✓

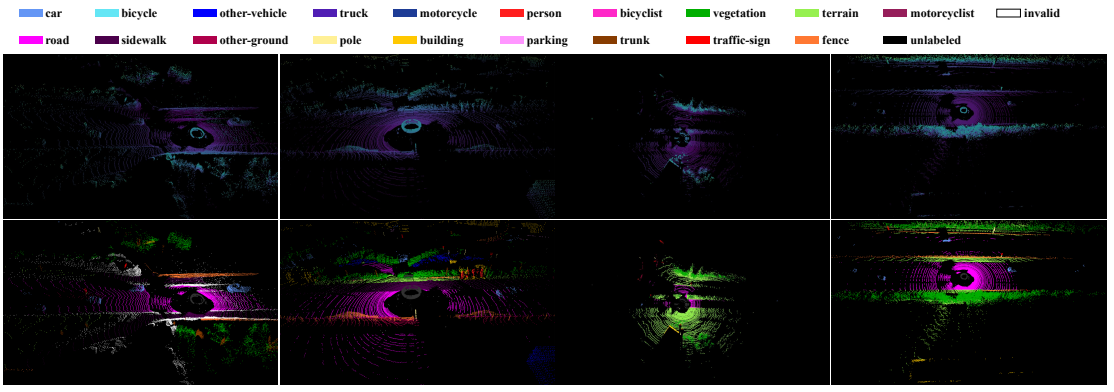


FIGURE 5.3: Examples of LiDAR point cloud scans captured under different adverse weather including snow, rain, dense fog, and light fog (the first row) and corresponding dense annotations in SemanticSTF (the second row).

row 2). Compared with normal-weather point clouds, point clouds captured under adverse weather exhibit four distinct properties: 1) Snow coverage and snowflakes under *snowy* weather introduce many white points (labeled as "invalid") as illustrated in Fig. 5.3(a). The thick snow coverage may lead to object deformation as well; *Rainy* conditions may cause specular reflection of laser signals from water on the ground and produce many noise points as shown in Fig.5.3(b); 3) *Dense fog* may greatly reduce the working range of LiDAR sensors, leading to small spatial distribution of the collected LiDAR points as illustrated in Fig. 5.3(c); 4) Point clouds under *light fog* have similar characteristics as normal-weather point clouds as illustrated in Fig. 5.3(d). The distinct properties of point clouds under different adverse weather introduce different types of domain shift from normal-weather point clouds which complicate 3DSS greatly as discussed in Section 5.5. They also verify the importance of developing universal 3DSS models that can perform well under all weather conditions.

5.4 Point Cloud Domain Randomization

Leveraging SemanticSTF, we explore domain generalization (DG) for semantic segmentation of LiDAR point clouds under all weather conditions. Specifically, we design PointDR, a domain randomization technique that helps to train a generalizable segmentation model from normal-weather point clouds that can work well for adverse-weather point clouds in SemanticSTF.

5.4.1 Problem Definition

Given labeled point clouds of a source domain $\mathcal{S} = \{S_k = \{x_k, y_k\}\}_{k=1}^{K}$ where x represents a LiDAR point cloud scan and y denotes its point-wise semantic annotations, the goal of domain generalization is to learn a segmentation model F by using the source-domain data only that can perform well on point clouds from an unseen target domain $\mathcal{T}$. We consider a 3D point cloud segmentation model F that consists of a feature extractor E and a classifier G. Note under the setup of domain generalization, target data will not be accessed in training as they could be hard and even impossible to acquire at the training stage.

5.4.2 Point Cloud Domain Randomization

Inspired by domain randomization studies in 2D computer vision research [234, 235], we explore how to employ domain randomization for learning domain generalizable models for point Specifically, we design PointDR, a point cloud randomization technique that consists of two complementary designs including *geometry style randomization* and *embedding aggregation* as illustrated in Fig. 5.4.

Geometry style randomization aims to enrich the geometry styles and expand the distribution of training point cloud data. Given a point-cloud scan x as input, we apply weak and strong spatial augmentation to obtain two copies of x including a weak-view $x^w = \mathcal{A}^W(x)$ and a strong-view $x^s = \mathcal{A}^S(x)$. For the augmentation schemes of $\mathcal{A}^W$, we follow existing supervised learning methods [6] and adopt the simple random rotation and random scaling. While for the augmentation schemes

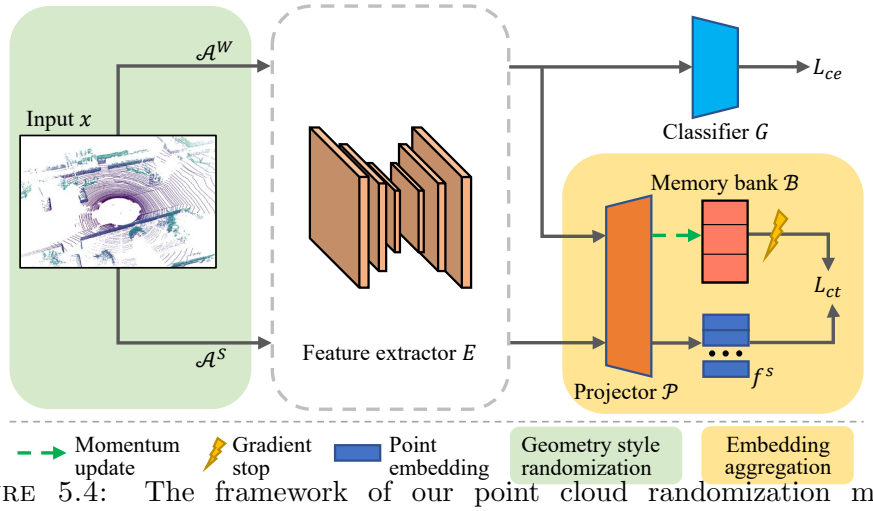


FIGURE 5.4: The framework of our point cloud randomization method (PointDR): *Geometry style randomization* creates different point cloud views with various spatial perturbations while *embedding aggregation* encourages the feature extractor to aggregate randomized point embeddings to learn perturbation-invariant representations, ultimately leading to a generalizable segmentation model.

of $\mathcal{A}^S$, we further adopt random dropout, random flipping, random noise perturbation, and random jittering on top of $\mathcal{A}^W$ to obtain a more diverse and complex copy of the input point cloud scan x.

Embedding aggregation aims to aggregate encoded embeddings of randomized point clouds for learning domain-invariant representations. We adopt contrastive learning [236] as illustrated in Fig. 5.4. Given the randomized point clouds x^w and x^s, we first feed them into the *feature extractor* E and a *projector* $\mathcal{P}$ (a two-layer MLP) which outputs normalized point feature embeddings f^w and f^s, respectively ($f = \mathcal{P}(E(x))$). $\overline{f}_C^w \in \mathbb{R}^{D\times C}$ (D: feature dimension; C: number of semantic classes) is then derived by class-wise averaging the feature embeddings f^w in a batch, which is stored in a memory bank $\mathcal{B} \in \mathbb{R}^{D\times C}$ that has no backpropagation and is momentum updated by iterations (i.e., $\mathcal{B} \leftarrow m \times \mathcal{B} + (1-m) \times \overline{f}_C^w$ with a momentum coefficient m). Finally, we employ each point feature embedding f_i^s of the strong-view f^s as query and feature embeddings in $\mathcal{B}$ as keys for contrastive learning, where the key sharing the same semantic class as the query is positive key $\mathcal{B}_+$ and the rest are negative keys. The contrastive loss is defined as

$$\mathcal{L}_{ct} = \frac{1}{N}\sum_{i=1}^{N} -\log \frac{\exp\left(f_i^s \mathcal{B}_+/\tau\right)}{\sum_{j=1}^{C}\exp\left(f_i^s \mathcal{B}_j/\tau\right)} \tag{5.1}$$

where τ is a temperature hyper-parameter [237]. Note there is no back-propagation for the "ignore" class in optimizing the contrastive loss.

Contrastive learning pulls point feature embeddings of the same classes closer while pushing away point feature embeddings of different classes. Therefore, optimizing the proposed contrastive loss will aggregate randomized point cloud features and learn perturbation-invariant representations, ultimately leading to a robust and generalizable segmentation model. The momentum-updated memory bank provides feature prototypes of each semantic class for more robust and stable contrastive learning.

Combining the supervised cross-entropy loss $\mathcal{L}_{ce}$ for weakly-augmented point clouds in Eq. 5.1, the overall training objective of PointDR can be formulated by:

$$\mathcal{L}_{\text{PointDR}} = \mathcal{L}_{ce} + \lambda_{ct}\mathcal{L}_{ct} \tag{5.2}$$

5.5 Experiments

SemanticSTF can be adopted for benchmarking different learning setups and network architectures on point cloud segmentation. We perform experiments over two typical learning setups including domain generalization and unsupervised domain adaptation. In addition, we evaluate several state-of-the-art point-cloud segmentation networks to examine their generalization capabilities.

5.5.1 Domain Generalization

We first study domain generalizable point cloud segmentation. For DG, we can only access an annotated source domain during training and the trained model is expected to generalize well to *unseen* target domains. Leveraging SemanticSTF, we build two DG benchmarks and examine how PointDR helps learn a universal 3DSS model that can work under different weather conditions.

The first benchmark is *SemanticKITTI [7]* $\rightarrow$ *SemanticSTF* where SemanticKITTI is a large-scale real-world 3DSS dataset collected under normal weather conditions.

This benchmark serves as a solid testing ground for evaluating domain generalization performance from normal to adverse weather conditions. The second benchmark is *SynLiDAR [2]* $\rightarrow$ *SemanticSTF* where SynLiDAR is a large-scale synthetic 3DSS dataset. The motivation of this benchmark is that learning a universal 3DSS model from synthetic point clouds that can work well across adverse weather is of high research and application value considering the challenges in point cloud collection and annotation. Note this benchmark is more challenging as the domain discrepancy comes from both normal-to-adverse weather distribution shift and synthetic-to-real distribution shift.

Setup. We use all 19 evaluating classes of SemanticKITTI in both domain generalization benchmarks. The category of *invalid* in SemanticSTF is mapped to the *ignored* since SemanticKITTI and SynLiDAR do not cover this category. We adopt MinkowskiNet [10] (with TorchSparse library [6]) as the backbone model, which is a sparse convolutional network that provides state-of-the-art performance with decent efficiency. We adopt the evaluation metrics of Intersection over the Union (IoU) for each segmentation class and the mean IoU (mIoU) over all classes. All experiments are run over a single NVIDIA 2080Ti (11GB). More implementation details are provided in the appendix.

Baseline Methods. Since domain generalizable 3DSS is far under-explored, there is little existing baseline that can be directly adopted for benchmarking. We thus select two closely related approaches as baseline to evaluate the proposed PointDR. The first approach is data augmentation and we select three related augmentation methods including *Dropout* [186] that randomly drops out points to simulate LiDAR points missing in adverse weather, *Noise perturbation* that adds random points in the 3D space to simulate noise points as introduced by particles like falling snow, and *PolarMix* [68] that mixes point clouds of different sources for augmentation. The second approach is to adapt 2D domain generalization methods for 3DSS. We select two 2D domain generalization methods including the widely studied *MMD* [224] and the recently proposed *PCL* [238].

Results. Table 5.2 shows experimental results over the validation set of SemanticSTF. For both benchmarks, the *Baseline* is a source-only model that is trained by using the training data of SemanticKITTI or SynLiDAR. We can see that the *Baseline* achieves very low mIoU while evaluated over the validation set of SemanticSTF, indicating the large domain discrepancy between point clouds of normal

TABLE 5.2: Experiments on domain generalization with SemanticKITTI [7] or SynLiDAR [2] as source and SemanticSTF as target.

Methods	car	bi.cle	mt.cle	truck	oth-v.	pers.	bi.clst	mt.clst	road	parki.	sidew.	oth-g.	build.	fence	veget.	trunk	terra.	pole	traf.	D-fog	L-fog	Rain	Snow	mIoU
Oracle	89.4	42.1	0.0	59.9	61.2	69.6	39.0	0.0	82.2	21.5	58.2	45.6	86.1	63.6	80.2	52.0	77.6	50.1	61.7	51.9	54.6	57.9	53.7	54.7
SemanticKITTI→SemanticSTF																								
Baseline	55.9	0.0	0.2	1.9	10.9	10.3	**6.0**	0.0	61.2	10.9	32.0	0.0	67.9	41.6	49.8	27.9	40.8	**29.6**	17.5	29.5	26.0	28.4	21.4	24.4
Dropout [186]	62.1	0.0	**15.5**	3.0	**11.5**	5.4	2.0	0.0	58.4	12.8	26.7	1.1	72.1	43.6	52.9	**34.2**	43.5	28.4	15.5	29.3	25.6	29.4	24.8	25.7
Perturbation	**74.4**	0.0	0.0	**23.3**	0.6	19.7	0.0	0.0	60.3	10.8	33.9	0.7	72.0	45.2	58.7	17.5	42.4	22.1	9.7	26.3	27.8	30.0	24.5	25.9
PolarMix [68]	57.8	**1.8**	3.8	16.7	3.7	26.5	0.0	**2.0**	65.7	2.9	32.5	0.3	71.0	**48.7**	53.8	20.5	45.4	25.9	15.8	29.7	25.0	28.6	25.6	26.0
MMD [224]	63.6	0.0	2.6	0.1	11.4	**28.1**	0.0	0.0	**67.0**	**14.1**	37.9	0.3	67.3	41.2	57.1	27.4	47.9	28.2	16.2	30.4	28.1	**32.8**	25.2	26.9
PCL [238]	65.9	0.0	0.0	17.7	0.4	8.4	0.0	0.0	59.6	12.0	35.0	**1.6**	**74.0**	47.5	**60.7**	15.8	**48.9**	26.1	**27.5**	28.9	27.6	30.1	24.6	26.4
PointDR (Ours)	67.3	0.0	4.5	19.6	9.0	18.8	2.7	0.0	62.6	12.9	**38.1**	0.6	73.3	43.8	56.4	32.2	45.7	28.7	27.4	**31.3**	**29.7**	31.9	**26.2**	**28.6**
SynLiDAR→SemanticSTF																								
Baseline	27.1	**3.0**	0.6	15.8	0.1	25.2	1.8	5.6	23.9	0.3	14.6	0.6	36.3	19.9	37.9	17.9	41.8	9.5	2.3	16.9	17.2	17.2	11.9	15.0
Dropout [186]	28.0	**3.0**	1.4	9.6	0.0	17.1	0.8	0.7	**34.2**	6.8	19.1	0.1	35.5	19.1	42.3	17.6	36.0	14.0	2.8	15.3	16.6	20.4	14.0	15.2
Perturbation	27.1	2.3	2.3	16.0	0.1	23.7	1.2	4.0	27.0	3.6	16.2	0.8	29.2	16.7	35.3	22.7	38.3	**17.9**	5.1	16.3	16.7	19.3	13.4	15.2
PolarMix [68]	**39.2**	1.1	1.2	8.3	**1.5**	17.8	0.8	0.7	23.3	1.3	17.5	0.4	45.2	24.8	**46.2**	20.1	38.7	7.6	1.9	16.1	15.5	19.2	15.6	15.7
MMD [224]	25.5	2.3	2.1	13.2	0.7	22.1	1.4	7.5	30.8	0.4	17.6	0.2	30.9	19.7	37.6	19.3	**43.5**	9.9	2.6	17.3	16.3	20.0	12.7	15.1
PCL [238]	30.9	0.8	1.4	10.0	0.4	23.3	**4.0**	**7.9**	28.5	1.3	**17.7**	**1.2**	39.4	18.5	40.0	16.0	38.6	12.1	2.3	17.8	16.7	19.3	14.1	15.5
PointDR (Ours)	37.8	2.5	**2.4**	**23.6**	0.1	**26.3**	2.2	3.3	27.9	**7.7**	17.5	0.5	**47.6**	**25.3**	45.7	**21.0**	37.5	**17.9**	**5.5**	**19.5**	**19.9**	**21.1**	**16.9**	**18.5**

and adverse weather conditions. In addition, all three data augmentation methods improve the model generalization consistently but the performance gains are limited especially for the challenging benchmark SynLiDAR→ SemanticSTF. The two 2D generalization methods both help SemanticKITTI → SemanticSTF clearly but show very limited improvement over SynLiDAR → SemanticSTF. The proposed PointDR achieves the best generalization consistently across both benchmarks, demonstrating its superior capability to learn perturbation-invariant point cloud representations and effectiveness while handling all-weather 3DSS tasks.

We also evaluate the compared domain generalization methods over each individual adverse weather condition as shown in Table 5.2. It can be observed that the three data augmentation methods work for data captured in rainy and snowy weather only. The 2D generalization method MMD shows clear effectiveness for point clouds under dense fog and rain while PCL works for point clouds under rainy and snowy weather instead. We conjecture that the performance variations are largely attributed to the different properties of point clouds captured under different weather conditions. For example, more points are missing in rain while object points often deform due to the covered snow (more illustrations are provided in the appendix). Such data variations lead to different domain discrepancies across weather which further leads to different performances of the compared methods. As PointDR learns perturbation-tolerant representations, it works effectively across different adverse weather conditions. We also provide qualitative results, please refer to the appendix for details.

TABLE 5.3: Ablation study of PointDR over domain generalized segmentation task SemanticKITTI→SemanticSTF.

Method	$\mathcal{L}_{ce}$	$\mathcal{L}_{ct}$	$\mathcal{B}$	mIoU
Baseline	✓			24.4
PointDR-CT	✓	✓		27.4
PointDR	✓	✓	✓	**28.6**

TABLE 5.4: Comparison of state-of-the-art domain adaptation methods on SemanticKITTI → SemanticSTF adaptation. SemanticKITTI serves as the source domain and the entire SemanticSTF including all four weather conditions serves as the target domain.

Methods	car	bi.cle	mt.cle	truck	oth-v.	pers.	bi.clst	mt.clst	road	parki.	sidew.	oth-g.	build.	fence	veget.	trunk	terra.	pole	traf.	mIoU
Oracle	89.4	42.1	0.0	59.9	61.2	69.6	39.0	0.0	82.2	21.5	58.2	45.6	86.1	63.6	80.2	52.0	77.6	50.1	61.7	54.7
Source-only	64.8	0.0	0.0	13.8	1.8	5.0	2.1	0.0	62.7	7.5	34.0	0.0	66.7	36.2	53.9	31.3	44.3	24.0	14.2	24.3
ADDA [190]	65.6	0.0	0.0	21.0	1.3	2.8	1.3	**16.7**	64.7	1.2	35.4	0.0	66.5	41.8	**57.2**	32.6	42.2	23.3	26.4	26.3
Ent-Min [149]	69.2	0.0	10.1	31.0	5.3	2.8	**2.6**	0.0	**65.9**	2.6	35.7	0.0	**72.5**	**42.8**	52.4	32.5	**44.7**	24.7	21.1	27.2
Self-training [84]	**71.5**	0.0	10.3	**33.1**	7.4	5.9	1.3	0.0	65.1	6.5	**36.6**	0.0	67.8	41.3	51.7	**32.9**	42.9	25.1	25.0	27.6
CoSMix [4]	65.0	**1.7**	**22.1**	25.2	**7.7**	**33.2**	0.0	0.0	64.7	**11.5**	31.1	**0.9**	62.5	37.8	44.6	30.5	41.1	**30.9**	**28.6**	**28.4**

Ablation study. We study different PointDR designs to examine how they contribute to the overall generalization performance. As Table 5.3 shows, we report three models over the benchmark "SemanticKITTI → SemanticSTF": 1) *Baseline* that is trained with $\mathcal{L}_{ce}$. 2) *PointDR-CT* that is jointly trained with $\mathcal{L}_{ce}$ and $\mathcal{L}_{ct}$ without using the memory bank $\mathcal{B}$. 3) The complete *PointDR* that is trained with $\mathcal{L}_{ce}$, $\mathcal{L}_{ct}$ and the memory bank $\mathcal{B}$. We evaluate the three models over the validation set of SemanticSTF and Table 5.3 shows experimental results. We can see that the *Baseline* performs poorly at 24.4% due to clear domain discrepancy between point clouds of normal weather and adverse weather. Leveraging the proposed contrastive loss, $\mathcal{L}_{ct}$ achieves clearly better performance at 27.4%, indicating that learning perturbation-invariance is helpful for universal LiDAR segmentation of all-weather conditions. On top of that, introducing the momentum-updated memory bank $\mathcal{B}$ further improves the segmentation performance at 28.6%. This is because the feature embeddings in $\mathcal{B}$ serve as the class prototypes which help the optimization of the segmentation network, finally leading to more robust representations of 3DSS that perform better over adverse weather point clouds.

5.5.2 Domain Adaptation

We also study SemanticSTF over a domain adaptive point cloud segmentation benchmark SemanticKITTI → SemanticSTF. Specifically, we select four representative UDA methods including ADDA [190], entropy minimization (Ent-Min) [149], self-training [84], and CoSMix [4] for adaptation from the source SemanticKITTI [7] toward the target SemanticSTF. Following the state-of-the-art [2, 4, 68] on synthetic-to-real adaptation, we adopt MinkowskiNet [10] as the segmentation backbone for all compared methods. Table 5.4 shows experimental results over the validation set of SemanticSTF. We can see that all UDA methods outperform the *Source-only* consistently under the normal-to-adverse adaptation setup. At the other end, the performance gains are still quite limited, showing the great improvement space along domain adaptive 3DSS from normal to adverse weather conditions.

In addition, we examined the adaptability of the four UDA methods in relation to each individual adverse weather condition. Specifically, we trained each of the four methods for adaptation from SemanticKITTI to SemanticSTF data for each adverse weather condition. Table 5.5 shows the experimental results over the validation set of SemanticSTF. We can see all four methods outperform the *Source-only* method under *Dense-fog* and *Light-fog*, demonstrating their effectiveness in mitigating domain discrepancies. However, for *rain* and *Snow*, only CoSMix achieved marginal performance gains while the other three UDA methods achieved limited performance improvements. We conjecture that snow and rain introduce large deformations on object surfaces or much noise, making adaptation from normal to adverse weather more challenging. CoSMix works in the input space by directly mixing source and target points, allowing it to perform better under heavy snow and rain which have larger domain gaps. However, all methods achieved relatively low segmentation performance, indicating the significance of our research and the large room for improvement in our constructed benchmarks.

5.5.3 Network Models vs All-Weather 3DSS

We also study how different 3DSS network architectures generalize when they are trained with normal-weather point clouds and evaluated over SemanticSTF. Specifically, we select five representative 3DSS networks [6, 47, 127, 164] that have been

TABLE 5.5: Comparison of state-of-the-art domain adaptation methods on SemanticKITTI → SemanticSTF adaptation for individual adverse weather conditions. We train a separate model for each weather-specific subset of SemanticSTF and evaluate the trained model on the weather condition it has been trained for.

Method	Dense-fog	Light-fog	Rain	Snow
Source-Only	26.9	25.2	27.7	23.5
ADDA [190]	31.5	27.9	27.4	23.4
Ent-Min [149]	31.4	28.6	30.3	24.9
Self-training [84]	**31.8**	29.3	27.9	25.1
CoSMix [4]	31.6	**30.3**	**33.1**	**32.9**

TABLE 5.6: Performance of state-of-the-art 3DSS models that are pre-trained over SemanticKITTI and tested on validation set of SemanticSTF for individual weather conditions and jointly for *all* weather conditions.

3DSS Model	D-fog	L-fog	Rain	Snow	All
RandLA-Net [47]	26.5	**26.0**	**25.1**	22.7	**25.3**
SalsaNext [164]	16.0	9.6	7.8	3.5	9.1
SPVCNN [6]	**30.4**	22.8	21.7	**18.3**	22.4
SPVNAS [6]	25.5	18.3	17.0	13.0	18.0
Cylinder3D [127]	14.8	7.4	5.7	4.0	7.3

widely adopted in 3D LiDAR segmentation studies. In the experiments, each selected network is first pre-trained with SemanticKITTI [7] and then evaluated over the validation set of SemanticSTF. We directly use the officially released code and the pre-trained weights for evaluation. Table 5.6 shows experimental results. We can observe that the five pre-trained models perform very differently though they all achieve superior segmentation over SemanticKITTI. Specifically, RandLA-Net [47], SPVCNN [6], and SPVNAS [6] perform clearly better than SalsaNext [164] and Cylinder3D [127]. In addition, none of the five pre-trained models perform well, verifying the clear domain discrepancy between point clouds of normal and adverse weather conditions. The experiments further indicate the great value of SemanticSTF in the future exploration of robust point cloud parsing under all weather conditions.

5.5.4 Supervised Learning on Adverse Weather Conditions

We use SemanticSTF to train five state-of-the-art 3DSS models in a supervised manner and report their segmentation performance in Table 5.7. Specifically, we use their officially released codes and default training configurations for model

TABLE 5.7: Comparison of state-of-the-art 3DSS methods (trained in a supervised manner) over the test set of SemanticSTF.

Methods	car	bi.cle	mt.cle	truck	oth-v.	pers.	bi.clst	mt.clst	road	parki.	sidew.	oth-g.	build.	fence	veget.	trunk	terra.	pole	traf.	invalid	mIoU
RandLA-Net [47]	75.2	0.0	0.0	25.8	0.0	47.3	0.0	0.0	73.3	7.8	48.7	57.5	68.2	48.3	61.5	27.3	49.5	39.7	27.5	56.5	35.7
SalsaNext [164]	77.3	31.2	0.0	47.5	30.5	64.2	26.6	5.0	76.3	18.2	55.2	64.9	79.2	50.4	56.8	27.8	55.8	36.8	36.7	62.2	45.1
MinkowskiNet [10]	87.4	42.5	0.0	51.2	40.3	73.6	29.1	0.0	79.5	15.0	57.7	63.4	78.6	56.8	64.4	40.4	53.3	47.6	47.6	67.7	49.8
SPVCNN [6]	87.1	45.5	0.0	53.1	42.7	74.1	21.9	0.0	78.9	16.3	57.9	57.0	78.6	56.5	65.6	40.9	50.3	49.4	45.9	66.4	49.4
Cylinder3D [127]	77.7	31.7	2.7	43.4	23.8	67.8	18.4	0.0	78.5	10.0	51.8	48.7	81.2	56.0	63.4	38.3	52.1	48.0	43.0	63.9	45.0

training. We can see that these state-of-the-art models achieve much lower segmentation performance over SemanticSTF as compared with their performance over SemanticKITTI. The results indicate that SemanticSTF is a more challenging benchmark for supervised methods due to the diverse data distribution and hard geometric domains. In addition, comparing Table 5.7 and Table 6 of the submitted paper, we notice that the rankings of the supervised and the pre-trained 3DSS models are not well aligned, indicating that the ability of supervised representation learning may not be highly correlated with the generalization ability. We also notice that the state-of-the-art network Cylinder3D [127] achieves much lower segmentation performance over SemanticSTF as compared with its performance over SemanticKITTI. This could be due to two major factors: 1) The design of Cylinder3D is sensitive to complicated and noisy geometries of point clouds as introduced by various adverse weather conditions; 2) Cylinder3D is sensitive to training parameters and the default training configurations for SemanticKITTI does not work well for SemanticSTF. The results further demonstrate the importance of studying universal 3DSS as well as the value of the proposed SemanticSTF dataset in steering the future endeavour along this very meaningful research direction.

5.6 Conclusion

This chapter presents SemanticSTF, a large-scale dataset and benchmark suite for semantic segmentation of LiDAR point clouds under adverse weather conditions. SemanticSTF provides high-quality point-level annotations for point clouds captured under adverse weather including dense fog, light fog, snow, and rain. Extensive studies have been conducted to examine how state-of-the-art 3DSS methods perform over SemanticSTF, demonstrating its significance in directing future research on domain adaptive and domain generalizable 3DSS under all-weather conditions.

We also design PointDR, a domain randomization technique that aims to use normal-weather point clouds to train a domain generalizable 3DSS model that can work well over adverse-weather point clouds. PointDR consists of two novel designs including geometry style randomization and embedding aggregation which jointly learn perturbation-invariant representations that generalize well to various new point-cloud domains. Extensive experiments show that PointDR achieves superior point cloud segmentation performance as compared with the state-of-the-art.

Chapter 6

Conclusion and Future Directions

6.1 Conclusion

This thesis investigated how to enable computers or systems to understand the 3D world in a resource-efficient manner. Our specific focus was on label-efficient learning of point cloud recognition, which involves using minimal annotation efforts to train deep recognition networks capable of accurately, generically, and robustly understanding 3D point clouds. This research has the potential to benefit both the deep learning community and society as a whole. By requiring fewer annotation efforts, we can reduce electric power consumption and labor costs. Additionally, this approach helps establish a more dependable artificial intelligent system, particularly in scenarios where limited annotated training data is available.

In particular, this thesis delved into the realm of label-efficient learning for point cloud recognition from two different angles: data augmentation and domain transfer learning. Data augmentation plays a pivotal role in expanding and diversifying a labelled training dataset, ultimately improving the model's ability to generalize and perform well on unseen data. This technique is particularly valuable in scenarios where the available training dataset is limited or where overfitting is a concern. By effectively reducing the need for extensive point cloud annotation, data augmentation facilitates a more efficient model training process, even when working with a smaller number of labelled samples.

In parallel, domain transfer learning plays a crucial role in leveraging knowledge from labelled source domains to develop networks that excel in recognizing unlabelled target domains, without requiring any additional annotations for the target domain. This approach is highly label-efficient as it enables us to fully utilize existing labelled datasets and eliminate the need for annotating new data. In this thesis, we investigate domain transfer learning from two perspectives including domain generalization and unsupervised domain adaptation. The research focuses on two representative types of domain transfer learning: learning from synthetic-to-real point clouds and learning from normal-to-adverse weather point clouds.

Firstly, for data augmentation, we presented a point cloud augmentation technique named PolarMix, which is simple and generic but can mitigate the data constraint effectively across different 3D perception tasks and scenarios. PolarMix enriches point cloud distributions and preserves point cloud fidelity via two cross-scan augmentation strategies that cut, edit, and mix point clouds along the scanning direction. The first is scene-level swapping which exchanges point cloud sectors of two LiDAR scans that are cut along the azimuth axis. The second is instance-level rotation and paste which crops point instances from one LiDAR scan, rotates them by multiple angles (to create multiple copies), and paste the rotated point instances into other scans. In this way, we enhance training and improve the generalization of point cloud recognition models in a label-efficient manner.

Secondly, for domain transfer learning from synthetic-to-real point clouds, we presented one dataset and two novel techniques for effective 3D knowledge transfer. Specifically, we first collected SynLiDAR, the first large-scale synthetic LiDAR point cloud dataset that provides point-wise annotations for 3D semantic segmentation (3DSS). On top of SynLiDAR, we design two techniques for domain transfer learning: The first one works on the input space. Specifically, we designed PCT, a point-cloud translator that translates synthetic point cloud to have real style. PCT disentangles the domain gap between synthetic and real point clouds into an appearance component and a sparsity component. It incorporates an appearance translation module and a sparsity translation module to handle the two gap components separately. The second technique works on the output space. Specifically, We presented a simple yet effective spatial consistency training framework (SCT) with three types of spatial consistency, namely, geometric-transform consistency, sparsity consistency, and mixing consistency, which captures the semantic invariance

of point clouds with respect to viewpoint changes, sparsity changes, and local context changes, respectively, ultimately leading networks to learn domain-invariant feature representations from unlabelled target point clouds.

Finally, for domain transfer learning from normal-to-adverse weather point clouds, we also presented a dataset and a technique. Specifically, we introduced SemanticSTF, an adverse-weather point cloud dataset that provides dense point-level annotations. Based on SemanticSTF, We study domain adaptive 3DSS that adapts from normal-weather data to adverse-weather data, as well as domain generalizable 3DSS that learns all-weather 3DSS models from normal-weather data. In addition, we design a domain randomization technique named PointDR that alternatively randomizes the geometry styles of point clouds and aggregates their embeddings, ultimately leading to a generalizable model that can improve 3DSS under various adverse weather effectively.

6.2 Future Directions

This thesis introduces a collection of innovative label-efficient learning methods that empower computers and systems to learn and recognize 3D environments using point cloud data while minimizing the need for extensive annotations. Extensive experiments conducted on various benchmarks demonstrate the advantages of the proposed label-efficient learning techniques in the continuous development of deep learning. For instance, reduced reliance on additional supervision helps lower labor costs, shorter learning times contribute to reduced electric power consumption, and improved performance facilitates the establishment of more reliable artificial intelligent systems. However, despite these advancements, there are still numerous unexplored label-efficient learning issues and setups. Therefore, effective label-efficient learning for point cloud recognition remains an open research field that requires further investigation and future work.

One direction is weakly-supervised learning (WSL). WSL, as an alternative to fully-supervised learning, leverages weak supervision for network training. Collecting weak annotations often reduces the annotation cost and time significantly, making WSL an important branch of label-efficient learning. Specifically, there are

three types of weak supervision that can exploit for training point cloud recognition models, including *incomplete* supervision, *inexact* supervision, and *inaccurate* supervision. Incomplete supervision involves only a small subset of training samples being labelled while the rest are unlabelled. Inexact supervision refers to only coarse-grained labels being provided, which may not match the model output. Inaccurate supervision occurs when the given labels are noisy and not always ground-truth. Preparing precise supervision is very challenging for unordered and unstructured point clouds, which showcases the importance of weakly supervised point cloud learning.

Furthermore, another noteworthy direction is self-supervised learning, which focuses on acquiring effective and comprehensive representation structures from extensive, unlabelled point clouds. By utilizing this approach, the learned parameters can then be utilized to initialize downstream networks, facilitating faster convergence and effective learning from small task data, hence reducing the effort of annotating downstream task-specific data significantly.

In addition, the recent advance of vision-language foundation models has yielded great breakthroughs in AI fields, which are trained with image-text pairs and have demonstrated remarkable zero-shot visual prediction performance, being able to recognize objects of novel concepts with impressive accuracy without involving any labelled images but only text illustrations in training. How can we transfer the knowledge gained from these large foundation models into the realm of point clouds? Is it possible to extend the image-text pretraining paradigm to incorporate point cloud-text relationships?

We plan to explore these novel and challenging tasks in our future research.

List of Author's Publications[1]

Journal Articles

- **Aoran Xiao***, Jiaxing Huang*, Dayan Guan, Xiaoqin Zhang, Shijian Lu. "Unsupervised Representation Learning for Point Clouds with Deep Neural Networks: A Survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2023, doi: 10.1109/TPAMI.2023.3262786. Accepted.

- **Aoran Xiao**, Xiaofei Yang, Shijian Lu, Dayan Guan, Jiaxing Huang . "FPS-Net: A convolutional fusion network for large-scale LiDAR point cloud segmentation", *ISPRS journal of Photogrammetry and Remote Sensing*, 176 (2021): 237-249. doi: https://doi.org/10.1016/j.isprsjprs.2021.04.011. Accepted.

- **Aoran Xiao**, Xiaoqin Zhang, Ling Shao, Shijian Lu. "A Survey of Label-Efficient Deep Learning for 3D Point Clouds". arXiv preprint arXiv:2305.19812, 2023. Submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, under review.

- **Aoran Xiao**, Dayan Guan, Shijian Lu. "Domain Adaptive LiDAR Point Cloud Segmentation with 3D Spatial Consistency". *IEEE Transactions on Multimedia (TMM)*, 2023. doi: 10.1109/TMM.2023.3335879. Accepted.

- **Aoran Xiao**, Jiaxing Huang, Kangcheng Liu, Dayan Guan, Xiaoqin Zhang, Shijian Lu. "Domain Adaptive LiDAR Point Cloud Segmentation via Density-Aware Self-Training". Submitted to *IEEE Transactions on Intelligent Transportation Systems (TITS)*, under review.

[1]The superscript * indicates joint first authors

Conference Proceedings

- **Aoran Xiao**, Jiaxing Huang, Weihao Xuan, Ruijie Ren, Kangcheng Liu, Dayan Guan, Abdulmotaleb El Saddik, Shijian Lu, Eric Xing. "3D Semantic Segmentation in the Wild: Learning Generalized Models for Adverse-Condition Point Clouds", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 9382-9392.

- **Aoran Xiao**, Jiaxing Huang, Dayan Guan, Kaiwen Cui, Shijian Lu, Ling Shao. "PolarMix: A General Data Augmentation Technique for LiDAR Point Clouds", *Advances in Neural Information Processing Systems (NeurIPS)*, 2022, 35: 11035-11048.

- **Aoran Xiao**, Jiaxing Huang, Dayan Guan, Fangneng Zhan, Shijian Lu. "Transfer Learning from Synthetic to Real LiDAR Point Cloud for Semantic Segmentation ", *AAAI Conference on Artificial Intelligence (AAAI), 2022. DOI:https://doi.org/10.1609/aaai.v36i3.20183.*.

Bibliography

[1] Yancheng Pan, Biao Gao, Jilin Mei, Sibo Geng, Chengkun Li, and Huijing Zhao. Semanticposs: A point cloud dataset with large quantity of dynamic instances. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 687–693. IEEE, 2020. xix, 28, 33, 44, 46, 47, 55, 61, 63, 65, 72

[2] Aoran Xiao, Jiaxing Huang, Dayan Guan, Fangneng Zhan, and Shijian Lu. Transfer learning from synthetic to real lidar point cloud for semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2795–2803, 2022. xix, xxiv, 14, 18, 20, 22, 29, 39, 40, 46, 47, 62, 63, 64, 65, 72, 73, 74, 75, 86, 89, 93, 97, 98, 100

[3] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017. xix, 64, 66, 67

[4] Cristiano Saltori, Fabio Galasso, Giuseppe Fiameni, Nicu Sebe, Elisa Ricci, and Fabio Poiesi. Cosmix: Compositional semantic mix for domain adaptation in 3d lidar segmentation. In *European Conference on Computer Vision*, pages 586–602. Springer, 2022. xix, 22, 39, 47, 62, 63, 65, 66, 72, 73, 74, 75, 76, 77, 80, 89, 99, 100, 101

[5] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. xix, 80

[6] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII*, pages 685–702. Springer, 2020. xxiv, 12, 26, 33, 34, 35, 41, 44, 47, 69, 73, 81, 82, 86, 88, 94, 97, 100, 101, 102

[7] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019. xxiv, 13, 14, 28, 33, 44, 46, 47, 49, 55, 64, 72, 86, 91, 92, 93, 96, 98, 100, 101

[8] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2, 10, 11, 16, 47, 56, 88

[9] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 2, 12

[10] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 2, 12, 33, 34, 35, 47, 69, 73, 88, 97, 100, 102

[11] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019. 10, 12

[12] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–779, 2019. 10

[13] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous robots*, 34(3):189–206, 2013. 11

[14] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)*, 32(6):1–11, 2013. 11

[15] 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. 11

[16] Tan Yu, Jingjing Meng, and Junsong Yuan. Multi-view harmonized bilinear network for 3d object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 186–194, 2018.

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

[18] Ze Yang and Liwei Wang. Learning relationships for multi-view 3d object recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7505–7514, 2019.

[19] Xin Wei, Ruixuan Yu, and Jian Sun. View-gcn: View-based graph convolutional network for 3d shape analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1850–1859, 2020.

[20] Aoran Xiao, Xiaofei Yang, Shijian Lu, Dayan Guan, and Jiaxing Huang. Fps-net: A convolutional fusion network for large-scale lidar point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 176: 237–249, 2021. 11, 16, 26, 47, 53, 88

[21] Qiangui Huang, Weiyue Wang, and Ulrich Neumann. Recurrent slice networks for 3d segmentation of point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2626–2635, 2018. 11

[22] Xiaoqing Ye, Jiamao Li, Hexiao Huang, Liang Du, and Xiaolin Zhang. 3d recurrent neural networks with context fusion for point cloud semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 403–417, 2018.

[23] Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 900–909, 2017. 11

[24] Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 3d point capsule networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1009–1018, 2019. 11

[25] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19313–19322, 2022. 11

[26] 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*, 2020. 11, 13, 65

[27] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 12, 16, 47

[28] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9224–9232, 2018. 12

[29] Haotian Tang, Zhijian Liu, Xiuyu Li, Yujun Lin, and Song Han. TorchSparse: Efficient Point Cloud Inference Engine. In *Conference on Machine Learning and Systems (MLSys)*, 2022. 12, 47

[30] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8895–8904, 2019. 13

[31] 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. 14, 20, 24, 28

[32] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 14, 20, 28

[33] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1588–1597, 2019. 14

[34] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015. 14

[35] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1543, 2016. 14

[36] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 14, 20, 28

[37] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 14, 93

[38] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 14, 47, 93

[39] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 14

[40] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11682–11692, 2020. 14, 86, 88, 90, 93

[41] Jiageng Mao, Minzhe Niu, Chenhan Jiang, Hanxue Liang, Jingheng Chen, Xiaodan Liang, Yamin Li, Chaoqiang Ye, Wei Zhang, Zhenguo Li, et al. One million scenes for autonomous driving: Once dataset. *arXiv preprint arXiv:2106.11037*, 2021. 14

[42] Timo Hackel, Nikolay Savinov, Jan D Wegner, Konrad Schindler, Marc Pollefeys, et al. Semantic3d. net: A new large-scale point cloud classification benchmark. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 4, pages 91–98. ISPRS Foundation, 2017. 14

[43] Qingyong Hu, Bo Yang, Sheikh Khalid, Wen Xiao, Niki Trigoni, and Andrew Markham. Towards semantic segmentation of urban-scale 3d point clouds: A dataset, benchmarks and challenges. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4977–4987, 2021. 14

[44] Aoran Xiao, Jiaxing Huang, Weihao Xuan, Ruijie Ren, Kangcheng Liu, Dayan Guan, Abdulmotaleb El Saddik, Shijian Lu, and Eric P. Xing. 3d semantic segmentation in the wild: Learning generalized models for adverse-condition point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9382–9392, June 2023. 14, 15, 24, 65

[45] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019. 15

[46] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019. 16

[47] 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. 26, 33, 38, 44, 47, 69, 86, 88, 100, 101, 102

[48] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *European conference on computer vision*, pages 574–591. Springer, 2020. 16, 66, 69

[49] Martin Hahner, Dengxin Dai, Alexander Liniger, and Luc Van Gool. Quantifying data augmentation for lidar based 3d object detection. *arXiv preprint arXiv:2004.01643*, 2020. 16

[50] Shivanand Venkanna Sheshappanavar, Vinit Veerendraveer Singh, and Chandra Kambhamettu. Patchaugment: Local neighborhood augmentation in point cloud classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2118–2127, 2021. 28

[51] Jaeseok Choi, Yeji Song, and Nojun Kwak. Part-aware data augmentation for 3d object detection in point cloud. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3391–3397. IEEE, 2021. 16

[52] Ruihui Li, Xianzhi Li, Pheng-Ann Heng, and Chi-Wing Fu. Pointaugment: an auto-augmentation framework for point cloud classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6378–6387, 2020. 16, 28, 48, 56

[53] Sihyeon Kim, Sanghyeok Lee, Dasol Hwang, Jaewon Lee, Seong Jae Hwang, and Hyunwoo J Kim. Point cloud augmentation with weighted local transformations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 548–557, 2021. 16, 28

[54] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 17, 26, 28, 41

[55] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR, 2019. 17, 28

[56] Yunlu Chen, Vincent Tao Hu, Efstratios Gavves, Thomas Mensink, Pascal Mettes, Pengwan Yang, and Cees GM Snoek. Pointmixup: Augmentation for point clouds. In *European Conference on Computer Vision*, pages 330–345. Springer, 2020. 17, 29, 41, 48

[57] Dogyoon Lee, Jaeha Lee, Junhyeop Lee, Hyeongmin Lee, Minhyeok Lee, Sungmin Woo, and Sangyoun Lee. Regularization strategy for point cloud via rigidly mixed sample. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15900–15909, 2021. 17, 29

[58] Jinlai Zhang, Lyujie Chen, Bo Ouyang, Binbin Liu, Jihong Zhu, Yujin Chen, Yanmei Meng, and Danfeng Wu. Pointcutmix: Regularization strategy for point cloud classification. *Neurocomputing*, 505:58–67, 2022. 17

[59] Sanghyeok Lee, Minkyu Jeon, Injae Kim, Yunyang Xiong, and Hyunwoo J Kim. Sagemix: Saliency-guided mixup for point clouds. *arXiv preprint arXiv:2210.06944*, 2022. 17

[60] Ardian Umam, Cheng-Kun Yang, Yung-Yu Chuang, Jen-Hui Chuang, and Yen-Yu Lin. Point mixswap: Attentional point cloud mixing via swapping matched structural divisions. In *European Conference on Computer Vision*, pages 596–611. Springer, 2022. 17

[61] Shuyang Cheng, Zhaoqi Leng, Ekin Dogus Cubuk, Barret Zoph, Chunyan Bai, Jiquan Ngiam, Yang Song, Benjamin Caine, Vijay Vasudevan, Congcong Li, et al. Improving 3d object detection through progressive population based augmentation. In *European Conference on Computer Vision*, pages 279–294. Springer, 2020. 17

[62] Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Wenqiang Zhang, Qian Zhang, Chang Huang, and Wenyu Liu. Azinorm: Exploiting the radial symmetry of point cloud for azimuth-normalized 3d perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6387–6396, 2022. 17

[63] Zhaoqi Leng, Shuyang Cheng, Benjamin Caine, Weiyue Wang, Xiao Zhang, Jonathon Shlens, Mingxing Tan, and Dragomir Anguelov. Pseudoaugment: Learning to use unlabeled data for data augmentation in point clouds. In *European Conference on Computer Vision*, pages 555–572. Springer, 2022. 17

[64] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 17, 26, 29, 39

[65] Jianhua Sun, Hao-Shu Fang, Xianghui Zhu, Jiefeng Li, and Cewu Lu. Correlation field for boosting 3d object detection in structured scenes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(2):2298–2306, Jun. 2022. 17

[66] Wu Zheng, Li Jiang, Fanbin Lu, Yangyang Ye, and Chi-Wing Fu. Boosting single-frame 3d object detection by simulating multi-frame point clouds. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4848–4856, 2022. 17

[67] Alexey Nekrasov, Jonas Schult, Or Litany, Bastian Leibe, and Francis Engelmann. Mix3d: Out-of-context data augmentation for 3d scenes. In *2021 International Conference on 3D Vision (3DV)*, pages 116–125. IEEE, 2021. 18, 29, 34, 35, 41

[68] Aoran Xiao, Jiaxing Huang, Dayan Guan, Kaiwen Cui, Shijian Lu, and Ling Shao. Polarmix: A general data augmentation technique for lidar point clouds. In *Advances in Neural Information Processing Systems*, 2022. 18, 22, 63, 65, 72, 73, 74, 75, 89, 97, 98, 100

[69] Jin Fang, Xinxin Zuo, Dingfu Zhou, Shengze Jin, Sen Wang, and Liangjun Zhang. Lidar-aug: A general rendering-based augmentation framework for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4710–4720, 2021. 18, 29

[70] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 18, 28

[71] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018.

[72] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 641–656, 2018.

[73] Tengteng Huang, Zhe Liu, Xiwu Chen, and Xiang Bai. Epnet: Enhancing point features with image semantics for 3d object detection. In *European Conference on Computer Vision*, pages 35–52. Springer, 2020.

[74] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11794–11803, 2021.

[75] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4604–4612, 2020. 18

[76] Xu Yan, Jiantao Gao, Chaoda Zheng, Chao Zheng, Ruimao Zhang, Shuguang Cui, and Zhen Li. 2dpass: 2d priors assisted semantic segmentation on lidar point clouds. In *European Conference on Computer Vision*, pages 677–695. Springer, 2022. 18

[77] Bin Yang, Runsheng Guo, Ming Liang, Sergio Casas, and Raquel Urtasun. Radarnet: Exploiting radar for robust perception of dynamic objects. In *European Conference on Computer Vision*, pages 496–512. Springer, 2020. 18

[78] Kun Qian, Shilin Zhu, Xinyu Zhang, and Li Erran Li. Robust multimodal vehicle detection in foggy weather using complementary lidar and radar signals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 444–453, 2021. 18

[79] Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern recognition*, 45(1):521–530, 2012. 19

[80] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010. 19

[81] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018. 19, 20

[82] 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. 19, 40

[83] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 289–305, 2018. 20, 39, 74, 75

[84] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5982–5991, 2019. 20, 39, 40, 99, 100, 101

[85] Aoran Xiao, Jiaxing Huang, Dayan Guan, Xiaoqin Zhang, Shijian Lu, and Ling Shao. Unsupervised point cloud representation learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2023. doi: 10.1109/TPAMI.2023.3262786. 20

[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, 2018. 20

[87] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1588–1597, 2019. 20, 24

[88] 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. 20, 39, 48

[89] Longkun Zou, Hui Tang, Ke Chen, and Kui Jia. Geometry-aware self-training for unsupervised domain adaptation on object point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6403–6412, 2021. 20, 21

[90] Yuefan Shen, Yanchao Yang, Mi Yan, He Wang, Youyi Zheng, and Leonidas J Guibas. Domain adaptation on point clouds via geometry-aware implicits. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7223–7232, 2022. 21

[91] Hanxue Liang, Hehe Fan, Zhiwen Fan, Yi Wang, Tianlong Chen, Yu Cheng, and Zhangyang Wang. Point cloud domain adaptation via masked local 3d structure prediction. In *European Conference on Computer Vision*, pages 156–172. Springer, 2022. 20

[92] Hehe Fan, Xiaojun Chang, Wanyue Zhang, Yi Cheng, Ying Sun, and Mohan Kankanhalli. Self-supervised global-local structure modeling for point cloud domain adaptation with reliable voted pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6377–6386, 2022. 20, 21

[93] Yongwei Chen, Zihao Wang, Longkun Zou, Ke Chen, and Kui Jia. Quasi-balanced self-training on noise-aware synthesis of object point clouds for closing domain gap. In *European Conference on Computer Vision*, pages 728–745. Springer, 2022. 20

[94] Adriano Cardace, Riccardo Spezialetti, Pierluigi Zama Ramirez, Samuele Salti, and Luigi Di Stefano. Refrec: Pseudo-labels refinement via shape reconstruction for unsupervised 3d domain adaptation. In *2021 International Conference on 3D Vision (3DV)*, pages 331–341. IEEE, 2021. 20

[95] 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. 21

[96] Peng Su, Kun Wang, Xingyu Zeng, Shixiang Tang, Dapeng Chen, Di Qiu, and Xiaogang Wang. Adapting object detectors with conditional domain normalization. In *European Conference on Computer Vision*, pages 403–419. Springer, 2020. 21, 89

[97] 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. 21, 89

[98] Benjamin Caine, Rebecca Roelofs, Vijay Vasudevan, Jiquan Ngiam, Yuning Chai, Zhifeng Chen, and Jonathon Shlens. Pseudo-labeling for scalable 3d object detection. *arXiv preprint arXiv:2103.02093*, 2021. 21

[99] 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. 21, 48, 89

[100] Zhipeng Luo, Zhongang Cai, Changqing Zhou, Gongjie Zhang, Haiyu Zhao, Shuai Yi, Shijian Lu, Hongsheng Li, Shanghang Zhang, and Ziwei Liu. Unsupervised domain adaptive 3d detection with multi-level consistency. In

Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 8866–8875, 2021. 21, 39, 66, 89

[101] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d++: Denoised self-training for unsupervised domain adaptation on 3d object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 21

[102] 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. 21, 48

[103] Martin Hahner, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Fog simulation on real lidar point clouds for 3d object detection in adverse weather. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15283–15292, 2021. 21, 88, 89

[104] Qiangeng Xu, Yin Zhou, Weiyue Wang, Charles R Qi, and Dragomir Anguelov. Spg: Unsupervised domain adaptation for 3d object detection via semantic point generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15446–15456, 2021. 21

[105] Martin Hahner, Christos Sakaridis, Mario Bijelic, Felix Heide, Fisher Yu, Dengxin Dai, and Luc Van Gool. Lidar snowfall simulation for robust 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16364–16374, 2022. 21, 88, 89

[106] Zeng Yihan, Chunwei Wang, Yunbo Wang, Hang Xu, Chaoqiang Ye, Zhen Yang, and Chao Ma. Learning transferable features for point cloud detection via 3d contrastive co-training. *Advances in Neural Information Processing Systems*, 34:21493–21504, 2021. 22

[107] Yi Wei, Zibu Wei, Yongming Rao, Jiaxin Li, Jie Zhou, and Jiwen Lu. Lidar distillation: Bridging the beam-induced domain gap for 3d object detection. *ECCV*, 2022. 22

[108] 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. 22, 48, 63, 65, 89

[109] Cristiano Saltori, Evgeny Krivosheev, Stéphane Lathuiliére, Nicu Sebe, Fabio Galasso, Giuseppe Fiameni, Elisa Ricci, and Fabio Poiesi. Gipso: Geometrically informed propagation for online adaptation in 3d lidar segmentation. In *European Conference on Computer Vision*, pages 567–585. Springer, 2022. 23, 39

[110] Runyu Ding, Jihan Yang, Li Jiang, and Xiaojuan Qi. Doda: Data-oriented sim-to-real domain adaptation for 3d indoor semantic segmentation. *arXiv preprint arXiv:2204.01599*, 2022. 22

[111] Maximilian Jaritz, Tuan-Hung Vu, Raoul de Charette, Emilie Wirbel, and Patrick Pérez. xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12605–12614, 2020. 22, 39, 65

[112] Duo Peng, Yinjie Lei, Wen Li, Pingping Zhang, and Yulan Guo. Sparse-to-dense feature matching: Intra and inter domain cross-modal learning in domain adaptation for 3d semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7108–7117, 2021. 23

[113] Inkyu Shin, Yi-Hsuan Tsai, Bingbing Zhuang, Samuel Schulter, Buyu Liu, Sparsh Garg, In So Kweon, and Kuk-Jin Yoon. Mm-tta: Multi-modal test-time adaptation for 3d semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16928–16937, 2022. 23

[114] Maximilian Jaritz, Tuan-Hung Vu, Raoul De Charette, Émilie Wirbel, and Patrick Pérez. Cross-modal learning for domain adaptation in 3d semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 22

[115] 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. 22, 48, 49, 53, 63, 65, 88, 89

[116] Ferdinand Langer, Andres Milioto, Alexandre Haag, Jens Behley, and Cyrill Stachniss. Domain transfer for semantic segmentation of lidar data using deep neural networks. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8263–8270. IEEE, 2020. 65

[117] 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. 63, 89

[118] Peng Jiang and Srikanth Saripalli. Lidarnet: A boundary-aware domain adaptation model for point cloud semantic segmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2457–2464. IEEE, 2021.

[119] Guangrui Li, Guoliang Kang, Xiaohan Wang, Yunchao Wei, and Yi Yang. Adversarially masking synthetic to mimic real: Adaptive noise injection for point cloud segmentation adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20464–20474, June 2023. 22

[120] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data. *Advances in Neural Information Processing Systems*, 34:3635–3649, 2021. 23, 44, 89

[121] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 23, 24, 89

[122] Chao Huang, Zhangjie Cao, Yunbo Wang, Jianmin Wang, and Mingsheng Long. Metasets: Meta-learning on point sets for generalizable representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8863–8872, 2021. 24

[123] Hao Huang, Cheng Chen, and Yi Fang. Manifold adversarial learning for cross-domain 3d shape representation. In *European Conference on Computer Vision*, pages 272–289. Springer, 2022. 24

[124] Alexander Lehner, Stefano Gasperini, Alvaro Marcos-Ramiro, Michael Schmidt, Mohammad-Ali Nikouei Mahani, Nassir Navab, Benjamin Busam, and Federico Tombari. 3d-vfield: Adversarial augmentation of point clouds for domain generalization in 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17295–17304, 2022. 24, 89

[125] Shuo Wang, Xinhai Zhao, Hai-Ming Xu, Zehui Chen, Dameng Yu, Jiahao Chang, Zhen Yang, and Feng Zhao. Towards domain generalization for multi-view 3d object detection in bird-eye-view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13333–13342, 2023. 24

[126] Hyeonseong Kim, Yoonsu Kang, Changgyoon Oh, and Kuk-Jin Yoon. Single domain generalization for lidar semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17587–17598, June 2023. 24

[127] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9939–9948, 2021. 26, 33, 38, 47, 69, 86, 88, 100, 101, 102

[128] Kangcheng Liu, Zhi Gao, Feng Lin, and Ben M Chen. Fg-conv: Large-scale lidar point clouds understanding leveraging feature correlation mining and geometric-aware modeling. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12896–12902. IEEE, 2021.

[129] Kangcheng Liu, Zhi Gao, Feng Lin, and Ben M Chen. Fg-net: A fast and accurate framework for large-scale lidar point cloud understanding. *IEEE Transactions on Cybernetics*, 2022. 26

[130] 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*, pages 12697–12705, 2019. 26, 39

[131] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6569–6578, 2019. 26, 39

[132] Patrice Y Simard, Yann A LeCun, John S Denker, and Bernard Victorri. Transformation invariance in pattern recognition—tangent distance and tangent propagation. In *Neural networks: tricks of the trade*, pages 239–274. Springer, 1998. 26

[133] Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. *Advances in neural information processing systems*, 13, 2000. 26

[134] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 26, 28, 34, 35, 41

[135] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2918–2928, 2021. 26, 28, 34, 35

[136] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 28

[137] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 28

[138] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 28

[139] 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. 28

[140] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 28

[141] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 28

[142] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 28

[143] 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. 28

[144] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020. 28

[145] Zhi Zhang, Tong He, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of freebies for training object detection neural networks. *arXiv preprint arXiv:1902.04103*, 2019. 28

[146] Hao-Shu Fang, Jianhua Sun, Runzhong Wang, Minghao Gou, Yong-Lu Li, and Cewu Lu. Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 682–691, 2019. 28

[147] Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. *IEEE Robotics and Automation Letters*, 7(2):3795–3802, 2022. 28, 33, 39, 46, 86, 93

[148] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. Rethinking pre-training and self-training. *Advances in neural information processing systems*, 33:3833–3845, 2020. 32

[149] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019. 40, 59, 74, 75, 99, 100, 101

[150] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Cross-view regularization for domain adaptive panoptic segmentation. *arXiv preprint arXiv:2103.02584*, 2021. 39, 44, 89

[151] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Rda: Robust domain adaptation via fourier adversarial attacking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8988–8999, 2021. 44

[152] Dayan Guan, Jiaxing Huang, Aoran Xiao, and Shijian Lu. Domain adaptive video segmentation via temporal consistency regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8053–8064, 2021. 89

[153] Jiaxing Huang, Dayan Guan, Aoran Xiao, Shijian Lu, and Ling Shao. Category contrast for unsupervised domain adaptation in visual tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1203–1214, 2022. 39, 89

[154] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4213–4220. IEEE, 2019. 44, 47, 53, 88

[155] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. 44

[156] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1534–1543, 2016.

[157] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349, 2016. 44

[158] Ruoqi Sun, Xinge Zhu, Chongruo Wu, Chen Huang, Jianping Shi, and Lizhuang Ma. Not all areas are equal: Transfer learning for semantic segmentation via hierarchical region selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4360–4369, 2019. 44

[159] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8050–8058, 2019. 57

[160] Dayan Guan, Jiaxing Huang, Shijian Lu, and Aoran Xiao. Scale variance minimization for unsupervised domain adaptation in image segmentation. *Pattern Recognition*, 112:107764, 2021. 44

[161] Romain Loiseau, Mathieu Aubry, and Loïc Landrieu. Online segmentation of lidar sequences: Dataset and algorithm. In *European Conference on Computer Vision*, pages 301–317. Springer, 2022. 46

[162] 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. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1887–1893. IEEE, 2018. 47, 48, 53

[163] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In *European Conference on Computer Vision*, pages 1–19. Springer, 2020.

[164] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds for autonomous driving. *arXiv preprint arXiv:2003.03653*, 2020. 88, 100, 101, 102

[165] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9601–9610, 2020. 47, 88

[166] Yuenan Hou, Xinge Zhu, Yuexin Ma, Chen Change Loy, and Yikang Li. Point-to-voxel knowledge distillation for lidar semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8479–8488, 2022. 47

[167] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018. 47

[168] Benjamin Graham and Laurens van der Maaten. Submanifold sparse convolutional networks. *arXiv preprint arXiv:1706.01307*, 2017. 47

[169] Braden Hurl, Krzysztof Czarnecki, and Steven Waslander. Precise synthetic image and lidar (presil) dataset for autonomous vehicle perception. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 2522–2529. IEEE, 2019. 47

[170] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European conference on computer vision*, pages 102–118. Springer, 2016. 47

[171] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International conference on artificial neural networks*, pages 270–279. Springer, 2018. 48

[172] 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. *arXiv preprint arXiv:2009.03456*, 2:3, 2020. 48

[173] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 48

[174] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 48, 54

[175] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *arXiv preprint arXiv:1711.11586*, 2017. 48

[176] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5162–5170, 2015. 48

[177] Riccardo Roveri, Lukas Rahmann, Cengiz Oztireli, and Markus Gross. A network architecture for point cloud classification via automatic depth images generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4176–4184, 2018. 48

[178] Peng Dai, Yinda Zhang, Zhuwen Li, Shuaicheng Liu, and Bing Zeng. Neural point cloud rendering via multi-plane projection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7830–7839, 2020. 48

[179] Ruihui Li, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-gan: a point cloud upsampling adversarial network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7203–7212, 2019. 48, 51, 54

[180] Chulin Xie, Chuxin Wang, Bo Zhang, Hao Yang, Dong Chen, and Fang Wen. Style-based point generator with adversarial rendering for point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4619–4628, 2021. 48

[181] UE4. Unreal game engine, 2014. URL `https://www.unrealengine.com/en-US/what-is-unreal-engine-4`. Accessed Aug 18, 2021. 49

[182] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 49, 54

[183] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. 51

[184] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. Pf-net: Point fractal network for 3d point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7662–7670, 2020. 51

[185] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 51, 53

[186] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 56, 97, 98

[187] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 58

[188] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. *ICCV*, 2019. 58

[189] Taekyung Kim and Changick Kim. Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation. In *European Conference on Computer Vision*, pages 591–607. Springer, 2020. 58

[190] 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. 59, 74, 75, 99, 100, 101

[191] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5982–5991, 2019. 59

[192] Nikita Araslanov and Stefan Roth. Self-supervised augmentation consistency for adapting semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15384–15394, 2021. 63, 65

[193] Luke Melas-Kyriazi and Arjun K Manrai. Pixmatch: Unsupervised domain adaptation via pixelwise consistency training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12435–12445, 2021. 63, 65, 66

[194] 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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(4):3500–3509, May 2021. doi: 10.1609/aaai.v35i4.16464. URL `https://ojs.aaai.org/index.php/AAAI/article/view/16464`. 65

[195] Aoran Xiao, Xiaoqin Zhang, Ling Shao, and Shijian Lu. A survey of label-efficient deep learning for 3d point clouds. *arXiv preprint arXiv:2305.19812*, 2023. 65

[196] Guangrui Li, Guoliang Kang, Xiaohan Wang, Yunchao Wei, and Yi Yang. Adversarially masking synthetic to mimic real: Adaptive noise injection for point cloud segmentation adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20464–20474, 2023. 65

[197] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016. 65

[198] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29, 2016.

[199] Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. In *International Conference on Learning Representations*, number 6, 2018.

[200] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018. 65

[201] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33:6256–6268, 2020. 65

[202] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations*, 2020.

[203] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 65

[204] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. Dacs: Domain adaptation via cross-domain mixed sampling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1379–1389, 2021. 65

[205] Yun Xing, Dayan Guan, Jiaxing Huang, and Shijian Lu. Domain adaptive video segmentation via temporal pseudo supervision. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX*, pages 621–639. Springer, 2022. 65

[206] Li Jiang, Shaoshuai Shi, Zhuotao Tian, Xin Lai, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Guided point contrastive learning for semi-supervised point cloud semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6423–6432, 2021. 65

[207] Zhipeng Luo, Zhongang Cai, Changqing Zhou, Gongjie Zhang, Haiyu Zhao, Shuai Yi, Shijian Lu, Hongsheng Li, Shanghang Zhang, and Ziwei Liu. Unsupervised domain adaptive 3d detection with multi-level consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8866–8875, October 2021. 65

[208] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15587–15597, 2021. 66

[209] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6535–6545, 2021. 66

[210] Yucen Luo, Jun Zhu, Mengxi Li, Yong Ren, and Bo Zhang. Smooth neighbors on teacher graphs for semi-supervised learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8896–8905, 2018. 66

[211] Lingdong Kong, Jiawei Ren, Liang Pan, and Ziwei Liu. Lasermix for semi-supervised lidar semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21705–21715, 2023. 66

[212] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 69

[213] Matthew Pitropov, Danson Evan Garcia, Jason Rebello, Michael Smart, Carlos Wang, Krzysztof Czarnecki, and Steven Waslander. Canadian adverse driving conditions dataset. *The International Journal of Robotics Research*, 40(4-5):681–690, 2021. 86, 88

[214] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. *Advances in Neural Information Processing Systems*, 32, 2019. 88

[215] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu. 2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12547–12556, 2021.

[216] Feihu Zhang, Jin Fang, Benjamin Wah, and Philip Torr. Deep fusionnet for point cloud semantic segmentation. In *European Conference on Computer Vision*, pages 644–663. Springer, 2020.

[217] Jianyun Xu, Ruixiang Zhang, Jian Dou, Yushi Zhu, Jie Sun, and Shiliang Pu. Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16024–16033, 2021. 88

[218] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017. 88

[219] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020. 88

[220] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10765–10775, 2021. 88, 91, 92

[221] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*, 24, 2011. 89

[222] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18, 2013. 89

[223] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. *Advances in neural information processing systems*, 31, 2018. 89

[224] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2018. 97, 98

[225] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Fsdr: Frequency space domain randomization for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6891–6902, 2021. 89

[226] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019. 89

[227] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015. 89

[228] Julian Ryde and Nick Hillier. Performance of laser and radar ranging devices in adverse environmental conditions. *Journal of Field Robotics*, 26(9):712–727, 2009. 89

[229] Thierry Peynot, James Underwood, and Steven Scheding. Towards reliable perception for unmanned ground vehicles in challenging conditions. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1170–1176. IEEE, 2009.

[230] A Filgueira, H González-Jorge, Susana Lagüela, L Díaz-Vilariño, and Pedro Arias. Quantifying the influence of rain in lidar performance. *Measurement*, 95:143–148, 2017.

[231] Robin Heinzler, Philipp Schindler, Jürgen Seekircher, Werner Ritter, and Wilhelm Stork. Weather influence and classification with automotive lidar sensors. In *2019 IEEE intelligent vehicles symposium (IV)*, pages 1527–1534. IEEE, 2019. 89

[232] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 92

[233] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 93

[234] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30. IEEE, 2017. 94

[235] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 969–977, 2018. 94

[236] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 95

[237] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018. 96

[238] Xufeng Yao, Yang Bai, Xinyun Zhang, Yuechen Zhang, Qi Sun, Ran Chen, Ruiyu Li, and Bei Yu. Pcl: Proxy-based contrastive learning for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7097–7107, 2022. 97, 98