Sparse2SOAP: Domain Adaptation for LiDAR-Based 3D Object Detection

by

Christopher Gus Mannes

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2023

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis involves joint work from the WISE lab at the University of Waterloo. Chengjie Huang developed the point cloud aggregation, quasi-stationary training, and the filtering/clustering techniques. I implemented the student-teacher framework, and the elliptical masking technique for handling dynamic objects. Furthermore, I performed all training, except for the SOAP and baseline models. Lastly, I performed all subsequent analysis of the results. Chengjie Huang, Dr. Vahdat Abdelzad, Dr. Sean Sedwards, and Professor Krzysztof Czarnecki provided valuable feedback on the presented ideas during their development and on the thesis manuscript.

Abstract

In this work, we propose Sparse2SOAP, an extension of the previous work in Sparse2Dense that uses knowledge distillation in a teacher-student framework to densify 3D features, to enable its uses for cross-domain LiDAR-based 3D object detection in autonomous driving. This is achieved by utilizing Stationary Object Aggregation Pseudo-labelling (SOAP) from prior work, to generate high-quality pseudo-labels for Quasi-Stationary (QS) dense point cloud objects in Simply Aggregated (SA) point clouds. The dense object pseudo-labels can then be paired with the corresponding sparse objects pseudo-labels creating dense-sparse pairs for knowledge distillation. We additionally propose a masking method for handling knowledge distillation for dynamic objects. We evaluate the proposed method using nuScenes and Waymo datasets for Unsupervised Domain Adaptation (UDA) tasks. We observe an increase in mAP and AP for classes with many QS objects. To the best of our knowledge, we are the first to perform feature alignment between sparse and dense point cloud representations using aggregated point clouds in the context of UDA.

Acknowledgements

I first want to thank Professor Krzysztof Czarnecki for his guidance and support as well as for the opportunity to develop my research skills and investigate interesting questions. Furthermore, I want to thank Dr. Vahdat Abdelzad, Dr. Sean Sedwards, Chengjie Huang, and Prarthana Bhattacharyya for their indispensable help in all facets of research. This work was supported by Huawei Corporation.

Dedication

This thesis is dedicated to my friends and family for their support in all my endeavors. Additionally, I want to dedicate this work to my all-time favourite scientist, Dr. Richard Feynman, who inspired me to ask questions and contribute to the development of science and engineering.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In Chapter 1 of this thesis, I will motivate the need to develop methods for domain adaptation, particularly for the autonomous vehicles setting. Then, I will describe the underlying causes for domain adaptation, and give a formal definition of the problem/task. Then, I will detail related work for 3D object detection and domain adaptation, and lastly summarize the contributions of this work.

1.1 Motivation

The realization of fully Autonomous Vehicles (AVs) promises safer and more efficient methods of transportation, and thus would be considered a landmark achievement in science and engineering. In order to achieve this goal, an AV needs to be able to execute a series of perception, planning, and control tasks in complex and dynamic scenarios. Additionally, AVs are considered to be safety-critical systems where failure to successfully perform the aforementioned tasks may result in the loss of human life. Figure 1.1 depicts a general software stack for an autonomous vehicle. The AV observes it environment using a series of sensors, then the collected data is processed by the perception module where 3D object detection is carried out to localize and classify other agents in the environment. In the subsequent module, long term prediction and motion forecasting is performed for all agents. Then, in the planning module, the ego vehicle performs it own path planning for a short time horizon. Lastly, the AV determines the control inputs needed to execute the desired path/action whilst operating within a set of given constraints.

The task of 3D object detection performed by the perception module is a critical component as it is charged with correctly detecting other agents in the scene such as cars,

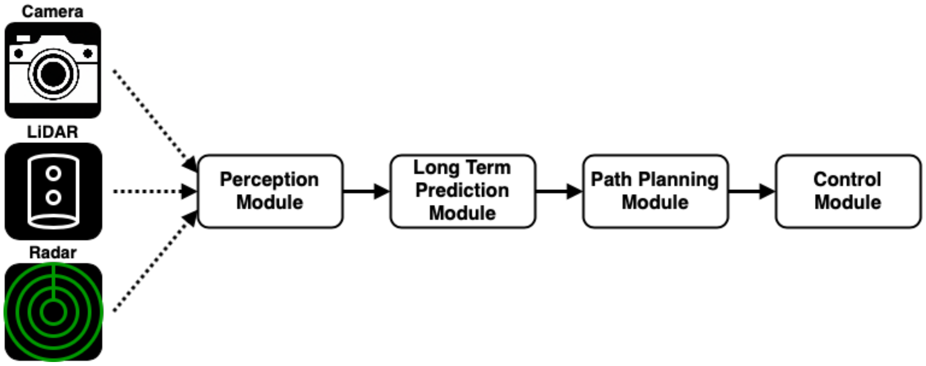

Figure 1.1: A general software stack for an autonomous vehicle. The AV observes its environment using sensors such as camera, LiDAR, and radar as shown on the left. This data is then passed as input to a series of modules to executed the required operations for navigating the environment.

cyclists, and pedestrians. As a result, this carries significant safety requirements. Additionally, since the perception module is executed first, the resulting information is relied upon by all subsequent modules. LiDAR-based measurements provide valuable 3D information by accurately recording the range and scale of objects independent of the lighting conditions. State-of-the-art performance for 3D object detection is obtained by training Deep Neural Networks (DNNs) in a supervised fashion [27, 12, 18, 31, 17]. However, supervised training relies heavily on the foundational principles regarding data for machine learning; that the data be i) large, ii) diverse, and iii) be correctly/accurately annotated. Additionally, supervised learning assumes that the training and testing data are identically distributed. If these requirements/assumptions are fulfilled then supervised training is highly effective. Many research institutions and companies have published large-scale LiDAR datasets for 3D object detection for the AV setting [9, 4, 11, 2, 19, 24, 15]. However, the available datasets only cover a small subset of domains encountered by an AV, where in the field of computer vision a domain is a particular distribution of values that make up the information content of the observed images/frames. A change in any factor that causes a non-negligible shift in the distribution of values is referred to as a domain shift. A domain shift can be caused by changes in the lighting conditions, view/perspective, object appearance, or object context. In 2D camera images, the domain is primarily dependent on the texture, which reflects the surface characteristics of objects. In 3D LiDAR point clouds, the domain is primarily determined by the geometric nature of the points [30].

It is essential to have methods for either learning domain invariant features with respect to certain domains or to be able to easily and efficiently map between two domains. This

Train Data	Val. Data	Average AP	Car AP	Truck AP	Bus AP	Motor. AP	Cyclist AP	Ped. AP
20% nuScenes	nuScenes	51.5	79.2	45.6	57.1	38.9	21.0	67.4
20% nuScenes	Waymo	10.4	24.8	1.4	9.5	17.5	7.3	2.3
100% nuScenes	Waymo	16.2	33.7	3.8	22.9	20.2	12.6	4.1
20% Waymo	Waymo	56.5	68.3	34.2	49.0	53.1	68.4	65.9

Table 1.1: Baseline performances for in-domain, cross-domain, and oracle models with nuScenes as the source domain and Waymo as the target domain using the CenterPoint detector.

is because it is observed that a model trained in one domain and validated in another domain experiences significant performance degradation as shown in Table 1.1. Table 1.1 shows good in-domain performance in Average Precision (AP) for a model trained on 20% nuScenes data and validated on nuScenes. However, a decrease in AP of 41.1% and 35.3% is observed for a model validated on Waymo, but trained on 20% nuScenes and 100% nuScenes, respectively. This is a significant decrease compared the oracle model, which sees values similar to the in-domain model for nuScenes, when trained and validated on Waymo. A naive solution would require collecting and labelling data for a new domain and re-training the DNN; however, this is infeasible as the collection of data and labelling by human annotators is extremely expensive and time-consuming. However, of the two operations the human- labelling is the most significant barrier, as collecting unlabelled data is far easier with the current number of AV research vehicles and engineers. Thus, the present work is performed under this consideration. To address the observed performance degradation, this work explores Unsupervised Domain Adaptation (UDA) for 3D object detection where the objective is to adapt a model trained in a source domain with labelled data to a target domain with unlabelled data to perform well in the target domain, thereby reducing to domain gap.

1.2 Formal Definition of the Problem and Task

Domain shifts in 3D point clouds for the AV setting can occur as a result of a change in any of the point cloud characteristic properties. Although a shift in the distribution of a single property is sufficient, in general a domain shift is often a result of a change in several properties co-occurring simultaneously. Some common properties that are known to cause domain shifts are: point cloud noise, occlusions, sensor type (LiDAR sensor model), sensor setup (LiDAR installation method), and variations in the object instances such as type, frequency, and characteristic attributes.

To formally describe the aforementioned task of domain adaptation, we follow the work of Pan *et al.* [16]. A domain is defined by the tuple $(\chi, P(X))$ where χ is a powerset of examples and referred to as the feature space and with $P(X)$ as the marginal probability distribution of X. Then, the dataset for the task, X, is a finite set in χ ($X \in \chi$) and x is a particular instance in X ($x \in X$). For a given domain $(\chi, P(X))$, a task is then defined by the tuple $(\mathcal{Y}, P(Y|X))$ where $\mathcal{Y}$ is the label space and is a powerset of all labels for the given task. The finite set $Y \in \mathcal{Y}$ containing of the labels corresponding to the dataset X with $y \in Y$ being a particular instance label. The objective predictive function that is learning during training approximates the conditional probability distribution, $P(Y|X)$. The source and target domains are then denoted as $D_S = (\chi_S, P_S(X_S)$ and $D_T = (\chi_T, P_T(X_T)$ with learning tasks $T_S = (\mathcal{Y}_S, P_S(Y_S|X_S)$ and $T_T = (\mathcal{Y}_T, P_T(Y_T|X_T)$, respectively. Models obtained when $D_S = D_T$, $T_S = T_T$, and $\mathcal{Y}_S, \mathcal{Y}_T$ are observable are considered to be trained in the conventional supervised fashion. However, DA scenarios occur when $D_S \neq D_T$, $T_S = T_T$, and $\mathcal{Y}_S, \mathcal{Y}_T$ are observable and unobservable, respectively.

To perform 3D object detection via a DNN approach, then it is necessary to learn an objective predictive function D to map the environment point cloud (P) of the ego vehicle scene to an output space (Y) that localizes and classifies m objects according to Equation 1.1:

$$D_\theta : P \rightarrow \widehat{Y} \tag{1.1}$$

where the set $P = \{p_1, ..., p_n\}$ contains n LiDAR points with $p_i|_{i=0}^{n} \in \mathbb{R}^4$ specifying the coordinate x,y,z, and reflectance r of the i^{th} point. The predicted annotations is the set $\widehat{Y} = \{y_1, ..., y_m\}$ contains m bounding box annotations with $y_j|_{j=0}^{m} \in \mathbb{R}^8$ that specifies the coordinate of the box center (x,y,z) and dimensions (l,w,h) of the bounding box shape, heading angle expressed as the yaw angle, α. The bounding box annotations are expressed in the Bird's-Eye-View (BEV) of the scene. The final element in the annotation y_j is the class label, c_k for $k \in \{\text{car}, \text{truck}, \text{bus}, \text{motorcycle}, \text{cyclist}, \text{pedestrian}\}$. In the conventional setting of supervised-learning, the model D parameterized by θ is optimized by minimizing the classification and regression loss over the source dataset between the predictions and ground truth, $\widehat{Y}$ and Y, respectively. In this work we aim to first establish D_θ by supervised training on the source domain and then adapt the model by introducing pseudo-labels, $\tilde{Y}$ and minimizing the classification and regression loss over the target dataset between the predictions and pseudo-label, $\widehat{Y}_T$ and $\tilde{Y}$, respectively. The resulting model localized and classifies objects according to Equation 1.2 on the target domain

$$D_\phi : P_T \rightarrow \widehat{Y}_T \tag{1.2}$$

The result of Equation 1.2 is the adaptation of a model parameterized by θ to model parameterized by ϕ, $D_\theta \rightarrow D_\phi$ such that the model D_ϕ performs well on the target domain.

1.3 Related Work

Domain adaptation has emerged as a highly active area of research regarding 3D object detection as it is an essential element for detector model deployment at scale. There are several prominent methods regarding DA for 3D object detection for autonomous driving. In this work, we detail 3 popular methods and describe example work for each method. The three methods detailed in this work are: *Domain Invariant Representations*, *Generative Methods*, and *Feature Alignment Methods*. Lastly, we detail the work *Sparse2Dense: Learning to Densify 3D Features* that inspired this research.

1.3.1 Domain Invariant Representations

The objective of *Domain Invariant Representations* is to move the domains into a common semi-canonical representation. The mapping can be applied to the input space or the output space, and thus generally take the form of an data-preprocessing or data-post-processing module that is employed in conjunction with a base model.

Semantic Point Generation (SPG): Xu *et al.* [26] proposes Semantic Point Generation (SPG) as a preprocessing point cloud module that takes the raw point cloud as input and generates a set of semantic points (points + point labels) to complete the partially observed foreground object point clouds. The SPG module performs point voxelization and subsequent voxel feature encoding to generate low-level BEV feature maps. The low-level BEV feature maps are then processed by 2D CNNs to propagate spatial features and learn high-level BEV features. The resulting BEV features are then used for prediction for each 3D voxel. The objective is to classify each voxel as occupied by a foreground object or not, and if so, to regress that mean value of the voxel points.

The learning of foreground regions is investigated using two proposed techniques: i) hide and predict, and ii) semantic area expansion. In the hide and predict method, some of the voxel points are dropped and the network is charged with predicting the missing points. In the semantic area expansion method, the set of supervised voxels is expanded to include voxels empty voxels that neighbour occupied voxels

Domain Invariant Representation methods have shown to be very effective. However, because they generally involve preprocessing and/or post-processing techniques on top of a base model there is significant computational overhead and therefore they often do not achieve real-time deployment.

1.3.2 Generative Methods

The objective of *Generative Methods* is to learn a mapping between domains by minimizing the distance between the distributions of each domain. These methods generally consist of mapping source domain to target domains in a way that preserves annotation information.

Cycle and Semantic Consistent Adversarial Domain Adaptation (CSCADA): Barrera *et al.* [1] propose to employ CycleGAN [37] to map simulated projected BEV LiDAR point cloud to the "real" BEV LiDAR point cloud domain. To address the issue of maintaining semantic consistency between domains, Barrera *et al.* propose applying SalsaNext [6] to each domain to perform semantic segmentation and ensure similarity by applying pixel-wise cross-entropy loss.

Generative mapping methods do not generally achieve state-of-the-art performance, however, they do show to be effective at generating additional data that can be used to improve model performance.

1.3.3 Feature Alignment Methods

The objective of *Feature Alignment Methods* is to align the feature of the target and source domain by employing the use of a similarity/distance losses. This results in obtaining domain-invariant feature representations.

3D Contrastive Co-Training (3D-CoCo): Yihan *et al.* [30] propose 3D-CoCo, which consists of two components based on two main insights for domain adaptation. The first is based on the observation that high-level BEV features are more transferable than low-level BEV features. This is because high-level features are more indicative of the semantic information whereas the low-level features encompass the geometric information of the voxel points. In order to exploit this knowledge, the authors employ separate domain-specific 3D encoder modules to generate feature maps that accurately encode the geometry of the points at each voxel and a single domain-agnostic module for learning

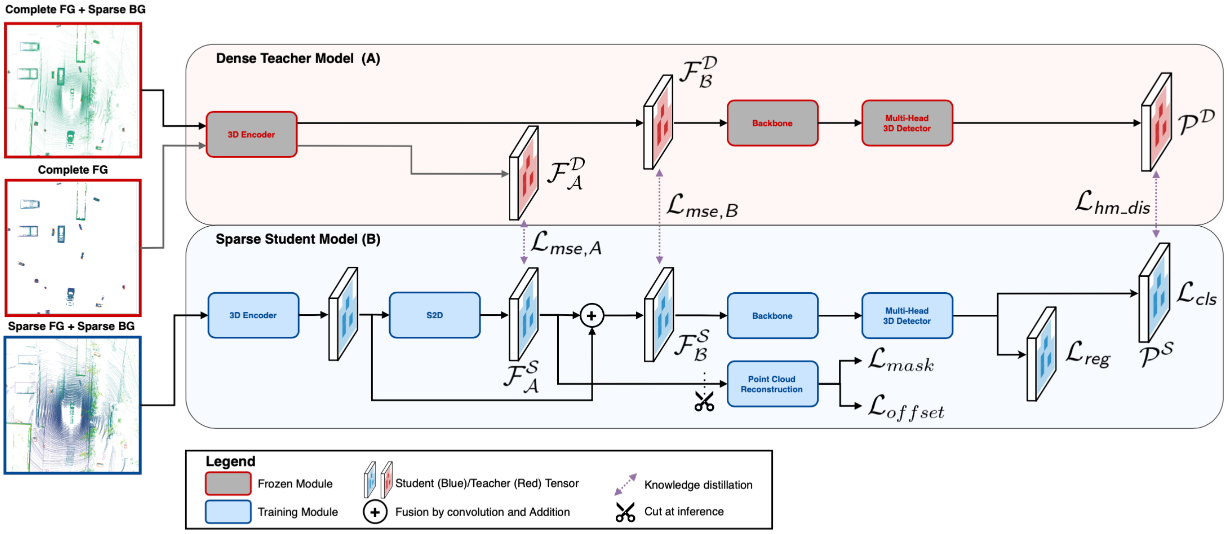

Figure 1.2: An overview of the Sparse2Dense framework. The dense teacher model (DDet) is encompassed by the red box and the sparse student model (SDet) is encompassed by the blue box. Knowledge distillation is performed using the feature maps at point A and B and at the detector heads. The Point Cloud Reconstruction (PCR) module is "cut at inference" since it is only implemented during training of the student model and not required during testing.

domain invariant BEV features. The second component performs contrastive instance alignment, which had been found to be ineffective in previous studies. The objective of contrastive instance alignment is to drive the feature centroids of similar samples from different domains closer to each other. The authors observe that a naive implementation of contrastive instance alignment creates a mismatch between the sample distribution for the source and target domain as the pseudo-labels are biased towards easy samples. Therefore, Yihan *et al.* propose hard sample mining to better align the distribution between the ground truth labels and the pseudo-labels.

Feature alignment has been found to be a relatively simple yet effective strategy for DA. However, 3D-CoCo performs domain alignment between sparse 32-beam and sparse 64-beam LiDAR sensor domains. This is suboptimal as we believe the domain alignment should be performed between less informative domains and more informative domains.

1.3.4 Sparse2Dense: Learning to Densify 3D Features

Wang *et al.* [22] propose Sparse2Dense, a teacher-student framework for densifying 3D features in the latent space to boost the network performance to detect small, distant, and partially observed objects. Figure 1.2 shows an overview of the Sparse2Dense framework. Densification is achieved by first training (stage A in Figure 1.2) a dense point cloud detector (DDet) on point clouds with complete Foreground (FG) objects (aggregated object point clouds using ground truth bounding boxes with symmetry completion) and sparse Background (BG) as shown by the red model with the black data pipeline in 1.2. This teacher model is then used to distill knowledge of dense objects into the student model trained (stage B in Figure 1.2) on sparse point clouds (SDet). During training of SDet on sparse point clouds, the teacher model takes 2 point cloud inputs: complete FG + sparse BG (black line) and complete FG only (grey line). Wang *et al.* propose 2 additional modules, S2D and Point Cloud Reconstruction (PCR), for training the student model (blue model in Figure 1.2) to densify the 3D features. The S2D module is used to filter out the background features and the PCR module is used to perform voxel-level point cloud reconstruction by predicting each voxel as occupied by FG points or not and if so, then regressing the mean position of the voxel points. Knowledge distillation is performed in the network body at points A and B by applying Mean Squared Error (MSE) loss to align the features of DDet and SDet. Additionally, knowledge distillation is performed at the detector heads by applying focal loss [13] using the DDet probability heatmap and the SDet probability heatmap as input. Focal loss is an extension of the conventional Cross Entropy loss that is used for applications with significant class imbalance. In this application there is significant class imbalance between FG and BG.

The work of Sparse2Dense is shown to be effective for boosting the performance of in-domain 3D object detection. However, it cannot be naively applied to DA for a target domain since the lack of ground truth bounding boxes does not allow for one to obtain the dense object point clouds. Thus, it is the work of this thesis, to propose a scheme for generating the dense FG + sparse BG, dense FG only, and sparse point cloud frames for an identical scene in the target domain and then extending the Sparse2Dense framework to align the dense and sparse features.

1.4 Contributions

In this work, we are the first to perform feature alignment between sparse and dense point cloud representations using aggregated point clouds in the context of Unsupervised Domain

Adaptation (UDA). Previous studies that perform sparse-to-dense feature alignment are not applicable to UDA [22, 7] while previous studies that perform feature alignment in UDA are only applicable for sparse-to-sparse feature alignment [30, 34, 5, 25]. Thus, we are able exploit all LiDAR frames in a sequence rather than a single or a few consecutive frames. Utilizing the SOAP method allows us to align the sparse and dense features for quasi-static objects, and we propose an elliptical masking technique for handling dynamic objects. Our evaluation using two large-scale well-known AV datasets, nuScenes [2] and Waymo [19] demonstrates that Sparse2SOAP can effectively align the sparse and dense feature increasing the Average Precision (AP), precision, and recall for 3D object detection.

Chapter 2

Sparse2SOAP

In Chapter 2 of this thesis, we first describe the data pipeline implemented to mimic the input point clouds utilized by Sparse2Dense for the target domain. Then, we will describe the proposed method for updating a source domain teacher model to effectively be utilized as a teacher model in the target domain. Lastly, we will describe the modifications made to Sparse2Dense to extend it for the task of DA for 3D object detection.

2.1 Data Pipeline

In the work of Sparse2Dense [22] and as illustrated in Figure 1.2, the dense point clouds used for in-domain 3d object detection feature complete FG objects that are created by aggregating sequential LiDAR frames that are corrected for the motion of moving objects and performing symmetry completion. However, in UDA this is not possible due to the lack of ground truth bounding boxes. Therefore, in this work we use Simple Aggregation (SA) as proposed by Huang *et al.* [10] for SOAP (Stationary Object Pseudo-labelling) objects, which aligns sequential LiDAR frames using the pose information of the ego vehicle. The resulting SA point cloud will contain aggregated points for quasi-stationary (QS) objects and background, and point cloud distortions that appear as smudges for dynamic objects. Ground points and smudges are removed by post-processing operations yielding a dense point cloud for QS objects. This substantially reduces the domain gap caused by different point densities and beam patterns between source and target domains. A source domain SOAP model is trained and applied to the SA target domain to produce high quality pseudo-labels for QS objects. The SOAP method is designed to detect quasi-static objects by calculating a Quasi-Stationary Score (QSS) based on a weighted Intersection Over

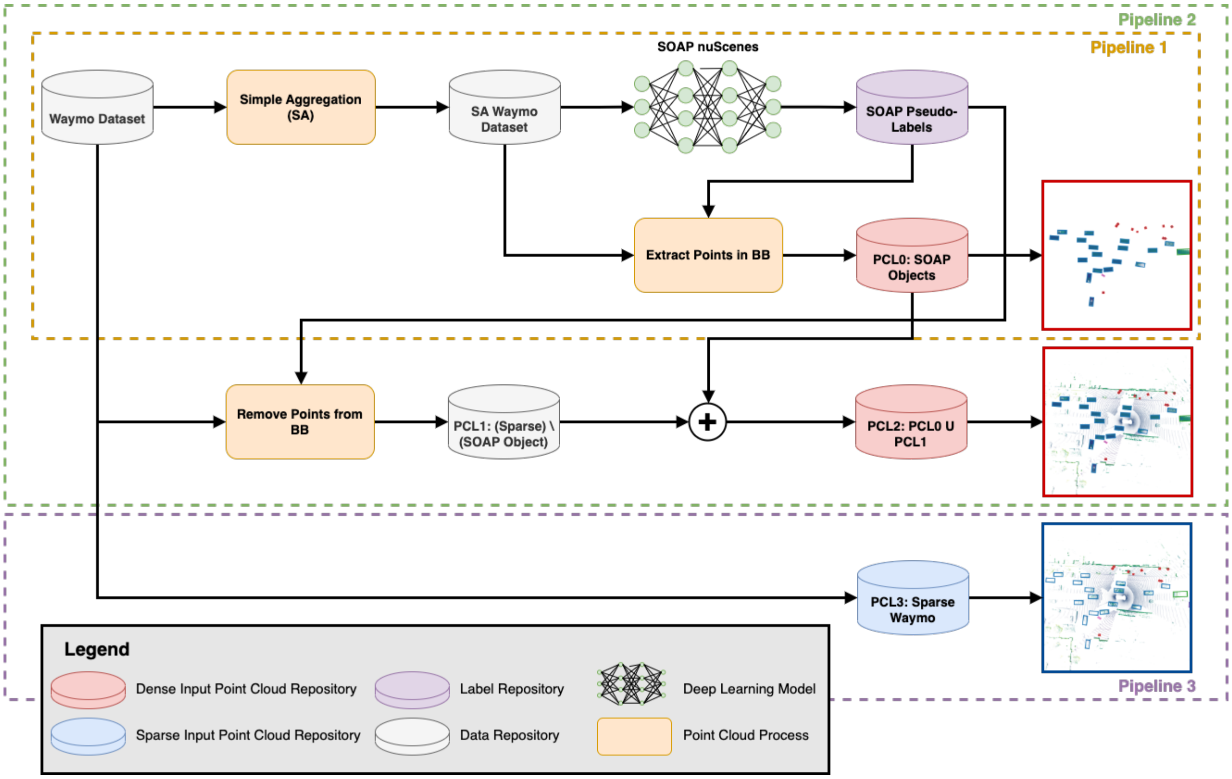

Figure 2.1: An overview of the Sparse2SOAP data pipeline for the target domain. Pipeline 1 (gold) highlights the modules used in generating PCL0 (SA FG only). Pipeline 2 (green) highlights the modules used in generating the PCL2 (SA FG + Sparse BG). Pipeline 3 (purple) highlights the modules used in generating the sparse point cloud.

Union (IoU) formulation during training and ensuring consistent labelling using a Spatial Consistency Post-processing (SCP) technique. Objects with a QSS $> \delta$ are taken to be quasi-stationary. SCP is performed by obtaining per-frame bounding box predictions for a sequence of SA point clouds. Then, all bounding box predictions are mapped into the global coordinate frame via ego pose matrix transformations where the predictions are clustered using an IoU threshold $> \mu$. That is, bounding boxes with IoU below μ are dropped. Then, the clusters are filtered based the number of SA bounding boxes, B_{SA} in each cluster, c according to $|B^{c}_{SA}| > \eta$ to ensure consistent pseudo-labels. The target domain SOAP pseudo-labels are then used to extract the dense object points from the SA point clouds generating Simply Aggregated Foreground (SA FG) object point clouds as shown in pipeline 1 (highlighted in gold) of Figure 2.1. The resulting SA FG points are

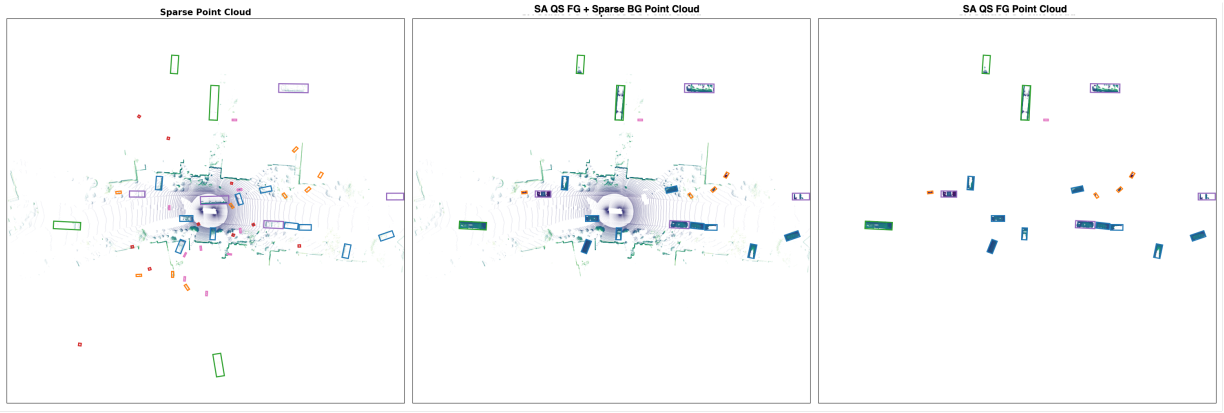

Figure 2.2: A 2D BEV scatter plot of the input point clouds for Sparse2SOAP. The sparse point cloud (left) is input to the student model while the SA QS FG + sparse BG and the SA FG point clouds are input to the teacher model.

denoted as PCL0 and stored in a point cloud repository.

Pipeline 1 is then extended to form pipeline 2, which takes a sparse point cloud frame and applies point removal to remove sparse object points inside the SOAP pseudo-labels to form point cloud, PCL1 (background only). The points of PCL0 are then concatenated with the points of PCL1 to generate the SA FG + sparse BG point clouds as shown in pipeline 2 (highlighted in green) of Figure 2.1.

Pipeline 3 is simply the general data loading procedure for sparse point cloud frames, which are stored in PCL3. Pipeline 3 is highlighted in purple in Figure 2.1. The 3 point clouds denoted as PCL0, PCL2, and PCL3 are then used as the input point clouds for Sparse2SOAP with PCL0 and PCL2 passed as input to the teacher and PCL3 passed as input to the student model. A BEV visualizaton of the input point clouds is shown in Figure 2.2.

2.1.1 Fine-Tune (FT) Source Domain Model and Pseudo-Label Refinement

There exists a domain mismatch between the teacher input point clouds, PCL2 and the SA point clouds used to train the source domain SOAP model. Therefore, the source domain SOAP model is Fine-Tuned (FT) on PCL2 point clouds with the initial pseudo-labels generated by the source domain SOAP model. This ensures the best performing model

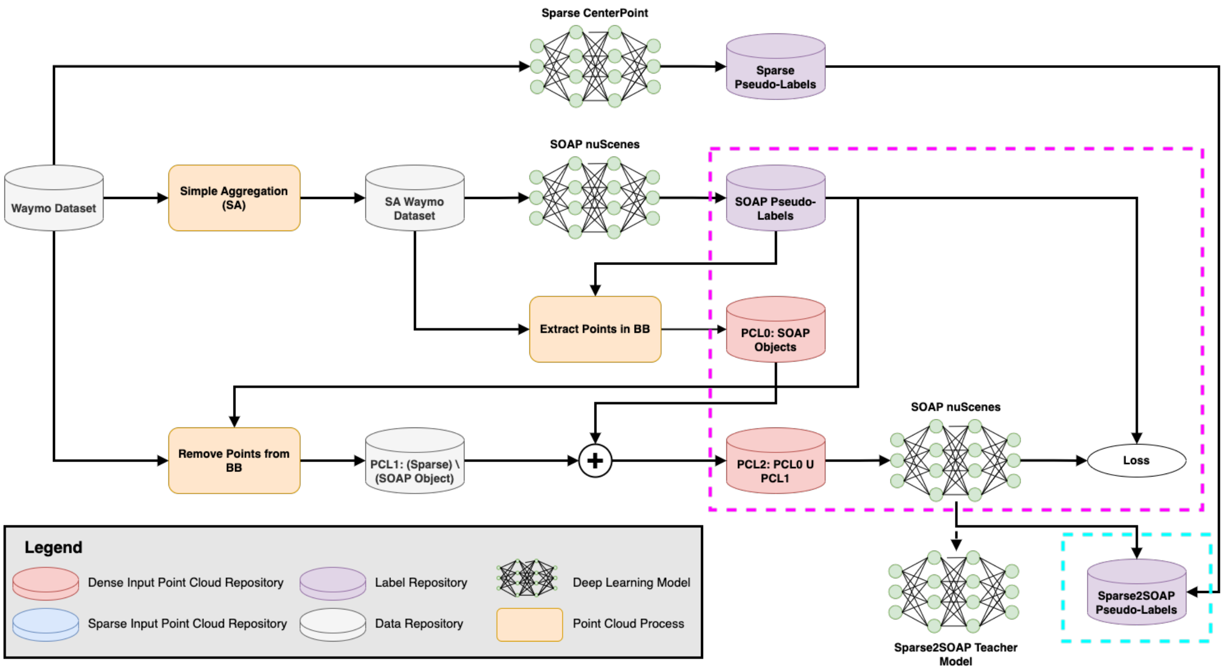

Figure 2.3: An overview of the Sparse2SOAP data pipeline.

is used as the teacher in Sparse2SOAP. The fine-tuning process is shown in Figure 2.3 and highlighted by the dashed magenta box. The SOAP pseudo-labels for the SA target domain generated by the source domain SOAP model are a good first order approximation of the object bounding boxes. However, they are still generated by a model optimized on the source domain. Therefore, the SOAP pseudo-labels for the PCL2 point clouds are then refined using the FT SOAP model. The source domain SOAP model proposed by Huang *et al.* [10] is only trained to detect QS objects in the SA point cloud yielding an incomplete set of pseudo-labels. Therefore, the SOAP method proposes to use a sparse model denoted as Sparse CenterPoint in Figure 2.3 to detect dynamic objects and recover the complete set of pseudo-labels. The SOAP pseudo-labels and the sparse pseudo-labels are then combined using SOAP's QSS evaluation, SCP, and Non-Maximum Suppression (NMS) to combine the two sets of pseudo-labels, which are denoted as Sparse2SOAP pseudo-labels in Figure 2.3 and highlighted by the dashed turquoise box.

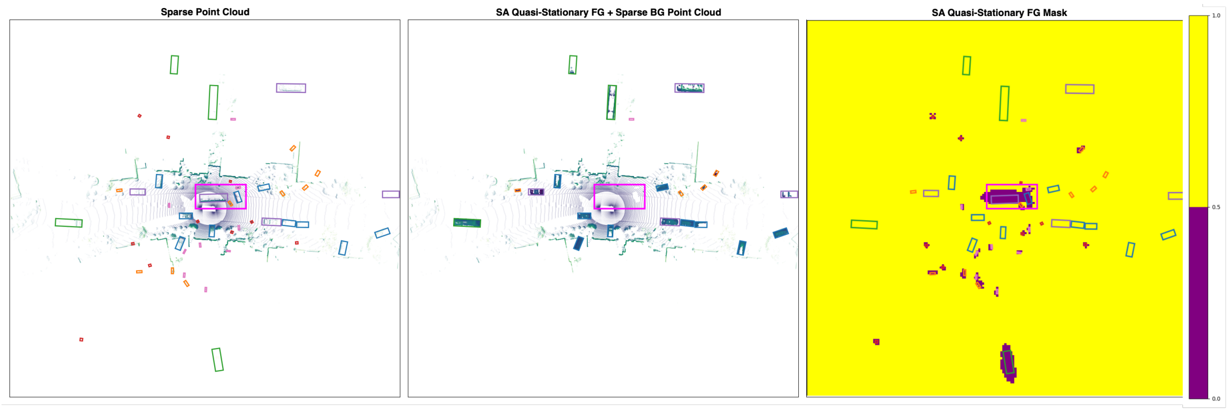

Figure 2.4: An overview of the Sparse2SOAP data pipeline.

2.1.2 Dynamic Object Masking

The knowledge distillation techniques proposed by Sparse2Dense requires additional consideration to be employed for DA in Sparse2SOAP. The SOAP teacher model is only trained to detect QS objects, however the Sparse2SOAP pseudo-labels contain both QS and dynamic objects. Therefore, we propose to assign a static ID to each pseudo-label indicating if the bounding box is generated by the Sparse CenterPoint model or the FT SOAP model. Then, the pseudo-labels used for knowledge distillation are only a subset of the pseudo-labels used for regression and classification. Figure 2.4 shows the BEV visualization of the sparse point cloud (left) containing all pseudo-labels, and the SA QS FG + sparse BG point cloud (middle). The magenta box highlights a cluster of pseudo-labels that are present in the sparse point cloud but absent in the SA QS FG + sparse BG point cloud. The corresponding regions are then masked in the feature maps that are used for knowledge distillation between the SOAP teacher model and the student model such that the student will not be penalized during feature alignment.

Chapter 3

Experiments

In Chapter 3 of this thesis, we first describe the datasets employed for DA and all dataset modifications made for the subsequent experiments. Then, we describe the models used for the evaluation and ablation experiments. Lastly, we present the results of the experiments.

3.1 Datasets

The evaluation of Sparse2SOAP is performed using the following two large-scale autonomous driving datasets: nuScenes [2] and Waymo [19]. Both, nuScenes and Waymo have been extensively used in previous studies.

3.1.1 nuScenes Dataset

The nuScenes dataset [2] is a multimodal dataset published by Nutonomy in 2019 for the development of object detection and tracking for autonomous vehicles. The LiDAR data is collected using a single roof-mounted Velodyne HDL-32E (32-beam) rotating LiDAR sensor at a sampling fequency of 20 Hz. The nuScenes LiDAR sensor has a maximum operating range of 70 m. The dataset consists of 1000 sequences that each span 20 s in length with 3D bounding boxes and labels for 23 classes with 8 attributes. The nuScenes data was collected from various diverse regions in Boston and Singapore, both of which feature dense and challenging traffic environments. Data is collected during night-time and day-time driving scenarios. The nuScenes LiDAR point features consist of the (x, y, z) spatial coordinates and the intensity of the return laser pulse. The models trained on nuScenes data use point clouds constructed from 10 consecutive LiDAR sweeps.

3.1.2 Waymo Open Dataset

The Waymo Open dataset [19] is a large-scale multi-modal dataset designed to facilitate research and development for a variety of machine perception tasks such as image classification, object detection, object tracking, semantic segmentation, and instance segmentation. Waymo data is collected using a 5-LiDAR sensor configuration at a sampling frequency of 10 Hz. The 5 sensor configuration consists of a single proprietary mid-range roof-mounted rotating 64-beam LiDAR sensor and four short-range side-mounted 200-beam LiDAR sensors. The maximum operating range of the LiDAR sensors is 75 m and 20 m for the mid-range and short-range, respectively. Waymo consists of 1,150 sequences that span 20-second time intervals. Waymo LiDAR features consist of the standard $(\mathrm{x}, \mathrm{y}, \mathrm{z})$ spatial coordinates, laser intensity, and Waymo's proprietary elongation measurement of the last return. Data is collected in the San Francisco, Phoenix, and Mountain View regions under a range of environmental conditions at various times of day. Additionally, there are rainy sequences obtained from Kirkland, WA. Ground truth annotations are provided for vehicles, pedestrians, and cyclists.

3.1.3 Object Classes

There is significant mismatch in terms of annotations between the nuScenes and Waymo datasets. nuScenes contains 3D object detection annotations for 10 classes, whereas Waymo only contains annotations for vehicles, cyclists, and pedestrians. Furthermore, the vehicle class in Waymo encompasses different types of vehicles that have distinct labels in nuScenes such as motorcycles. Therefore, many previous works label all Waymo vehicles as "car", and perform 3D object detection for a single car class [23, 29, 28, 21, 20, 32, 14, 30]. In this work, we follow Huang *et al.* and create distinct car, truck, bus, and motorcycle labels for the Waymo vehicle objects using the Waymo semantic segmentation labels in order to properly compare with nuScenes. As a results, the proposed 3D object detector is tasked with detecting the 6 classes that are common to both nuScenes and Waymo: {car, truck, bus, motorcycle (motor.), cyclist, and pedestrian (ped.)}.

3.1.4 Domain Gaps

Significant domain shifts occur between the nuScenes and Waymo data. In this section we will describe the primary sources for the observed shifts. The first domain shift is caused by the different LiDAR configurations as described in Section 3.1.1 and Section 3.1.2. The Waymo LiDAR configuration yields a significantly different LiDAR beam pattern and high point densities as a result of the high beam number side-mounted sensors. The second domain shift is caused by the different locations sampled in each dataset. The nuScenes dataset contains scenes from Singapore, whereas Waymo only contains scenes from American cities. This results in different object distributions and different background features due to the environments.

3.2 Models

We employ CenterPoint [31] as the base 3D object detector in this work for both the student and the teacher models. CenterPoint was found to be the best architecture investigated in Sparse2Dense [22]. We used the implementation developed by mmdetection3d [35] library and performed the necessary modification and adaptations to implement the proposed method. All models were trained using 100% nuScenes data and 20% Waymo data with a batch size of 8 across 4 GPUs. Optimization during training is performed using the AdamW optimizer with the one cycle cyclic learning rate schedule. In this section we will give the training details for the 4 models used in the Sparse2SOAP method: Sparse model, CenterPoint with SOAP model, FT with SOAP model, and the Sparse2SOAP (S2S) Teacher-Student (TS) model.

3.2.1 Sparse Model

The Sparse CenterPoint model is obtained by training CenterPoint on sparse point clouds from random initial weights. The model is trained with SN augmentations [23] for domain adaptations, which uses the statistics of the source and target domains to resize object bounding boxes and object point clouds. The Sparse CenterPoint was trained on nuScenes for 20 epoch with CBGS sampling [36] applied to the data and a maximum learning rate of 10^{-3}.

3.2.2 CenterPoint with SOAP Model

The CenterPoint with SOAP model is initialized with the Sparse CenterPoint weights and trained on SA source domain point clouds for 52,700 iterations with a maximum learning rate of 10^{-3}. Since the CenterPoint with SOAP model is trained to only detect QS objects, the CenterPoint with SOAP training does not provide supervision for cyclist and pedestrians as these objects are almost never considered to be quasi-stationary. Quasi-stationary training is implemented with a threshold of $\delta = 0.8$ for QSS. Additionally, SCP is implemented with $\mu = 0.5$ for clustering and $\eta = 2$ for filtering [10].

3.2.3 FT with SOAP Model

The FT CenterPoint with SOAP model is initialized with the Sparse CenterPoint model weights and fine-tuned on the sparse target-domain point clouds for 5 epochs. The FT CenterPoint with SOAP model regression is performed on the SOAP pseudo-labels with a fixed learning rate of 10^{-4}. We follow Caine *et al.* [3] and only use pseudo-labels with a confidence scores > 0.5.

3.2.4 Sparse2SOAP (S2S) Model

Sparse2SOAP (S2S) consists of a teacher and a student model. The teacher is optimized to operate on PCL2 and the student is optimized to operate on sparse point clouds. To obtain the teacher model, refined pseudo-labels are first constructed by performing SCP only for the QS objects detected by the CenterPoint with SOAP model. The S2S teacher is then initialized with the CenterPoint with SOAP model weights and fine-tuned with the same hyper-parameters as the FT with SOAP model described in Section 3.2.3.

The teacher and student components of the Sparse2SOAP model are initialized with the S2S teacher weights. S2S is then trained for 20 epochs on the sparse target domain point clouds (PCL3) with PCL0 and PCL2 processed by the teacher for knowledge distillation. The maximum learning rate employed during S2S training is 10^{-3}. We also train an ablation model called S2S Distill that is treated identically to S2S, but we remove the S2D module and the distillation loss at point A. The purpose of this ablation is to investigate performance of only knowledge distillation without the extra capacity gained by the S2D module.

3.3 Metrics

In this section we will cover the two metrics employed in this thesis to quantify the performance of the experiments. The metrics that are used are: the Percent Gain (% Gain) and the Percent Domain Gap Closed (% Closed).

3.3.1 Percent Gain

The percent gain is used to quantify the experiments performed regarding the quality of the pseudo-labels. The percent gain is simply the difference between the model being analyzed and the baseline model and is given by Equation 3.1:

$$\%Gain = AP_{model} - AP_{baseline} \tag{3.1}$$

3.3.2 Percent Domain Gap Closed

The percent domain gap closed is used to quantify the amount of the domain gap that the model being analyzed has recovered using the baseline as the lower bound and the oracle model as the upper bound. The percent domain gap closed is given by Equation 3.2:

$$\%Closed = \frac{AP_{model} - AP_{baseline}}{AP_{oracle} - AP_{baseline}} \cdot 100 \tag{3.2}$$

3.4 Pseudo-Label Quality

Here we demonstrate the improvement obtained by fine-tuning the CenterPoint with SOAP model on the PCL2 formatted Waymo point clouds to obtain the S2S Teacher model. This is important as the CenterPoint models employed in the work of [10] *et al.* only operate on either the SA point clouds or the sparse point clouds, whereas this work requires a teacher model that is optimized to operate on the PCL2 point clouds.

We observe in this work that an overall increase in pseudo-label quality for the FT S2S teacher model. The pseudo-label quality for the car class is given in Table 3.1. All models were trained on nuScenes as the source domain and evaluated on Waymo as the target domain while only using unlabeled target domain data as described in Chapter 2 of this thesis.

Model	**Overall**		**0-30 m**		**30-50 m**		**50+ m**	
	Level 1	% Gain	Level 1	% Gain	Level 1	% Gain	Level 1	% Gain
Sparse	28.9	-	59.9	-	13.8	-	2.2	-
FT with SOAP	43.4	+55.8	62.8	+15.4	**38.6**	**+70.5**	**20.1**	**+71.3**
S2S Teacher	**44.0**	**+58.1**	**69.6**	**+51.6**	37.0	+66.0	16.9	+58.6
Oracle	54.9	-	78.7	-	49.0	-	27.3	-

Table 3.1: Pseudo-label quality for the car class for the 3 distance intervals by performing prediction on PCL2 formatted point clouds. The source domain is nuScenes and the target domain is Waymo. Evaluation is w.r.t. the Waymo validation set. The table gives the performance on Level 1 objects, and the +/- sign indicates an increase/decrease in performance relative to the sparse model. **Bold** indicates the best performing model.

The pseudo-label quality is increased in the short distance regime and in the overall distance performance for Sparse2SOAP where the % Gain is observed to be 51.6% and 58.1%, respectively. However, the FT with SOAP model out performs Sparse2SOAP in the moderate (30-50 m) and far (50+ m) distance regimes. This is most likely due to the fact that the dense object point clouds in PCL2 are constructed using pseudo-labels, which are more accurate in the short (0-30 m) distance regime.

3.5 Domain Adaptation for 3D Object Detection

In Table 3.1 we observed an increase in pseudo-label quality for Sparse2SOAP. Analyzing Table 3.2, we observe a consistent increase in domain adaptation performance in all distance regimes for the car class. Table 3.2 shows an approx. 2%, 3%, and 1% increase for 0-30 m, 50-80 m, and 50+ m regimes, respectively.

The DA results for all 6 classes is given Table 3.3 and Table 3.4. The average column in both tables is the same and is the mAP (Mean Average Precision) across all 6 classes. In general, the average performance for FT with SOAP is better than the Sparse2SOAP model by a small margin (0.9%).

In Table 3.3 it is observed that the bus and truck classes see a degradation in performance. It should be noted that all other models have poor performance on truck. The difficulties in truck most likely stem from differences in the label definitions between nuScenes and Waymo.

Model	**Overall**		**0-30 m**		**30-50 m**		**50+ m**	
	Level 1	% Closed	Level 1	% Closed	Level 1	% Closed	Level 1	% Closed
Sparse	33.2	-	63.3	-	19.6	-	3.6	-
FT with SOAP	47.8	+41.4	74.3	+43.8	37.6	+40.1	15.4	+32.5
S2S Distill	48.8	+44.2	75.0	+46.7	39.7	+44.6	16.2	+34.9
S2S	**49.8**	**+47.3**	**76.2**	**+51.2**	**40.9**	**+47.5**	**16.9**	**+36.8**
Oracle	68.3	-	88.5	-	64.6	-	39.7	-

Table 3.2: Domain adaptation results for Sparse2SOAP on the car class for the 3 distance intervals. The source domain is nuScenes and the target domain is Waymo. Evaluation is w.r.t. the Waymo validation set. The table gives the performance on Level 1 objects, and the sign +/- indicates a increase/decrease in performance relative to the sparse model. **Bold** indicates the best performing model.

The DA results for motorcycle, cyclist, and pedestrian are given in Table 3.4. It is observed that motorcycle sees an increase in Waymo Level 1 performance and that the domain gap is closed by nearly 30%. However, Sparse2SOAP does not perform nearly as well on cyclist and pedestrian. The performance is better than the baseline but does not surpass the FT with SOAP model performance. This is most likely due to the lack of supervision from the teacher during training. Most of the cyclist and pedestrian instances are dynamic and as described in Section 2.1.2 those region will not receive any guidance from the teacher on how to map the sparse features to the dense features. However, since there is a small domain gap between motorcycles, cyclists, and pedestrians, then these results suggest that more supervision for cyclist and pedestrian may result in similar gains to motorcycles.

3.6 Precision-Recall Curves

Figure 3.1 shows the precision-recall (PR) curves for the car class for the models analyzed in this work. PR curves demonstrate the model's performance as a function of the decision thresholds. There are regions on the PR curve in which the FT with SOAP model out performs the Sparse2SOAP model. To analyze if the Sparse2SOAP model is indeed more optimal than FT with SOAP, the F1-score was calculated using Equation 3.3:

Model	**Average**		**Car**		**Truck**		**Bus**	
	Level 1	% Closed	Level 1	% Closed	Level 1	% Closed	Level 1	% Closed
Sparse	15.5	-	33.2	-	4.0	-	10.8	-
FT with SOAP	**24.0**	**+20.8**	47.8	+41.5	**6.7**	**+8.9**	**24.6**	**+36.1**
S2S Distill	22.2	+16.5	48.8	+44.2	5.2	+3.9	16.2	+14.1
S2S	23.1	+18.7	**49.8**	**+47.3**	5.7	+5.5	18.4	+19.9
Oracle	68.3	-	88.5	-	64.6	-	39.7	-

Table 3.3: Domain adaptation results for Sparse2SOAP on the car, truck, and bus classes. The source domain is nuScenes and the target domain is Waymo. Evaluation is w.r.t. the Waymo validation set. The table gives the performance on Level 1 objects, and the +/- sign indicates an increase/decrease in performance relative to the sparse model. The average is calculated over car, truck, bus, motorcycle, cyclist, and pedestrian in Table 3.3 and 3.4. **Bold** indicates the best performing model.

Model	**Average**		**Motorcycle**		**Cyclist**		**Pedestrian**	
	Level 1	% Closed	Level 1	% Closed	Level 1	% Closed	Level 1	% Closed
Sparse	15.5	-	21.4	-	5.5	-	17.9	-
FT with SOAP	**24.0**	**+20.8**	27.5	+19.3	**13.0**	**+11.9**	**24.4**	**+13.6**
S2S Distill	22.2	+16.5	**30.7**	**+29.5**	9.8	+6.8	22.8	+10.3
S2S	23.1	+18.7	30.6	+29.2	11.2	+9.1	22.9	+10.5
Oracle	68.3	-	88.5	-	64.6	-	39.7	-

Table 3.4: Domain adaptation results for Sparse2SOAP on the motorcycle, cyclists, and pedestrian classes. The source domain is nuScenes and the target domain is Waymo. Evaluation is w.r.t. the Waymo validation set. The table gives the performance on Level 1 objects, and the +/- sign indicates an increase/decrease in performance relative to the sparse model. The average is calculated over car, truck, bus, motorcycle, cyclist, and pedestrian in Table 3.3 and 3.4. **Bold** indicates the best performing model.

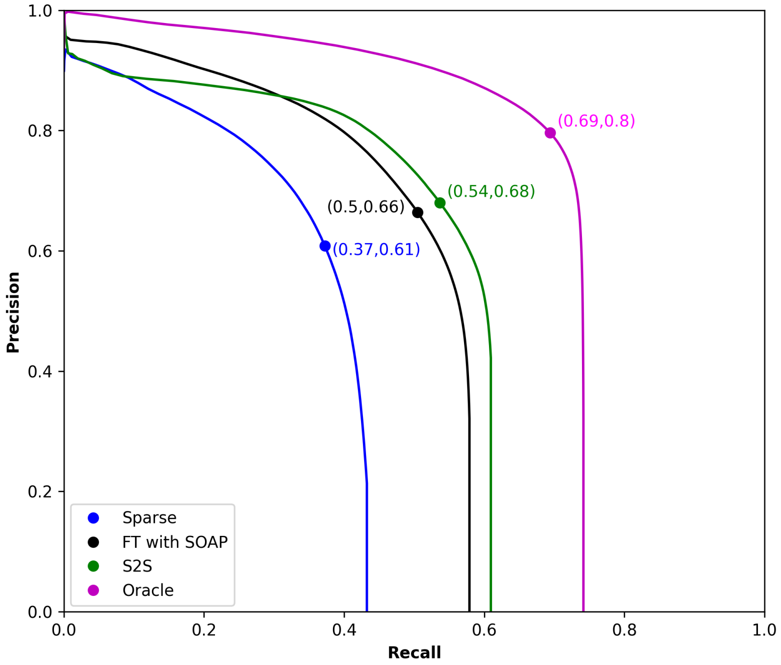

Figure 3.1: Precision-recall curves for the car class by models trained with nuScenes as the source domain and evaluated with Waymo as the target domain.

$$F1 = \frac{2 \cdot p \cdot r}{p + r} \cdot 100 \tag{3.3}$$

Determining the threshold that corresponds to the maximum in F1-score is the threshold that gives the best trade-off between precision and recall. That threshold is plotted as a single scatter point for each model in Figure 3.1 and we see that at the optimal threshold, the Sparse2SOAP model outperforms SOAP and the baseline model in terms of precision and recall.

Chapter 4

Limitations, Future Work and Conclusion

In Chapter 4 of this thesis, we first describe the limitations of the proposed Sparse2SOAP method. Then, we discuss some possible future work regarding additional experiments on other DA appropriate dataset and better techniques for object sampling. Lastly, we conclude this thesis.

4.1 Limitations

Our Sparse2SOAP method has three fundamental limitations. The first two limitations occur as a result of utilizing the SOAP method for generating pseudo-labels, while the third is a more general limitation of DA. Firstly, as noted by Huang *et al.* [10] the construction of the SA point clouds in the SOAP method requires that the LiDAR data be obtained in a sequential fashion and that the corresponding ego pose information be available. This requirement can be a limiting factor if it is desirable to follow this approach.

Secondly, the Sparse2SOAP teacher is based on the SOAP model, which is explicitly trained to detect QS objects. Therefore, we apply a mask as detailed in Section 2.1.2 and visualized in Figure 2.4 such that the Sparse2SOAP student is not penalized for the lack of dense knowledge for dynamic objects. As a result, the Sparse2SOAP student receives less supervision for these objects and correspondingly does not experience the same gain in performance for classes with all or mostly dynamic objects compared to classes with mostly QS objects.

Model	**0-30 m**		**30-50 m**		**50+ m**	
	mAP	% Gain	mAP	% Gain	mAP	% Gain
Baseline	70.6	-	52.3	-	32.6	-
SECOND	72.9	+2.3	55.3	+3.0	36.3	+3.7
Baseline	74.2	-	61.6	-	42.9	-
CenterPoint-Pillars	77.9	+3.7	66.7	+5.1	49.4	+6.5
Baseline	80.8	-	64.2	-	45.4	-
CenterPoint-Voxels	**82.7**	**+1.9**	**67.6**	**+3.4**	**49.5**	**+4.1**

Table 4.1: Performance gain over baseline approaches on Waymo validation set in different ranges from Sparse2Dense [22]. **Bold** indicates the best performing model.

Type	**Distance (r)**	**Percentage**	**Number of Labels**
Ground Truth Labels	$r < 30$	33.97	279,093
	$30 \leq r < 50$	32.38	266,091
	$r \geq 50$	33.65	276,471
S2S Pseudo Labels	$r < 30$	42.63	263,169
	$30 \leq r < 50$	33.93	209,469
	$r \geq 50$	23.44	144,665

Table 4.2: The percentage and number of labels in the three ranges analyzed for the ground truth and pseudo-labels.

The third limitation is observed when comparing the Sparse2SOAP results in Table 3.2 to the Sparse2Dense results in Table 4.1. The authors of Sparse2Dense observe a performance gain in all three distance intervals, but measure the greatest gain over baseline for far range objects. Thus, the Sparse2Dense method is shown to be effective at densifying features for difficult far range objects. However, we do not see the trend in Table 3.2. This can be explained by analyzing the number and percentage of labels in the three distance intervals for ground truth labels and pseudo-labels as shown in Table 4.2. The ground truth labels are well distributed over the three distance intervals, whereas the pseudo-labels are biased towards the closer distance interval. This is expected as the pseudo-labels are obtained by a DNN model, which will be more likely to produce high confidence pseudo-labels for higher point density objects in the close range regime. Therefore, this is a limitation that arises in the DA task that is not encountered when using ground truth labels.

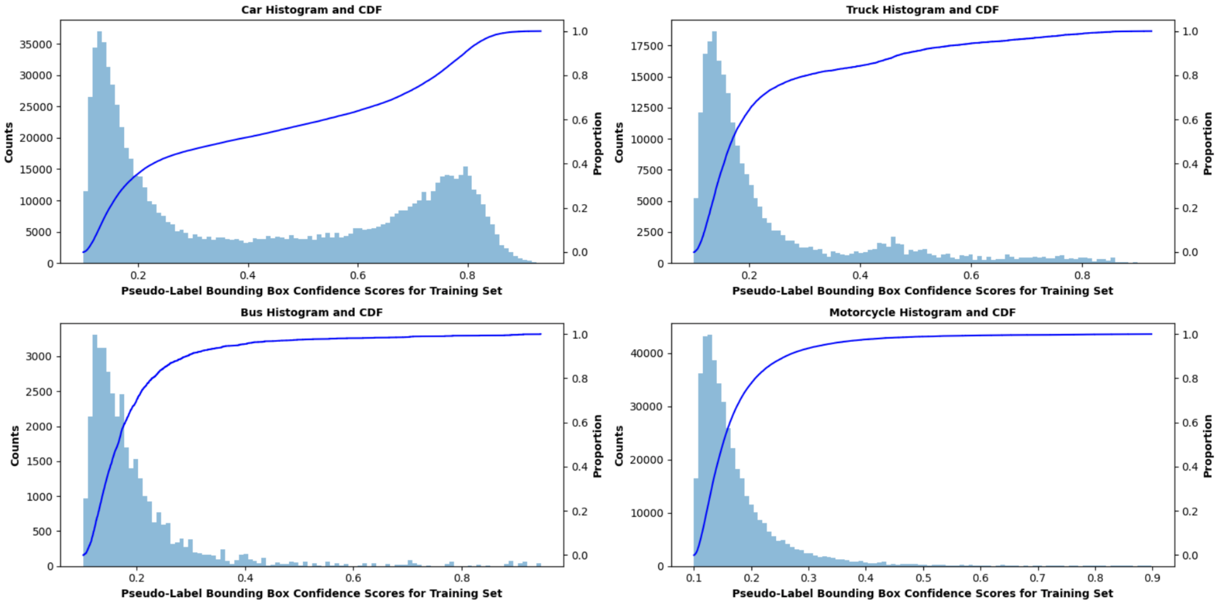

Figure 4.1: CDF/histogram plots for the number of predicted bounding boxes (pseudo-labels) as a function of their confidence scores.

4.2 Future Work

In this work, we only investigated the scenario for when nuScenes is the source domain and Waymo is the target domain. In this scenario, we found that the Sparse2SOAP model is effective for the task of DA and can surpass the performance of other similar models in terms of %Gain or %Closed. Thus, we intend to further investigate the ability of Sparse2SOAP and analyze the scenario when Waymo is the source domain and nuScenes is the target domain. We also hope to incorporate other datasets that are appropriate for domain adaptation.

It was noted previously in this work that we follow Caine *et al.* [3] and only use pseudo-labels with a confidence score greater than 0.5. These pseudo-labels are used to extract object point clouds for sampling additional instances into each training frame. This is a common practice as many frames will only have a few object examples of each object type and sampling more examples can help the model to converge much faster. Furthermore, object augmentations are applied to increase the diversity of the objects seen by the model. The Cumulative Distribution Function (CDF)/histogram of the number of predicted bounding boxes as a function of the confidence scores is shown in Figure 4.1. This figure shows that the vast majority of the predicted bounding boxes have low confidence scores. Also, it is feasible that a lot of the pseudo-labels used that are close to the threshold may be noisy or low quality. Therefore, there is a trade-off between the quality and the diversity of the objects used. A high threshold applied to the predicted pseudo-labels will yield a small number of high quality objects with little diversity, whereas a low threshold will yield a diverse set of objects but many will be poor in quality. Most work regarding object point clouds only focus on upsampling object points [33] or use CAD model to create sparse object point clouds to be sampled into scene point clouds [8]. We think there is more opportunity to intelligently inject object point clouds to increase data diversity. This would allow one to use a high threshold for the pseudo-labels but maintain the diversity of the samples.

4.3 Conclusion

In this work, we present the work Sparse2SOAP that extends Sparse2Dense, an in-domain 3d object detector framework, and shows that it can be adapted for DA. To achieve this, we employ SOAP for generating high quality pseudo-labels. Then, we develop a data pipeline for constructing target domain versions of the point clouds proposed by Sparse2Dense and modify the Sparse2Dense knowledge distillation techniques to account for differences in QS and dynamic objects. We use two large-scale well known autonomous driving datasets in our evaluation and showed the effectiveness of the proposed method when nuScenes is the source domain and Waymo is the target domain. The proposed method is shown to perform well on objects with well-defined labels and many QS object instances. However, objects with poorly defined labels (trucks) and objects that consist of mostly dynamic instances (cyclists and pedestrians) do not experience the same increase in performance.

References

[1] Alejandro Barrera, Jorge Beltrán, Carlos Guindel, Jose Antonio Iglesias, and Fernando García. Cycle and semantic consistent adversarial domain adaptation for reducing simulation-to-real domain shift in lidar bird's eye view, 2021.

[2] 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. 2020.

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

[4] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3D tracking and forecasting with rich maps. 2019.

[5] Eduardo R. Corral-Soto, Mrigank Rochan, Yannis Y. He, Shubhra Aich, Yang Liu, and Liu Bingbing. Domain adaptation in lidar semantic segmentation via alternating skip connections and hybrid learning, 2022.

[6] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds for autonomous driving, 2020.

[7] Liang Du, Xiaoqing Ye, Xiao Tan, Jianfeng Feng, Zhenbo Xu, Errui Ding, and Shilei Wen. Associate-3ddet: Perceptual-to-conceptual association for 3d point cloud object detection, 2020.

[8] Jin Fang, Dingfu Zhou, Jingjing Zhao, Chulin Tang, Cheng-Zhong Xu, and Liangjun Zhang. Lidar-cs dataset: Lidar point cloud dataset with cross-sensors for 3d object detection, 2023.

[9] 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.

[10] Chengjie Huang, Vahdat Abdelzad, Sean Sedward, and Krzysztof Czarnecki. Soap: Stationary object aggregation pseudo-labelling for lidar-based 3d object detection. Unpublished paper, WISE Lab, University of Waterloo, Waterloo, ON, Canada, 2023.

[11] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet. Level 5 perception dataset 2020. https://level-5.global/level5/data/, 2019.

[12] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds, 2019.

[13] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018.

[14] 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. pages 8866–8875, 2021.

[15] Jiageng Mao, Minzhe Niu, Chenhan Jiang, hanxue liang, Jingheng Chen, Xiaodan Liang, Yamin Li, Chaoqiang Ye, Wei Zhang, Zhenguo Li, Jie Yu, Hang Xu, and Chunjing Xu. One million scenes for autonomous driving: ONCE dataset. In *35th Conference on Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track (Round 1)*, 2021.

[16] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[17] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection, 2021.

[18] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud, 2019.

[19] 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. 2020.

[20] Darren Tsai, Julie Stephany Berrio, Mao Shan, Eduardo Nebot, and Stewart Worrall. Viewer-centred surface completion for unsupervised domain adaptation in 3D object detection, 2022. arXiv: 2209.06407.

[21] Darren Tsai, Julie Stephany Berrio, Mao Shan, Stewart Worrall, and Eduardo Nebot. See eye to eye: A LiDAR-agnostic 3D detection framework for unsupervised multi-target domain adaptation. *IEEE Robotics and Automation Letters*, 7(3):7904–7911, 2022.

[22] Tianyu Wang, Xiaowei Hu, Zhengzhe Liu, and Chi-Wing Fu. Sparse2dense: Learning to densify 3d features for 3d object detection, 2022.

[23] 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. 2020.

[24] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[25] 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, 2018.

[26] 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 *ICCV*, 2021.

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

[28] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. ST3D++: Denoised self-training for unsupervised domain adaptation on 3D object detection, 2021. arXiv: 2103.05346.

[29] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. ST3D: Self-training for unsupervised domain adaptation on 3D object detection. 2021.

[30] 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. volume 34, pages 21493–21504. Curran Associates, Inc., 2021.

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

[32] Yurong You, Carlos Andres Diaz-Ruiz, Yan Wang, Wei-Lun Chao, Bharath Hariharan, Mark Campbell, and Kilian Q Weinbergert. Exploiting playbacks in unsupervised domain adaptation for 3D object detection in self-driving cars. 2022.

[33] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[34] 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 (CVPR)*, pages 6769–6779, June 2021.

[35] Yihong Zhao, Shiyi Huang, Ting Wang, Yeming Wang, Yixiao Zhang, Di Xie, Chen Chen, Xinge Guo, Yu Qiao, and Dahua Lin. Mmdetection3d: Openmmlab's next-generation platform for 3d object detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12257–12266, 2021.

[36] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection, 2019.

[37] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2020.