

# CL3D: Unsupervised Domain Adaptation for Cross-LiDAR 3D Detection

Xidong Peng<sup>1</sup>, Xinge Zhu<sup>3</sup>, Yuexin Ma<sup>1,2\*</sup>

<sup>1</sup>ShanghaiTech University

<sup>2</sup>Shanghai Engineering Research Center of Intelligent Vision and Imaging

<sup>3</sup>The Chinese University of Hong Kong

{pengxd, mayuexin}@shanghaitech.edu.cn

## Abstract

Domain adaptation for Cross-LiDAR 3D detection is challenging due to the large gap on the raw data representation with disparate point densities and point arrangements. By exploring domain-invariant 3D geometric characteristics and motion patterns, we present an unsupervised domain adaptation method that overcomes above difficulties. First, we propose the Spatial Geometry Alignment module to extract similar 3D shape geometric features of the same object class to align two domains, while eliminating the effect of distinct point distributions. Second, we present Temporal Motion Alignment module to utilize motion features in sequential frames of data to match two domains. Prototypes generated from two modules are incorporated into the pseudo-label reweighting procedure and contribute to our effective self-training framework for the target domain. Extensive experiments show that our method achieves state-of-the-art performance on cross-device datasets, especially for the datasets with large gaps captured by mechanical scanning LiDARs and solid-state LiDARs in various scenes. Project homepage is at <https://github.com/4DVLab/CL3D.git>.

## Introduction

Due to the advantages of accurately capturing depth information of large-scale scenes, LiDARs become crucial sensors for the 3D perception of autonomous driving and robotics. Boosted by deep learning techniques, LiDAR-based 3D detection (Yan, Mao, and Li 2018; Zhu et al. 2021; Qi et al. 2019; Zhu et al. 2020; Chen et al. 2020; Shi and Rajkumar 2020; Zhang, Hu, and Xu 2022; Cong et al. 2022; Hou et al. 2022) has made great progress and becomes the main solution for many autonomous driving companies. However, deep learning-based methods rely heavily on massive annotated data, which is time-consuming and expensive, especially for labeling 3D point clouds of large scenes. Moreover, the domain adaptation for point clouds gets more challenging compared to image-based adaptation tasks (Chen et al. 2018; Saito et al. 2019; Zhu et al. 2018; Cui et al. 2020; Li, Ji, and Qu 2022) where the image-based domain gap is more about the explicit appearance including lighting and weather while the domain gap existing in

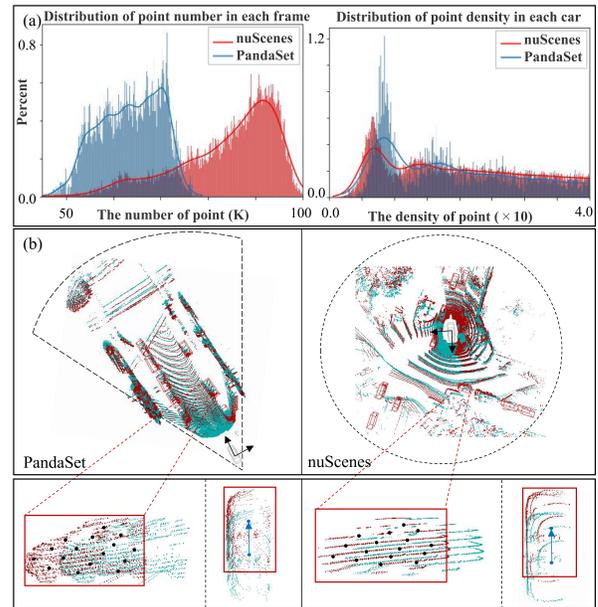


Figure 1: We show several key domain gaps between mechanical scanning LiDAR (from nuScenes) and solid-state LiDAR (from PandaSet). (a) shows the comparison of statistics of point numbers and point densities. (b) describes domain gaps of perception range and instance-level point distribution. Two adjacent frames are shown in green and red. The geometric structure represented by the black sampling points in the side-view and the motion pattern in the bird-eye-view are domain-invariant for cars.

LiDAR is mainly reflected in the raw point representation, which causes image-based domain adaptation strategies inapplicable.

Different types of LiDAR have obvious difference on the point representation. For the widely-used mechanical scanning LiDARs, they have various beams with different point densities. Compared with them, solid-state LiDARs are based on another principle of physics, having disparate perception ranges, point densities, and point arrangements. We show the statistics and visualization to describe the domain gaps in Figure. 1. Actually, solid-state LiDARs become more and more popular because it is cheaper and has longer

\*Corresponding author

perception distance and longer longevity, which makes autonomous vehicles installed with them more feasible for quantity production. However, most current open large-scale 3D datasets (Geiger, Lenz, and Urtasun 2012; Caesar et al. 2020; Sun et al. 2020) are captured by mechanical LiDARs. How to solve the domain adaptation problem between these two distinct LiDARs becomes extremely urgent and significant. Previous works (Yang et al. 2021; Wang et al. 2020; Luo et al. 2021; Zhang, Li, and Xu 2021) mainly consider the consistent objects’ sizes for the domain alignment. Actually, the more important domain-invariant features in 3D point cloud are geometric characteristics including shapes, scales, etc., and temporal motion information. For example, sedan cars bear resemblance in shape and motion representation; the poses and actions of pedestrians are also alike.

Motivated by this, we propose **CL3D**, an unsupervised domain adaptation method for LiDAR-based 3D detection, especially for solving the cross-device setting, via exploring the spatial geometry information and temporal motion representation. Specifically, we develop a framework consisting of two key components, **Spatial Geometry Alignment (SGA)** and **Temporal Motion Alignment (TMA)**, to extract the geometric features of both local structure and global context, and model the motion pattern in consecutive point clouds, respectively. Based on SGA and TMA, a prototype representation with geometry and motion constraints for each specific class is proposed, where the similarity between current sample and average target prototype (updated via exponential moving average) is used to reweight the confidence of pseudo labels generated by the self-training framework. Different from existing rigid pseudo-label selection process, this proposed soft-selection mechanism could reduce the effect of incorrect labels and avoid directly discarding correct labels with low confidence. In addition, we explore several strategies to solve the perception range gaps between mechanical scanning LiDARs and solid-state LiDARs and give conclusions to facilitate the data pre-processing.

We conduct extensive experiments on various cross-LiDAR and synthetic-to-real domain adaptation tasks, and all get state-of-the-art performance. We also conduct detailed ablation studies quantitatively and qualitatively to demonstrate the effectiveness of different modules of our method. To our knowledge, we are the first to explore the point cloud-based 3D detection domain adaptation for the challenging mechanical-to-solid-state cross-LiDAR setting.

Our contributions are summarized as follows.

- We investigate the cross-LiDAR domain gap and propose an unsupervised domain adaptation method for 3D detection, especially for the challenging mechanical to solid-state LiDAR adaptation.
- We propose SGA and TMA to extract similar 3D shape geometric characteristics and motion patterns belonging to objects of the same category to align two domains by reweighting pseudo labels with soft constraints.
- Our method achieves state-of-the-art performance on unsupervised domain adaptation for cross-LiDAR 3D Detection.

## Related Work

**LiDAR-based 3D Object Detection** Due to the advantages of capturing depth information in large-scale scenes, more and more LiDAR-based methods for 3D object detection are proposed in recent years. These methods can be divided into point-based methods and grid-based methods. The former (Yang et al. 2020b; Qi et al. 2019; Chen et al. 2020; Shi and Rajkumar 2020; He et al. 2022) directly extracts the features from unordered raw point clouds data with PointNet (Qi et al. 2017a) or PointNet++ (Qi et al. 2017b) to generate 3D proposals. These methods preserve the original geometric features of point cloud but become time-consuming for processing large-scale outdoor data. The latter (He et al. 2020; Shi et al. 2020b,a; Yan, Mao, and Li 2018; Zhu et al. 2021, 2020; Hu et al. 2022; Zhang, Hu, and Xu 2022) utilizes structured representation to quantize the LiDAR data into the fix-sized voxel or pillar grids and then use this representation to further extract semantic features for 3D object detection. Such methods perform well in efficiency and are usually adopted in autonomous driving scenarios. In our work, we chose the state-of-the-art grid-based 3D detector CenterPoint (Yin, Zhou, and Krahenbuhl 2021) as base network, which is a one-stage anchor-free method with high efficiency and accuracy. Instead of generating a new detector, we aim at adding new modules to basic detector to adapt it to the unsupervised domain adaptation task.

**Domain Adaptation for 2D image** There are already a lot of investigations (Hoffman et al. 2018; Chen et al. 2018; Saito et al. 2019; Zhu et al. 2018; Li, Ji, and Qu 2022; Yu et al. 2022) about domain adaptation for various 2D computer vision tasks such as image-based detection and segmentation. Many domain adaptation methods (Hoffman et al. 2016; Ganin et al. 2016; Bousmalis, Trigeorgis, and et al. 2016; Cui et al. 2020; Hu et al. 2020; Yang et al. 2020a) use adversarial learning to align feature distributions across different domains inspired by GANs (Goodfellow et al. 2014), while some statistic-based methods (Mancini et al. 2018; Maria Carlucci et al. 2017; Long et al. 2017; Sun and et al. 2016; Xu et al. 2020; Long et al. 2015) employ the statistic-based metrics to the domain gap between two different data distributions. Moreover, pseudo-label-based self-training is becoming a more and more popular approach (Khodabandeh et al. 2019; RoyChowdhury et al. 2019; Seibold et al. 2022; Yao, Hu, and Li 2022) for unsupervised domain adaptation, which is easier to implement compared with previous two kinds of methods. Our method also adopts such self-training mechanism for the target domain. However, unlike images with unchanged regular pixel representations for different environments and devices, LiDAR point cloud is changing apparently on the raw data representation with diverse point densities and distributions, which is not applicable to directly extend the image-based adaptation methods to LiDAR point cloud. Our method fully explores the specific properties of 3D data and achieves superior performance.

**Domain Adaptation for 3D point cloud** To bridge the domain gap on point clouds captured in various environments and by different LiDAR sensors, some works appear recently for shape classification (Qin et al. 2019), semantic

segmentation (Wu et al. 2018; Yi, Gong, and Funkhouser 2021; Jaritz et al. 2020; Xiao et al. 2022; He, Yang, and Qi 2021) and 3D detection (Hegde and Patel 2021; Luo et al. 2021; Xu et al. 2021; You et al. 2022). We focus on the 3D detection task. To align geometry features, (Yihan et al. 2021) transforms the 3D feature to the bird’s eye view with regular structures as images. However, dimensionality reduction usually misses detailed 3D geometry characteristics and is not friendly for small objects. (Wang et al. 2020) utilizes the object size statistics of two domains to narrow the gap, but the performance relies on the source and target data distributions. SRDN (Zhang, Li, and Xu 2021) aligns the features according to the instance sizes and distances to the LiDAR sensor. It is reasonable for the data captured by the same LiDAR, but not applicable for the cross-device data, where even the same object at the same location will be presented differently with different densities and arrangements. Compared with these works using the object size, our work considers a more important geometric characteristic, the shape, consisting of both the size and structure information, as well as the temporal features to pull two domains closer. ST3D (Yang et al. 2021) generates pseudo labels of the target domain by the model trained on source domain and then selects high-quality pseudo labels for self-supervision. However, the threshold-based method for pseudo-label selection usually leads to involving incorrect labels that have high confidence and discarding correct labels with low confidence in the self-training procedure, which may mislead the network. Our method uses a soft-selection mechanism for pseudo labels by reweighting the confidence, which can alleviate above situations to a certain extent.

## Methodology

### Problem Statement

Given the source domain, denoted as  $D_S = \{P^S, L^S\}$  with point cloud data  $P^S$  and labeled samples  $L^S$ , and the target domain  $D_T = \{P^T\}$  without any annotation, our task is to train a 3D detector that can generalize well to the target domain, utilizing the data in both domains. Specifically,  $P^S$  and  $P^T$  are captured by different types of LiDAR in various environments with different numbers, densities, arrangements, and ranges of points.

### Framework Overview

To align two distinct point cloud domains, we fully explore the domain-invariant features existing in raw data representations, including the geometric characteristics and motion patterns. We design two important modules in CL3D. The first is **Spatial Geometry Alignment (SGA)**, which extracts geometric features of both local structure and global context for objects. The second is **Temporal Motion Alignment (TMA)**, which aims at learning the motion features of the same kind of objects from consecutive data. Then, we generate a **feature-fusion prototype** representation based on SGA and TMA to align the source domain and target domain with geometry and motion constraints. Specifically, the target prototype is updated via exponential moving average

(EMA) and the similarity between current sample and prototype is used to reweight the confidence of pseudo labels to optimize the detection results via this soft-selection mechanism. In particular, the perception range also differs a lot for mechanical scanning LiDAR and solid-state LiDAR. We further propose an effective strategy of **range normalization** for better pre-processing. The pipeline of our method, CL3D, is illustrated in Figure 2.

### Spatial Geometry Alignment

Different from the dense and regular representation of image pixels, points captured by LiDARs appear sparsity-varying and unordered distributions in 3D space. For various kinds of LiDAR, like widely-used mechanical scanning LiDARs with 32, 63, and 128 beams and solid-state LiDARs, there exist obvious differences in the raw data representations, leading to huge domain gaps. However, the object’s geometric structure is invariant even in diverse environments and captured by disparate devices, which inspires us to extract the essential geometry features from arbitrary point cloud representations. By the 3D detection backbone trained on the source domain, we can obtain pseudo labels to localize potential objects of the target domain. To eliminate the influence brought by diverse densities and arrangements of points and make the shape feature more robust, we adopt the farthest point sampling (FPS) method to uniformly collect 16 points from the point cloud belonging to each object as the shape representation. After that, we normalize these points by transforming the coordinates from the LiDAR coordinate system to the self-coordinate system. A two-layer multi-layer perceptron (MLP) is attached to extract the point-wise geometry feature,  $f_{local}$ , for the local structure of objects. Considering that feature map generated by the detection backbone involves the features of the global context, which contains high-level geometry information, we sample on the feature map by interpolation to get the  $f_{global}$ . Then we obtain the final geometric feature  $f_{geo}$  for each object by  $f_{geo} = \text{concat}\{f_{local}, f_{global}\}$ , which contains both local structure and global context information.

### Temporal Motion Alignment

Additionally, the dynamic properties in objects’ movement are also invariant across various domains, which can benefit the domain alignment as well. First, we input two consecutive frames of LiDAR point cloud to acquire dynamic information. Then, we add an extra motion head for the backbone network to extract the temporal feature and get the motion feature map. Finally, we conduct the feature sampling on the motion map according to the pseudo label to obtain the corresponding motion feature  $f_{mo}$  of each object. Both of the features obtained by SGA and TMA will be used to generate prototype to align two domains by prototype learning.

### Prototype Learning

In previous works, rigid filtering methods (Khodabandeh et al. 2019; RoyChowdhury et al. 2019; Yang et al. 2021) are popular for the pseudo-label-based self-supervised frameworks. Commonly, rigid filtering methods are used by set-

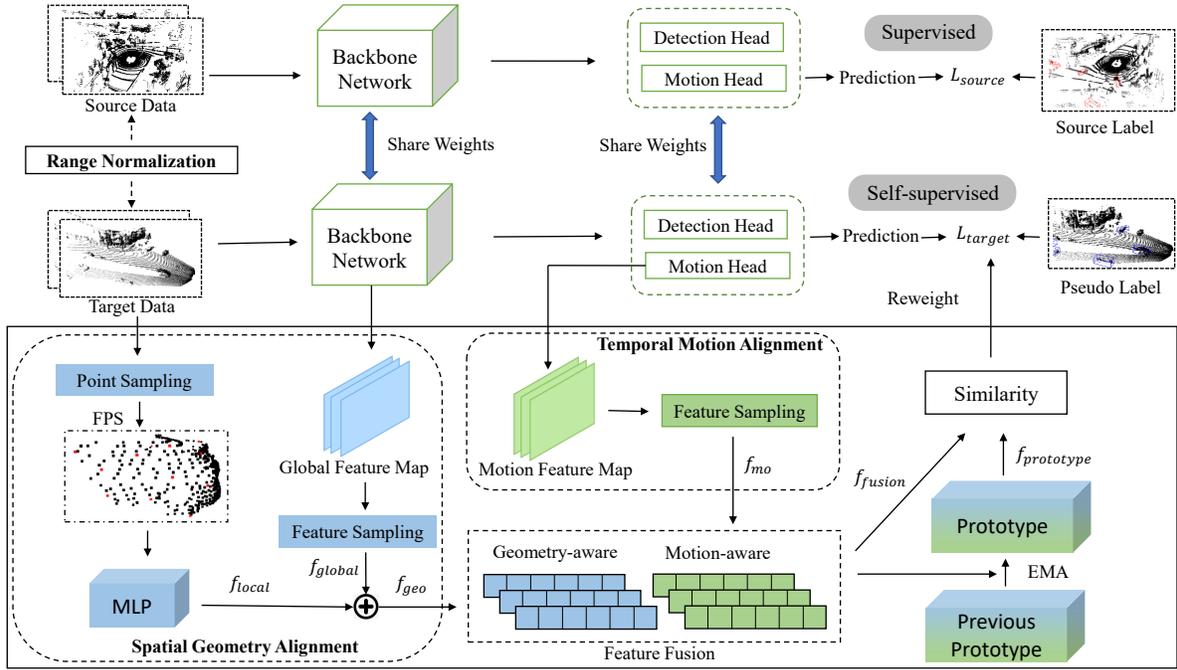


Figure 2: The pipeline of our method CL3D. Firstly, a 3D detection network is trained on source data to generate the pseudo label on target domain. During re-training on target domain, Spatial Geometry Alignment and Temporal Motion Alignment are used to extract the domain-invariant geometry-aware and motion-aware features, which are used to generate prototype for the soft selection of pseudo labels. Moreover, Range Normalization is designed to ease the range gap for better pre-processing.

ting different confidence thresholds to filter prediction results, but it may mistakenly exclude some correct but low-confidence predictions. Therefore, we hope to calculate and learn the corresponding prototype representation for each specific class in the re-training process, then reweight the information from each pseudo-label by comparing the corresponding feature of pseudo-label with the prototype representation, in order to achieve a soft limitation effect for incorrect pseudo labels in the re-training process.

Usually, the quality of pseudo label on target data would dominate the effectiveness of the network on the target domain in the self-supervised framework. Particularly, although there exist large variances between different types of LiDARs, these instance-level geometry structure and temporal motion information are consistent. To this end, we employ the spatial geometry alignment and temporal motion alignment acting as a soft selection mechanism for the pseudo labels as shown in the half bottom of Figure 2. In this way, only pseudo labels with high similarity between prototypes (current labels and template prototypes) are remained.

**Feature Fusion for Prototype.** In the traditional 2D object detection (Jiang et al. 2018; Yang et al. 2018), it is considered that the extracted semantic features of specific class from the backbone network are similar, which can be used to compute the prototype representation. However, for LiDAR-based 3D target detection, the input data is point cloud, where no texture and color information exists but the shape-aware geometric representation and dynamic motion information are presented. Therefore, we process and obtain

the underlying consistent shape information and temporal motion pattern for each object to match different domains.

Based on the geometry feature  $f_{geo}$  acquired from SGA and the motion feature  $f_{mo}$  obtained from TMA, we can update the prototype representation in each training iteration. Instead of averaging the fused feature information directly to obtain the prototype, we adjust the weights of these fused features by the confidence  $d$  corresponding to each pseudo label as  $f_{fusion} = \frac{1}{N} \sum_{i=1}^N d_i \cdot \text{concat}\{f_{geo}^i, f_{mo}^i\}$ , where  $N$  means the number of sampled features. Additionally, the prototype computed in an iteration is combined with the previous prototype through exponential moving average (EMA), so the final attentive prototype at each iteration  $j$  is  $f_{prototype}^j = \alpha f_{prototype}^{j-1} + (1 - \alpha) f_{fusion}^j$ , where  $\alpha = 0.99$ .

**Similarity-based Reweight.** Finally, the classification loss is multiplied by these cosine similarity scores between each fused feature and the prototype representation. Since we use CenterPoint (Yin, Zhou, and Krahenbuhl 2021) as the original detector. It uses the class-balanced focal loss (Lin, Goyal, and etc. 2017) as the classification loss with the calculation of classification scores prediction heatmap and ground truth heatmap. Therefore, the generation of weight  $W \in [0, 1]^{w \times h \times c}$  is similar to the ground truth heatmap  $Y \in [0, 1]^{w \times h \times c}$ , where  $w, h, c$  denotes the width, height and channels of the ground truth heatmap for classification. We use a modified Gaussian kernel function (Zhou, Wang, and Krähenbühl 2019) for each pseudo label  $p$  as shown in the function below, where  $W_{xyc}$  denotes the weight value in

the location  $(x, y)$  and channel  $c$  for reweighting,  $p_x, p_y$  denote the location of the pseudo label in heatmap,  $s_p$  denotes the corresponding cosine similarity score and  $\sigma$  is an object size-adaptive standard deviation.

$$W_{xyc} = s_p \cdot \exp\left(-\frac{(x - p_x)^2 + (y - p_y)^2}{2\sigma^2}\right)$$

With this similarity-based reweighting, the losses will pay more attention to the corresponding regions that have been identified as correct through prototype matching and therefore the pseudo-label selection would bias to these consistent patterns, resulting in the final loss for target domain  $L_{target} = W \cdot L_{target}^{cls} + L_{target}^{reg}$ , which contains a class-balanced focal loss  $L_{target}^{cls}$  for object classification, and a smooth-L1 loss  $L_{target}^{reg}$  for bounding box regression. In this way, a soft-selection-based self-supervised framework is completed.

### Range Normalization

We also observe that mechanical LiDAR and solid-state LiDAR usually capture the scene with different perception ranges. Taking the mechanical LiDAR-based nuScenes dataset (Caesar et al. 2020) and solid-state LiDAR-based PandaSet dataset (Xiao et al. 2021) for example, the former’s perception range is [-50m, 50m], and the perspective view is a 360-degree ring area, while perception range of the latter is [0m, 100m], and the visual range is only a fan area of 60 degrees straight ahead. This difference in range also leads to a domain gap.

We propose range normalization (RN) as a data pre-processing to ease the difference in sensor range from different datasets. Specifically, we centralize all the non-centralized point cloud sample data to ensure that the origin of the point cloud coordinate system is in the center of the point cloud perception range. In other words, for PandaSet dataset whose original perception range is [0, 100m], we translate the overall point cloud data and corresponding annotated bounding box information, so that its perception range becomes [-50m, 50m] which is consistent with the data perception range of nuScenes dataset. It is a simple yet efficient normalization approach and considerable improvement can be observed in the 3D detection domain adaption task as shown in the ablation studies.

## Experiment

We first introduce all datasets and evaluation metrics used in the experiments and implementation details. After that, we explore cross-LiDAR domain shift scenarios and show compared 3D detection results to demonstrate the state-of-the-art performance of CL3D. Finally, we conduct extensive ablation studies to give a comprehensive assessment of sub-modules of CL3D.

### Experimental Setup

**Datasets** We consider five widely-used large-scale autonomous driving datasets to simulate the various domain shifts, which are Waymo (Sun et al. 2020), nuScenes (Caesar et al. 2020), KITTI (Geiger, Lenz, and Urtasun 2012),

PandaSet (Xiao et al. 2021), and PreSIL (Hurl, Czarnecki, and Waslander 2019). Among them, Waymo is the largest dataset with more than 230K annotated 64-beam mechanical lidar frames collected across six US cities. nuScenes consists of 28130 training samples and 6019 validation samples collected by the 32-beam mechanical LiDAR and KITTI consists of 7,481 annotated lidar frames collected by the 64-beam mechanical LiDAR. PandaSet is the only dataset whose data is captured by solid-state LiDAR, including 5520 training samples and 2720 validation samples. Particularly, the synthetic dataset PreSIL contains 51075 synthetic LiDAR data generated from the Grand Theft Auto V (GTA V) game. Note that KITTI and PreSIL do not contain consecutive frames of data, so we will delete our TMA module for related experiments.

**Evaluation metric** We adopt the nuScenes evaluation metric for evaluating our methods on the commonly used car category in most of our experiments. The Average Precision (AP) is used as the metric and a match is defined by thresholding the 2D center distance  $d$  on the ground plane, then we average over matching thresholds of  $d = \{0.5, 1, 2, 4\}$  meters and get the mean Average Precision (mAP). As for the synthetic-to-real domain adaptation from PreSIL to KITTI, we use the KITTI evaluation metric of intersection over union (IOU) of 0.7 in order to align other methods’ performance. Refer to (Yang et al. 2021), we use **closed gap**  $= \frac{AP_{model} - AP_{DT}}{AP_{Oracle} - AP_{DT}}$  to report how much the performance gap between Direct Transfer(DT) to Oracle is closed.

**3D detection network** We use the CenterPoint detector (Yin, Zhou, and Krahenbuhl 2021) as the base network, which is a one-stage anchor-free 3D object detector. A motion head is attached for extracting motion features by the supervision of object velocities (position offset between two adjacent frames) during the training on the source domain. The classification loss is reweighed during the computation in the domain adaptation process.

**Implementation details** As for the implementation, we use the public pyTorch (Paszke, Gross, and Massa 2019) repository MMDetection3D (Contributors 2020) and we perform experiments with a 24GB GeForce RTX 3090 GPU. During both the pre-training and self-training processes, we adopt the widely adopted data augmentation, including random flipping, scaling, and rotation. The source data in pre-training process are trained for 20 epoch and target data in self-training process are trained for 1 epoch. Other settings are the same as official implementation of CenterPoint.

### Performance

We mainly demonstrate the performance on cross-LiDAR 3D detection domain adaptation tasks. We compare with several approaches, including Direct Transfer (DT), Self-Training (ST), and current published SOTA ST3D (Yang et al. 2021) by running its released code on our experimental settings. As for another SOTA work SRDAN (Zhang, Li, and Xu 2021), we compare with it by aligning the performance on synthetic-to-real domain adaptation task according to its reported results for fair comparison. In particular, Direct Transfer(DT) indicates directly evaluating pre-

Source	Target	Method	mAP	Closed Gap
nuScenes	PandaSet	DT	0.363	-
		ST	0.379	2.60%
		ST3D	0.617	57.21%
		CL3D	<b>0.705</b>	<b>77.03%</b>
		Oracle	0.807	-
PandaSet	nuScenes	DT	0.089	-
		ST	0.244	27.63%
		ST3D	0.349	46.35%
		CL3D	<b>0.467</b>	<b>68.98%</b>
		Oracle	0.650	-
Waymo	PandaSet	DT	0.272	-
		ST	0.383	20.75%
		ST3D	0.704	80.75%
		CL3D	<b>0.729</b>	<b>86.42%</b>
		Oracle	0.807	-
PandaSet	Waymo	DT	0.115	-
		ST	0.312	37.10%
		ST3D	0.423	58.00%
		CL3D	<b>0.492</b>	<b>71.00%</b>
		Oracle	0.646	-

Table 1: Comparison on 3D car detection adaptation task between mechanical LiDAR dataset (nuScenes and Waymo) and solid-state LiDAR dataset (PandaSet). Oracle indicates the fully supervised model trained on the target dataset standing for upper bound of performance after adaptation.

trained model from the source dataset on the target dataset, and Self-Training(ST) indicates re-training the object detector supervised only by the pseudo-label generated by source-model. For adaptation experiments between mechanical LiDAR and solid-state LiDAR, we also apply Range Normalization (RN) to other SOTA methods. For other cross-domain settings without perception range gap, we will ignore the RN procedure.

Table. 1 shows the adaptation results between solid-state LiDAR dataset PandaSet and mechanical LiDAR datasets, including nuScenes and Waymo, and our method attains an obvious improvement over other methods. ST is superior to DT because of the retraining under the guidance of pseudo labels. Our method outperforms ST due to the domain alignment process with the assistance of domain-invariant geometry and motion features. ST3D utilizes threshold strategy to select high-quality pseudo-labels, which may lead to involving incorrect labels that have high confidence and discarding correct labels with low confidence in the self-training procedure. Our method is superior to ST3D, which is mainly due to the soft selection mechanism by reweighting the pseudo labels via prototypes, which avoids the misjudgement of pseudo labels by hard constraints and further benefits the learning of the network. Furthermore, to demonstrate the effectiveness of our method on the domain shift caused by different beams of mechanical LiDARs, we conduct the experiment between KITTI and nuScenes in Table 2 and also get the state-of-the-art performance.

Beyond cross-LiDAR domain adaptation problems,

Source	Target	Method	mAP	Closed Gap
nuScenes	KITTI	DT	0.395	-
		ST	0.605	47.83%
		ST3D	0.625	52.39%
		CL3D	<b>0.682</b>	<b>65.38%</b>
		Oracle	0.834	-
KITTI	nuScenes	DT	0.116	-
		ST	0.179	11.80%
		ST3D	0.289	32.40%
		CL3D	<b>0.305</b>	<b>35.39%</b>
		Oracle	0.650	-

Table 2: Comparison on 3D car detection adaptation task between mechanical LiDARs with different beams, where nuScenes is 32-beam and KITTI is 64-beam.

PreSIL ->KITTI	mAP(Car)		
	Easy	Mod.	Hard
DABEV	-	17.1	-
CDN	-	19.0	-
SWDA-3D	22.6	18.7	16.3
SRDAN	25.9	22.1	18.7
CL3D	<b>28.0</b>	<b>25.3</b>	<b>23.4</b>

Table 3: Comparison of different methods under the synthetic-to-real scenario from PreSIL to KITTI dataset using the metric mAP adopted by KITTI. ‘-’ indicates the results are not available in their works.

synthetic-to-real domain adaptation is also significant due to the difficulty in collecting and annotating large-scale data in real-world scenarios. Therefore, we further validate our method under the synthetic-to-real setting using the synthetic PreSIL dataset and real KITTI dataset. We use the IOU metric of KITTI in order to align other methods’ results (Zhang, Li, and Xu 2021; Saito et al. 2019; Su et al. 2020; Saleh and Abobakr 2019). The results are shown in Tabel 3. Our method outperforms all methods by convincing margins, indicating that our method is also suitable for bridging the domain gap between the synthetic and real domain. That is because, no matter the cross-LiDAR domains or the synthetic-to-real domains, domain-invariant features are all about the geometric characteristics and motion patterns and our method exploits the invariant information to solve the domain gap essentially.

### Ablation Studies

To evaluate the effectiveness of submodules of our method, we conduct ablation studies and analyze their contributions to the unsupervised domain adaptation task from nuScenes to PandaSet. We also show the performance of using different detector backbones and on other categories of objects to further illustrate our method’s generalization capability.

**Effectiveness of SGA and TMA** We first study the effects of spatial geometry alignment (SGA) and temporal motion alignment (TMA). Table 5 and Figure. 3 show quantitative and qualitative results, respectively. SGA extracts the

Method	w/o TMA & SGA	w/o TMA	w/o SGA	CL3D
mAP	0.641	0.686	0.662	<b>0.705</b>

Table 4: Ablation study for SGA and TMA of our method.

Method	ST	ST+RSym	ST+RSp	ST+ RN
mAP	0.379	0.361	0.343	0.641

Table 5: Comparison of different solutions for range gap.

domain-invariant geometry-aware features and TMA learns the motion-aware features for the same kind of objects in different domains, which are all used during the self-training process to generate prototype representation for specific class to select high-confidence pseudo labels with soft constraints. It is obvious that both of them play critical roles.

**Effectiveness of Range Normalization** To solve the huge range gap between mechanical LiDAR and solid-state LiDAR, we design RN in data pre-processing for the range alignment. It’s simple but effective. Actually, we have tested several solutions to reduce the range gap, as Table. 5 shows. Range Symmetrization (RSym) copies the fan area of solid-state LiDARs to complete the symmetrical area as mechanical LiDARs. Range Split (RSp) splits the circular area of mechanical LiDARs into fans to align solid-state LiDARs. These intuitive methods do not work and even have negative effects, while RN produces significant performance gains by regularizing all point cloud to a general range, which eases the learning difficulty and has good generalization capability for current LiDAR categories.

**Different detection backbones** Additionally, we show the performance of using different backbones in the Center-Point detector, namely voxel-based, pillar-based and point-based backbones, which are three main types of backbones in 3D perception area, to verify the generalization capability of CL3D. Voxel-based backbone utilizes the structured voxel representation to quantize the LiDAR data while pillar-based backbone utilizes the pillar representation for efficient point process. Point-based backbone extract features from raw point cloud data directly. Results in Table

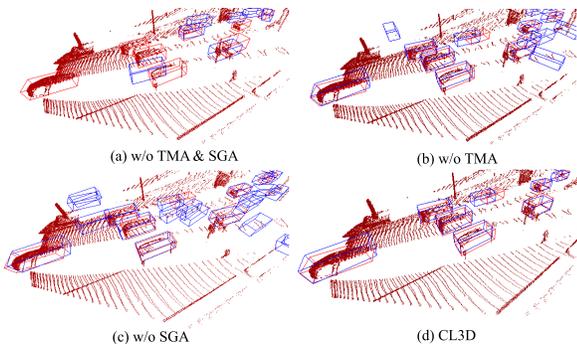


Figure 3: Visualization of 3D detection for the domain adaptation from nuScenes to PandaSet, where red boxes are ground truth and blue boxes are predictions.

Method	Backbone	voxel-based	pillar-based	point-based
	DT	0.363	0.247	0.624
ST	0.379	0.303	0.632	
CL3D		0.686	0.445	0.691

Table 6: Ablation study for different backbones of detector.

Source → Target	Method	mAP(tr.)	mAP(pe.)
nuScenes → PandaSet	DT	0.000	0.030
	ST	0.096	0.086
	CL3D	0.131	0.124
	Oracle	0.284	0.295
PandaSet → nuScenes	DT	0.000	0.003
	ST	0.072	0.153
	CL3D	0.113	0.294
	Oracle	0.275	0.524

Table 7: Results of CL3D on more challenging categories, where tr. represents the truck category and pe. represents the pedestrian category.

6 demonstrate that our method can boost the domain adaptation performance based on different detector backbones. Among them, the voxel-based backbone is a good choice with high-efficient processing for large-scale point cloud and precise detection performance.

**More challenging categories** Except for the normal car category concerned in most 3D detection domain adaptation methods, we also conduct experiments on other two important types of traffic agents, including trucks and pedestrians. Table. 7 shows experimental results. Direct Transfer can hardly obtain predictions. Our method gets improvement by a large margin, demonstrating that our method is solid for different types of detection objects. Specifically, due to the limited training samples on these challenging categories, even the result of oracle is not good.

## Conclusions

We propose an unsupervised domain adaptation method to bridge the domain gap for LiDAR-based 3D detection caused by the differences in perception range, point cloud density, and point arrangement. In particular, we design Spatial Geometry Alignment to extract similar 3D shape geometric features and Temporal Motion Alignment to extract similar motion patterns of the same category from distinct instance-level distributions to align two domains. Extensive experiments and comprehensive ablation study demonstrate the effectiveness of our approach for cross-LiDAR 3D object detection. Although prototype representation solves the false classification, the deviation caused by scale and location error still exists, which we aim to solve in the future.

## Acknowledgements

This work was supported by NSFC (No.62206173), Shanghai Sailing Program (No.22YF1428700), and Shanghai Frontiers Science Center of Human-centered Artificial Intelligence (ShangHAI).

## References

- Bousmalis, K.; Trigeorgis, G.; and etc. 2016. Domain separation networks. *NeurIPS*, 29.
- Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 11621–11631.
- Chen, J.; Lei, B.; Song, Q.; Ying, H.; Chen, D. Z.; and Wu, J. 2020. A hierarchical graph network for 3d object detection on point clouds. In *CVPR*, 392–401.
- Chen, Y.; Li, W.; Sakaridis, C.; Dai, D.; and Van Gool, L. 2018. Domain adaptive faster r-cnn for object detection in the wild. In *CVPR*, 3339–3348.
- Cong, P.; Zhu, X.; Qiao, F.; Ren, Y.; Peng, X.; Hou, Y.; Xu, L.; Yang, R.; Manocha, D.; and Ma, Y. 2022. STCrowd: A Multimodal Dataset for Pedestrian Perception in Crowded Scenes. In *CVPR*, 19608–19617.
- Contributors, M. 2020. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. <https://github.com/open-mmlab/mmdetection3d>. Accessed: 2020-12-30.
- Cui, S.; Wang, S.; Zhuo, J.; Su, C.; Huang, Q.; and Tian, Q. 2020. Gradually vanishing bridge for adversarial domain adaptation. In *CVPR*, 12455–12464.
- Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1): 2096–2030.
- Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 3354–3361. IEEE.
- Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *NeurIPS*, 27.
- He, C.; Zeng, H.; Huang, J.; Hua, X.-S.; and Zhang, L. 2020. Structure aware single-stage 3d object detection from point cloud. In *CVPR*, 11873–11882.
- He, Q.; Wang, Z.; Zeng, H.; Zeng, Y.; and Liu, Y. 2022. Svga-net: Sparse voxel-graph attention network for 3d object detection from point clouds. In *AAAI*, volume 36, 870–878.
- He, R.; Yang, J.; and Qi, X. 2021. Re-distributing biased pseudo labels for semi-supervised semantic segmentation: A baseline investigation. In *ICCV*, 6930–6940.
- Hegde, D.; and Patel, V. 2021. Attentive Prototypes for Source-free Unsupervised Domain Adaptive 3D Object Detection. *arXiv preprint arXiv:2111.15656*.
- Hoffman, J.; Tzeng, E.; Park, T.; Zhu, J.-Y.; Isola, P.; Saenko, K.; Efros, A.; and Darrell, T. 2018. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 1989–1998. PMLR.
- Hoffman, J.; Wang, D.; Yu, F.; and Darrell, T. 2016. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*.
- Hou, Y.; Zhu, X.; Ma, Y.; Loy, C. C.; and Li, Y. 2022. Point-to-Voxel Knowledge Distillation for LiDAR Semantic Segmentation. In *CVPR*, 8479–8488.
- Hu, L.; Kan, M.; Shan, S.; and Chen, X. 2020. Unsupervised domain adaptation with hierarchical gradient synchronization. In *CVPR*, 4043–4052.
- Hu, Y.; Ding, Z.; Ge, R.; Shao, W.; Huang, L.; Li, K.; and Liu, Q. 2022. Afdetv2: Rethinking the necessity of the second stage for object detection from point clouds. In *AAAI*, volume 36, 969–979.
- Hurl, B.; Czarnecki, K.; and Waslander, S. 2019. Precise synthetic image and lidar (presil) dataset for autonomous vehicle perception. In *IV*, 2522–2529. IEEE.
- Jaritz, M.; Vu, T.-H.; Charette, R. d.; Wirbel, E.; and Pérez, P. 2020. xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation. In *CVPR*, 12605–12614.
- Jiang, H.; Wang, R.; Shan, S.; and Chen, X. 2018. Learning class prototypes via structure alignment for zero-shot recognition. In *ECCV*, 118–134.
- Khodabandeh, M.; Vahdat, A.; Ranjbar, M.; and Macready, W. G. 2019. A robust learning approach to domain adaptive object detection. In *CVPR*, 480–490.
- Li, G.; Ji, Z.; and Qu, X. 2022. Stepwise Domain Adaptation (SDA) for Object Detection in Autonomous Vehicles Using an Adaptive CenterNet. *T-ITS*.
- Lin, T.-Y.; Goyal, P.; and etc. 2017. Focal loss for dense object detection. In *ICCV*, 2980–2988.
- Long, M.; Cao, Y.; Wang, J.; and Jordan, M. 2015. Learning transferable features with deep adaptation networks. In *ICML*, 97–105. PMLR.
- Long, M.; Zhu, H.; Wang, J.; and Jordan, M. I. 2017. Deep transfer learning with joint adaptation networks. In *ICML*, 2208–2217. PMLR.
- Luo, Z.; Cai, Z.; Zhou, C.; Zhang, G.; Zhao, H.; Yi, S.; Lu, S.; Li, H.; Zhang, S.; and Liu, Z. 2021. Unsupervised domain adaptive 3d detection with multi-level consistency. In *ICCV*, 8866–8875.
- Mancini, M.; Porzi, L.; Bulò, S. R.; Caputo, B.; and Ricci, E. 2018. Boosting domain adaptation by discovering latent domains. In *CVPR*, 3771–3780.
- Maria Carlucci, F.; Porzi, L.; Caputo, B.; Ricci, E.; and Rota Bulò, S. 2017. Autodial: Automatic domain alignment layers. In *ICCV*, 5067–5075.
- Paszke, A.; Gross, S.; and Massa, e. 2019. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32.
- Qi, C. R.; Litany, O.; He, K.; and Guibas, L. J. 2019. Deep hough voting for 3d object detection in point clouds. In *ICCV*, 9277–9286.
- Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 652–660.
- Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

- Qin, C.; You, H.; Wang, L.; Kuo, C.-C. J.; and Fu, Y. 2019. Pointdan: A multi-scale 3d domain adaption network for point cloud representation. *NeurIPS*, 32.
- RoyChowdhury, A.; Chakrabarty, P.; Singh, A.; Jin, S.; Jiang, H.; Cao, L.; and Learned-Miller, E. 2019. Automatic adaptation of object detectors to new domains using self-training. In *CVPR*, 780–790.
- Saito, K.; Ushiku, Y.; Harada, T.; and Saenko, K. 2019. Strong-weak distribution alignment for adaptive object detection. In *CVPR*, 6956–6965.
- Saleh, K.; and Abobakr, e. 2019. Domain adaptation for vehicle detection from bird’s eye view LiDAR point cloud data. In *ICCV Workshops*, 0–0.
- Seibold, C. M.; Reiß, S.; Kleesiek, J.; and Stiefelhagen, R. 2022. Reference-guided pseudo-label generation for medical semantic segmentation. In *AAAI*, volume 36, 2171–2179.
- Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; and Li, H. 2020a. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *CVPR*, 10529–10538.
- Shi, S.; Wang, Z.; Shi, J.; Wang, X.; and Li, H. 2020b. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *TPAMI*, 43(8): 2647–2664.
- Shi, W.; and Rajkumar, R. 2020. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *CVPR*, 1711–1719.
- Su, P.; Wang, K.; Zeng, X.; Tang, S.; Chen, D.; Qiu, D.; and Wang, X. 2020. Adapting object detectors with conditional domain normalization. In *ECCV*, 403–419. Springer.
- Sun, B.; and etc. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 443–450. Springer.
- Sun, P.; Kretschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2446–2454.
- Wang, Y.; Chen, X.; You, Y.; Li, L. E.; Hariharan, B.; Campbell, M.; Weinberger, K. Q.; and Chao, W.-L. 2020. Train in germany, test in the usa: Making 3d object detectors generalize. In *CVPR*, 11713–11723.
- Wu, B.; Wan, A.; Yue, X.; and Keutzer, K. 2018. Squeeze-seg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *ICRA*, 1887–1893. IEEE.
- Xiao, A.; Huang, J.; Guan, D.; Zhan, F.; and Lu, S. 2022. Transfer learning from synthetic to real LiDAR point cloud for semantic segmentation. In *AAAI*, volume 36, 2795–2803.
- Xiao, P.; Shao, Z.; Hao, S.; Zhang, Z.; Chai, X.; Jiao, J.; Li, Z.; Wu, J.; Sun, K.; Jiang, K.; et al. 2021. PandaSet: Advanced Sensor Suite Dataset for Autonomous Driving. In *ITSC*, 3095–3101. IEEE.
- Xu, Q.; Zhou, Y.; Wang, W.; Qi, C. R.; and Anguelov, D. 2021. Spg: Unsupervised domain adaptation for 3d object detection via semantic point generation. In *ICCV*, 15446–15456.
- Xu, R.; Liu, P.; Wang, L.; Chen, C.; and Wang, J. 2020. Reliable weighted optimal transport for unsupervised domain adaptation. In *CVPR*, 4394–4403.
- Yan, Y.; Mao, Y.; and Li, B. 2018. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10): 3337.
- Yang, H.-M.; Zhang, X.-Y.; Yin, F.; and Liu, C.-L. 2018. Robust classification with convolutional prototype learning. In *CVPR*, 3474–3482.
- Yang, J.; Shi, S.; Wang, Z.; Li, H.; and Qi, X. 2021. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In *CVPR*, 10368–10378.
- Yang, J.; Zou, H.; Zhou, Y.; Zeng, Z.; and Xie, L. 2020a. Mind the discriminability: Asymmetric adversarial domain adaptation. In *ECCV*, 589–606. Springer.
- Yang, Z.; Sun, Y.; Liu, S.; and Jia, J. 2020b. 3dssd: Point-based 3d single stage object detector. In *CVPR*, 11040–11048.
- Yao, H.; Hu, X.; and Li, X. 2022. Enhancing Pseudo Label Quality for Semi-Supervised Domain-Generalized Medical Image Segmentation. *arXiv preprint arXiv:2201.08657*.
- Yi, L.; Gong, B.; and Funkhouser, T. 2021. Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds. In *CVPR*, 15363–15373.
- Yihan, Z.; Wang, C.; Wang, Y.; Xu, H.; Ye, C.; Yang, Z.; and Ma, C. 2021. Learning transferable features for point cloud detection via 3d contrastive co-training. *NeurIPS*, 34: 21493–21504.
- Yin, T.; Zhou, X.; and Krahenbuhl, P. 2021. Center-based 3d object detection and tracking. In *CVPR*, 11784–11793.
- You, Y.; Diaz-Ruiz, C. A.; Wang, Y.; Chao, W.-L.; Hariharan, B.; Campbell, M.; and Weinbergert, K. Q. 2022. Exploiting Playbacks in Unsupervised Domain Adaptation for 3D Object Detection in Self-Driving Cars. In *ICRA*, 5070–5077. IEEE.
- Yu, F.; Wang, D.; Chen, Y.; Karianakis, N.; Shen, T.; Yu, P.; Lymberopoulos, D.; Lu, S.; Shi, W.; and Chen, X. 2022. SC-UDA: Style and Content Gaps aware Unsupervised Domain Adaptation for Object Detection. In *WACV*, 382–391.
- Zhang, W.; Li, W.; and Xu, D. 2021. SRDAN: Scale-aware and range-aware domain adaptation network for cross-dataset 3D object detection. In *CVPR*, 6769–6779.
- Zhang, Y.; Hu, Q.; and Xu, e. 2022. Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In *CVPR*, 18953–18962.
- Zhou, X.; Wang, D.; and Krähenbühl, P. 2019. Objects as points. *arXiv preprint arXiv:1904.07850*.
- Zhu, X.; Ma, Y.; Wang, T.; Xu, Y.; Shi, J.; and Lin, D. 2020. Ssn: Shape signature networks for multi-class object detection from point clouds. In *ECCV*, 581–597. Springer.
- Zhu, X.; Zhou, H.; Wang, T.; Hong, F.; Li, W.; Ma, Y.; Li, H.; Yang, R.; and Lin, D. 2021. Cylindrical and asymmetrical 3d convolution networks for lidar-based perception. *TPAMI*.
- Zhu, X.; Zhou, H.; Yang, C.; Shi, J.; and Lin, D. 2018. Penalizing top performers: Conservative loss for semantic segmentation adaptation. In *ECCV*, 568–583.