SSDA3D: Semi-supervised Domain Adaptation for 3D Object Detection from Point Cloud

Yan Wang[1*], **Junbo Yin**[1*], **Wei Li**[2], **Pascal Frossard**[3], **Ruigang Yang**[2], **Jianbing Shen**[4†]

[1]Beijing Institute of Technology
[2]Inceptio
[3]École Polytechnique Fédérale de Lausanne (EPFL)
[4]SKL-IOTSC, CIS, University of Macau
yanwang@bit.edu.cn, yinjunbocn@gmail.com

Abstract

LiDAR-based 3D object detection is an indispensable task in advanced autonomous driving systems. Though impressive detection results have been achieved by superior 3D detectors, they suffer from significant performance degeneration when facing unseen domains, such as different LiDAR configurations, different cities, and weather conditions. The mainstream approaches tend to solve these challenges by leveraging unsupervised domain adaptation (UDA) techniques. However, these UDA solutions just yield unsatisfactory 3D detection results when there is a severe domain shift, *e.g.*, from Waymo (64-beam) to nuScenes (32-beam). To address this, we present a novel **S**emi-**S**upervised **D**omain **A**daptation method for **3D** object detection (*SSDA3D*), where only a few labeled target data is available, yet can significantly improve the adaptation performance. In particular, our *SSDA3D* includes an Inter-domain Adaptation stage and an Intra-domain Generalization stage. In the first stage, an Inter-domain Point-CutMix module is presented to efficiently align the point cloud distribution across domains. The Point-CutMix generates mixed samples of an intermediate domain, thus encouraging to learn domain-invariant knowledge. Then, in the second stage, we further enhance the model for better generalization on the unlabeled target set. This is achieved by exploring Intra-domain Point-MixUp in semi-supervised learning, which essentially regularizes the pseudo label distribution. Experiments from Waymo to nuScenes show that, with only 10% labeled target data, our *SSDA3D* can surpass the fully-supervised oracle model with 100% target label. Our code is available at https://github.com/yinjunbo/SSDA3D.

1 Introduction

3D object detection from LiDAR point cloud has been a central task in applications such as robotics and autonomous driving. It aims to localize and classify the obstacles in the 3D space accurately and timely, where the obstacles are often formulated as orientated 3D bounding boxes. In the past few years, substantial progress has been made in 3D object detection thanks to the well-developed 3D neural networks

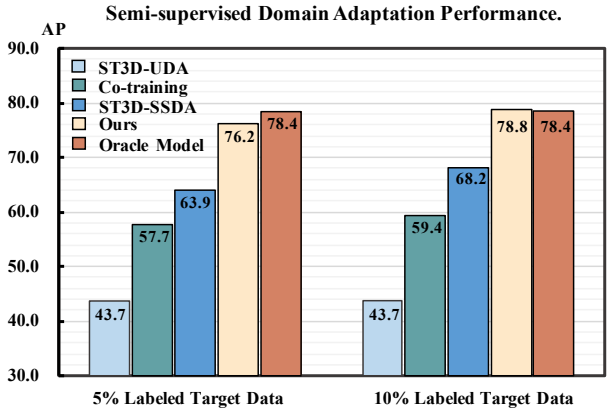


Figure 1: An example of semi-supervised domain adaptation on Waymo→nuScenes (5% and 10% label) based on the CenterPoint (Yin, Zhou, and Krahenbuhl 2021) detector. Our *SSDA3D* significantly surpasses advanced methods such as Co-training (*i.e.*, jointly train the labeled source and target data) and ST3D-SSDA (*i.e.*, adapt ST3D (Yang et al. 2021) with the target label). Moreover, with only 10% target label, we also exceed the *Oracle* model that is trained with 100% target label.

such as VoxelNet (Zhou and Tuzel 2018) and PointNet (Qi et al. 2017). The advanced 3D object detectors (Yan, Mao, and Li 2018; Yin, Zhou, and Krahenbuhl 2021; Shi et al. 2020; Xu et al. 2021b) can gain promising detection results when given massive well-labeled training samples. Unfortunately, these 3D detectors may incur dramatic performance drops when they are deployed in unseen environments. The domain shift (Sun, Feng, and Saenko 2016; Tzeng et al. 2017) indicates that there is an obvious distribution discrepancy between the training data (source domain) and test data (target domain). One solution to overcome this challenge is to further manually annotate the target domain data.

A recent trend is to exploit unsupervised domain adaptation (UDA) algorithms to mitigate the domain gap in 3D object detection, where SRDAN (Zhang, Li, and Xu 2021), SN (Wang et al. 2020) and ST3D (Yang et al. 2021) are

[*]Equal Contribution. Work done during internship at Inceptio.
[†]Corresponding author. Email: jianbingshen@um.edu.mo

the prior works. Though impressive domain adaptation ability has been demonstrated between domains with similar LiDAR configuration, *e.g.*, from Waymo (Sun et al. 2020) to KITTI (Geiger, Lenz, and Urtasun 2012), where both are based on 64-beam Velodyne LiDAR, none of them is capable of tackling the domain gap derived from various LiDAR configurations. For example, on the Waymo (64-beam)→nuScenes (32-beam) (Caesar et al. 2019) adaptation setting, ST3D only gains 2 to 3 points improvement over the *source only* model (*i.e.*, trained with only the Waymo data), even with the assistance of target statistics information. Nevertheless, it is often the case that a self-driving car will upgrade its LiDAR sensor to a new type, and it seems that the well-labeled source data from the old LiDAR sensor is not able to contribute much with current UDA algorithms, which leads to a huge waste.

The above observations motivate us to devise a more realistic setting, *i.e.*, semi-supervised domain adaptation (SSDA). Compared to UDA, SSDA smartly uses only a small set of labeled target data, together with the large-scale labeled source data. It can greatly close the gap to the fully-supervised *Oracle* model, meanwhile maintaining a rather lower annotation cost that is readily acceptable in the practical application. An example of SSDA is shown in Figure 1, where we aim to adapt Waymo to nuScenes and only a few labeled nuScenes data is available. An intuitive way for addressing this is to jointly train the labeled source and target data, which is called Co-training. This obtains suboptimal performance due to the inappropriate domain mixing, where the large-scale source domain dominates the learning process. An alternative solution is to adapt UDA techniques to SSDA. We enhance ST3D with the labeled part of nuScenes and name it ST3D-SSDA. Despite the better performance than Co-training, it is still far behind the fully-supervised *Oracle* that is trained with the full target label.

In this work, we propose a new framework to tackle SSDA in 3D object detection, named *SSDA3D*. The core idea of *SSDA3D* is to reduce the Inter-domain discrepancy (*e.g.*, labeled source→ labeled target) as well as enhance the intra-domain generalization (*e.g.*, labeled target→unlabeled target). This is realized by solving domain adaptation task and semi-supervised learning (SSL) task in a unified framework by two-stage learning, which includes an **Inter-domain Adaptation Stage** and an **Intra-domain Generalization Stage**. Concretely, in the first stage, we aim to generate mixed samples from the labeled source and target data, which acts as intermediate domain data that can mitigate the domain bias and learn domain-invariant representation. This is achieved by an Inter-domain Point-CutMix module, *e.g.*, we randomly remove a local point cloud region in the source sample, and then replace it with another region from the target sample to close the distribution discrepancy. Regions from different point cloud range present various patterns, *e.g.*, the points become more sparse in distant regions. Thus, a constraint is enforced that the inpainting regions should be from the same range to preserve the geometrical nature. The model in the first stage has significantly boosted the performance thanks to the efficient domain mixing. In the second stage, we aim to further strengthen the intra-domain generalization on the unlabeled target data. In particular, the pseudo-labeling technique is typically used in SSL for learning from the unlabeled data, while we find that inaccurately pseudo-labeled samples essentially account for a higher percentage, which will dominate the learning and undermine the performance. To this end, an Intra-domain Point-MixUp module is introduced by globally mixing the labeled scenes and the pseudo-labeled scenes. In this way, the mixed data and labels can take the role of regularizer to implicitly improve the learning process.

To the best of our knowledge, *SSDA3D* is the first effort for semi-supervised domain adaptation in the context of 3D object detection. This is achieved by a novel framework that jointly addresses inter-domain adaptation and intra-domain generalization. The proposed Inter-domain Point-CutMix module largely reduces the domain discrepancy, while the Intra-domain Point-MixUp module essentially regularizes the pseudo label distribution of unlabeled target data. Our model is evaluated in a challenging domain adaptation setting, *e.g.*, Waymo→nuScenes. As shown in Figure 1, with only 10% labeled target data, we achieve competitive performance to the fully-supervised *Oracle* model. It turns out that our method can save almost 90% annotation cost for cross-domain 3D object detection.

2 Related Works

3D Object Detection from LiDAR Point Cloud. LiDAR-based 3D object detection has received great attention due to the rapid development of autonomous driving. The mainstream 3D object detection approaches can be categorized into three groups, *e.g.*, point-based methods (Qi et al. 2017; Shi, Wang, and Li 2019; Yang et al. 2020; Zhang et al. 2022; Zhou et al. 2020; Chen et al. 2022a), voxel-based methods (Lang et al. 2019; Yan, Mao, and Li 2018; Yin, Zhou, and Krahenbuhl 2021; Zheng et al. 2021) and hybrid methods with joint point-voxel representation (Shi et al. 2020; Mao et al. 2021; Deng et al. 2021). Point or hybrid approaches are often restricted to the inference speed when the point scale becomes large. Voxel-based approaches are more prevalent in practical applications, since they can engage trade-offs between accuracy and speed. They usually discretize the 3D space into regular voxels, and then apply sparse convolutional networks (Graham, Engelcke, and Van Der Maaten 2018; Chen et al. 2022c) for abstracting 3D features. Apart from the single-frame 3D object detection paradigm, there are also some attempts to leverage the temporal point cloud information (Yin et al. 2021; Chen et al. 2022b). However, all these 3D detectors work well when given large-scale precisely annotated point cloud samples, but are incapable of handling unseen domains without sufficient annotations. In this paper, we investigate the semi-supervised domain adaptation in 3D object detection, which effectively improves the domain generalization of leading 3D detectors.

Semi-supervised Learning in 3D Object Detection. To achieve promising detection performance, prevalent 3D object detectors resort to large-scale 3D annotations. However, collecting such annotations is extremely time-consuming

and expensive. This motivates researchers to apply semi-supervised learning(SSL) methods (Tang and Lee 2019; Yin et al. 2022b) to cut the expense. In order to boost the performance with limited annotations, SESS (Zhao, Chua, and Lee 2020) proposes a self-ensembling framework to enhance the generalization ability on unlabeled data. After that, 3DIoUMatch (Wang et al. 2021) proposes an IoU-based filtering mechanism to improve the quality of the pseudo labels. Later, ProficientTeachers (Yin et al. 2022a) leverages a spatial-temporal ensemble module and a clustering-based box voting module to further refine the pseudo labels. There are also some works exploring weakly semi-supervised 3D object detection to save annotation cost. Meng *et al*. (Meng et al. 2021) propose to learn from a few weakly annotated point cloud samples as well as some precisely annotated instances. It achieves 97% performance to the oracle model by a two-stage network architecture. These methods mainly focus on intra-domain 3D SSL, while ours aims to address cross-domain 3D SSL which needs to transfer knowledge from a well-annotated source domain to a few-annotated target one and faces more challenges.

Domain Adaptation in Point Cloud. Recent years have witnessed an increased interest in domain adaptation in 3D point cloud (Achituve, Maron, and Chechik 2021; Xiao et al. 2022; Peng et al. 2020; Jiang and Saripalli 2021), which mainly focuses on unsupervised domain adaptation (UDA). For UDA in 3D semantic segmentation (Xu et al. 2021a), SqueezeSegV2 (Wu et al. 2019) proposes several techniques to adapt simulated data to the real world, such as intensity rendering, geodesic alignment and layer-wise calibration between domains. ePointDA (Zhao et al. 2021) also handles the simulation-to-real setting. It employs CycleGAN for dropout noise rendering and performs statistics-invariant feature alignment to close the gap. To address domain discrepancies caused by different sensors, Langer *et al*. (Langer et al. 2020) generate semi-synthetic LiDAR scans to simulate the target LiDAR sensor, and also align outputs between different domains via the geodesic loss. Later, SVCN (Yi, Gong, and Funkhouser 2021) designs a point cloud completion network to recover both the source and target points to a canonical domain (*e.g*., the complete point cloud), and then performs segmentation on this canonical domain. Several studies have also explored UDA in 3D object detection (Saleh et al. 2019). SRDAN (Zhang, Li, and Xu 2021) performs domain alignment by applying adversarial learning on different feature levels. SN (Wang et al. 2020) leverages the target statistics such as 3D object sizes to adjust the source domain data. ST3D (Yang et al. 2021) further adapts self-training on 3D UDA by assigning pseudo labels on the target domain. The experiments indicate that it is extremely challenging for these UDA approaches to catch the oracle performance, especially when the domain gap is large (e.g., cross LiDAR configurations). By contrast, our *SSDA3D* gains much better performance, meanwhile saving 90% annotation cost.

Mixing Augmentation Strategies. Data augmentation (Cubuk et al. 2020, 2019), which has been proved an indispensable component in training deep neural networks, aims to generate new training samples lie in the vicinity distribution of the original samples. Different from the traditional augmentation strategies such as random crop, flip, rotation and scaling, MixUp (Zhang et al. 2018) constructs new training samples by linearly combing two samples drawn from the training set. CutMix (Yun et al. 2019) also aims to combine two different training samples. Instead of directly mixing the two samples globally, it replaces the local image patch of one sample with another patch from a different sample. In essence, both MixUp and CutMix are designed for the image classification task. In this work, we adapt these techniques to 3D object detection in point cloud, which is an early effort that exploits mixing augmentation in 3D object detection. The analysis in Sec. 3.1 reveals their capacity in addressing inter-domain discrepancy and intra-domain generalization.

3 Our Semi-supervised Domain Adaptation Learning Approach

Previous approaches tend to tackle the point cloud domain adaptation by UDA, while they are incapable of handling large domain discrepancies. A prior work, ST3D (Yang et al. 2021), reports an improvement of 3.0 AP with the help of nuScenes object statistics when adapting from Waymo, which is far from satisfactory for practical deployment in self-driving vehicles. Thus, a better way is to explore SSDA for addressing this challenge.

SSDA has limited access to a small set of labeled target samples, thus it is of great importance to fully leverage these training samples. In this paper, we develop a new SSDA framework, *SSDA3D*, for 3D object detection. In Sec. 3.1, we detail the overall learning pipeline of our *SSDA3D*. Then, in Sec. 3.2, a Point-CutMix module is advocated to address domain adaptation problem between source and target. Afterwards, a Point-MixUp is presented in Sec. 3.3 to further improve the learning on the unlabeled target data.

3.1 Two-stage Learning for SSDA

For SSDA in 3D Object detection, we take as input three types of point cloud data from two different domains, *e.g*., the labeled source data $D_{\mathrm{S}} = \left\{(x_{\mathrm{S}}^i, y_{\mathrm{S}}^i)\right\}_{\mathrm{i=1}}^{N_{\mathrm{S}}}$, the labeled target data $D_{\mathrm{TL}} = \left\{(x_{\mathrm{TL}}^i, y_{\mathrm{TL}}^i)\right\}_{\mathrm{i=1}}^{N_{\mathrm{TL}}}$ and the unlabeled target data $D_{\mathrm{TU}} = \left\{x_{\mathrm{TU}}^i\right\}_{i=1}^{N_{\mathrm{TU}}}$, where N_{S}, N_{TL} and N_{TU} are the number of data samples. A crucial characteristic of SSDA is to leverage the well-labeled source data to save annotations of the target, thus both N_{S} and N_{TU} are much bigger than N_{TL}, *i.e*., $N_{\mathrm{S}} \gg N_{\mathrm{TL}}$ and $N_{\mathrm{TU}} \gg N_{\mathrm{TL}}$. Besides, x^i and y^i represent the *i*-th point cloud sample and corresponding 3D detection label, respectively.

Since the SSDA task requires to resolve both cross-domain adaptation and intra-domain generalization, we thus advocate a two-stage learning strategy. As illustrated in Figure 2, the first stage is the Inter-domain Adaptation stage, which is used to tackle the domain discrepancy between the source D_{S} and labeled target D_{TL}. To achieve this, mixed point cloud samples $D_{\mathrm{STL}} = \left\{(\tilde{x}_{\mathrm{STL}}^i, \tilde{y}_{\mathrm{STL}}^i)\right\}_{i=1}^{N_{\mathrm{STL}}}$ are generated from D_{S} and D_{TL}, based on the proposed Inter-domain

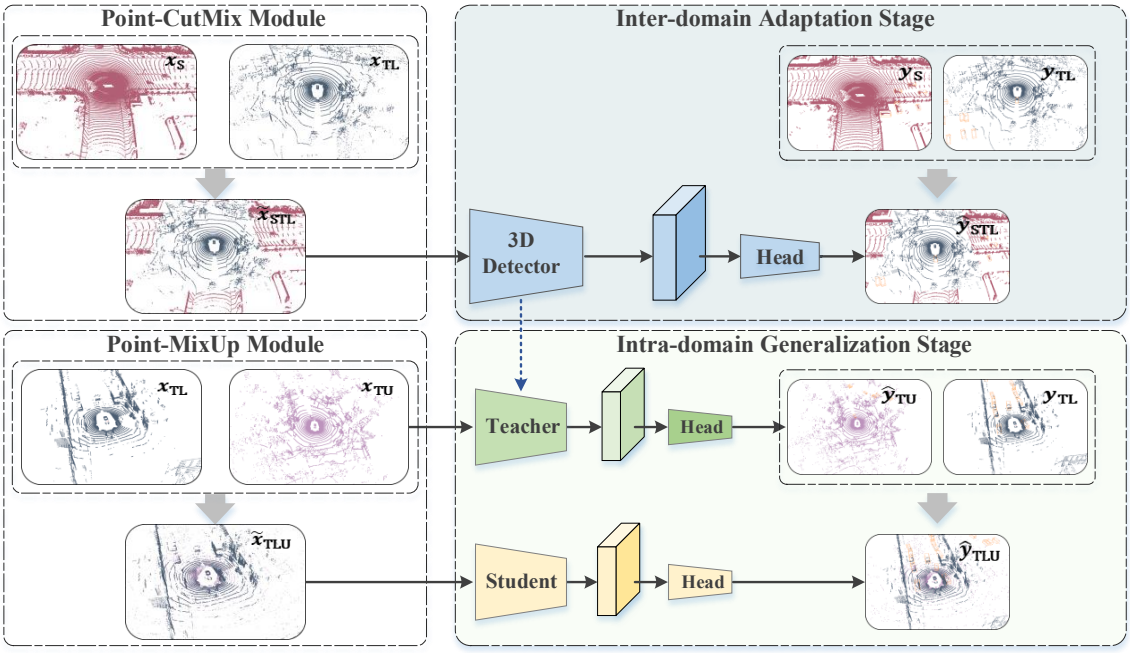


Figure 2: The overall framework of our *SSDA3D*. It comprises an Inter-domain Adaptation stage and an Intra-domain Generalization stage. The Point-CutMix in the former stage generates region-wise mixed samples from source and target to address the domain discrepancy. The Point-MixUp in the latter stage further constructs scene-wise mixed samples for improving the learning on the unlabeled set.

Point-CutMix module. The mixed samples D_{STL} serve as the intermediate domain samples to help the model learn domain-invariant features and thus close the inter-domain gap. By training on D_{STL}, the knowledge has been transferred from D_{S} to D_{TL}, and the obtained model achieves impressive results on the target domain. Therefore, in the second stage, we utilize this model as the teacher detector to learn towards Intra-domain Generalization. Specifically, we use the teacher model to generate pseudo labels $\hat{y}$ for unlabeled target data, which is denoted as $D_{\text{TU}} = \left\{(x^i_{\text{TU}}, \hat{y}^i_{\text{TU}})\right\}_{i=1}^{N_{\text{TU}}}$. Then, the pseudo labeled point cloud D_{TU}, together with the real-labeled point cloud D_{TL}, are globally mixed by our Intra-Domain Point-MixUp module to obtain $D_{\text{TLU}} = \left\{(\tilde{x}^i_{\text{TLU}}, \hat{y}^i_{\text{TLU}})\right\}_{i=1}^{N_{\text{TLU}}}$. The mixed samples D_{TLU} are then used to train the student detector. In particular, D_{TLU} tactfully regularizes the pseudo label distribution and thus obtain better performance.

3.2 Inter-domain Point-CutMix

To resolve the inter-domain discrepancy, we propose to mix the LiDAR points from different domains to transfer the data distribution from their original domains to an intermediate domain. This can make the model learn domain-invariant features and integrate the knowledge from both domains. To this end, we present the Point-CutMix module, which is inspired by CutMix (Yun et al. 2019). Basically, there are three design choices when performing point cloud mixing, *e.g.*, the object level, the region level or the scene level. In this work, we prefer the region level domain mixing. The intuition is that, the global scene-wise features from different domains may be quite different and focusing on the object-level features is yet too strict, making both too difficult to learn. The local region features, however, usually containing similar contexts like the combinations of cars, road planes or buildings, are much easier to learn. Thus it would be more favorable to learn from the locally mixed point cloud samples.

Specifically, given a point cloud x_{TL} from the target domain, we first randomly choose a point in x_{TL} as the region center c_{T}. Then, we randomly select a rectangle region around c_{T}, based on the bird's eye view (x-y plane). After that, we remove all the points outside the selected region and only keep the points inside the region, which is referred to as P_{T}. P_{T} is then used to mix the source point cloud x_{S}. In particular, we sample another center c_{S} in x_{S}, where c_{S} is constrained to have the same range to c_{T}. Then, we remove the points around c_{S} with the same region as that in the target, and use P_{T} to inpaint this region. This leads to a new point cloud $\tilde{x}_{STL}$. Similarly, the new label $\tilde{y}_{STL}$ is also obtained by combining the 3D boxes in P_{T} and the remained 3D boxes in x_{S}. Formally, this process can be denoted as:

$$\tilde{x}_{\text{STL}} = Concat(M_x \odot x^i_{\text{TL}}, (1 - M_x) \odot x^j_{\text{S}}) \quad (1)$$

$$\tilde{y}_{\text{STL}} = Concat(M_y \odot y^i_{\text{TL}}, (1 - M_y) \odot y^j_{\text{S}}) \quad (2)$$

where M_x, M_y denote two binary masks indicating whether this point or label (*i.e.*, 3D box) is inside the selected point cloud region, $\odot$ is the element-wise multiplication operation, and $Concat(\cdot, \cdot)$ denotes the concatenation operation.

In this way, we can update the original point cloud distribution to the mixed distribution. The newly generated point clouds contain both source and target local region information, and thus both domain knowledge is reserved. Furthermore, by enforcing the model to learn on the mixed point cloud at each iteration step, we also ensure the data balance of different domains. The Point-CutMix module is applied with a fixed probability during training. After the learning of the Inter-domain Adaptation stage, we can get a domain adaptive model F_{Tea}, which will be exploited as a teacher model in the next stage.

3.3 Intra-Domain Point-MixUp

The main purpose of the second stage is to strengthen the intra-domain generalization capacity to improve the learning on the unlabeled data, which is also known as semi-supervised learning (SSL). Self-training, also called Psuedo-labeling, is a commonly used technique for SSL. Since we have obtained an excellent detector from the first stage, we can utilize it to produce pseudo labels on the unlabeled point cloud. However, there is a potential problem if directly using the pseudo labels, *e.g.*, the existence of inaccurate predictions will inevitably make the distribution of the pseudo labels inconsistent with the real labels. To this end, we propose the Intra-Domain Point-MixUp in this stage to overcome the inconsistent label distribution. Our Point-MixUp is motivated by MixUp (Zhang et al. 2018), aiming to globally mix the point cloud scenes from the labeled and unlabeled sets. Though scene-wise global mixing is not suitable for cross-domain learning, we find that it is extremely beneficial to intra-domain learning, since there is little domain gap between the data from the same domain. By applying the Point-MixUp, the pseudo label can be regularized by the real label, thus improving the label distribution. Moreover, since the unlabeled scale N_{TU} is much larger than the labeled scale N_{TL}, the Point-MixUp also enforces a balanced learning between D_{TL} and D_{TU}.

More specifically, given an unlabeled point cloud, we first exploit the teacher model F_{Tea} to generate the pseudo labels, *i.e.*, $\hat{y}_{TU} = F_{\text{Tea}}(x_{TU})$, where $\hat{y}_{TU}$ denotes the pseudo labels. Here, we follow SSL methods such as FixMatch (Sohn et al. 2020) to utilize a score threshold to filter the pseudo boxes with low confidence. Then, we apply the Point-MixUp on both the data and label levels. Concretely, given a real-labeled point cloud (x^i_{TL}, y^i_{TL}) and a pseudo-labeled point cloud $(x^i_{TU}, \hat{y}^i_{TU})$, we randomly reserve partial points of x^i_{TL} and x^i_{TU}, respectively. This can be implemented by first shuffling the points, and then selecting the top point indices, which are denoted as:

$$\tilde{x}_{\text{TLU}} = Concat(P \odot x^i_{\text{TL}}, Q \odot x^j_{\text{TU}}) \tag{3}$$

$$\hat{y}_{\text{TLU}} = Concat(y^i_{\text{TL}}, \hat{y}^j_{\text{TU}}) \tag{4}$$

where P and Q are two binary masks used for selecting points and are subjected to $\frac{|P=1|}{|P|} + \frac{|Q=1|}{|Q|} = 1$, which ensures the density of the mixed point cloud remains similar as the original ones.

Also, before merging the 3D boxes, collision detection will be performed. If there is a collision between the boxes from real and pseudo labels, the pseudo boxes and the near points will be replaced by the real boxes and corresponding points. Afterwards, the mixed samples $D_{TLU} = \left\{(\tilde{x}^i_{TLU}, \hat{y}^i_{TLU})\right\}_{i=1}^{N_{TLU}}$ will be used to train the student detector F_{Stu} with a fixed probability, where F_{Stu} is initialized by the model in the first stage. We find that the updated label distribution of $\hat{y}_{\text{TLU}}$ essentially improves the distribution of original pseudo labels $\hat{y}_{\text{TU}}$ by involving the real labels. By learning on the mixed samples, the model is also encouraged to be more robust on corrupt point cloud samples.

4 Experiments

4.1 Experiment Setup

Datasets. Our experiments are conducted on two widely used datasets: Waymo(Sun et al. 2020) with 64-beam LiDAR and nuScenes(Caesar et al. 2019) with 32-beam LiDAR. We adapt from Waymo to nuScenes, *i.e.*, 100% Waymo annotations together with partial nuScenes annotations are used. In particular, we uniformly downsample the nuScenes training samples into 1%, 5%, 10% and 100% (resulting in 282, 1407, 2813 and 28130 frames), and the rest of the samples remain unlabeled.

Comparison Methods. We compare *SSDA3D* with six methods: (***i***) **Source Only** indicates that we directly evaluate the model on target after training on source. (***ii***) **ST3D-UDA** indicates the ST3D under UDA setting where no real target labels are available. (***iii***) **Labeled Targets** indicates that only partial target labels (*i.e.*, 1%, 5%, 10% or 100%) are used to train the model with fully-supervised learning. (***iv***) **Co-training** represents that we train labeled source and target data jointly. (***v***) **ST3D-SSDA** is the extension of **ST3D-UDA** with extra supervision on limited real labeled target data. (***vi***) **Oracle** indicates the fully supervised model trained on target domain.

Evaluation Metric. Following nuScenes, we also select commonly used average precision (AP) and the specific metric nuScenes detection score (NDS) in nuScenes as our evaluation metric on the car category, which is also named Vehicle in Waymo. We follow the official nuScenes protocol to average over matching thresholds of $\mathbb{D}$=$\{0.5, 1, 2, 4\}$ meters on the car category. What's more, following ST3D, we also report closed gap which is defined as $\frac{AP_{\text{model}} - AP_{\text{source only}}}{AP_{\text{oracle}} - AP_{\text{source only}}} \times 100\%$ to show how much the performance gap is closed.

Implementation Details. All the methods are implemented based on an advanced 3D detector, CenterPoint(Yin, Zhou, and Krahenbuhl 2021). The learning schedule follows the popular codebase OpenPCDet(Team 2020), where the training epochs are set to 20 for both stages. For both Inter-domain Point-CutMix and Intra-Domain Point-MixUp, there is a probability to decide whether to utilize the corresponding technique.

The labeled target samples are downsampled in a frame-level. For example, 1% labeled target data means that only

Methods	Semi-supervised Domain Adaptation 3D Detection Performance with Different Target Label Amounts							
	1%		5%		10%		100%	
	AP/NDS	Closed Gap	AP/NDS	Closed Gap	AP/NDS	Closed Gap	AP/NDS	Closed Gap
Source Only	42.6/50.3	+0/+0	42.6/50.3	+0/+0	42.6/50.3	+0/+0	42.6/50.3	+0/+0
ST3D-UDA	43.7/50.2	+3.07/-0.51	43.7/50.2	+3.07/-0.51	43.7/50.2	+3.07/-0.51	43.7/50.2	+3.07 / -0.51
Labeled Target	37.2/38.1	-15.1/-62.2	61.0/53.2	+51.4/+14.8	65.6/58.2	+64.2/+40.3	78.4/69.9	+100/+100
Co-training	51.4/54.6	+24.6/+21.9	57.7/58.0	+42.2/+39.3	59.4/58.9	+46.9/+43.9	66.5/63.5	+66.8/+67.3
ST3D-SSDA	55.2/55.8	+35.2/+28.1	63.9/60.8	+59.5/+53.6	68.2/63.2	+71.5/+65.8	79.7/71.3	+103.6/+107.1
Ours	**73.4/67.1**	**+86.0/+85.7**	**76.2/68.8**	**+93.9/+94.4**	**78.8/70.9**	**+101.1/+105.1**	**79.8/71.8**	**+103.9/109.7**
Oracle	78.4/69.9	+100/+100	78.4/69.9	+100/+100	78.4/69.9	+100/+100	78.4/69.9	+100/+100

Table 1: Domain adaptation performance with different amounts of target labels. We report AP, NDS and their corresponding Closed Gap (%) on the car category for all methods. The best adaptation performance is indicated by bold.

	Components					AP / NDS
	Target Sup.	Source Sup.	Inter-domain CutMix	Naive SSL Learning	Intra-Domain MixUp	
(a)	✓					61.0 / 53.2
(b)	✓	✓				57.7 / 58.0
(c)	✓	✓	✓			74.5 / 67.8
(d)	✓	✓	✓	✓		73.8 / 67.7
(e)	✓	✓	✓	✓	✓	76.2 / 68.8
(f)	✓			✓	✓	65.4 / 58.2

Table 2: Ablation studies for each component of *SSDA3D*. These experimental results are reported based on 5% target label.

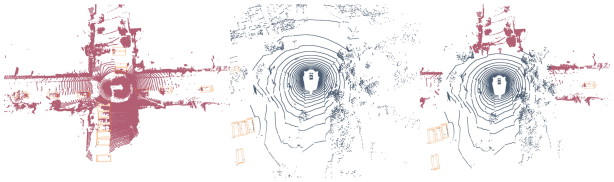
1% of the total samples are available and the GT-database for GT-sampling augmentation is also determined accordingly. The detection range is set to [-54.0, 54.0]m for X and Y axes, and [-5.0, 4.8]m for Z axis. We set the voxel size to [0.075, 0.075, 0.2]. As there is a difference in the range of intensity between Waymo and nuScenes, we normalize it to $0 \sim 1$ for both datasets. For augmentation techniques, we adopt widely used random world flip, random world rotation and random world scaling for both learning stages.

4.2 Performance Comparison

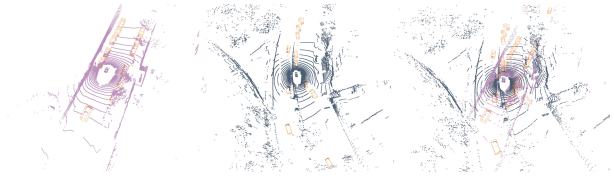
We perform domain adaptation experiments from Waymo to nuScenes with different amounts of nuScenes label . The 3D detection performance of different methods is shown in Table 1. As can be seen, our method achieves the best performance and surpasses all the other methods by a large margin with all settings. Specifically, with only 1% labeled target data, we improve the performance of Source Only and ST3D-UDA by around 30% AP and 17% NDS. We can still gain noticeable performance improvements of 18.2% AP and 11.3% NDS compared with ST3D-SSDA. Besides, with the increasing number of labeled target data, our method continuously produces better results. Compared with other methods, our method can greatly close the gap between *Oracle* and Source Only. With only 5% labeled target data, we achieve 94% Closed Gap. What's more, it is worth noting that with only 10% labeled target data, we even surpass *Oracle* in terms of AP and NDS (*i.e.*, 78.8% *vs* 78.4% AP and 70.9% *vs* 69.9% NDS). To sum up, our method can save about 90% annotation cost in cross-domain LiDAR-based 3D object detection.

4.3 Ablation Studies

Effectiveness of Each Component. In this section, we conduct ablation experiments to validate the effectiveness of each individual component of our proposed method, which is shown in Table 2. All the experiments are conducted with 5% labeled target data. From the comparison of (a) and (b), we can see that simply training the combined data of labeled source and target data will not yield performance



(a) Qualitative results of Inter-domain Point-CutMix.



(b) Qualitative results of Intra-domain Point-MixUp.

Figure 3: Qualitative results of different types of mixed samples.

gains. This proves that the giant domain gap between source and target could hinder the model in learning reasonable features and the large-scale source data also dominates the learning. By contrast, after applying Inter-domain Point-CutMix, an obvious improvement can be seen from the comparison between (b) and (c), *i.e.*, AP and NDS are improved by nearly 20% and 10%, respectively. It turns out that the Inter-domain Point-CutMix can help to learn domain-invariant features that are helpful in improving the 3D detection performance in the target domain.
Afterwards, we investigate the effectiveness by involving the unlabeled target data. In (d), we first utilize a naive semi-supervised learning method that simply trains the model using both the real- and pseudo-labeled target samples. However, this obtains nearly no gains compared with (c). While after we apply the Intra-Domain Point-MixUp, the performance improves from 73.8% to 76.2% in terms of AP and from 67.7% to 68.8% in terms of NDS. It clearly suggests the plausibility of the Intra-Domain Point-MixUp in regularizing the learning. From (e), we can further tell that learning from source domain contributes most to the final performance. Transferring knowledge from source domain to target domain largely improves the performance on the target domain.

Qualitative Results To better understand the mixed point cloud samples after the Inter-domain Point-CutMix and Intra-domain Point-MixUp modules. We give some examples of them in Figure 3. As can be seen in Figure 3 (a), the Inter-domain Point-CutMix module produces new scenes that resemble the nature scenes due to the constraining of the range of the mixed regions. Further, the Intra-domain Point-MixUp module yields sparser point clouds from the perspective of each involved sample, which also effectively regularizes the point cloud distribution.

Probability of Inter-domain Point-CutMix. In this section, we validate the effectiveness of the Inter-domain Point-CutMix module by adjusting its probability. The results are

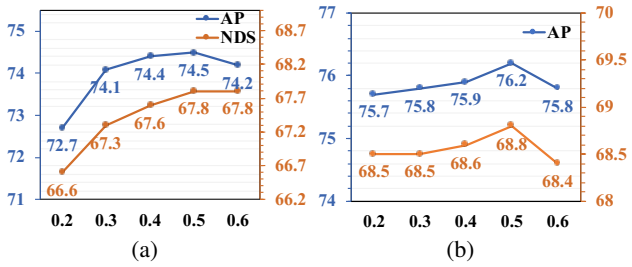


Figure 4: Ablation studies for mixing probability. (a) and (b) indicate the impact of Inter-domain Point-CutMix probability and Intra-domain Point-MixUp probability, respectively.

shown in Figure 4 (a). As seen, the probability of applying Inter-domain Point-CutMix has an obvious impact on the final performance. In general, the AP and NDS continue to get better as the probability increases, and the best performance is obtained when the probability is set to 0.5. It is interesting that even with a probability of 0.2, the performance is still better than the model without the Point-CutMix module, *i.e.*, the Co-training model (57.7% AP and 58.0% NDS). This further proves the necessity of our Inter-domain Point-CutMix module.

Probability of Intra-domain Point-MixUp. In this section, we conduct experiments to validate the impact of the probability to apply Intra-Domain Point-MixUp. As shown in Figure 4 (b), a similar trend to (a) can be seen. As the probability increases, the performance keeps improving and reaches the peak with 76.2% in AP and 68.8% in NDS at the probability of 0.5. Higher probability doesn't give better results. All the performances are better than the model without the Point-MixUp (73.8% AP and 67.7% NDS). This also indicates the robustness and generalization ability of our model.

5 Conclusion

In this paper, we presented a new task, semi-supervised domain adaptation (SSDA) in the context of 3D object detection, as well as a novel framework *SSDA3D*. We decoupled the learning of SSDA-based 3D object detection into two stages, which are the Inter-domain Adaptation stage and the Intra-domain Generalization stage. The purposes are to address the cross-domain discrepancy and improve semi-supervised learning, respectively. During the Inter-domain Adaptation stage, we proposed a Point-CutMix module to construct mixed point cloud samples to learn domain-invariant features. Then, in the Intra-domain Generalization stage, a Point-MixUp module was advocated to regularize the pseudo label distribution. Extensive experiments prove the effectiveness of *SSDA3D*, *e.g.*, on Waymo→nuScenes, and we significantly outperform previous methods. Moreover, with 10% labeled target data, *SSDA3D* achieves superior performance to the fully-supervised *Oracle* model.

Acknowledgements

This work was supported in part by the FDCT grant SKL-IOTSC(UM)-2021-2023, and the Start-up Research Grant (SRG) of University of Macau (SRG2022-00023-IOTSC). We would like to thank Inceptio for their infra supports.

References

Achituve, I.; Maron, H.; and Chechik, G. 2021. Self-supervised learning for domain adaptation on point clouds. In *WACV*, 123–133.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2019. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*, 11618–11628.

Chen, C.; Chen, Z.; Zhang, J.; and Tao, D. 2022a. Sasa: Semantics-augmented set abstraction for point-based 3d object detection. In *AAAI*, 221–229.

Chen, X.; Shi, S.; Zhu, B.; Cheung, K. C.; Xu, H.; and Li, H. 2022b. MPPNet: Multi-Frame Feature Intertwining with Proxy Points for 3D Temporal Object Detection. *ECCV*.

Chen, Y.; Li, Y.; Zhang, X.; Sun, J.; and Jia, J. 2022c. Focal Sparse Convolutional Networks for 3D Object Detection. In *CVPR*, 5418–5427.

Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, Q. V. 2019. Autoaugment: Learning augmentation strategies from data. In *CVPR*, 113–123.

Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. V. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshops*, 3008–3017.

Deng, J.; Shi, S.; Li, P.; Zhou, W.; Zhang, Y.; and Li, H. 2021. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *AAAI*.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 3354–3361.

Graham, B.; Engelcke, M.; and Van Der Maaten, L. 2018. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, 9224–9232.

Jiang, P.; and Saripalli, S. 2021. LiDARNet: A boundary-aware domain adaptation model for point cloud semantic segmentation. In *ICRA*, 2457–2464. IEEE.

Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. PointPillars: Fast encoders for object detection from point clouds. *CVPR*, 12697–12705.

Langer, F.; Milioto, A.; Haag, A.; Behley, J.; and Stachniss, C. 2020. Domain transfer for semantic segmentation of LiDAR data using deep neural networks. In *IROS*, 8263–8270. IEEE.

Mao, J.; Niu, M.; Bai, H.; Liang, X.; Xu, H.; and Xu, C. 2021. Pyramid r-cnn: Towards better performance and adaptability for 3d object detection. In *CVPR*, 2703–2712.

Meng, Q.; Wang, W.; Zhou, T.; Shen, J.; Jia, Y.; and Van Gool, L. 2021. Towards a weakly supervised framework for 3d point cloud object detection and annotation. *IEEE TPAMI*, 4454–4468.

Peng, S.; Xi, X.; Wang, C.; Xie, R.; Wang, P.; and Tan, H. 2020. Point-based multilevel domain adaptation for point cloud segmentation. *IEEE Geoscience and Remote Sensing Letters*, 1–5.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *NeurIPS*, 5099–5108.

Saleh, K.; Abobakr, A.; Attia, M.; Iskander, J.; Nahavandi, D.; Hossny, M.; and Nahvandi, S. 2019. Domain adaptation for vehicle detection from bird's eye view LiDAR point cloud data. In *ICCV Workshops*.

Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; and Li, H. 2020. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *CVPR*, 10526–10535.

Shi, S.; Wang, X.; and Li, H. 2019. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. In *CVPR*, 770–779.

Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C. A.; Cubuk, E. D.; Kurakin, A.; and Li, C.-L. 2020. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. In *NeurIPS*.

Sun, B.; Feng, J.; and Saenko, K. 2016. Return of frustratingly easy domain adaptation. In *AAAI*.

Sun, P.; Kretzschmar, 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*, 2443–2451.

Tang, Y. S.; and Lee, G. H. 2019. Transferable semi-supervised 3d object detection from rgb-d data. In *ICCV*, 1931–1940.

Team, O. D. 2020. OpenPCDet: An Open-source Toolbox for 3D Object Detection from Point Clouds. https://github.com/open-mmlab/OpenPCDet. Accessed: 2022-07-10.

Tzeng, E.; Hoffman, J.; Saenko, K.; and Darrell, T. 2017. Adversarial discriminative domain adaptation. In *CVPR*, 2962–2971.

Wang, H.; Cong, Y.; Litany, O.; Gao, Y.; and Guibas, L. J. 2021. 3DIoUMatch: Leveraging iou prediction for semi-supervised 3d object detection. In *CVPR*, 14615–14624.

Wang, Y.; Chen, X.; You, Y.; Li, L. E.; Hariharan, B.; Campbell, M. E.; Weinberger, K. Q.; and Chao, W. 2020. Train in Germany, Test in the USA: Making 3D Object Detectors Generalize. In *CVPR*, 11710–11720. Computer Vision Foundation / IEEE.

Wu, B.; Zhou, X.; Zhao, S.; Yue, X.; and Keutzer, K. 2019. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *ICRA*.

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*, 2795–2803.

Xu, Q.; Zhou, Y.; Wang, W.; Qi, C. R.; and Anguelov, D. 2021a. Spg: Unsupervised domain adaptation for 3d object detection via semantic point generation. In *ICCV*.

Xu, S.; Zhou, D.; Fang, J.; Yin, J.; Bin, Z.; and Zhang, L. 2021b. FusionPainting: Multimodal fusion with adaptive attention for 3d object detection. In *ITSC*, 3047–3054. IEEE.

Yan, Y.; Mao, Y.; and Li, B. 2018. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10): 3337.

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, Z.; Sun, Y.; Liu, S.; and Jia, J. 2020. 3dssd: Point-based 3d single stage object detector. In *CVPR*, 11037–11045.

Yi, L.; Gong, B.; and Funkhouser, T. A. 2021. Complete & Label: A Domain Adaptation Approach to Semantic Segmentation of LiDAR Point Clouds. In *CVPR*, 15363–15373. Computer Vision Foundation / IEEE.

Yin, J.; Fang, J.; Zhou, D.; Zhang, L.; Xu, C.-Z.; Shen, J.; and Wang, W. 2022a. Semi-supervised 3D object detection with proficient teachers. In *ECCV*, 727–743. Springer.

Yin, J.; Shen, J.; Gao, X.; Crandall, D.; and Yang, R. 2021. Graph neural network and spatiotemporal transformer attention for 3D video object detection from point clouds. *IEEE TPAMI*.

Yin, J.; Zhou, D.; Zhang, L.; Fang, J.; Xu, C.-Z.; Shen, J.; and Wang, W. 2022b. Proposalcontrast: Unsupervised pre-training for lidar-based 3D object detection. In *ECCV*, 17–33. Springer.

Yin, T.; Zhou, X.; and Krahenbuhl, P. 2021. Center-based 3d object detection and tracking. In *CVPR*, 11784–11793.

Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 6022–6031.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond empirical risk minimization. In *ICLR*.

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.; Xu, G.; Ma, Y.; Wan, J.; and Guo, Y. 2022. Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In *CVPR*, 18931–18940.

Zhao, N.; Chua, T.-S.; and Lee, G. H. 2020. Sess: Self-ensembling semi-supervised 3d object detection. In *CVPR*, 11076–11084.

Zhao, S.; Wang, Y.; Li, B.; Wu, B.; Gao, Y.; Xu, P.; Darrell, T.; and Keutzer, K. 2021. ePointDA: An end-to-end simulation-to-real domain adaptation framework for LiDAR point cloud segmentation. In *AAAI*.

Zheng, W.; Tang, W.; Chen, S.; Jiang, L.; and Fu, C.-W. 2021. Cia-ssd: Confident iou-aware single-stage object detector from point cloud. In *AAAI*, 3555–3562.

Zhou, D.; Fang, J.; Song, X.; Liu, L.; Yin, J.; Dai, Y.; Li, H.; and Yang, R. 2020. Joint 3d instance segmentation and object detection for autonomous driving. In *CVPR*, 1836–1846.

Zhou, Y.; and Tuzel, O. 2018. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, 4490–4499.