Density-Insensitive Unsupervised Domain Adaption on 3D Object Detection

Qianjiang Hu Daizong Liu Wei Hu✉
Wangxuan Institute of Computer Technology, Peking University
No. 128, Zhongguancun North Street, Beijing, China
hqjpku@pku.edu.cn, dzliu@stu.pku.edu.cn, forhuwei@pku.edu.cn

Abstract

3D object detection from point clouds is crucial in safety-critical autonomous driving. Although many works have made great efforts and achieved significant progress on this task, most of them suffer from expensive annotation cost and poor transferability to unknown data due to the domain gap. Recently, few works attempt to tackle the domain gap in objects, but still fail to adapt to the gap of varying beam-densities between two domains, which is critical to mitigate the characteristic differences of the LiDAR collectors. To this end, we make the attempt to propose a density-insensitive domain adaption framework to address the density-induced domain gap. In particular, we first introduce Random Beam Re-Sampling (RBRS) to enhance the robustness of 3D detectors trained on the source domain to the varying beam-density. Then, we take this pre-trained detector as the backbone model, and feed the unlabeled target domain data into our newly designed task-specific teacher-student framework for predicting its high-quality pseudo labels. To further adapt the property of density-insensitivity into the target domain, we feed the teacher and student branches with the same sample of different densities, and propose an Object Graph Alignment (OGA) module to construct two object-graphs between the two branches for enforcing the consistency in both the attribute and relation of cross-density objects. Experimental results on three widely adopted 3D object detection datasets demonstrate that our proposed domain adaption method outperforms the state-of-the-art methods, especially over varying-density data. Code is available at https://github.com/WoodwindHu/DTS.

1. Introduction

3D object detection is a fundamental task in various real-world scenarios, such as autonomous driving [24, 35] and robot navigation [29], aiming to detect and localize traffic-related objects such as cars, pedestrians, and cyclists in 3D point clouds [16, 25, 26]. With the advent of deep learning, this task has obtained remarkable advances [24, 35–37, 49, 56, 60] in recent years, which however requires costly dense annotations of point clouds. Further, in real-world scenarios, upgrading LiDARs to other product models can be time-consuming and labor-intensive to collect and annotate massive data for each kind of product, while it is reasonable to use labeled data from previous sensors. Also, the number of LiDAR points used in mass-produced robots and vehicles is usually fewer than that in large-scale public datasets [44]. To bridge the domain gap caused by different LiDAR beams, it is essential to develop methods that address these differences. However, the generalization ability of existing methods is proved to be limited [43] when the 3D models trained on a specific dataset are directly applied to an unknown dataset collected with a different LiDAR, which prevents the wide applicability of 3D object detection in autonomous driving.

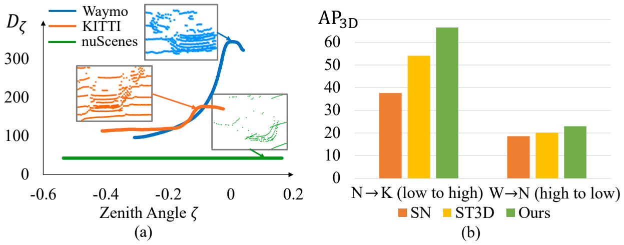

Figure 1. (a) The significant difference of beam densities among Waymo, KITTI, and nuScenes datasets. The beam density D_ζ represents the number of beams per unit zenith angle. The beams are evenly distributed in nuScenes, while the density of beams in Waymo and KITTI increases as the zenith angle ζ increases, with the highest density near the horizontal direction. (b) Compared to previous works (SN [43] and ST3D [51]), our method is more effective in transferring the knowledge from low density to high density or high density to low density (N: nuScenes, K: KITTI, W: Waymo.).

To reduce the domain gap between different datasets, some works [13, 14, 27, 34, 43, 44, 47, 51, 55, 57] proposed unsupervised domain adaptation (UDA) methods to transfer knowledge from a labeled source domain to an unlabeled target domain. However, most of them focus on reducing

[✉] Corresponding author: W. Hu. This work was supported by National Natural Science Foundation of China (61972009).

the domain gap introduced by the bias in object sizes on the labeled source domain, which neglect another important domain gap induced by *varying densities* of point clouds acquired from different types of LiDAR. We argue that this domain gap is crucial for 3D object detection in two aspects: 1) As demonstrated in Figure 1(a), different LiDAR collectors generally produce point cloud data with distinctive densities and distributions, leading to huge density-induced domain gap. 2) Most 3D detectors are directly trained on a single environment and thus sensitive to the cross-domain density variation. As shown in Figure 1(b), existing domain adaption methods suffer from performance bottlenecks in cross-density scenarios. Although few works [44] attempt to downsample point clouds of high density and transfer its knowledge to the low-density domain, they are limited to the model design that cannot realize the knowledge transfer from a low-density domain to a high-density domain. Hence, it is demanded to train robust 3D feature representations that can adapt to point cloud data of varying densities.

To this end, we make the attempt to propose a novel Density-insensitive Teacher-Student (DTS) framework to address the domain gap induced by varying point densities and distributions. The key idea of DTS is to first pre-train a density-insensitive object detector on the source domain, and then employ a self-training strategy [20, 51, 58] to fine-tune this detector on the unlabeled target domain by iteratively predicting and updating its pseudo results. However, there still remain two concerns: 1) Previous self-training methods may be prone to its mistake by using single-branch prediction. 2) How to adapt and improve the property of density-insensitivity of the pre-trained 3D detector on the target domain is important. Therefore, we introduce a task-specific teacher-student framework in order to provide more reliable and robust supervision, in which the teacher and student branches are fed with variants of the same sample in different densities. Further, considering the object prediction should be invariant in the two branches, we propose to capture their cross-density object-aware consistency for enhancing the density-insensitivity on the target domain.

To be specific, we first introduce Random Beam Re-Sampling (RBRS) to train the density-invariant 3D object detector on the labeled source domain, by randomly masking or interpolating the beams of the point clouds. Then, we take this pre-trained 3D detector as the backbone model to build a teacher-student framework to iteratively predict and update the pseudo labels on the unlabeled target domain. To achieve the goal of density-insensitivity, we feed the student and teacher models with the RBRS-augmented sample and the original sample, respectively. Moreover, in order to enforce the consistency in attributes and relations of detected objects in the teacher and student branches for more reliable supervision, we construct two graphs based on the objects predicted from the teacher and student models, and propose a novel Object Graph Alignment (OGA) to keep consistent cross-density object-attributes (node-level) and object-relations (edge-level) between the two graphs. During the training, the student model is optimized based on the predictions of the teacher while the weights of the teacher model are updated by taking the exponential moving average of the weights of the student model. In this way, our DTS is effective in reducing the density-induced domain gap and achieving state-of-the-art performance on the unknown target data.

In summary, our main contributions include

- We propose a density-insensitive unsupervised domain adaption framework to alleviate the influence of the domain gap caused by varying density distributions. We develop beam re-sampling to randomize the density of point clouds, which effectively enhances the robustness of 3D object detection to varying densities.
- We exploit a task-specific teacher-student framework to fine-tune the pre-trained 3D detector on the target domain. To adapt and improve the density-insensitivity on the target domain, we introduce an object graph alignment module to keep the cross-density object-aware consistency.
- Experimental results demonstrate our model significantly outperforms the state-of-the-art methods on three widely adopted 3D object detection datasets including NuScenes [4], KITTI [11], and Waymo [39].

2. Related Work

Point-cloud-based 3D Object Detection: Point-cloud-based 3D object detection [6, 10, 23, 24, 30, 35–37, 49, 50, 54, 59, 60] aims to localize and classify objects from point clouds. Depending on representation learning strategies, existing works can be divided into three categories: voxel based, point based, and voxel-point based. Voxel based methods [10,24,49,59,60] voxelize point clouds into 2D/3D compact grids and then collapse it to a bird's-eye-view representation. They are computationally effective but the desertion of fine-grained patterns degrades further refinement. Point based methods [36, 53] directly process point clouds without voxelization. These methods wholly preserve the irregularity and locality of a point cloud but have relatively higher latency. Point-voxel based methods [35, 54] integrate the advantages of both voxel based methods and point based methods together. Following previous works [44,51], we adopt voxel based PointPillars [24], SECOND [49] and point-voxel based PV-RCNN [35] as our detectors.

Unsupervised Domain Adaptation for 2D/3D Object Detection: UDA aims to transfer the model trained on the fully annotated source domain to the unannotated target domain. A variety of solutions [5, 7, 9, 15, 19–21, 31–33, 38, 42, 58] have been proposed in the 2D object detection task. As the pioneer, Ben *et al.* [2] designed a $\mathcal{H}\Delta\mathcal{H}$-distance

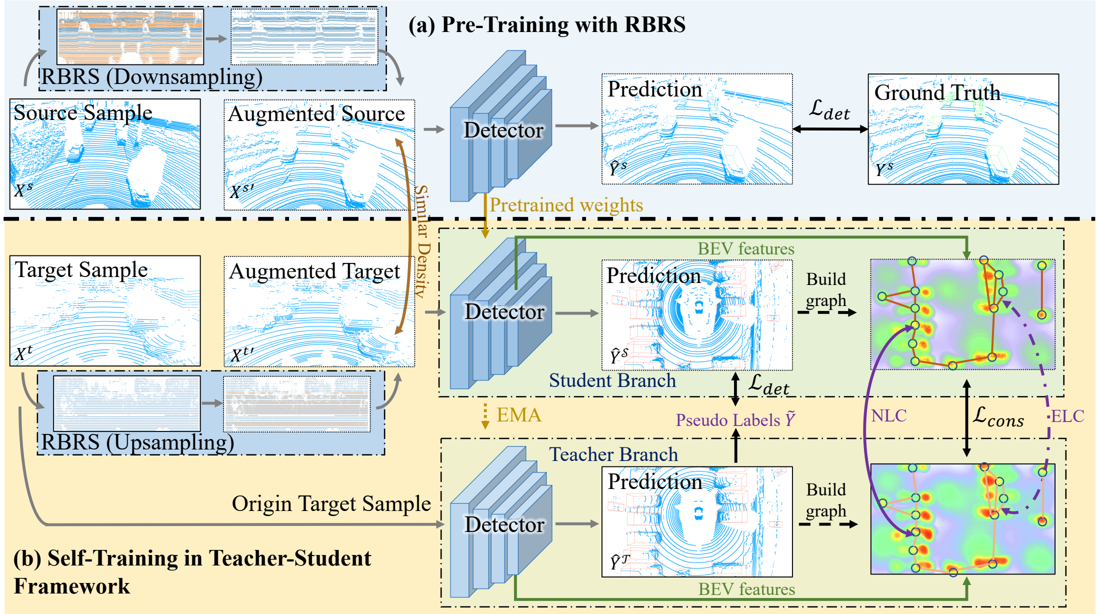

Figure 2. The overall pipeline of the proposed DTS. Here, we take the case of transferring the high-density domain into low-density domain as an example. Given the labeled source data and the unlabeled target data with different point density distributions, (a) our DTS first pre-trains a 3D detector on the source data with random beam re-sampling (RBRS), where the Yellow beams mark the masked/interpolated beams; (b) then DTS builds a teacher-student self-training framework with the pre-trained 3D detector. During the self-training process, the student and teacher branches are fed with different density variants of the same input. An object graph alignment module is further deployed to capture the cross-density object-aware consistency between the two branches.

to measure the divergence between two domains that have different data distributions and proposed a general framework to perform domain adaptation. Inspired by GAN [12], many methods in the literature [7, 15, 33, 38, 42] generate a domain discriminator to correctly classify the source/target domain while training a detector model to fool the domain discriminator. Some other methods [21, 31] follow the domain randomization strategy to devoid all source-style bias on the source detector.

While a lot of research has been conducted on UDA for object detection with 2D image data, there is relatively little literature in the field of UDA for 3D object detection. Wang *et al.* proposed statistical normalization (SN) [43] to normalize the object sizes of the source and target domains so they could bridge the domain gap introduced by the difference in object sizes. SPG [48] utilized the semantic point generation to tackle the domain gaps induced by deteriorated point cloud quality. SF-UDA [34] uses the temporal coherency to estimate the object size in the target domain while getting rid of the target domain statistics. 3D-CoCo [55] explores a contrastive co-training framework including separate 3D encoders to provide more stable supervisions from the labeled source data while avoiding the biased knowledge of the source domain. MLC-Net [27] implements a mean-teacher paradigm and exploits the point-, instance- and neural statistics-level consistency to facilitate the cross-domain transfer. Yang *et al.* proposed ST3D [51] which redesigns the self-training pipeline to improve the quality of pseudo-labels for 3D object detection. Although these methods successfully achieve performance improvement compared to direct transfer, they neglect the density-induced domain gap. Observing that ST3D is hard to adapt detectors from data with more beams to data with fewer beams, Wei *et al.* proposed LiDAR Distillation [44], which downsamples the high-density data to align the point cloud density of the source and target domains. Then they progressively distill the knowledge from the high-density data to the low-density data. Different from LiDAR Distillation that is limited to transferring from high density to low density, our proposed DTS is able to transfer knowledge under various settings of point densities.

3. Method

3.1. Problem Statement and Overview

Unsupervised domain adaption for 3D object detection aims to transfer a model trained on a labeled source domain to an unlabeled target domain. Generally, the source domain is a point cloud dataset $\{\boldsymbol{X}_i^s\}_{i=1}^{N_s}$ labeled with the corresponding class a_s and bounding box $\{\boldsymbol{Y}_i^s\}_{i=1}^{N_s}$, while the target domain is an unlabeled point cloud dataset $\{\boldsymbol{X}_i^t\}_{i=1}^{N_t}$, where s and t represent source and target domains respectively, and i means the i-th instance. N_s and N_t are the number of source and target point clouds, respectively. Generally, the label Y_i^s is a seven-dimensional vector, which is parameterized by its center location $\mathbf{c} = \{c_x, c_y, c_z\}$, bounding box size $\mathbf{b} = \{l, w, h\}$, and the yaw angle ξ, respectively. Since point clouds of different domains are often collected by different LiDAR equipments with various density distributions, it is demanded to develop a density-insensitive domain adaptation model to reduce this density-induced domain gap.

To this end, we propose a novel Density-insensitive Teacher-Student (DTS) framework, as shown in Figure 2, which mainly consists of three components.

- Given the source and target domain data, DTS first pre-trains a density-insensitive 3D detector on the source data with Random Beam Re-Sampling (RBRS) to reduce the density-induced domain gap.
- Then the pre-trained 3D detector is taken as the backbone to build a task-specific teacher-student self-training framework, in which the teacher model generates the pseudo labels of the target data and provides high-quality supervision to update network weights of the student model. To adapt the property of density-insensitivity on the target domain, we feed the student and teacher models with the RBRS-augmented and the original target samples, respectively.
- We further propose an Object Graph Alignment (OGA) module to capture and learn the cross-density object-aware consistency between student and teacher models for improving the density-insensitivity.

In the following, we provide the details of each component.

3.2. Pre-training with Random Beam Re-Sampling

Before transferring the knowledge from the source domain into the target domain, we need to pre-train a 3D detector to extract a wealth of knowledge on the annotated source data $\{(\boldsymbol{X}_i^s, \boldsymbol{Y}_i^s)\}_{i=1}^{N_s}$. However, this learned knowledge contains the domain-specific bias due to different object sizes and point densities collected by different LiDAR, leading to the poor generalization ability on the target domain. Although Yang *et al.* [51] seek to overcome the bias in object sizes via random object scaling (ROS), the density bias has been seldom investigated. Therefore, we propose to reduce this density gap among different domains and provide a density-insensitive 3D detector.

Given point clouds collected with M-beam LiDAR, we first denote the zenith angle of the j-th beamas as $\zeta(j)$. The density of beams, *i.e.*, the count of beams in the unit zenith angle, could be approximated by $D_\zeta(j) = 1/(\zeta(j+1) - \zeta(j))$. Inspired by the success of ROS in overcoming the bias in object sizes, we propose random beam re-sampling (RBRS), a simple yet effective strategy, to train a *density-insensitive* 3D detector.

Random Beam Re-Sampling. RBRS aims to randomly down-sample the dense data and up-sample the sparse data. Before re-sampling, we first transfer cartesian coordinates (x, y, z) of points to the spherical coordinates as:

$$\begin{aligned} \zeta &= \arctan \frac{z}{\sqrt{x^2+y^2}}, \\ \phi &= \arcsin \frac{y}{\sqrt{x^2+y^2}}, \\ r &= \sqrt{x^2+y^2+z^2}, \end{aligned} \tag{1}$$

where ζ and ϕ are zenith and azimuth angles, r is the distance from each point to the LiDAR sensor. By taking the zenith angle as the vertical coordinate and the azimuth angle as the horizontal coordinate, point clouds could be transformed to range images (like RBRS blocks in Figure 2). We use the K-Means algorithm [18, 28] to mark the the ID of beams in the range images according to its zenith angle ζ. Then we perform RBRS by randomly down-sampling or up-sampling the range images and reverse them into point clouds.

Specifically, to down-sample the dense data, for the j-th beam with the beam density of $D_\zeta(j)$, RBRS randomly masks this beam with the probability of η_j according to the beam density $D_\zeta(j)$. Considering a beam with a larger beam density is more likely to be masked, we formulate this mask probability η_j as $\eta_j = 1 - \gamma_1 / D_\zeta(j)$, where γ_1 is a factor to control the overall density of the sampled point cloud.

To up-sample the sparse data, RBRS randomly interpolates artificial beams between the original beams. Specifically, an artificial beam is interpolated with a probability of η'_j between the j-th beam and the $(j+1)$-th beam. Considering a beam with a larger $D_\zeta(j)$ has smaller η'_j, , the interpolation probability η'_j is set as $\eta'_j = \gamma_2 / D_\zeta(j)$, where γ_2 is the factor, and a larger γ_2 indicates more beams to be interpolated. Then, if a new beam is to be interpolated between the j-th beam and the $(j+1)$-th beam, every point in the j-th beam is selected as datum mark. Taking a point k in the j-th beam as an example, assuming its spherical coordinate is (ζ_k, ϕ_k, r_k), we first find a point in the $(j+1)$-th beam, marked as k', which is closest to point k measured in the azimuth angle. Then the spherical coordinate of the newly interpolated point is defined as:

$$\zeta = \frac{\zeta_k + \zeta_{k'}}{2}, \quad \phi = \frac{\phi_k + \phi_{k'}}{2}, \quad r = \frac{r_k + r_{k'}}{2}. \tag{2}$$

Subsequently, the newly interpolated point is transformed back to Cartesian coordinates for concatenating with the original points. At last, we utilize this RBRS strategy to pre-train the 3D detector on the source data.

3.3. Our Basic Teacher-Student Architecture

After obtaining the pre-trained density-insensitive 3D detector on the source domain, we take it as the backbone model to build a self-training framework for fine-tuning the source domain knowledge into the target domain. Motivated by the success of the teacher-student paradigm [3, 8, 40, 45] in semi-supervised learning, we design a task-specific teacher-student paradigm to fine-tune the 3D detector on the target domain.

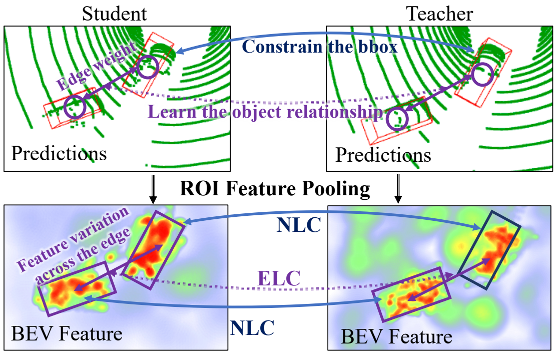

Figure 3. Illustration of our proposed Node-Level Consistency (NLC) and Edge-Level Consistency (ELC). We maintain the NLC to constrain the bounding-box regression and utilize the ELC to learn the object relations for better discriminating the objects.

Specifically, the teacher-student framework consists of two separate branches, *i.e.*, a non-trainable teacher detector $D_{\mathcal{T}}$ and a trainable student detector $D_{\mathcal{S}}$, sharing the same architecture. For each input $\boldsymbol{X}_i^t$ of the unlabeled target domain, the teacher detector is fed with its original data $\boldsymbol{X}_i^t$, while the student detector is fed with the augmented version $\boldsymbol{X}_i^{t'}$ via our RBRS. This operation is to adapt the property of density-insensitivity of the source domain into the target domain. We denote the generated bounding boxes of the teacher branch and the student branch as $\hat{\boldsymbol{Y}}_i^{\mathcal{T}}$ and $\hat{\boldsymbol{Y}}_i^{\mathcal{S}}$, respectively.

During the teacher-student framework learning, the teacher detector is used to generate pseudo labels of the input $\boldsymbol{X}_t$ and provide the supervision signal to train the student detector. In particular, the predictions of the teacher branch with confidence higher than a threshold c_{th} are chosen to generate the pseudo labels as:

$$\tilde{\boldsymbol{Y}}_i = \{\hat{\mathbf{y}}_j \in \hat{\boldsymbol{Y}}_i^{\mathcal{T}} | c_j > c_{th}\}, \tag{3}$$

where $\hat{\mathbf{y}}_j$ is the i-th predicted bounding box in $\hat{\boldsymbol{Y}}_i^{\mathcal{T}}$ and c_j is the confidence of $\hat{\mathbf{y}}_j$. The student detector takes these pseudo labels for supervision as:

$$\mathcal{L} = \mathcal{L}_{\text{det}} + \mathcal{L}_{\text{cons}}, \tag{4}$$

where $\mathcal{L}_{\text{det}}$ is the detection loss of the target-domain samples and is supervised by the pseudo label $\tilde{\boldsymbol{Y}}_i$, and $\mathcal{L}_{\text{cons}}$ is the consistency loss between the teacher and the student branch, which will be elaborated in Section 3.4.

To further improve the quality of the pseudo labels, we also deploy the exponential moving average (EMA) technique [40] to update the weight of the teacher detector as:

$$\theta^{\mathcal{T}} = \alpha\theta^{\mathcal{T}} + (1-\alpha)\theta^{\mathcal{S}}, \tag{5}$$

where α is a smoothing coefficient hyperparameter. The moving average in Eq. 5 makes $\theta^{\mathcal{T}}$ evolve more smoothly than $\theta^{\mathcal{S}}$. As a result, the teacher can aggregate information after every step and generate stable predictions of the input.

3.4. Object-Graph Consistency Learning between the Teacher and Student Branches

To further enhance the property of density-insensitivity in the target domain, we propose to capture the cross-density consistency among the student and teacher branches. Considering the detected objects of different-density variants of the same sample should be invariant, we build a contextual graph based on the predicted objects of each branch for aligning both node-level (object-attribute) and edge-level (object-relation) information between the two branches, as shown in Figure 3. Different from MLC-Net which also explores the multi-level consistency, we enforce consistency from objects' relations for capturing their similarity, by building graphs to model the object-level consistency via NLC and the **relation-level consistency** via ELC, which considers both local and global features.

Object Graph Construction According to the predicted objects of each branch, we construct a fully-connected undirected graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ to model the object relationship. For the graph of each branch, $\mathcal{N} = \{\mathbf{y}_i \in \hat{\mathbf{Y}}_t^{\mathcal{T}} | c_i > c_{th_{\mathcal{G}}}\}$ is the node set where each node represents an object prediction. $\mathcal{E}$ is the edge set where each edge represents the relationship between the connected objects. Here, we utilize a heuristic method to weigh the edges between the objects by 1) the close-distance objects have a stronger connection; 2) the objects share similar sizes have a stronger connection; 3) the objects in the same direction indicate they are driving in the same direction, thus they have a stronger connection. Based on these, we set the edge weight corresponding to the above connected objects' location $\mathbf{c}$, bounding box size $\mathbf{b}$ and yaw angle ξ as:

$$w_{ij} = \exp\left(-\frac{||\mathbf{c}_i - \mathbf{c}_j||_2^2 + \epsilon_1||\mathbf{b}_i - \mathbf{b}_j||_2^2 + \epsilon_2(\xi_i - \xi_j)^2}{\tau^2}\right), \tag{6}$$

where ϵ_1 and ϵ_2 control the importance of the object size and yaw angle, and τ is a temperature hyperparameter. $||\cdot||_2$ is the L2 norm operation. We denote the graphs of the teacher and the student as $\mathcal{G}^{\mathcal{S}} = \{\mathcal{N}^{\mathcal{S}}, \mathcal{E}^{\mathcal{S}}\}$ and $\mathcal{G}^{\mathcal{T}} = \{\mathcal{N}^{\mathcal{T}}, \mathcal{E}^{\mathcal{T}}\}$, respectively.

Node-Level Consistency Since the inputs of both teacher and student branches are generated from the same sample with different densities, their object predictions should be density-invariant, *i.e.*, $\mathcal{N}^{\mathcal{S}}$ and $\mathcal{N}^{\mathcal{T}}$ is fully matched. Therefore, we aim to capture the node-level consistency between the graphs of two branches for aligning the predicted bounding boxes of the same object from inputs of different densities.

Specifically, we first calculate the IoU between the student predictions $\mathcal{N}^{\mathcal{S}}$ and the teacher prediction $\mathcal{N}^{\mathcal{T}}$. If the IoU is larger than a threshold IoU_{th}, we consider these object detection results are matched and belong to the same object. The detection results with lower IoU are filtered out. For each matched object, we utilize its bird's eye

Algorithm 1 Training Process of our DTS

```
Require: Labeled source domain data $\{\boldsymbol{X}_i^s, \boldsymbol{Y}_i^s\}_{i=1}^{N_s}$, and
    unlabeled target domain data $\{\boldsymbol{X}_i^t\}_{i=1}^{N_t}$
1: Pre-train the object detector on $\{\boldsymbol{X}_i^s, \boldsymbol{Y}_i^s\}_{i=1}^{N_s}$ with
   RBRS as detailed in Sec. 3.2.
2: Take the pre-trained model as the backbone to build the
   teacher-student architecture.
3: for $i = 1$ to $N_t$ do
4:     Forward the teacher-student network demonstrated
       in Figure 2 with $\boldsymbol{X}_i^t$.
5:     Generate pseudo labels $\tilde{\boldsymbol{Y}}_i$ using the teacher's output
       $\hat{\boldsymbol{Y}}_i$ with Eq. 3.
6:     Calculate the bounding-box supervision $\mathcal{L}_{det}$ of the
       student branch with $\tilde{\boldsymbol{Y}}_i$.
7:     Construct object graphs over the two branches as detailed in Sec. 3.4
8:     Calculate the cross-graph node-level consistency loss
       $\mathcal{L}_{node}$ and edge-level consistency loss $\mathcal{L}_{edge}$ as detailed in Sec. 3.4
9:     Back-forward the total loss in Eq. 4 to update the student network.
10:    Update the teacher's weight using Eq. 5.
11: end for
12: Go back to Line 3 until convergence.
Output: The object detection model for the target domain.
```

view (BEV) features sampled via region of interest (ROI) as the guidance to pull its predicted bounding boxes of two branches closer. Considering that the matched object features of the two branches should be similar, we design the node-level consistency loss $\mathcal{L}_{\text{node}}$ to maximize their similarity in attributes as:

$$\mathcal{L}_{\text{node}} = \frac{1}{N}\sum_{i=1}^{N} \exp\left(-\frac{(\mathbf{f}_i^{\mathcal{S}})^\top \mathbf{f}_i^{\mathcal{T}}}{||\mathbf{f}_i^{\mathcal{S}}||_2 ||\mathbf{f}_i^{\mathcal{T}}||_2}\right), \tag{7}$$

where N is the number of the matched object pairs, $\mathbf{f}_i^{\mathcal{S}} \in R^C$ and $\mathbf{f}_i^{\mathcal{T}} \in R^C$ are the C-channel ROI features of the i-th object from the student and the teacher detectors.

Edge-Level Consistency In addition to the object-level consistency, the edge-level consistency is also worth exploring since the same object in two branches should contribute similarly to its neighboring objects. To align the edges of the graphs in two branches, we consider to not only match the corresponding edge weights but also constrain the same feature variation across the edges. Specifically, the edge weight alignment could be implemented by minimizing the difference between the teacher's edge weight and the student's edge weight. Besides, to measure the feature variation across the edges, we introduce the graph Laplacian regularization (GLR) [1, 17] as:

$$GLR = \mathbf{tr}(\boldsymbol{F}^T(\mathbf{D} - \mathbf{W})\boldsymbol{F}), \tag{8}$$

where $\mathbf{tr}(\cdot)$ calculates the trace of the matrix, $\boldsymbol{F} \in R^{N\times C}$ is the concatenated features of nodes, $\mathbf{W} \in R^{N\times N}$ is the adjacency matrix with each entry denoting an edge weight $w_{i,j}$, $\mathbf{D}$ is the degree matrix—a diagonal matrix where $d_{ij} = \sum_{j=1}^{N} w_{ij}$. GLR measures the smoothness of features with respect to the graph: a smaller GLR represents smaller variation in node features across edges. Hence, we define the edge-level consistency loss $\mathcal{L}_{\text{edge}}$ as:

Datasets	Size	LiDAR Type	Vertical Field
Waymo	230K	1 × 64 + 4 × 200-Beam	$[-17.6°, 2.4°]$
KITTI	15K	1 × 64-Beam	$[-23.6°, 3.2°]$
nuScenes	40K	1 × 32-Beam	$[-30.0°, 10, 0°]$

Table 1. Datasets overview. The dataset size refers to the number of annotated point cloud frames.

$$\mathcal{L}_{\text{edge}} = \frac{1}{N^2}\left(\gamma||\mathbf{W}^{\mathcal{T}} - \mathbf{W}^{\mathcal{S}}||_2^2 + (1-\gamma)\mathbf{tr}(GLR^{\mathcal{S}} - GLR^{\mathcal{T}})\right), \tag{9}$$

where γ strikes a balance between the two terms. The first term represents the difference in edge weights between the two graphs, while the second term aims to enforce the feature variation of the student branch to be similar with that of the teacher branch.

Overall, the joint consistency loss $\mathcal{L}_{\text{cons}}$ is defined as:

$$\mathcal{L}_{\text{cons}} = \beta_1\mathcal{L}_{\text{node}} + \beta_2\mathcal{L}_{\text{edge}}, \tag{10}$$

where β_1 and β_2 are the hyperparameters to control the involvement of the two consistency losses. The training process of the whole DTS is summarized in Algorithm 1.

4. Experiments

4.1. Experimental Settings

Datasets. We conduct experiments on three widely used autonomous driving datasets: KITTI [11], Waymo [39], and nuScenses [4]. The KITTI [11] contains 7481 frames of point clouds for training and validation, and all the data is collected with 64-beam Velodyne LiDAR. The Waymo [39] dataset contains 122000 training and 30407 validation frames of point clouds collected with five LiDAR sensors, *i.e.*, one 64-beam LiDAR and four 200-beam LiDAR. The nuScenes dataset [4] contains 28130 training and 6019 validation point clouds collected with a 32-beam roof LiDAR. Table 1 shows an overview of the three datasets. Note that there is a huge difference in their densities (also shown in Figure 1(a)). Following previous works [51, 52], we evaluate our DTS by adapting across domains with different LiDAR-beam densities (Waymo → nuScenes, Waymo → KITTI and nuScenes → KITTI).

Evaluation metrics. We follow [51] and adopt the KITTI evaluation metric for evaluating our methods on the commonly used car category (the vehicle category in Waymo). In detail, we use the average precision (AP) as the evaluation metric for both BEV IoUs and 3D IoUs under an IoU threshold of 0.7 over 40 recall positions. We also adopt the domain adaptation metric (*i.e.*, Closed Gap) [51] to demonstrate the effectiveness on domain adaption, which is defined as $Closed\ Gap = \frac{\text{AP}_{model} - \text{AP}_{source}}{\text{AP}_{oracle} - \text{AP}_{source}} \times 100\%$.

Implementation details. We validate the proposed DTS on three detection backbones SECOND-IoU [51], PV-RCNN [35] and PointPillars [24]. We adopte the training

Task	Method	SECOND-IoU [51]		PV-RCNN [35]		PointPillars [24]	
		$\text{AP}_{\text{BEV}}/\text{AP}_{\text{3D}}$	Closed Gap	$\text{AP}_{\text{BEV}}/\text{AP}_{\text{3D}}$	Closed Gap	$\text{AP}_{\text{BEV}}/\text{AP}_{\text{3D}}$	Closed Gap
N→K	Source Only	51.8/17.9	-/-	68.2/37.2	-/-	22.8/0.5	-/-
	SN† [43]	59.7/37.6	+25.1%/+35.4%	60.5/49.5	+36.8%/+27.1%	39.3/2.0	+26.6%/+2.1%
	ST3D [51]	75.9/54.1	+76.6%/+59.5%	78.4/70.9	+49.0%/+74.3%	60.4/11.1	+60.6%/+14.9%
	ST3D++ [52]	80.5/ 62.4	+91.1%/+80.0%	-/-	-/-	-/-	-/-
	3D-CoCo [55]	-/-	-/-	-/-	-/-	77.0/47.2	+87.4%/+65.7%
	Ours	**81.4/66.6**	**+94.0%/+87.6%**	**83.9/71.8**	**+75.8%/+76.4%**	**79.5/51.8**	**+91.5%/+72.2%**
	Oracle	83.3/73.5	-/-	88.9/82.5	-/-	84.8/71.6	-/-
W→K	Source Only	67.6/27.5	-/-	61.2/22.0	-/-	47.8/11.5	-/-
	SN† [43]	79.0/59.2	+72.3%/+69.0%	79.8/63.6	+66.9%/+68.7%	27.4/6.4	-55.1%/-8.5%
	ST3D [51]	82.2/61.8	+93.0%/+74.7%	84.1/64.8	+82.4%/+70.7%	58.1/23.2	+27.8%/+19.5%
	ST3D++ [52]	80.8/65.6	+84.1%/+82.8%	-/-	-/-	-/-	-/-
	3D-CoCo [55]	-/-	-/-	-/-	-/-	**76.1**/42.9	**+76.5%**/+52.2%
	Ours	**85.8/71.5**	**+115.9%/+95.7%**	**86.4/68.1**	**+90.6%/+76.2%**	**76.1/50.2**	**+76.5%/+64.4%**
	Oracle	83.3/73.5	-/-	89.0/82.5	-/-	84.8/71.6	-/-
W→N	Source Only	32.9/17.2	-/-	34.5/21.5	-/-	27.8/12.1	-/-
	SN† [43]	33.2/18.6	+1.7%/+7.5%	34.2/22.3	-1.5%/+4.8%	28.3/13.0	+2.4%/+4.7%
	ST3D [51]	35.9/20.2	+15.9%/+16.7%	36.4/23.0	+10.3%/+8.8%	30.6/15.6	+13.2%/+18.2%
	ST3D++ [52]	35.7/20.9	+14.7%/+20.9%	-/-	-/-	-/-	-/-
	3D-CoCo [55]	-/-	-/-	-/-	-/-	33.1/20.7	+25.0%/+44.8%
	L· D [44]	40.7/22.9	+41.1%/+32.2%	43.3/25.6	+47.3%/+24.0%	40.2/19.1	+58.4%/+36.5%
	Ours	**41.2/23.0**	**+43.7%/+32.8%**	**44.0/26.2**	**+51.1%/+27.5%**	**42.2/21.5**	**+67.9%/+49.0%**
	Oracle	51.9/34.9	-/-	53.1/38.6	-/-	49.0/31.3	-/-

Table 2. Performance comparison of different methods on different domain adaptation tasks. †: SN is weakly supervised with target domain statistics. Source Only indicates that the model trained on the source dataset is directly tested on the target dataset. Oracle indicates that the model is trained with labeled target data. We report AP_{BEV} and AP_{3D} over 40 recall positions of the car category at IoU = 0.7.

setup of the popular point cloud detection codebase open pcdet [41] to pre-train our detectors on the source domain. For the following target domain self-training stage, we use Adam [22] and one cycle scheduler to fine-tune the detectors for 30 epochs. The learning rate is set to 1.5×10^{-3}. The EMA smoothing coefficient hyperparameter α is set to 0.999. The confidence threshold c_{th} of pseudo labels as well as the confidence threshold c_{th_g} for node construction are both set to 0.5. We set λ_1 and λ_2 to 5.0 and 20.0 empirically, which control the participation of object sizes and yaw angles to build object graphs. The temperature τ is set to 13.0. The involvement hyperparameters of the two consistency losses β_1 and β_2 are set to 0.05 and 0.3 respectively, while γ is set to 0.5. The threshold to match nodes IoU_{th} is set to 0.1. We set the beam interpolation probability factor as $\epsilon_2 = 25.0$ to up-sample nuScenes and the beam mask probability factor as $\epsilon_1 = 75.0$ and $\epsilon_1 = 100.0$ to down-sample KITTI and Waymo respectively.

4.2. Comparison with State-of-the-Arts

Main Results. We compare our proposed DTS with SN [43], ST3D [51], ST3D++ [52], 3D-CoCo [55] and L.D [44]. As shown in Table 2, DTS outperforms all compared methods by large margins on all domain adaptation settings. SN, ST3D and ST3D++ overcome the bias of object sizes in the source domain effectively, however, their ignorance of the density-induced domain gap sacrifices some of their performance. 3D-CoCo includes separate 3D encoders for the source and target data, making it hard to utilize the useful knowledge from the 3D encoder of the source branch, thus leading to worse performance. Compared to L.D. that only transfers knowledge from high density into low density, our DTS is more density-insensitive.

Further, while our method is proposed for UDA, we observed one needs to provide around 50% labels to reach parity with the oracle detector, thus validating the potential applicability (Supplementary Sec. S3).

Domain Adaption on Different Densities. To demonstrate the effectiveness of our DTS on overcoming the domain gap induced by LiDAR densities, we implement experimental comparison by adapting the nuScenes-trained model to different down-sampled KITTI dataset of different densities. As nuScenes data is collected with 32-beam LiDAR, we down-sample KITTI data to 16-, 32-, 48-beams for simulation of different density situations, including the high-density to low-density or similar density, and the low-density to high-density. As shown in Figure 4, we compare our method with Source Only, SN [43] and ST3D [54] with the metric AP_{3D}. We observe that, with the increase in density from 32-beam to 64-beam, the performance of ST3D has almost no improvement, while the performance of SN even becomes worse. This is because SN and ST3D suffer from density gaps since their general detectors are sensitive to density of points without any special design. Instead, our DTS trains density-insensitive detectors to overcome the density-induced domain gap, thus outperforming others in all density settings.

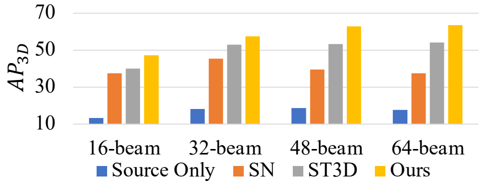

Figure 4. Adapting the model trained on nuScenes data to different down-sampled KITTI data of different densities.

	Pre-Training	Self-Training			AP_{BEV}	AP_{3D}
	RBRS	RBRS	NLC	ELC		
(a)					73.3	55.8
(b)	✓				77.6	60.7
(c)		✓			74.7	60.7
(d)	✓	✓			79.0	61.7
(e)	✓	✓	✓		79.5	64.7
(f)	✓	✓	✓	✓	**81.4**	**66.6**

Table 3. Main ablations on the architecture design. NLC and ELC represent node-level consistency and edge-level consistency.

4.3. Ablation Studies

All ablation studies are conducted on nuScenes → KITTI, using SECOND-IoU as the network backbone.

Main ablation. As demonstrated in Table 3, we investigate the contribution of each component. Starting from the backbone model (a), we pre-train the 3D detector without RBRS in the source domain, then directly fine-tune the model through our basic teacher-student framework without RBRS and object graph alignment. By applying our proposed RBRS into the pre-training on the source data, the variant (b) brings significant improvement since it is beneficial to density-insensitivity. Compared to (b), (c) does not introduce any boost as there is no annotation supervision in the target domain. Moreover, by applying the RBRS and both NLC, ELC of the object graph alignment into the self-training, the performances of variants (d), (e), (f) improve a lot, demonstrating the effectiveness of each component.

Ablations on the re-sampling strategy. As in Table 4, we test different re-sampling strategies on both source only method and our DTS. Since KITTI is denser than nuScenes, all strategies are designed to up-sample the nuScenes data or down-sample the KITTI data to obtain similar density distribution. In the experiments for Source Only, we see that RBRS performs better than directly up-sampling the point cloud. In the experiments for DTS, RBRS also achieves the best performance among different re-sampling strategies. We also observe that there is almost no improvement by simply up-sampling the source domain or down-sampling the target domain. We think the reason is that a simple up-sampling or down-sampling strategy could neither introduce more information from data nor train a density-insensitive detector.

Ablations on the teacher-student framework. To investigate the effectiveness of our object-graph based teacher-student framework, we also compare our model with different self-training pipelines, like naive Teacher-Student [46] and ST3D [51]. As shown in Table 5, our object-graph based teacher-student framework outperforms the two compared self-training frameworks, since we explore the object consistency for learning density-invariant bounding boxes.

Method	Source Data		Target Data			AP_{BEV}	AP_{3D}
	Beam	RBRS	Beam	Point	RBRS		
Source Only						60.5	42.4
	✓					70.9	48.2
		✓				**74.7**	**54.6**
DTS						73.9	56.8
	✓					74.2	56.4
		✓				80.9	64.0
			✓			74.8	56.0
				✓		74.1	54.4
					✓	80.2	61.9
		✓			✓	**81.4**	**66.6**

Table 4. Ablation on the re-sampling strategy. Here we take the case of transferring from low-density into high-density as example. The sampling strategies include beam-level up-sampling and RBRS on the source domain, as well as beam-level down-sampling, point-level down-sampling and RBRS on the target domain.

Self-Training Framework	AP_{BEV}	AP_{3D}
naive Teacher-Student [46]	78.8	55.5
ST3D [51]	78.7	59.1
Ours	**81.4**	**66.6**

Table 5. Ablation on different self-training frameworks. All the frameworks are implemented with RBRS in both the pre-training stage and self-training stage.

Sensitive analysis of node-level consistency and edge-level consistency. As shown in Table 6, we evaluate different β_1 and β_2 to control the weights of NLC and ELC. When β_1 and β_2 changed, the overall performance remains relatively good while AP_{3D} is more sensitive to NLC.

β_1	β_2	AP_{BEV}	AP_{3D}	β_1	β_2	AP_{BEV}	AP_{3D}
0.05	0.0	79.5	64.7	0.00	0.3	79.5	62.7
0.05	0.1	81.0	63.9	0.02	0.3	81.1	64.0
0.05	0.2	81.0	64.3	0.04	0.3	81.3	64.5
0.05	0.3	**81.4**	**66.6**	0.05	0.3	**81.4**	**66.6**
0.05	0.4	81.1	64.1	0.06	0.3	81.0	64.0
0.05	0.5	80.6	61.5	0.08	0.3	81.1	63.8

Table 6. Sensitivity analysis of NLC and ELC.

5. Conclusion

We propose a novel DTS model to bridge the density-induced domain gap for unsupervised domain adaption on 3D object detection. In particular, we design Random Beam Re-Sampling to train a density-insensitive detector on the source domain. To adapt the property of density-insensitivity into the target domain, we then develop a teacher-student framework with Object Graph Alignment to maintain the consistency in both cross-density object attributes and object relations. Experiments over three datasets demonstrate the effectiveness of the proposed DTS.

References

[1] Rie Ando and Tong Zhang. Learning on graph with laplacian regularization. *Advances in neural information processing systems*, 19, 2006. 6

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010. 2

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019. 4

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2, 6

[5] Qi Cai, Yingwei Pan, Chong-Wah Ngo, Xinmei Tian, Lingyu Duan, and Ting Yao. Exploring object relation in mean teacher for cross-domain detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11457–11466, 2019. 2

[6] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 2

[7] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018. 2, 3

[8] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018. 4

[9] Jinhong Deng, Wen Li, Yuhua Chen, and Lixin Duan. Unbiased mean teacher for cross-domain object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4091–4101, 2021. 2

[10] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1201–1209, 2021. 2

[11] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2, 6

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 3

[13] Deepti Hegde and Vishal Patel. Attentive prototypes for source-free unsupervised domain adaptive 3d object detection. *arXiv preprint arXiv:2111.15656*, 2021. 1

[14] Deepti Hegde, Vishwanath Sindagi, Velat Kilic, A Brinton Cooper, Mark Foster, and Vishal Patel. Uncertainty-aware mean teacher for source-free unsupervised domain adaptive 3d object detection. *arXiv preprint arXiv:2109.14651*, 2021. 1

[15] Cheng-Chun Hsu, Yi-Hsuan Tsai, Yen-Yu Lin, and Ming-Hsuan Yang. Every pixel matters: Center-aware feature alignment for domain adaptive object detector. In *European Conference on Computer Vision*, pages 733–748. Springer, 2020. 2, 3

[16] Qianjiang Hu, Daizong Liu, and Wei Hu. Exploring the devil in graph spectral domain for 3d point cloud attacks. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*, pages 229–248. Springer, 2022. 1

[17] Wei Hu, Jiahao Pang, Xianming Liu, Dong Tian, Chia-Wen Lin, and Anthony Vetro. Graph signal processing for geometric data and beyond: Theory and applications. *IEEE Transactions on Multimedia*, 2021. 6

[18] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988. 4

[19] Mehran Khodabandeh, Arash Vahdat, Mani Ranjbar, and William G Macready. A robust learning approach to domain adaptive object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 480–490, 2019. 2

[20] Seunghyeon Kim, Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-training and adversarial background regularization for unsupervised domain adaptive one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6092–6101, 2019. 2

[21] Taekyung Kim, Minki Jeong, Seunghyeon Kim, Seokeon Choi, and Changick Kim. Diversify and match: A domain adaptive representation learning paradigm for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12456–12465, 2019. 2, 3

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[23] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018. 2

[24] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019. 1, 2, 6, 7

[25] Daizong Liu and Wei Hu. Imperceptible transfer attack and defense on 3d point cloud classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1

[26] Daizong Liu, Wei Hu, and Xin Li. Point cloud attacks in graph spectral domain: When 3d geometry meets graph signal processing. *arXiv preprint arXiv:2207.13326*, 2022. 1

[27] Zhipeng Luo, Zhongang Cai, Changqing Zhou, Gongjie Zhang, Haiyu Zhao, Shuai Yi, Shijian Lu, Hongsheng

Li, Shanghang Zhang, and Ziwei Liu. Unsupervised domain adaptive 3d detection with multi-level consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8866–8875, 2021. 1, 3

[28] J MacQueen. Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, pages 281–297, 1967. 4

[29] Flavio BP Malavazi, Remy Guyonneau, Jean-Baptiste Fasquel, Sebastien Lagrange, and Franck Mercier. Lidar-only based navigation algorithm for an autonomous agricultural robot. *Computers and electronics in agriculture*, 154:71–79, 2018. 1

[30] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018. 2

[31] Adrian Lopez Rodriguez and Krystian Mikolajczyk. Domain adaptation for object detection via style consistency. *arXiv preprint arXiv:1911.10033*, 2019. 2, 3

[32] Aruni RoyChowdhury, Prithvijit Chakrabarty, Ashish Singh, SouYoung Jin, Huaizu Jiang, Liangliang Cao, and Erik Learned-Miller. Automatic adaptation of object detectors to new domains using self-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 780–790, 2019. 2

[33] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6956–6965, 2019. 2, 3

[34] Cristiano Saltori, Stéphane Lathuiliére, Nicu Sebe, Elisa Ricci, and Fabio Galasso. Sf-uda 3d: Source-free unsupervised domain adaptation for lidar-based 3d object detection. In *2020 International Conference on 3D Vision (3DV)*, pages 771–780. IEEE, 2020. 1, 3

[35] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. 1, 2, 6, 7

[36] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–779, 2019. 1, 2

[37] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE transactions on pattern analysis and machine intelligence*, 43(8):2647–2664, 2020. 1, 2

[38] Vishwanath A Sindagi, Poojan Oza, Rajeev Yasarla, and Vishal M Patel. Prior-based domain adaptive object detection for hazy and rainy conditions. In *European Conference on Computer Vision*, pages 763–780. Springer, 2020. 2, 3

[39] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 2, 6

[40] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. 4, 5

[41] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020. 7

[42] Vibashan Vs, Vikram Gupta, Poojan Oza, Vishwanath A Sindagi, and Vishal M Patel. Mega-cda: Memory guided attention for category-aware unsupervised domain adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4516–4526, 2021. 2, 3

[43] Yan Wang, Xiangyu Chen, Yurong You, Li Erran Li, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. Train in germany, test in the usa: Making 3d object detectors generalize. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11713–11723, 2020. 1, 3, 7

[44] Yi Wei, Zibu Wei, Yongming Rao, Jiaxin Li, Jie Zhou, and Jiwen Lu. Lidar distillation: Bridging the beam-induced domain gap for 3d object detection. *arXiv preprint arXiv:2203.14956*, 2022. 1, 2, 3, 7

[45] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268, 2020. 4

[46] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020. 8

[47] Chang-Dong Xu, Xing-Ran Zhao, Xin Jin, and Xiu-Shen Wei. Exploring categorical regularization for domain adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11724–11733, 2020. 1

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

[49] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 1, 2

[50] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018. 2

[51] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10368–10378, 2021. 1, 2, 3, 4, 6, 7, 8

[52] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d++: Denoised self-training for unsupervised domain adaptation on 3d object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 6, 7

[53] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11040–11048, 2020. 2

[54] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1951–1960, 2019. 2, 7

[55] Zeng Yihan, Chunwei Wang, Yunbo Wang, Hang Xu, Chaoqiang Ye, Zhen Yang, and Chao Ma. Learning transferable features for point cloud detection via 3d contrastive co-training. *Advances in Neural Information Processing Systems*, 34:21493–21504, 2021. 1, 3, 7

[56] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021. 1

[57] Weichen Zhang, Wen Li, and Dong Xu. Srdan: Scale-aware and range-aware domain adaptation network for cross-dataset 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6769–6779, 2021. 1

[58] Ganlong Zhao, Guanbin Li, Ruijia Xu, and Liang Lin. Collaborative training between region proposal localization and classification for domain adaptive object detection. In *European Conference on Computer Vision*, pages 86–102. Springer, 2020. 2

[59] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu. Se-ssd: Self-ensembling single-stage object detector from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14494–14503, 2021. 2

[60] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018. 1, 2