ST3D: Self-training for Unsupervised Domain Adaptation on 3D Object Detection

Jihan Yang[1]*, Shaoshuai Shi[2]*, Zhe Wang[3,4], Hongsheng Li[2,5], Xiaojuan Qi[1]†
[1]The University of Hong Kong [2]CUHK-SenseTime Joint Laboratory, The Chinese University of Hong Kong
[3]SenseTime Research [4]Shanghai AI Laboratory [5]School of CST, Xidian University
{jhyang, xjqi}@eee.hku.hk, shaoshuaics@gmail.com, wangzhe@sensetime.com, hsli@ee.cuhk.edu.hk

Abstract

We present a new domain adaptive self-training pipeline, named ST3D, for unsupervised domain adaptation on 3D object detection from point clouds. First, we pre-train the 3D detector on the source domain with our proposed random object scaling strategy for mitigating the negative effects of source domain bias. Then, the detector is iteratively improved on the target domain by alternatively conducting two steps, which are the pseudo label updating with the developed quality-aware triplet memory bank and the model training with curriculum data augmentation. These specific designs for 3D object detection enable the detector to be trained with consistent and high-quality pseudo labels and to avoid overfitting to the large number of easy examples in pseudo labeled data. Our ST3D achieves state-of-the-art performance on all evaluated datasets and even surpasses fully supervised results on KITTI 3D object detection benchmark. Code will be available at https://github.com/CVMI-Lab/ST3D.

1. Introduction

3D object detection aims to categorize and localize objects from 3D sensor data (*e.g.* LiDAR point clouds) with many applications in autonomous driving, robotics, virtual reality, to name a few. Recently, this field has obtained remarkable advancements [46, 24, 36, 37, 34, 35] driven by deep neural networks and large-scale human-annotated datasets [13, 38].

However, 3D detectors developed on one specific domain (*i.e.* source domain) might not generalize well to novel testing domains (*i.e.* target domains) due to unavoidable domain-shifts arising from different types of 3D sensors, weather conditions and geographical locations, etc. For instance, a 3D detector trained on data collected in USA cities with Waymo LiDAR (*i.e.* Waymo dataset [38]) suffers from a dramatic performance drop (of over 45%) [41] when evaluated on data from European cities captured by Velodyne LiDAR (*i.e.* KITTI dataset [13]). Though collecting more training data from different domains could alleviate this problem, it unfortunately might be infeasible given various real-world scenarios and enormous costs for 3D annotation. Therefore, approaches to effectively adapting 3D detector trained on labeled source domain to a new unlabeled target domain is highly demanded in practical applications. This task is also known as unsupervised domain adaptation (UDA) for 3D object detection.

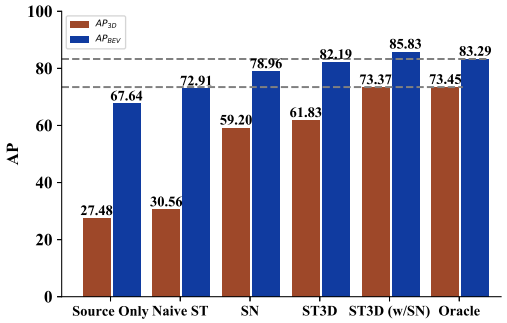

Figure 1. Performance of ST3D on Waymo → KITTI task using SECOND-IoU [46], compared to other unsupervised (*i.e.* source only, naive ST), weakly-supervised (*i.e.* SN [41]) and fully supervised (*i.e.* oracle) approaches. Dashed line denotes fully supervised target labeled data trained SECOND-IoU.

In contrast to the intensive studies on UDA of the 2D image setting [10, 26, 17, 7, 32, 11, 12], few efforts [41] have been made to explore UDA for 3D detection. Meanwhile, the fundamental differences in data structures and network architectures render UDA approaches for image tasks not readily applicable to this problem. For DA on 3D detection, while promising results have been obtained in [41], the method requires object size statistics of the target domain, and its efficacy largely depends on data distributions.

Recently, self-training has emerged as a simple and effective technique for UDA, attaining state-of-the-art performance on many image recognition tasks [51, 54, 21]. This motivates us to study self-training for UDA on 3D object detection. Self-training starts from pre-training a model on

*equal contribution
†corresponding author

source labeled data and further iterating between pseudo label generation and model training on unlabeled target data until convergence is achieved. The pseudo label for 3D object detection includes oriented 3D bounding boxes for localization and object category information. Despite of the encouraging results in image tasks, our study illustrates that naive self-training [44] does not work well in UDA for 3D detection as shown in Fig. 1 ("source only" *vs*. "naive ST").

In this paper, we propose ST3D, redesigning the self-training pipeline, for UDA on 3D object detection. First, in model pre-training, we develop *random object scaling* (ROS), a simple 3D object augmentation technique, randomly scaling the 3D objects to overcome the bias in object size on the labeled source domain. Second, for pseudo label generation, we develop a *quality-aware triplet memory bank* (QTMB) which encompasses an IoU-based box scoring criterion to directly assess the quality of pseudo boxes, a triplet box partition scheme to avoid assigning pseudo labels to ambiguous examples, and a memory bank, integrating historical pseudo labels via ensemble and voting, to reduce pseudo label noise and stabilize training. Finally, in the model training process, we design a *curriculum data augmentation* (CDA) strategy, progressively increasing the intensity of augmentation, to guarantee effective learning at the beginning and gradually simulate hard examples to improve the model, preventing it from overfitting to easy examples – pseudo-labeled data with high confidence.

Experimental results on four 3D object detection datasets KITTI [13], Waymo [38], nuSenses [4], and Lyft [20] demonstrate the effectiveness of our approach, where the performance gaps between source only results and fully supervised oracle results are closed by a large percentage ($16\% \sim 75\%$). Besides, we outperform the existing approach [41] by a notable margin on all evaluated settings. It's also noteworthy that our approach even outperforms the oracle results on the Waymo $\rightarrow$ KITTI setting when further combined with existing approach [41] as shown in Fig. 1.

2. Related Work

3D Object Detection from Point Clouds aims to localize and classify 3D objects from point clouds, which is a challenging task due to the irregularity and sparsity of 3D point clouds. Some previous work [6, 23, 47] directly projects the irregular point clouds to 2D bird-view maps such that the task could be resolved by previous 2D detection methods. Another line of research [46, 53, 37, 15, 34] adopts 3D convolutional networks to learn 3D features from voxelized point clouds, and the extracted 3D feature volumes are also further compressed to bird-view feature maps as the above. Recently, point-based approaches [36, 49] propose to directly generate 3D proposals from raw point clouds by adopting PointNet++ [29] to extract point-wise features. There are also some other methods [28, 42] that utilize 2D images for generating 2D box proposals which are further employed to crop the object-level point clouds for generating 3D bounding boxes. In our work, we adopt SECOND [46] and PV-RCNN [34] as our 3D object detectors.

Unsupervised Domain Adaptation aims to generalize the model trained on source domain to unlabeled target domains. [26, 27] explore domain-invariant feature learning by minimizing Maximum Mean Discrepancy [1]. Inspired by GANs [14], adversarial learning was employed to align feature distributions across different domains on various 2D vision tasks [10, 17, 7, 32]. Besides, [16, 52] try to eliminate the domain gap on pixel-level by translating images. Other approaches [31, 55, 21, 5] utilize the self-training strategy to generate pseudo labels for unlabeled target domains. Saito *et al*. [33] adopt a two branch classifier to reduce the $\mathcal{H}\Delta\mathcal{H}$ discrepancy. [39, 9, 8] employ curriculum learning [2] and separate cases by their difficulties to realize local sample-level curriculum. Xu *et al*. [45] propose a progressive feature-norm enlarging method to reduce the domain gap. [25, 48] inject feature perturbations to obtain a robust classifier through adversarial training.

On par with the developments on domain adaptation for image recognition tasks, some recent works also aim to address the domain shift on point clouds for shape classification [30] and semantic segmentation [43, 50, 19]. However, despite of intensive studies on the 3D object detection task [53, 36, 46, 37, 49, 34], only very few approaches have been proposed to solve UDA for 3D object detection. Wang *et al*. propose SN [41] to normalize the object size of the source domain leveraging the object statistics of the target domain to close the size-level domain gap. Though the performance has been improved, the method needs the target statistics information, and its effectiveness depends on the source and target data distributions. In contrast, we propose a novel self-training pipeline for domain adaptive 3D object detection which achieves superior performance on all evaluated settings without target object statistics as a prior.

3. Method

3.1. Overview

Our goal is to adapt a 3D object detector trained on source labeled data $\{(P_i^s, L_i^s)\}_{i=1}^{n_s}$ of n_s samples to unlabeled target domain given target unlabeled data $\{P_i^t\}_{i=1}^{n_t}$ of n_t samples. Here, P_i^s and L_i^s represent the i-th source input point cloud and its corresponding label. L_i^s contains the category and 3D bounding box information for each object in the i-th point clouds, and each box is parameterized by its size (l, w, h), center (c_x, c_y, c_z), and heading angle θ. Similarly, P_i^t denotes the i-th unlabeled target point cloud.

In this section, we present ST3D, a self-training framework for adapting the 3D detector trained on source domain to target domain, which is shown in Fig. 2 and described in Algo. 1. Starting from pre-training a detector on source labeled data with random object scaling (ROS) (see Fig. 2 (a)), ST3D alternates between generating pseudo labels for target data via quality-aware triplet memory bank (QTMB)

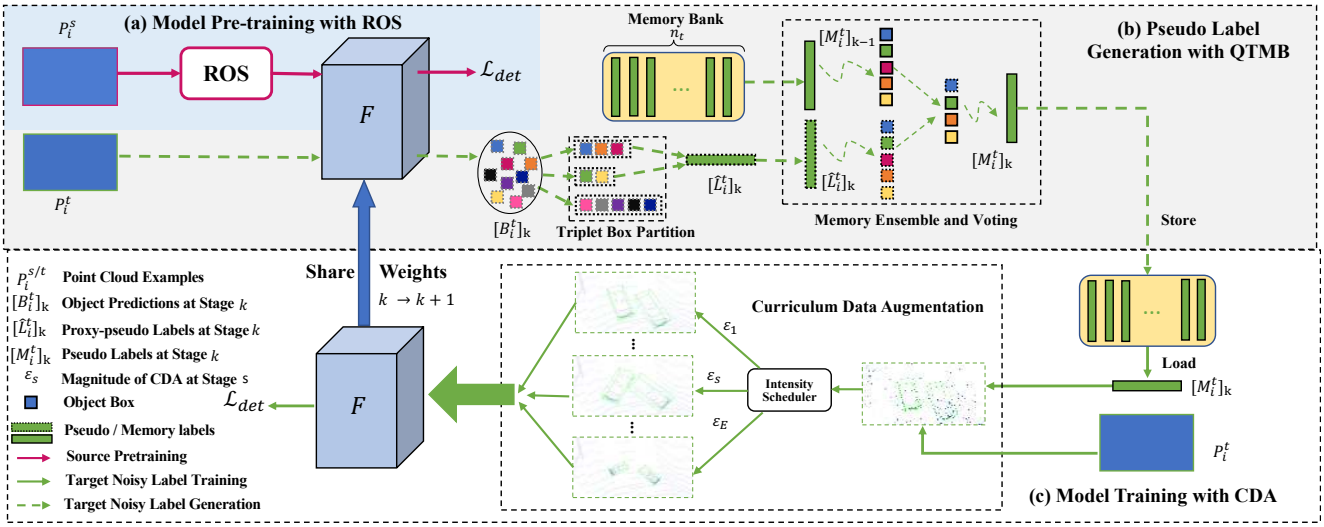

Figure 2. Our ST3D framework consists of three phases: (a) Pre-train the object detector F with ROS in source domain to mitigate object-size bias. (b) Generate high-quality and consistent pseudo labels on target unlabeled data with our QTMB. (c) Train model effectively on pseudo-labeled target data with CDA to progressively simulate hard examples. Best viewed in color.

Algorithm 1 Overview of our ST3D.

Require: Source domain labeled data $\{(P_i^s, L_i^s)\}_{i=1}^{n_s}$, and target domain unlabeled data $\{P_i^t\}_{i=1}^{n_t}$.

Output: The object detection model for target domain.

1: Pre-train the object detector on $\{(P_i^s, L_i^s)\}_{i=1}^{n_s}$ with ROS as detailed in Sec. 3.2.
2: Utilize the current model to generate raw object proposals $[B_i^t]_k$ for every sample P_i^t, where k is the current number of times for pseudo label generation.
3: Generate quality-aware pseudo labels $[\hat{L}_i^t]_k$ by triplet box partition given $[B_i^t]_k$ in Sec. 3.3.1.
4: Update the memory (*i.e.* pseudo labels) $[M_i^t]_k$ given pseudo labels $[\hat{L}_i^t]_k$ from the detection model and historical pseudo labels $[M_i^t]_{k-1}$ ($[M_i^t]_0 = \emptyset$) in the memory with memory ensemble-and-voting (MEV) as elaborated in Sec. 3.3.2. The memory bank $\{[M_i^t]_k\}_{i=1}^{n_t}$ contains the pseudo labels for all unlabeled examples.
5: Train the model on $\{P_i^t, [M_i^t]_k\}_{i=1}^{n_t}$ with CDA for several epochs as detailed in Sec. 3.4.
6: Go back to Line 2 until convergence.

(see Fig. 2 (b)) and training the detector with our curriculum data augmentation (CDA) (see Fig. 2 (c)) until convergence.

3.2. Model Pre-training with ROS

Our ST3D starts from training a 3D object detector on labeled source data $\{(P_i^s, L_i^s)\}_{i=1}^{n_s}$. The pre-trained model learns how to perform 3D detection on source labeled data and is further adopted to initialize object predictions for the target domain unlabeled data.

Motivation. However, despite of the useful knowledge, the pre-trained detector also learns the bias from the source data, such as object size and point densities due to domain shift. Among them, the bias in object size has direct negative impacts on 3D object detection, and results in incorrect size for pseudo-labeled target domain bounding boxes. This is also in line with the findings in [41]. To mitigate the issue, we propose a very simple yet effective per-object augmentation strategy, *i.e. random object scaling* (ROS), fully leveraging the high degree of freedom of 3D spaces.

Random Object Scaling. Given an annotated 3D bounding box with size (l, w, h), center (c_x, c_y, c_z) and heading angle θ, ROS scales the box in the length, width and height dimensions with random scale factors (r_l, r_w, r_h) through transforming the points inside the box. We denote the points inside the box as $\{p_i\}_{i=1}^{n_p}$ with a total of n_p points, and the coordinate of p_i is represented as (p_i^x, p_i^y, p_i^z). First, we transform the points to the local coordinate system of the box along its length, width and height dimensions via

$$(p_i^l, p_i^w, p_i^h) = (p_i^x - c_x, p_i^y - c_y, p_i^z - c_z) \cdot R,$$
$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

where $\cdot$ is matrix multiplication. Second, to derive the scaled object, the point coordinates inside the box are scaled to be $(r_l p_i^l, r_w p_i^w, r_h p_i^h)$ with object size $(r_l l, r_w w, r_h h)$. Third, to derive the augmented data $\{p_i^{\text{aug}}\}_{i=1}^{n_p}$, the points inside the scaled box are transformed back to the ego-car coordinate system and shifted to the center (c_x, c_y, c_z) as

$$p_i^{\text{aug}} = (r_l p_i^l, r_w p_i^w, r_h p_i^h) \cdot R^T + (c_x, c_y, c_z). \quad (2)$$

Albeit simple, ROS effectively simulates objects with diverse object sizes to address the size bias and hence facilitates to train size-robust detectors that produce more accurate initial pseudo boxes for subsequent self-training.

3.3. Pseudo label Generation with QTMB

With the trained detector, the next step is to generate pseudo labels for the unlabeled target data. Given the target sample P_i^t, the output B_i^t of the object detector is a group of predicted boxes containing category confidence scores, regressed box sizes, box centers and heading angles, where non-maximum-suppression (NMS) has already been conducted to remove the redundant boxes. For clarity, we call B_i^t as the object predictions for a scene.

Motivation. Different from classification and segmentation tasks, 3D object detection needs to jointly consider the classification and localization information, which poses great challenges for high-quality pseudo label generation. First, the confidence of object category prediction may not necessarily reflect the precision of location as shown by the blue line in Fig. 3 (a). Second, the fraction of false labels is much increased in confidence score intervals with medium values as illustrated in Fig. 3 (b). Third, model fluctuations induce inconsistent pseudo labels as demonstrated in Fig. 3 (c). The above factors will undoubtedly have negative impacts on the pseudo-labeled objects, leading to noisy supervisory information and instability for self-training.

To address the above challenges, we design *quality-aware triplet memory bank* (QTMB) to parse object predictions to pseudo labels for self-training. The memory bank at the k-th pseudo label generation stage, denoted as $\{[M_i^t]_k\}_{i=1}^{n_t}$, contains pseudo labels for all target domain data. $\{[M_i^t]_k\}_{i=1}^{n_t}$ is derived by combining pseudo labels $\{[\hat{L}_i^t]_k\}_{i=1}^{n_t}$ from the object detector and historical pseudo labels $\{[M_i^t]_{k-1}\}_{i=1}^{n_t}$ in the memory via ensemble and voting. Meanwhile, given the object predictions $\{B_i^t\}_{i=1}^{t}$ from the detector, $\{[\hat{L}_i^t]_k\}_{i=1}^{n_t}$ is constructed with an IoU-based scoring criterion to ensure the localization quality and a triplet box partition scheme to safely avoid assigning different labels to objects predictions with ambiguous confidence. To differentiate pseudo labels $\{[\hat{L}_i^t]_k\}_{i=1}^{n_t}$ from the object detector and pseudo labels $\{[M_i^t]_{k-1}\}_{i=1}^{n_t}$ in the memory, we call $\{[\hat{L}_i^t]_k\}_{i=1}^{n_t}$ "proxy-pseudo label" in what follows.

3.3.1 Proxy-pseudo Labels from the Object Detector

Firstly, to obtain high-quality and accurate proxy-pseudo labels $\{[\hat{L}_i^t]_k\}_{i=1}^{n_t}$ from the detection model, we introduce an IoU-based quality-aware criterion to directly assess the quality of the box, and a triplet box partition scheme to reduce noise from ambiguous objects predictions.

IoU-based Quality-aware Criterion for Scoring. To assess the localization quality of pseudo labels, we propose to augment the original object detection model with a lightweight IoU regression head. Specifically, given the feature derived from RoI pooling, we append two fully connected layers to directly predict the 3D box IoU between RoIs and their ground truths (GTs) or pseudo labels. A sigmoid function is adopted to map the output into range $[0, 1]$.

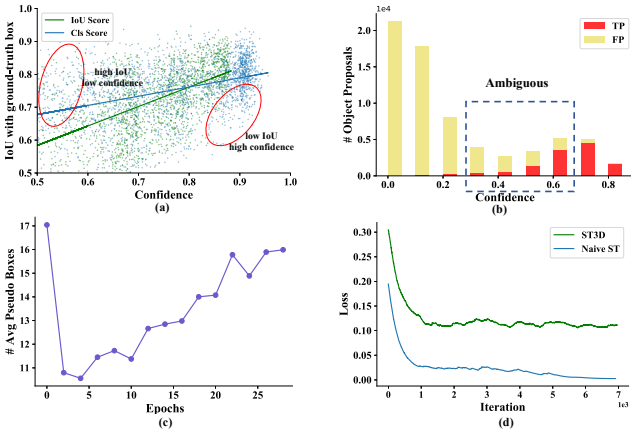

Figure 3. (a) Correlation between confidence value and box IoU with ground-truth (b) Lots of boxes with medium confidence may be assigned with ambiguous labels. (c) The average number of pseudo boxes fluctuates at different epochs. (d) Training loss curve comparison between naive ST and our ST3D with CDA.

During model training, the IoU branch is optimized by a binary cross entropy loss as

$$\mathcal{L}_{iou} = -\hat{u}\log u - (1-\hat{u})\log(1-u), \tag{3}$$

where u is the predicted IoU and $\hat{u}$ is the IoU between the ground truth (or pseudo label) box and the predicted 3D box. The correlation between the IoU score and localization quality (see green line in Fig. 3 (a)) is much increased in comparison with the classification confidence. Though IoU regression has been tried to improve supervised image object detection performance [18, 3], to the best of our knowledge, we are the first to demonstrate that it can serve as a good criterion to assess the quality of pseudo box for UDA self-training with encouraging results.

Triplet Box Partition to Avoid Ambiguous Samples. Now, we are equipped with a better IoU-based quality assessment criterion and object predictions $[B_i^t]_k$ (for the i-th sample at stage k) from the detector after NMS. Here, we present a triplet box partition scheme to obtain the proxy-pseudo labels $[\hat{L}_i^t]_k$ to avoid assigning labels to ambiguous examples. Given an object box b from $[B_i^t]_k$ with IoU prediction score u_b, we create a margin $[T_{\text{neg}}, T_{\text{pos}}]$ to ignore boxes with score u_b inside the margin, preventing them from contributing to training, as follows:

$$\text{state}_b = \begin{cases} \text{Positive (Store to } [\hat{L}_i^t]_k), & T_{\text{pos}} \le u_b, \\ \text{Ignored (Store to } [\hat{L}_i^t]_k), & T_{\text{neg}} \le u_b < T_{\text{pos}}, \\ \text{Negative (Discard)}, & u_b < T_{\text{neg}}. \end{cases} \tag{4}$$

If state_b is positive, b will be cached into $[\hat{L}_i^t]_k$ as a positive sample with its category label and pseudo box. Similarly, the ignored boxes will also be incorporated into the $[\hat{L}_i^t]_k$ to identify regions that should be ignored during model training due to its high uncertainty. Box b with negative state_b will be discarded, corresponding to backgrounds.

Our triplet box partition scheme reduces noisy pseudo labels from ambiguous boxes and ensures the quality of

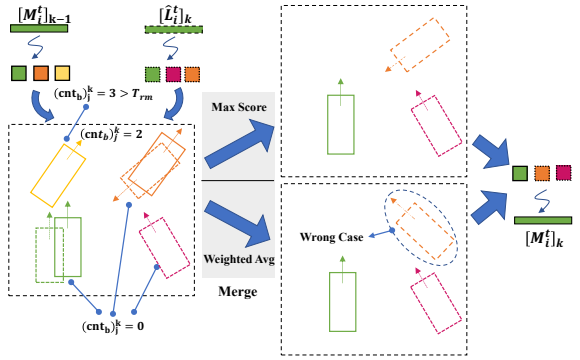

Figure 4. An instance of memory ensemble and voting (MEV). Given proxy-pseudo labels and historical memory labels, MEV automatically matches and merges boxes while ignoring or discarding successively unmatched boxes. The weighted average boxes merging strategy could produce wrong final box for boxes with very different heading angles.

pseudo-labeled boxes. To be noted, objects on the ignored regions may be evoked later if their scores are improved.

3.3.2 Memory Update and Pseudo Label Generation

Here, we combine proxy-pseudo labels $\{[\hat{L}_i^t]_k\}_{i=1}^{n_t}$ at stage k and the historical pseudo labels $\{[M_i^t]_{k-1}\}_{i=1}^{n_t}$ ($[M_i^t]_0 = \emptyset$) in the memory bank via memory ensemble and voting. The outputs are the updated pseudo labels $\{[M_i^t]_k\}_{i=1}^{n_t}$ that also serve as the labels for the subsequent model training. During this memory update process, each pseudo box b from $[\hat{L}_i^t]_k$ and $[M_i^t]_{k-1}$ has three attributes $(u_b, \text{state}_b, \text{cnt}_b)$, which are the confidence score, state (positive or ignored) and an unmatched memory counter (UMC) (for memory voting), respectively. We assume that $[\hat{L}_i^t]_k$ contains n_l boxes denoted as $[\hat{L}_i^t]_k = \{(u_l, \text{state}_l, \text{cnt}_l)_j^k\}_{j=1}^{n_l}$ and $[M_i^t]_{k-1}$ has n_m boxes represented as $[M_i^t]_{k-1} = \{(u_m, \text{state}_m, \text{cnt}_m)_j^{k-1}\}_{j=1}^{n_m}$.

Memory Ensemble. Instead of directly replacing $[M_i^t]_{k-1}$ with the latest proxy-pseudo labels $[\hat{L}_i^t]_k$, we propose the memory ensemble operation to combine $[M_i^t]_{k-1}$ and $[\hat{L}_i^t]_k$ to produce more consistent and high-quality pseudo labels.

The memory ensemble operation matches two object boxes with similar locations, sizes and angles from $[M_i^t]_{k-1}$ and $[\hat{L}_i^t]_k$, and merges them to produce a new object box. By default, we adopt the consistency ensemble strategy for box matching. Specifically, it calculates the pair-wise 3D IoU matrix $A = \{a_{jv}\} \in \mathbb{R}^{n_m \times n_l}$ between each box in $[M_i^t]_{k-1}$ and each box in $[\hat{L}_i^t]_k$. For the j-th object box in $[M_i^t]_{k-1}$, its matched box index $\hat{j}$ in $[\hat{L}_i^t]_k$ is derived by,

$$\hat{j} = \text{argmax}_j\ (a_{jv}),\ v = 1, \cdots, n_l. \tag{5}$$

Note that if $a_{j\hat{j}} < 0.1$, we denote each of these two paired boxes as unmatched boxes that will be further processed by the memory voting operation.

We assume the successfully matched pair-wise object boxes as $(u_l, \text{state}_l, \text{cnt}_l)_j^k$ and $(u_m, \text{state}_m, \text{cnt}_m)_j^{k-1}$. They are further merged to cache the pseudo labeled box with a higher confidence value into the $[M_i^t]_k$ and update its corresponding attributes as

$$(u_m, \text{state}_m, 0)_j^k = \begin{cases} (u_l, \text{state}_l, \text{cnt}_l)_j^k, & \text{if } u_m \le u_l, \\ (u_m, \text{state}_m, \text{cnt}_m)_j^{k-1}, & \text{otherwise}, \end{cases} \tag{6}$$

Here, we adopt to choose box instead of a weighted combination is because weighted combination has the potential to produce an unreasonable final box if the matched boxes have very different heading angles (see Fig. 4 "wrong case"). We also explore two alternative strategies for box matching, which are discussed in Sec. 4.3.

Memory Voting. The memory ensemble operation can effectively select better matched pseudo boxes. However, it cannot handle the unmatched pseudo boxes from either $[M_i^t]_{k-1}$ or $[\hat{L}_i^t]_k$. As the unmatched boxes often contain both false positive boxes and high-quality true positive boxes, either caching them into the memory or discarding them all is suboptimal. To address the above problem, we propose a novel memory voting approach, which leverages history information of unmatched object boxes to robustly determine their status (**cache**, **discard** or **ignore**). For the j-th unmatched pseudo boxes b from $[M_i^t]_{k-1}$ or $[\hat{L}_i^t]_k$, its UMC $(\text{cnt}_b)_j^k$ will be updated as follows:

$$(\text{cnt}_b)_j^k = \begin{cases} 0 & , \text{ if } b \in [\hat{L}_i^t]_k, \\ (\text{cnt}_b)_j^{k-1} + 1 & , \text{ if } b \in [M_i^t]_{k-1}, \end{cases} \tag{7}$$

We update the UMC for unmatched boxes in $[M_i^t]_{k-1}$ by adding 1 and initialize the UMC of the newly generated boxes in $[\hat{L}_i^t]_k$ as 0. The UMC records the successive unmatched times of a box, which are combined with two thresholds T_{ign} and T_{rm} ($T_{\text{ign}} = 2$ and $T_{\text{rm}} = 3$ by default) to select the subsequent operation for unmatched boxes as

$$\begin{cases} \text{Discard} & , \quad (\text{cnt}_b)_j^k \ge T_{\text{rm}}, \\ \text{Ignore (Store to} [M_i^t]_k) & , \quad T_{\text{ign}} \le (\text{cnt}_b)_j^k < T_{\text{rm}}, \\ \text{Cache (Store to} [M_i^t]_k) & , \quad (\text{cnt}_b)_j^k < T_{\text{ign}}. \end{cases} \tag{8}$$

Benefited from our memory voting, we could generate more robust and consistent pseudo boxes by caching the occasionally unmatched box in the memory bank.

3.4. Model training with CDA

Our proposed QTMB can produce consistent and stable pseudo labels $[M_i^t]_k$ for the i-th point clouds. Now, the detection model can be trained on $\{P_i^t, [M_i^t]_k\}_{i=1}^{n_t}$ at stage k as described in Algo. 1 (Line 5).

Motivation. However, our observations show that most of positive pseudo boxes are easy examples since they are generated from previous high-confident object predictions. Consequently, during training, model is prone to overfitting to these easy examples with low loss values (see Fig. 3 (d)), unable to further mine hard examples to improve the detector [2]. To prevent model from being trapped by bad

local minimal, strong data augmentations could be an alternative to generate diverse and potentially hard examples to improve the model. However, this might confuse the learner and hence be harmful to model training at the initial stage.

Curriculum Data Augmentation. Motivated by the above observation, we design a curriculum data augmentation (CDA) strategy to progressively increase the intensity ϵ of data augmentation and gradually generate increasingly harder examples to facilitate improving the model and ensure effective learning at the early stages.

To progressively increase the intensity ϵ of data augmentations $\{D_i\}_{i=1}^{n_d}$ with n_d types (*i.e.* world coordinate system transformation and per-object coordinate system transformation), we design a multi-step intensity scheduler with initial intensity ϵ_0^i for the i-th data augmentation. Specifically, we split the total training epochs into E stages. After each stage, the data augmentation intensity is multiplied by an enlarging ratio α ($\alpha > 1$, we use $\alpha = 1.2$ by default). Thus, the data augmentation intensity for i-th data augmentation at stage s ($1 \leq s \leq E$) is derived as $\epsilon_s^i = \epsilon_0^i \alpha^{s-1}$. Hence, the random sampling range of the i-th data augmentation could be calculated as follows:

$$\begin{cases} [-\epsilon_s^i, \epsilon_s^i] & , \quad \text{if } D_i \text{ belongs to rotation,} \\ [1-\epsilon_s^i, 1+\epsilon_s^i] & , \quad \text{if } D_i \text{ belongs to scaling.} \end{cases} \tag{9}$$

CDA enables the model to learn from the challenging samples while making the difficulty of examples be within the capability of the learner during the whole training process.

4. Experiments

4.1. Experimental Setup

Datasets. We conduct experiments on four widely used autonomous driving datasets: KITTI [13], Waymo [38], nuSenses [4], and Lyft [20]. Our experiments lie in two aspects: Adaptation from label rich domains to label insufficient domains (i.e., Waymo to other datasets) and across domains with different number of the LiDAR beams (i.e., Waymo $\rightarrow$ nuScenes and nuScenes $\rightarrow$ KITTI).

Comparison Methods. We compare ST3D with three methods: (i) **Source Only** indicates directly evaluating the source domain pre-trained model on the target domain. (ii) **SN** [41] is the SOTA domain adaptation method on 3D object detection with target domain statistical object size as extra information. (iii) **Oracle** indicates the fully supervised model trained on the target domain.

Evaluation Metric. We follow [41] and adopt the KITTI evaluation metric for evaluating our methods on the commonly used car category (also named vehicle in the Waymo Open Dataset). We evaluate all settings on ring view point clouds since it is more useful in real-world applications, except for the KITTI dataset which only provides the annotations in the front view. We follow the official KITTI evaluation metric and report the average precision (AP) over 40 recall positions, and the IoU thresholds are 0.7 for both the bird's eye view (BEV) IoUs and 3D IoUs. To further demonstrate the effectiveness of different methods for adaptation, we also report how much the performance gap between Source Only to Oracle is closed, which is represented as **closed gap** $= \frac{\text{AP}_{\text{model}} - \text{AP}_{\text{source only}}}{\text{AP}_{\text{oracle}} - \text{AP}_{\text{source only}}} \times 100\%$.

Implementation Details. We validate our proposed ST3D on two detection backbones SECOND [46] and PV-RCNN [34]. Specifically, we improve the SECOND detector with an extra IoU head to estimate the IoU between the object proposals and their GTs, and name this detector as SECOND-IoU. We adopt the training settings of the popular point cloud detection codebase OpenPCDet [40] to pre-train our detectors on the source domain with our proposed random object scaling (ROS) data augmentation strategy. For the following target domain self-training stage, we use Adam [22] with learning rate 1.5×10^{-3} and one cycle scheduler to finetune the detectors for 30 epochs with curriculum data augmentation (CDA). We update the pseudo label with QTMB after every 2 epochs. For all the above datasets, the detection range is set to $[-75.2, 75.2]m$ for X and Y axes, and $[-2, 4]m$ for Z axis (the origins of coordinates of different datasets have been shifted to the ground plane). We set the voxel size of both SECOND-IoU and PV-RCNN to $(0.1m, 0.1m, 0.15m)$ on all datasets.

During both the pre-training and self-training processes, we adopt the widely adopted data augmentation, including random flipping, random world scaling, random world rotation, random object scaling and random object rotation. CDA is utilized in the self-training process to provide proper hard examples for promoting the training process.

4.2. Main results and Comparison with SOTA

Main results of our ST3D. As shown in Table 1, we compare the performance of our ST3D with Source Only, SN [41] and Oracle. Since SN employs extra statistical supervision on the target domain, we compare our method with other approaches in terms of two settings, the Unsupervised DA (UDA) and Weakly-supervised DA setting (with target domain size statistics).

For the UDA setting, our method outperforms the Source Only baseline on all evaluated UDA settings. Specifically, without leveraging the target domain size statistics, we improve the performance on Waymo $\rightarrow$ KITTI and nuScenes $\rightarrow$ KITTI tasks by a large margin of around 34% $\sim$ 43% in AP_{3D}, which largely closes the performance gap between Source Only and Oracle. Furthermore, when transferring Waymo models to other domains that have full ring view annotations for evaluation (*i.e.*, Waymo $\rightarrow$ nuSenses and Waymo $\rightarrow$ Lyft [1]), our ST3D also attains a considerable performance gain which closes the Oracle and Source Only performance gap by up to 33.93% on SECOND-IoU and 15.20% on PV-RCNN. These encouraging results validate

[1]Lyft dataset is constructed with different label rules from the other 3 datasets which enlarges the domain gaps and we will detail this in the supplementary materials

Task	Method	SECOND-IoU		PV-RCNN	
		AP_{BEV} / AP_{3D}	Closed Gap	AP_{BEV} / AP_{3D}	Closed Gap
Waymo → KITTI	Source Only	67.64 / 27.48	-	61.18 / 22.01	-
	SN [41]	78.96 / 59.20	+72.33% / +69.00%	79.78 / 63.60	+66.91% / +68.76%
	ST3D	82.19 / 61.83	+92.97% / +74.72%	84.10 / 64.78	+82.45% / +70.71%
	ST3D (w/ SN)	**85.83 / 73.37**	+116.23% / +99.83%	**86.65 / 76.86**	+91.62% / +90.68%
	Oracle	83.29 / 73.45	-	88.98 / 82.50	-
Waymo → Lyft	Source Only	72.92 / 54.34	-	75.49 / 58.53	-
	SN [41]	72.33 / 54.34	-05.11% / +00.00%	72.82 / 56.64	-24.34% / -14.36%
	ST3D	76.32 / **59.24**	+29.44% / +33.93%	**77.68 / 60.53**	+19.96% / +15.20%
	ST3D (w/ SN)	**76.35** / 57.99	+15.71% / +17.81%	74.95 / 58.54	-04.92% / +00.08%
	Oracle	84.47 / 68.78	-	86.46 / 71.69	-
Waymo → nuScenes	Source Only	32.91 / 17.24	-	34.50 / 21.47	-
	SN [41]	33.23 / 18.57	+01.69% / +07.54%	34.22 / 22.29	-01.50% / +04.80%
	ST3D	**35.92** / 20.19	+15.87% / +16.73%	36.42 / 22.99	+10.32% / +08.89%
	ST3D (w/ SN)	35.89 / **20.38**	+15.71% / +17.81%	**36.62 / 23.67**	+11.39% / +12.87%
	Oracle	51.88 / 34.87	-	53.11 / 38.56	-
nuScenes → KITTI	Source Only	51.84 / 17.92	-	68.15 / 37.17	-
	SN [41]	40.03 / 21.23	-37.55% / +05.96%	60.48 / 49.47	-36.82% / +27.13%
	ST3D	75.94 / 54.13	+76.63% / +59.50%	78.36 / 70.85	+49.02% / +74.30%
	ST3D (w/ SN)	**79.02 / 62.55**	+86.42% / +80.37%	**84.29 / 72.94**	+77.48% / +78.91%
	Oracle	83.29 / 73.45	-	88.98 / 82.50	-

Table 1. Result of different adaptation tasks. We report AP_{BEV} and AP_{3D} of the car category at IoU = 0.7 as well as the domain gap closed by various approaches along Source Only and Oracle. The reported AP is moderate case for the adaptation tasks for to KITTI tasks, and is the overall result for other adaptation tasks. We indicate the best adaptation result by **bold**.

that our method can effectively adapt 3D object detectors trained on the source domain to the target domain and perform generally well on different detection architectures.

For the weakly-supervised DA setting, we equip our ST3D with the SN [41] (denoted as ST3D (w/SN)) to obtain the pre-trained detector. We observe that our ST3D approach and SN can work collaboratively to further boost the performance on Waymo → KITTI where ST3D improves SN by 14% (SECOND-IoU) and 13% (PV-RCNN) in AP_{3D}. Notably, our ST3D (w/ SN) performs on par with the fully supervised 3D detector on this setting as shown in Table 1. Moreover, our approach with SECOND-IoU obtains over 40% AP_{3D} improvement on the nuScenes → KITTI setting compared with SN. For Waymo → nuScenes and Waymo → Lyft tasks, despite performance gains are still obtained compared to SN, only minor performance gains or even performance degradation are observed compared to our UDA setting ST3D due to the minor domain shifts in object size. In contrast, our ST3D still demonstrates consistent improvements on these settings.

We also observe that it is hard to adapt detectors from the point clouds with more LiDAR beams (*e.g.* Waymo) to the point clouds with fewer LiDAR beams (*e.g.* NuScenes), while the opposite adaptation is relatively easy as shown in Table 1 nuScenes → KITTI. It demonstrates that the point density of target domain is more important than the point density of source domain, and our ST3D could effectively improve the performance on target domain even with a relatively worse pre-trained detector on source domain.

Method	AP_{BEV} / AP_{3D}
(a) Source Only	67.64 / 27.48
(b) Random Object Scaling (ROS)	78.07 / 54.67
(c) SN	78.96 / 59.20
(d) ST3D (w/o ROS)	75.54 / 34.76
(e) ST3D (w/ ROS)	82.19 / 61.83
(f) ST3D (w/ SN)	**85.83 / 73.37**

Table 2. Effectiveness analysis of Random Object Scaling.

4.3. Ablation Studies

In this section, we conduct extensive ablation experiments to investigate the individual components of our ST3D. All experiments are conducted with the 3D detector SECOND-IoU on the task of Waymo → KITTI.

Random Object Scaling. As mentioned in Sec. 3.2, by employing our random object scaling for pre-training, the detectors could be more robust to the variations of object size in different domains. Table 2 (a), (b), (c) show that our unsupervised ROS improves the performance by around 27.2% in AP_{3D} and is only 4.5% lower than the weakly-supervised SN method. Furthermore, as shown in Table 2 (d), (e), the ROS pre-trained model also greatly benefits the subsequent self-training process. We also observe that there still exists a gap between the performance of ST3D (w/ ROS) and ST3D (w/ SN) in AP_{3D}, potentially due to that the KITTI dataset has a larger domain gap over object size compared with other datasets, and in this situation, the weakly supervised SN could provide more accurate object size information than our fully unsupervised ROS.

Method	AP_{BEV} / AP_{3D}
SN (baseline)	78.96 / 59.20
ST (w/ SN)	79.74 / 65.88
ST (w/ SN) + Triplet	79.81 / 67.39
ST (w/ SN) + Triplet + QAC	83.76 / 70.64
ST (w/ SN) + Triplet + QAC + MEV-C	85.35 / 72.52
ST (w/ SN) + Triplet + QAC + MEV-C + CDA	**85.83 / 73.37**

Table 3. Component ablation studies. **ST** represents naive self-training. **Triplet** means the triplet box partition. **QAC** indicates the quality-aware criterion. **MEV-C** is consistency memory ensemble-and-voting. **CDA** means curriculum data augmentation.

T_{neg}	T_{pos}	AP_{BEV} / AP_{3D}	T_{neg}	T_{pos}	AP_{BEV} / AP_{3D}
0.20	0.60	**86.44** / 72.23	0.25	0.25	83.06 / 67.97
0.25	0.60	85.83 / **73.37**	0.25	0.30	83.21 / 69.51
0.30	0.60	85.30 / 72.73	0.25	0.40	83.69 / 69.98
0.40	0.60	84.59 / 72.25	0.25	0.50	84.30 / 70.17
0.50	0.60	84.96 / 72.11	0.25	0.60	**85.83 / 73.37**
0.60	0.60	83.66 / 70.10	0.25	0.70	76.81 / 66.23

Table 4. Sensitivity analysis for $[T_{neg}, T_{pos}]$ of triplet box partition.

Method	Memory Voting	Merge	AP_{BEV} / AP_{3D}
ST3D (w/ ME-N)	√	Max	**85.93** / 73.17
ST3D (w/ ME-B)	√	Max	85.65 / 72.37
ST3D (w/ ME-C)	√	Max	85.83 / **73.37**
	√	Avg	84.08 / 72.07
	×	Max	84.23 / 70.86
	×	Avg	83.92 / 70.96

Table 5. Ablation study of memory ensemble (different variants and merge strategies for matched boxes) and memory voting. We denote three memory ensemble variants: consistency, NMS and bipartite as ME-C, ME-N, ME-B separately.

Method	World	Object	Intensity	AP_{BEV} / AP_{3D}
ST3D	×	×	-	83.31 / 66.73
	√	×	Normal	84.47 / 70.60
	×	√	Normal	81.81 / 67.91
	√	√	Normal	85.35 / 72.52
	√	√	Strong	84.84 / 72.23
	√	√	Curriculum	**85.83 / 73.37**

Table 6. Analysis of data augmentation type and intensity.

Component Analysis in Self-training. As demonstrated in Table 3, we investigate the effectiveness of our individual components. Our ST3D (last line) outperforms the SN baseline and naive ST (w/ SN) by around 14.2% and 7.5% in AP_{3D}. Specifically, on the pseudo label generation stage, Triplet box partition and quality-aware IoU criterion provide around 1.5% and 3.3% performance gains on AP_{3D}, respectively. MEV-C and CDA separately further yield around 1.9% and 0.9% improvements, respectively.

Sensitivity Analysis of Triplet Box Partition. In this part, we investigate the importance of the ignore margin $[T_{pos}, T_{neg}]$ for our triplet box partition. As shown in Table 4, without triplet box partition (i.e., $T_{pos} = T_{neg}$), our ST3D drops by 3.3% and 5.4% for $T_{pos} = T_{neg} = 0.6$ and 0.25 respectively. Furthermore, our method is more sensitive to T_{pos} than T_{neg}. Lower T_{pos} could introduce excessive noisy labels while higher T_{pos} gives rise to a small number of positive examples that harm the self-training process.

Analysis of Memory Ensemble and Voting. As shown in Table 5, we further investigate the memory ensemble and memory voting schemes for updating memory bank and generating pseudo labels. On the one hand, we propose the other two memory ensemble strategies including NMS ensemble and bipartite ensemble, which use NMS and bipartite matching separately. For the comparison of different memory ensemble variants, ME-N and ME-C achieve similar performance and outperform 0.8% $\sim$ 1% than ME-B in terms of 3D AP. For the paired box merging strategy in the memory ensemble stage, we compare two merging approaches max score and weighted average, where max score obtains a 1.3% performance gain than weighted average. This validates our analysis in Sec. 3.3.2 that the weighted average strategy may generate inappropriate pseudo labels when matched boxes have very different heading angles.

On the other hand, without memory voting, the performance drops over 2.4% since the unmatched boxes along different memories could not be well handled. Our memory voting strategy could robustly mine high-quality boxes and discard low-quality boxes.

Data Augmentation Analysis. As shown in Table 6, we also investigate the effects of data augmentation in the self-training pipeline, where both the type (world-level and object-level) and the intensity of augmentation are explored. We observe that without any data augmentation, ST3D suffers from over 6.6% performance degradation. Both world-level and object-level augmentation provide improvements and their combination can further boost the performance. When it comes to the intensity of data augmentation, compared to the normal intensity, stronger data augmentation magnitude confuses the deep learner and slightly drops performance while our CDA can bring around 0.9% gains.

5. Conclusion

We have presented ST3D – a redesigned self-training pipeline – for unsupervised domain adaptive 3D object detection from point clouds. ST3D involves random object scaling, a quality-aware triplet memory bank, and curriculum data augmentation to address fundamental challenges stemming from the self-training on 3D object detection. Experiments demonstrate that ST3D substantially advance the state of the art. Our future work will be to extend our method to other UDA tasks on image and video data.

Acknowledgement

This work has been supported in part by HKU Start-up Fund, HKU Seed Fund for Basic Research, Centre for Perceptual and Interactive Intelligence Limited, the General Research Fund through the Research Grants Council of Hong Kong under Grants (Nos. 14208417 and 14207319), and CUHK Strategic Fund.

References

[1] Shai Ben-David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. In *International Conference on Artificial Intelligence and Statistics*, pages 129–136, 2010. 2

[2] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual International Conference on Machine Learning*, pages 41–48, 2009. 2, 5

[3] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact++: Better real-time instance segmentation. *arXiv preprint arXiv:1912.06218*, 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 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. 1, 2

[8] Kashyap Chitta, Jianwei Feng, and Martial Hebert. Adaptive semantic segmentation with a strategic curriculum of proxy labels. *arXiv preprint arXiv:1811.03542*, 2018. 2

[9] Jaehoon Choi, Minki Jeong, Taekyung Kim, and Changick Kim. Pseudo-labeling curriculum for unsupervised domain adaptation. *arXiv preprint arXiv:1908.00262*, 2019. 2

[10] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015. 1, 2

[11] Yixiao Ge, Dapeng Chen, and Hongsheng Li. Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification. In *International Conference on Learning Representations*, 2019. 1

[12] Yixiao Ge, Dapeng Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Self-paced contrastive learning with hybrid memory for domain adaptive object re-id. *arXiv preprint arXiv:2006.02713*, 2020. 1

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition*, 2012. 1, 2, 6

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 2

[15] Chenhang He, Hui Zeng, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Structure aware single-stage 3d object detection from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11873–11882, 2020. 2

[16] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017. 2

[17] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016. 1, 2

[18] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6409–6418, 2019. 4

[19] Maximilian Jaritz, Tuan-Hung Vu, Raoul de Charette, Emilie Wirbel, and Patrick Pérez. xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12605–12614, 2020. 2

[20] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet. Lyft level 5 perception dataset 2020. https://level5.lyft.com/dataset/, 2019. 2, 6

[21] Mehran Khodabandeh, Arash Vahdat, Mani Ranjbar, and William G Macready. A robust learning approach to domain adaptive object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 480–490, 2019. 1, 2

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[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 Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019. 1

[25] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In *International Conference on Machine Learning*, pages 4013–4022, 2019. 2

[26] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pages 97–105, 2015. 1, 2

[27] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adapta-

tion. In *Advances in Neural Information Processing Systems*, pages 1647–1657, 2018. 2

[28] 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

[29] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017. 2

[30] Can Qin, Haoxuan You, Lichen Wang, C-C Jay Kuo, and Yun Fu. Pointdan: A multi-scale 3d domain adaption network for point cloud representation. In *Advances in Neural Information Processing Systems*, pages 7192–7203, 2019. 2

[31] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 2988–2997. JMLR. org, 2017. 2

[32] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6956–6965, 2019. 1, 2

[33] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018. 2

[34] 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

[35] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. *arXiv preprint arXiv:2102.00463*, 2021. 1

[36] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE 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. *arXiv preprint arXiv:1907.03670*, 2019. 1, 2

[38] 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. 1, 2, 6

[39] James S Supancic and Deva Ramanan. Self-paced learning for long-term tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2379–2386, 2013. 2

[40] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020. 6

[41] 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, 2, 3, 6, 7

[42] Zhixin Wang and Kui Jia. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. *arXiv preprint arXiv:1903.01864*, 2019. 2

[43] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *2018 IEEE International Conference on Robotics and Automation*, pages 1887–1893. IEEE, 2018. 2

[44] 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. 2

[45] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *The IEEE International Conference on Computer Vision*, October 2019. 2

[46] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 1, 2, 6

[47] 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

[48] Jihan Yang, Ruijia Xu, Ruiyu Li, Xiaojuan Qi, Xiaoyong Shen, Guanbin Li, and Liang Lin. An adversarial perturbation oriented domain adaptation approach for semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12613–12620, 2020. 2

[49] 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 International Conference on Computer Vision*, pages 1951–1960, 2019. 2

[50] Li Yi, Boqing Gong, and Thomas Funkhouser. Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds. *arXiv preprint arXiv:2007.08488*, 2020. 2

[51] Yabin Zhang, Bin Deng, Kui Jia, and Lei Zhang. Label propagation with augmented anchors: A simple semi-supervised learning baseline for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 781–797. Springer, 2020. 1

[52] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Fully convolutional adaptation networks for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6810–6818, 2018. 2

[53] 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. 2

[54] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5982–5991, 2019. 1

[55] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *European Conference on Computer Vision*, pages 289–305, 2018. 2