

Pseudo Label Refinery for Unsupervised Domain Adaptation on Cross-dataset 3D Object Detection

Zhanwei Zhang^{1,4*} Minghao Chen^{3*} Shuai Xiao⁵ Liang Peng⁴ Hengjia Li¹
Binbin Lin^{2,6†} Ping Li³ Wenxiao Wang² Boxi Wu² Deng Cai¹

¹State Key Lab of CAD&CG, Zhejiang University

²School of Software Technology, Zhejiang University

³School of Computer Science and Technology, Hangzhou Dianzi University

⁴Fabu Inc. ⁵Alibaba Group ⁶Fullong Inc.

Abstract

Recent self-training techniques have shown notable improvements in unsupervised domain adaptation for 3D object detection (3D UDA). These techniques typically select pseudo labels, i.e., 3D boxes, to supervise models for the target domain. However, this selection process inevitably introduces unreliable 3D boxes, in which 3D points cannot be definitively assigned as foreground or background. Previous techniques mitigate this by reweighting these boxes as pseudo labels, but these boxes can still poison the training process. To resolve this problem, in this paper, we propose a novel pseudo label refinery framework. Specifically, in the selection process, to improve the reliability of pseudo boxes, we propose a complementary augmentation strategy. This strategy involves either removing all points within an unreliable box or replacing it with a high-confidence box. Moreover, the point numbers of instances in high-beam datasets are considerably higher than those in low-beam datasets, also degrading the quality of pseudo labels during the training process. We alleviate this issue by generating additional proposals and aligning RoI features across different domains. Experimental results demonstrate that our method effectively enhances the quality of pseudo labels and consistently surpasses the state-of-the-art methods on six autonomous driving benchmarks. Code will be available at <https://github.com/Zhanwei-Z/PERE>.

1. Introduction

Three dimension LIDAR-based detection has prominent significance in perceiving objects in 3D scenarios. This task is driven by the availability of the large-scale annotated data

*Equal contribution. This work was done when Zhanwei Zhang was an intern at Fabu Inc.

†Corresponding author

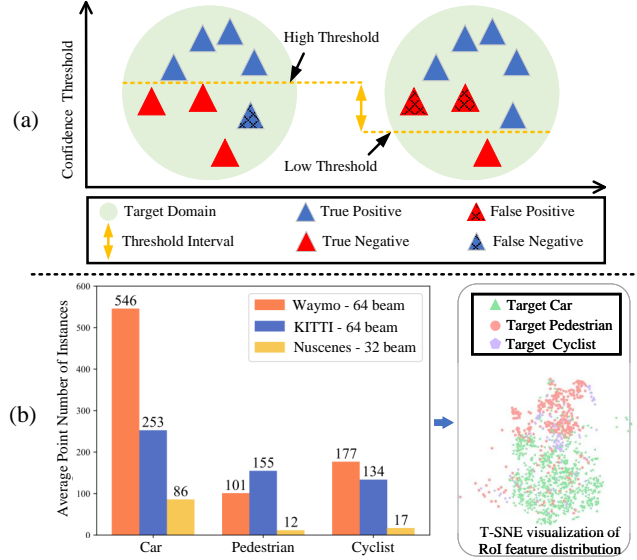


Figure 1. Self-training methods generally consist of the selection and the training process. (a) In the selection process, setting the threshold whether high or low would lead to inevitable false negatives or false positives during the threshold interval. (b) In the training process, the point numbers of instances in high-beam datasets are markedly higher than those in low-beam datasets, which causes RoI feature confusion across different categories.

[2, 8, 32] and the advancements in deep neural networks [24, 29, 41]. However, due to cross-dataset domain discrepancies, models trained on the labeled source domain often have notorious generalization performance when applied to the unlabeled target domain [38]. Consequently, 3D UDA has emerged as a hot topic [39, 42, 43, 45].

Recently, several self-training methods [4, 11, 20, 22, 42, 43, 45, 47, 48, 49] have achieved state-of-the-art (SOTA) performance in the field of 3D UDA. These methods commonly involve pre-training models in the source domain,

followed by two iterative processes: 1) the selection process: a predetermined confidence threshold is employed to select qualified 3D pseudo boxes. 2) the training process: the selected pseudo boxes are utilized to train the target domain and then update pseudo labels.

However, during the selection process, as shown in Fig. 1(a), setting the threshold, whether high or low, would induce inevitable false negatives or false positives within the threshold interval [42]. In other words, these boxes within the interval are unreliable, *inside which 3D points cannot be definitively assigned as foreground or background points by the threshold*. To address the unreliable boxes, CL3D [22] reweights them by soft-selection, while ST3D [42] uses a voting strategy to select a portion of them. However, they essentially still exploit unreliable boxes to supervise the target domain during the training process. Another naive solution is to remove all unreliable boxes along with their internal points directly. However, this method would misclassify such points as background points during testing, thereby trapping the model in local minima.

In this paper, we propose a novel **P**seudo **l**abel **R**efin**E**ry framework, named PERE, to enhance the reliability of pseudo boxes. During the selection process, this framework adopts a complementary augmentation strategy, leveraging the editability of point clouds [41]. Specifically, given an unreliable 3D box b , rather than merely removing all points within it, our strategy probabilistically replaces b and its contained points with a high-confidence box and associated points. This replacement ensures that the points within the newly integrated, reliable box are utilized as effective foreground points, preventing the detector F from getting stuck in local minima. After augmentation, the unreliable box b is excluded from the subsequent training process.

Moreover, during the training process, there exists the *cross-dataset Instance-level Point Number Inconsistency* (IPNI), which also worsens the quality of pseudo labels. Specifically, as shown in Fig. 1(b), the average point number of instances from each category in the 64-beam datasets is significantly higher than that in the 32-beam datasets. Notably, the average point numbers of instances from pedestrians and cyclists in Waymo even surpass that of cars in NuScenes. In the target domain, IPNI potentially causes the proposals to inaccurately cluster around the regions with similar point numbers as the source domain instances. As a result, the RoI features derived from these imprecise proposals, regardless of their categories, are confused together as shown in Fig. 1(b). LD [39] and DTS [11] primarily focus on addressing the general point density inconsistency by either downsampling or upsampling to alter the point beams. However, downsampling inevitably leads to information loss, while upsampling would introduce unrealistic points, thereby compromising the data credibility.

Since directly altering the point beams is suboptimal, we

address the issue of IPNI from two perspectives. 1) To mitigate the adverse impact of IPNI on proposals, we propose the interpolation and extrapolation operations on proposals. Specifically, interpolation exploits the ensemble of existing proposals, while extrapolation pushes the detection boundary toward regions with sparse point clouds. Both operations aim to generate extra proposals that are not exclusively focused on regions with similar point numbers as source instances. 2) To dilute the confusion of RoI features caused by IPNI, we align cross-domain RoI features of the same category by reformulating the intra-domain and the inter-domain triplet losses for the field of 3D UDA.

Our contributions can be summarized as follows:

- We propose a pseudo label refinery framework (PERE) that is specifically designed for cross-dataset 3D UDA.
- To enhance the reliability of pseudo labels, we develop a complementary augmentation, which either removes all points within an unreliable box or replaces them with a high-confidence box and associated points.
- To further boost the quality of pseudo labels, we alleviate the negative impact of IPNI by generating additional proposals and aligning cross-domain RoI features.
- Extensive experimental results on multi-category object detection tasks across Waymo [32], nuScenes [2] and KITTI [8] datasets validate the effectiveness of PERE.

2. Related Work

2.1. LiDAR-based 3D Object Detection

LiDAR-based 3D detectors concentrate on the challenging task of detecting objects within disorderly and sparse 3D point clouds. These detectors can be broadly categorized into two groups: point-based methods and voxel-based methods. Point-based methods [30, 31, 44] involve feeding raw points into neural networks and commonly employ PointNet or PointNet++ [23, 24] to extract point-wise features from the original geometric attributes. On the other hand, voxel-based methods [14, 29, 41, 46, 52] convert point clouds into regular voxels. These methods then utilize convolutional networks to learn feature representations. Another branch of voxel-based approaches process point clouds into various 2D views, such as the bird-eye view [15] and the range view [6, 35]. However, as revealed in [38], due to the domain gaps, few detectors can generalize well when directly applied to the target domain.

2.2. Unsupervised Domain Adaptation for 2D Tasks

In recent years, UDA has been extensively studied in various 2D tasks [10, 25, 26]. One line of research exploits Generative Adversarial Networks (GANs) [9] to align feature distributions across domains [1, 3, 40]. Some previous works [5, 18, 19, 27, 37] extend self-training [16] to supervise the target domain. Another branch of works [7, 26, 34, 50] resort to the triplet loss [28] and contrastive

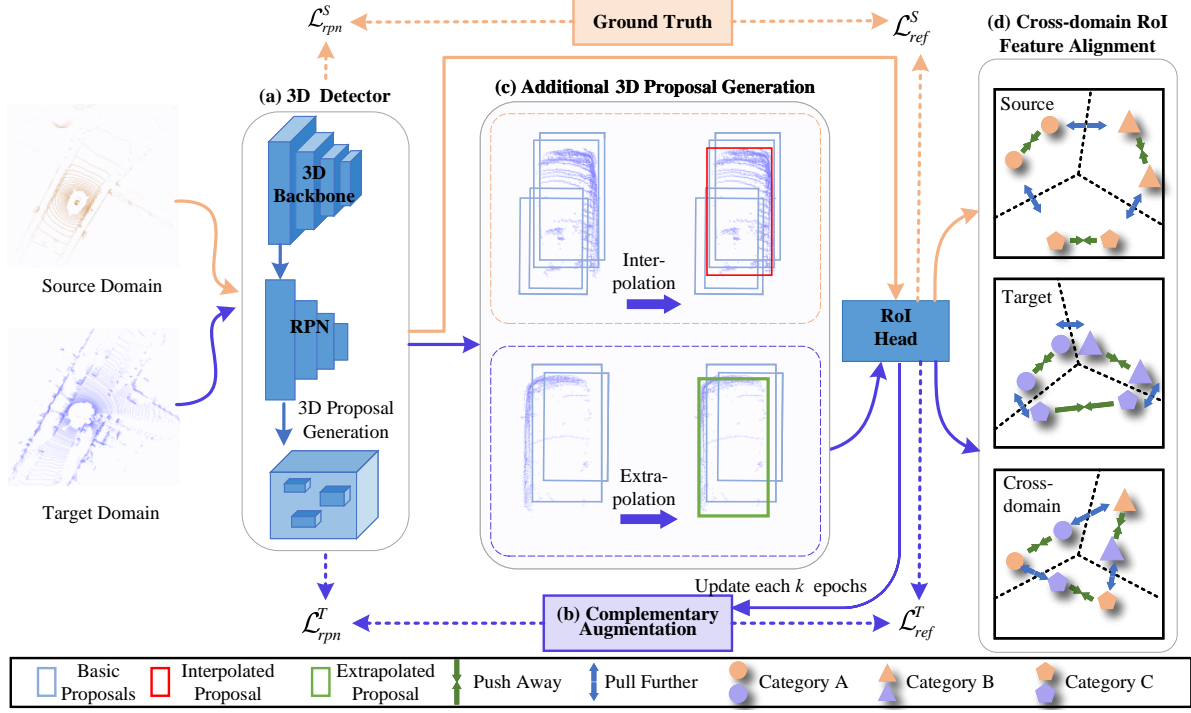


Figure 2. The overall framework of our PERE. (a) We pre-train an existing two-stage 3D detector in the source domain and then generate the basic pseudo labels in the target domain, followed by two iterative processes. (b) During the selection process, these labels are processed by Complementary Augmentation (Sec. 3.3) to boost the reliability of pseudo boxes. (c) During the training process, we implement Additional Proposal Generation Based on Interpolation and Extrapolation (Sec. 3.4), (d) and perform Cross-Domain RoI Feature Alignment (Sec. 3.5) to progressively address the issue of IPNI. After training k epochs, we update the basic pseudo labels.

learning to achieve feature alignment. However, most of these UDA mechanisms are specially developed for image tasks. When directly transferred to the sparse and unordered 3D point clouds, their detection performances are significantly exacerbated due to the fundamental discrepancies in data structures and model architectures [44]. Notably, our work introduces a complementary augmentation tailored specifically for 3D point clouds. Additionally, we address the issue of IPNI, which is particularly prevalent in cross-dataset 3D unsupervised domain adaptation.

2.3. Unsupervised Domain Adaptation for 3D Object Detection

Recently, several approaches have been proposed to address the 3D UDA. [38] relies on partial statistics information of the target domain to provide weak supervision. [39] presents LiDAR distillation to bridge the domain shift caused by different LiDAR beams. [21] employs a teacher-student detector, while [51] aligns cross-domain distribution to mitigate domain gaps. Recent works [11, 22, 42, 43, 45, 47, 48, 49] based on the self-training mechanism [16] have achieved SOTA performance in the field of 3D UDA. Compared to these works, our method aims to improve the reliability of pseudo boxes and IPNI

for consistently improving the quality of pseudo labels.

3. Methodology

3.1. Problem Statement and Preliminary

Problem Statement. The objective of the 3D UDA task is to train a 3D object detector F based on \mathbb{D}_s and \mathbb{D}_t , and minimize F 's classification and localization errors on \mathbb{D}_t , where $\mathbb{D}_s = \{(P_i^s, L_i^s)\}_{i=1}^{N_s}$ denotes the labeled source domain containing N_s point cloud samples, $\mathbb{D}_t = \{P_i^t\}_{i=1}^{N_t}$ denotes the unlabeled target domain containing N_t point cloud samples. P_i^s denotes the i -th point cloud sample, and L_i^s denotes its corresponding label, including the size (l, w, h) , the center location (o_x, o_y, o_z) , the heading angle θ and the category c of each object in P_i^s . Similarly, P_i^t denotes the i -th point cloud sample in the target domain.

Preliminary. Our work builds upon a two-stage voxel-based detector F . In the first stage, the point clouds are processed through the 3D backbone and the region proposal network (RPN) of F . This process generates basic proposals, along with their corresponding IoU confidence scores. In the second stage, the RoI head utilizes these proposals to derive RoI features. Detection results are subsequently refined based on these RoI features. Then F is optimized

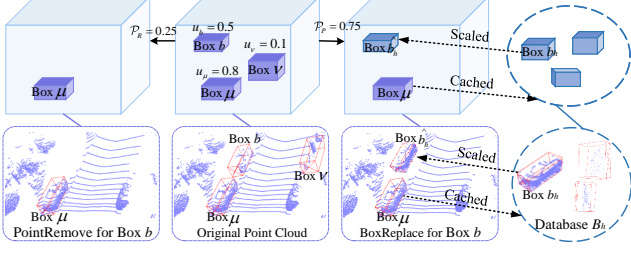


Figure 3. An example of how CA works. Here, the margin $[T_{neg}, T_{pos}]$ is set as $[0.2, 0.6]$. $u_\nu \leq T_{neg}$, so box ν is discarded. $u_\mu \geq T_{pos}$, so box μ is cached in the database B_h . $T_{neg} < u_b < T_{pos}$, so box b is performed by either BoxReplace or PointRemove according to Eq. (2).

using the detection loss \mathcal{L}_{det} , which can be written as

$$\mathcal{L}_{det} = \mathcal{L}_{rpn} + \mathcal{L}_{ref}, \quad (1)$$

where \mathcal{L}_{rpn} denotes the classification and regression loss of the RPN in the first stage, and \mathcal{L}_{ref} denotes the second stage refinery loss, which incorporates the Intersection-over-Union (IoU) regression loss \mathcal{L}_{IoU} . Here, \mathcal{L}_{IoU} is utilized to optimize the IoU confidence scores [12, 17, 29].

3.2. Framework Overview

As shown in Fig. 2, initially, we utilize the pre-trained model derived from the source domain to generate the basic pseudo labels in the target domain, followed by two iterative processes. During the selection process, these pseudo labels are then processed by Complementary Augmentation (Sec. 3.3). Subsequently, \mathbb{D}_t , equipped with the processed pseudo labels, and \mathbb{D}_s are fed into the detector F for training. During the training process, we implement Additional Proposal Generation (Sec. 3.4) in the first stage and perform RoI Feature Alignment (Sec. 3.5) in the second stage to progressively tackle the issue of IPNI. After training k epochs, we update the basic pseudo labels.

3.3. Complementary Augmentation

During the selection process, the presence of unreliable 3D boxes is unavoidable. Previous works still utilize them for training [4, 20, 22, 42], resulting in suboptimal performance. Instead, we propose a complementary augmentation (CA), to strengthen the reliability of pseudo boxes.

Concretely, given a pseudo box b generated from the detector F , we follow [12, 17, 29] to evaluate its quality through its 3D IoU confidence score u_b . Subsequently, we follow [42] to set a threshold margin $[T_{neg}, T_{pos}]$ to compare against u_b . If $u_b \leq T_{neg}$, we classify b as a low-confidence box and discard it. If $u_b \geq T_{pos}$, we classify b as a high-confidence box and store it in the database B_h . Conversely, if $T_{neg} < u_b < T_{pos}$, we classify b as an unreliable box and subject it to either PointRemove or BoxRe-

place through weighted sampling. To determine the sampling probability of BoxReplace \mathcal{P}_P , we normalize the confidence score u_b by

$$\mathcal{P}_P = (u_b - T_{neg}) / (T_{pos} - T_{neg}). \quad (2)$$

And $\mathcal{P}_R = 1 - \mathcal{P}_P$ denotes the sampling probability of PointRemove. As shown in Eq. (2), when u_b is lower, box b is prone to be tackled by PointRemove. Conversely, when u_b is higher, box b is inclined to be handled by BoxReplace. To present how CA works intuitively, we give a comprehensive example in Fig. 3. Notably, regardless of whether PointRemove or BoxReplace is performed, box b is no longer utilized in the subsequent training process. The detailed operations of BoxReplace and PointRemove are as follows: **1) BoxReplace.** We randomly select a high-confidence box b_h with the same category as b from B_h . Subsequently, for each point inside b_h , we follow [29, 41, 42] to transform its coordinate (x, y, z) from the ego-car coordinate system to the local coordinate system by

$$(x^l, y^l, z^l) = (x - o_x^h, y - o_y^h, z - o_z^h) M_{\theta_h}, \quad (3)$$

$$M_{\theta} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (4)$$

where $\theta \in \{\theta_h, \theta_b\}$, M_{θ} denotes the rotation matrix of θ , (x^l, y^l, z^l) denotes the point coordinate in the local coordinate system, (o_x^h, o_y^h, o_z^h) and θ_h denote the center coordinate and the heading angle of b_h . Next, we remove all points inside b , and scale the size of b_h to align with that of b . Let \hat{b}_h denote the scaled b_h , we then replace b with \hat{b}_h . Specifically, for each point inside \hat{b}_h , its coordinate $(\hat{x}, \hat{y}, \hat{z})$ under the ego-car coordinate system can be calculated by

$$(\hat{x}, \hat{y}, \hat{z}) = \left(\frac{l_b}{l_h} x^l, \frac{w_b}{w_h} y^l, \frac{h_b}{h_h} z^l \right) (M_{\theta_b})^T + (o_x^b, o_y^b, o_z^b), \quad (5)$$

where (l_b, w_b, h_b) and (l_h, w_h, h_h) denote the sizes of box b and b_h , respectively. (o_x^b, o_y^b, o_z^b) and θ_b denote the center coordinate and the heading angle of box b . In this way, we store the valuable localization and categorization information from box b into \hat{b}_h . Box \hat{b}_h is then cached into the database \hat{B}_h . The points within \hat{b}_h can serve as foreground points at unreliable locations, aiming to prevent the detector F from getting stuck in local minima. **2) PointRemove** removes all points within box b . After processing unreliable pseudo labels, we employ pseudo labels from $(B_h \cup \hat{B}_h)$ to supervise \mathbb{D}_t in the subsequent training process.

3.4. Additional Proposal Generation Based on Interpolation and Extrapolation

During the training process, IPNI potentially leads to proposals clustering around regions with similar point numbers

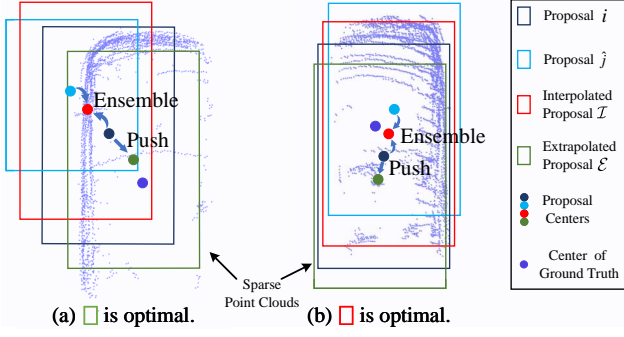


Figure 4. We adopt bird-eye view (BEV) and omit other basic low-confidence proposals to present the interpolation and extrapolation operations more intuitively. (a) and (b) demonstrate that the extrapolated and the interpolated proposals exhibit the closest alignment with their corresponding instances, respectively.

as instances in \mathbb{D}_s in the first stage. A naive manner to strengthen the proposal accuracy is to generate additional dense proposals around the highest-confidence proposals, as they are more likely to cover the corresponding instances. However, this way would introduce a substantial computational burden. Instead, we propose interpolation and extrapolation (I&E) to tackle this issue.

Specifically, given a point cloud sample in \mathbb{D}_t , let \mathbb{P} denote the set of its all basic proposals. We gather the highest-confidence proposal subset \mathbb{P}^h from \mathbb{P} by leveraging a NMS threshold of 0.01. The set of the remaining proposals is denoted as \mathbb{P}^r . Given a proposal $i \in \mathbb{P}^h$, its closest proposal \hat{j} can be derived by

$$\hat{j} = \operatorname{argmax}_j (\sigma_{ij}), j \in \mathbb{P}^r, \quad (6)$$

where σ_{ij} denotes the pair-wise 3D IoU between i and j . If $\sigma_{i\hat{j}} < T_{iou}$, where T_{iou} is a pre-defined threshold, we assume the basic proposals i and \hat{j} represent different instances. In this case, no extra proposals will be generated. If $\sigma_{i\hat{j}} > T_{iou}$, we assume i and \hat{j} represent the same instance. Subsequently, we implement the interpolation and extrapolation operations. The interpolated proposal \mathcal{I} and the extrapolated proposal \mathcal{E} can be calculated by

$$o^{\mathcal{I}} = o^i - \lambda (o^i - o^{\hat{j}}), \text{ Interpolation}, \quad (7)$$

$$o^{\mathcal{E}} = o^i + \lambda (o^i - o^{\hat{j}}), \text{ Extrapolation}, \quad (8)$$

where $o \in (o_x, o_y, o_z)$ denotes the coordinate of the corresponding proposal center. Additionally, $\lambda \in (0, 1)$ denotes the deviation level. With a larger λ , the generated proposals exhibit greater deviations from i . \mathcal{I} and \mathcal{E} inherit (l, w, h, θ) from i , which offer higher precision compared to those of j . Then we cache \mathcal{I} and \mathcal{E} into the set \mathbb{P}^a . Notably, instead

of generating dense proposals for each instance, we generate only **two** additional proposals. In this way, we aim to balance computational burden and model accuracy.

In Fig. 4, we visually illustrate the interpolation and extrapolation operations. Specifically, in Fig. 4(a), \mathcal{E} is the closest to the corresponding instance, where the extrapolation operation could push the detection boundary of i towards regions with sparse point clouds, rather than focusing solely on regions with similar point numbers as instances in \mathbb{D}_s . On the other hand, in Fig. 4(b), \mathcal{I} aligns most closely with the corresponding instance, where the interpolation operation could exploit an ensemble of i and \hat{j} to comprehensively combine their location information. Both operations present opportunities for generating superior proposals. Conversely, there might be cases where the basic proposals align most closely with the instances. Considering the above factors, we feed the proposals from $(\mathbb{P}^a \cup \mathbb{P})$ along with their RoI features into the second stage. Subsequently, a non-maximum-suppression (NMS) is performed to select the optimal proposals.

3.5. Cross-Domain RoI Feature Alignment

In the second stage, RoI feature confusion caused by IPNI also exacerbates the quality of pseudo labels during the training process. Consequently, how to effectively align cross-domain 3D RoI features of the same category remains a challenge. Recent studies in the triplet loss [7, 26, 28] have shown its capability of feature alignment in person re-identification, image retrieval and *etc.* In light of this, we redesign the triplet input (*i.e.* anchor, positive and negative samples) of the triplet loss for cross-domain 3D RoI feature alignment. Our core idea is to enhance the RoI feature compactness in the same category and strengthen the RoI feature separability in different categories, no matter whether they are from the same domain or not.

Specifically, given a proposal (anchor) a in domain d_1 derived from the first stage, we obtain its corresponding RoI feature representation x_a , where $d_1, d_2 \in \{\mathbb{D}_s, \mathbb{D}_t\}$. In order to ensure fast convergence, we select the hardest positive representation x_p and the hardest negative representation x_n of a from domain d_2 . The indices of these representations can be calculated by

$$p = \operatorname{argmax}_{i \in I_c^{d_2}} \{\|x_a - x_i\|\}, a \in I_c^{d_1}, \quad (9)$$

$$n = \operatorname{argmin}_{j \in I_{\{1, 2, \dots, |C|\} \setminus c}^{d_2}} \{\|x_a - x_j\|\}, a \in I_c^{d_1}, \quad (10)$$

where $I_c^{d_1}$ and $I_c^{d_2}$ denote the total proposals of category c in domain d_1 and d_2 at current batch, $c \in \{1, 2, \dots, |C|\}$, $|C|$ denotes the total number of categories, $\setminus c$ means excluding c , $\|x - y\|$ denotes the Euclidean Distance between x and

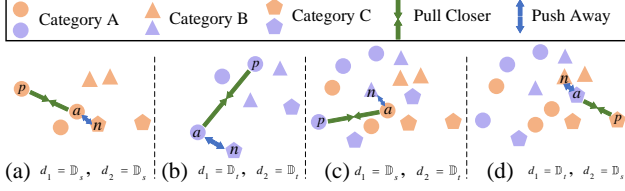


Figure 5. In (a) and (b), where $d_1 = d_2$, we implement the intra-domain loss within the same domain, whereas the inter-domain loss is implemented across different domains, as depicted in (c) and (d), where $d_1 \neq d_2$.

y. The cross-domain triplet loss for RoI feature alignment in the 3D UDA field can be formulated as

$$\mathcal{L}_{(d_1, d_2)} = \sum_{c=1}^{|C|} \sum_{a \in I_c^{d_1}} \max(\|x_a - x_p\| - \|x_a - x_n\| + \alpha, 0), \quad (11)$$

where $\alpha > 0$ denotes the margin, controlling the distance between positive and negative samples. $\mathcal{L}_{(d_1, d_2)}$ aims to enforce the margin between $\|x_a - x_n\|$ and $\|x_a - x_p\|$ by pulling x_a and x_p closer and pulling x_a and x_n away. In addition to aligning features in the same domain by the intra-domain triplet loss \mathcal{L}_{intra} , we also align RoI features across different domains by the inter-domain triplet loss \mathcal{L}_{inter} , as shown in Fig. 5. Specifically, $\mathcal{L}_{intra} = \sum_{d_1=d_2} \mathcal{L}_{(d_1, d_2)}$, and

$\mathcal{L}_{inter} = \sum_{d_1 \neq d_2} \mathcal{L}_{(d_1, d_2)}$. The final triplet loss is written as

$$\mathcal{L}_{triplet} = \mathcal{L}_{intra} + \mathcal{L}_{inter}. \quad (12)$$

By redesigning the cross-domain triplet loss, we aim to mitigate RoI feature confusion in 3D UDA, thereby enhancing the quality of pseudo labels.

The final loss function can be denoted as

$$\mathcal{L} = \mathcal{L}_{det} + \eta \mathcal{L}_{triplet}, \quad (13)$$

where $\eta \in (0, 1)$ is a trade-off hyper-parameter.

4. Experiments

4.1. Experimental Setup

4.1.1 Dataset

We evaluate our proposed PERE on three widely used autonomous driving datasets including Nuscenes [2], Waymo [32] and KITTI [8]. NuScenes is a massively annotated 32-beam dataset collected in America and Singapore. KITTI is a popular 64-beam dataset collected in Germany. Waymo is a large-scale 64-beam dataset collected in America and we randomly subsample 50% of its training set. Each dataset has distinctive idiosyncrasies in sensor configurations and

etc. We denote each dataset as a separate domain. Cross-dataset experiments are conducted to detect multiple categories simultaneously, which account for the majority of the total objects. The detailed settings are as follows: NuScenes \rightarrow KITTI (N \rightarrow K), Waymo \rightarrow KITTI (W \rightarrow K), and Waymo \rightarrow NuScenes (W \rightarrow N). We follow [39, 42] to evaluate all models on the validation set of each dataset.

4.1.2 Evaluation Metrics

We evaluate all experimental results from BEV and 3D perspectives by the official KITTI average precision over 40 recall positions with IoU thresholds of 0.7, 0.5 and 0.5 for *Car*, *Pedestrian* and *Cyclist*, as well as leveraging Closed Gap = $\frac{AP_{model} - AP_{source\ only}}{AP_{oracle} - AP_{source\ only}} \times 100\%$ [42] to compare the effectiveness of all models.

4.1.3 Implementation details

Hyperparameters. To ensure a fair comparison with SOTA methods, we adopt the same detection range (*i.e.* [-75.2, 75.2]m for X and Y axes, and [-2, 4]m for Z axis), voxel size (*i.e.* (0.1m, 0.1m, 0.15m)) and threshold margin (*i.e.* [0.25, 0.6]) as [42]. In the first stage, we generate 512 (100) basic proposals during the training (test) process in each sample. In Sec 3.4, we set T_{iou} to 0.01 and λ to 0.5. In Sec 3.5, we set α to 1.0 and η to 0.1. Pseudo labels are updated every $k = 2$ epochs.

Training details. We adopt the OpenPCDet [33] toolbox to obtain the pre-training model in the source domain with the intra-domain triplet loss $\mathcal{L}_{(D_s, D_s)}$. We have done all experiments with four NVIDIA 3080Ti GPUs. In the self-training process, we use Adam [13] with a learning rate 1.5×10^{-3} to optimize PERE in the target domain for 30 epochs.

4.2. Performance Comparison of Main Results

As demonstrated in Table 1, PERE outperforms all competitive methods by convincing margins across six source-target benchmarks in most cases. Specifically, on N \rightarrow K, W \rightarrow K and W \rightarrow N, PERE surpasses the second-bests by around 3.68%, 4.15% and 1.21% (12.25%, 7.84% and 5.60%) in terms of AP_{BEV} (AP_{3D}) in *Car* using PVRCNN, while the results using SECOND-IOU are also remarkable. For *Pedestrian*, PERE performs the best in terms of AP_{BEV} on all tasks, while we achieve the second-bests in terms of AP_{3D} on W \rightarrow K and W \rightarrow N using PVRCNN. For *Cyclist*, PERE also exhibits the highest AP_{BEV} and AP_{3D} values. Furthermore, PERE markedly narrows the performance gaps between the source only model and the oracle model. For instance, for *Car*, PERE closes AP_{3D} gaps by 68.23%, 53.65%, 17.61%, 82.09%, 53.50% and 38.53% across all six benchmarks. Notably, the overall performance

Task (Backbone)	Method	Car				Pedestrian				Cyclist				Average	
		AP _{BEV}	Closed Gap	AP _{3D}	Closed Gap	AP _{BEV}	Closed Gap	AP _{3D}	Closed Gap	AP _{BEV}	Closed Gap	AP _{3D}	Closed Gap	AP _{BEV}	AP _{3D}
N → K (PVR-CNN)	Source Only	68.53	-	42.52	-	28.08	-	23.87	-	14.72	-	8.31	-	37.11	24.90
	ST3D [42]	79.18*	55.58%	58.64	42.60%	47.41*	70.01%	41.06	60.54%	20.61	10.80%	16.42	14.43%	49.07	38.71
	ST3D++ [43]	78.46	51.83%	60.88*	48.52%	47.04	68.67%	41.20*	61.04%	22.65*	14.54%	18.75*	18.56%	49.38	40.28
	DTS [11]	77.65	47.60%	57.82	40.43%	45.74	63.96%	36.30	43.78%	19.76	9.24%	14.83	11.60%	47.72	36.32
	PERE (ours)	82.09	70.77%	68.34	68.23%	48.37	73.48%	42.24	64.71%	26.42	21.46%	23.96	27.84%	52.29	44.85
	Oracle	87.69	-	80.36	-	55.69	-	52.26	-	69.25	-	64.53	-	70.88	65.72
W → K (PVR-CNN)	Source Only	64.71	-	23.86	-	43.75	-	38.59	-	48.57	-	45.32	-	52.34	35.92
	ST3D [42]	70.88	26.85%	46.79	40.58%	48.57	40.37%	42.38	27.72%	54.93	30.75%	51.17*	30.45%	58.12	46.78
	ST3D++ [43]	71.65*	30.20%	50.23*	46.67%	50.94*	60.22%	47.23	63.20%	56.23*	37.04%	50.78	28.42%	59.61	49.41
	DTS [11]	69.38	20.32%	47.06	41.06%	46.11	19.77%	42.27	26.92%	49.75	5.70%	45.70	1.98%	55.08	45.01
	PERE (ours)	74.62	43.12%	54.17	53.65%	51.26	62.90%	46.91*	60.86%	60.47	57.54%	56.82	59.86%	62.12	52.63
	Oracle	87.69	-	80.36	-	55.69	-	52.26	-	69.25	-	64.53	-	70.88	65.72
W → N (PVR-CNN)	Source Only	33.54	-	19.86	-	12.78	-	9.46	-	2.67	-	2.06	-	16.33	10.46
	ST3D [42]	34.79*	6.98%	21.62*	10.43%	15.89*	20.91%	13.93	37.65%	6.17	22.70%	3.90	15.01%	18.95	13.15
	ST3D++ [43]	33.46	-0.45%	20.57	4.21%	14.76	13.31%	12.41	24.85%	6.23	23.09%	4.29*	18.19%	18.15	12.42
	DTS [11]	34.55	5.65%	20.64	4.63%	14.73	13.11%	13.03	30.08%	6.59*	25.42%	4.11	16.72%	18.62	12.59
	LD [39]	33.87	1.84%	20.12	1.54%	15.20	16.27%	13.47	33.78%	6.05	21.92%	3.83	14.44%	18.37	12.47
	PERE (ours)	35.21	9.33%	22.83	17.61%	16.18	22.86%	13.78*	36.39%	8.63	38.65%	6.47	35.97%	20.01	14.36
	Oracle	51.43	-	36.72	-	27.65	-	21.33	-	18.09	-	14.32	-	32.39	24.12
N → K (SECOND)	Source Only	49.27	-	25.13	-	24.96	-	21.68	-	12.29	-	6.74	-	28.84	17.85
	ST3D [42]	69.32	60.07%	49.66	48.28%	40.90*	74.84%	31.55	54.32%	17.86	12.26%	14.33	16.50%	42.69	31.85
	ST3D++ [43]	72.01*	68.12%	50.54	50.01%	40.08	70.98%	34.16*	68.68%	18.75*	14.21%	16.90*	22.02%	43.61	33.87
	DTS [11]	71.96	67.97%	58.07*	64.83%	40.27	71.88%	33.82	66.81%	17.38	11.20%	15.95	19.96%	43.20	35.95
	PERE (ours)	73.65	73.04%	66.84	82.09%	42.69	83.24%	35.47	75.89%	21.74	20.79%	19.39	27.42%	46.03	40.57
	Oracle	82.65	-	75.94	-	46.26	-	39.85	-	57.74	-	52.88	-	62.22	56.22
W → K (SECOND)	Source Only	46.38	-	19.12	-	41.28	-	34.91	-	43.37	-	41.06	-	43.68	31.70
	ST3D [42]	66.83	56.38%	42.67	41.45%	43.02*	34.94%	35.79*	17.81%	45.59*	15.45%	42.70	13.87%	51.81	40.39
	ST3D++ [43]	69.28*	63.14%	46.40*	42.67%	42.35	21.49%	35.31	8.10%	44.86	10.36%	43.04*	16.75%	52.16	41.58
	DTS [11]	64.38	49.63%	39.46	35.80%	41.94	13.25%	34.93	0.40%	43.90	3.69%	41.76	5.92%	50.07	38.72
	PERE (ours)	71.02	67.93%	49.52	53.50%	43.86	51.81%	36.67	35.63%	48.22	33.75%	43.70	22.36%	54.37	43.30
	Oracle	82.65	-	75.94	-	46.26	-	39.85	-	57.74	-	52.88	-	62.22	56.22
W → N (SECOND)	Source Only	28.73	-	16.32	-	8.42	-	5.31	-	3.09	-	2.57	-	13.41	8.07
	ST3D [42]	32.07*	16.71%	22.49*	31.95%	13.45*	34.45%	8.92	24.78%	7.40*	35.13%	4.22	17.33%	17.64	11.88
	ST3D++ [43]	31.80	15.36%	21.32	25.89%	12.78	29.86%	9.31*	27.45%	7.27	34.07%	4.36*	18.80%	17.28	11.66
	DTS [11]	29.85	5.60%	21.39	26.26%	11.40	20.41%	8.71	23.34%	6.85	30.64%	3.68	11.65%	16.03	11.26
	LD [39]	30.95	11.10%	22.03	29.57%	12.55	28.29%	8.34	20.80%	7.19	33.41%	4.01	15.13%	16.90	11.46
	PERE (ours)	34.48	28.76%	23.76	38.53%	15.45	48.15%	11.47	42.28%	8.79	46.45%	5.84	34.35%	19.57	13.69
	Oracle	48.72	-	35.63	-	23.02	-	19.88	-	15.36	-	12.09	-	29.03	22.53

Table 1. Performance comparison on six adaptation benchmarks. The best performances are in bold and the second-bests are marked by *. **Oracle** with gray values represents that the detector is fully supervised by the labeled target domain data. **Source Only** means directly applying the pre-trained model of the source domain to the target domain. [39] is limited to the transition from the high-beam dataset to the low-beam dataset. [11, 39, 42, 43] were originally designed for detecting a single category in a model. To ensure a fair comparison, we have modified their open-source code to cater to the multi-category object detection task, which is common and hard in real scenarios. Besides, all competitors build upon the same backbone detectors (*i.e.* PVR-CNN [29] and SECOND-IOU [41]) as ours. Notably, we follow [42, 43] to select the best models to generate the preliminary pseudo labels for different categories during pre-training.

of both PERE and its competitors in the multi-category object detection task is slightly lower than that of the single-category task mentioned in respective papers [11, 39, 42], illustrating the greater difficulty of the former task.

To sum up, when compared with the self-training methods [11, 42, 43], PERE presents superior performance for effectively tackling the unreliable pseudo labels. Additionally, unlike previous works [11, 39], we take into consideration the issue of IPNI. The overall results validate the generalization and effectiveness of PERE.

4.3. Ablation Studies

In this section, all ablation experiments are based on PVR-CNN, conducted on $N \rightarrow K$, and evaluated for *Car*.

4.3.1 Architecture Designs

As shown in Table 2, we compare the results of using different configurations equipped with particular components. Generally, each module in PERE contributes to the per-

Method	AP _{BEV} / Closed Gap	AP _{3D} / Closed Gap
ST (baseline)	75.23 / 34.97%	57.79 / 40.35%
ST + $\mathcal{L}_{(\mathbb{D}_s, \mathbb{D}_t)}$	75.94 / 38.67%	58.21 / 41.46%
ST + RFA	77.81 / 48.43%	59.92 / 45.98%
ST + RFA + I&E	79.66 / 58.09%	62.93 / 53.94%
ST + RFA + I&E + CA	82.09 / 70.77%	68.34 / 68.23%

Table 2. Component studies of different network configurations. ST denotes the naive self-training technique, $\mathcal{L}_{(\mathbb{D}_s, \mathbb{D}_t)}$ indicates that we pre-train the model in the source domain using the intra-domain triplet loss, I&E represents interpolation and extrapolation operations, CA denotes the complementary augmentation, and RFA denotes the cross-domain ROI feature alignment.

formance, demonstrating their effectiveness. Specifically, comparing with ST, $\mathcal{L}_{(\mathbb{D}_s, \mathbb{D}_t)}$, RFA, I&E and CA yield performance gains of 0.73%, 2.96%, 5.21%, and 9.36% (0.94%, 2.49%, 2.46%, and 3.23%) in terms of AP_{3D} (AP_{BEV}). Remarkably, CA contributes the most to the model’s performance, highlighting the superiority of our

Method	AP _{BEV} / Closed Gap	AP _{3D} / Closed Gap
w/o BoxReplace	80.19 / 60.86%	64.01 / 56.79%
w/o PointRemove	81.21 / 66.18%	64.32 / 57.61%
Random sampling	79.95 / 59.60%	62.37 / 52.46%
CA (ours)	82.09 / 70.77%	68.34 / 68.23%

Table 3. Effectiveness analysis of Each Module in CA. Without BoxReplace, all unreliable boxes are directly removed. Random sampling denotes the sampling process without assigning weights.

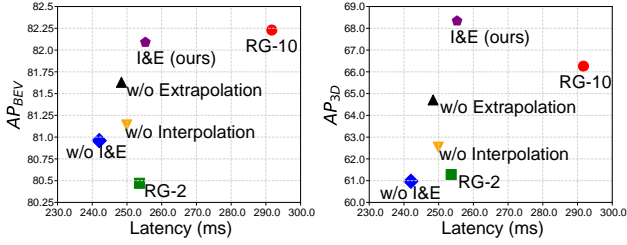


Figure 6. Accuracy-latency Analysis of Additional Proposal Generation. RG-10 denotes randomly generating ten additional proposals for each instance (*i.e.* a dense set), while RG-2 denotes randomly generating two additional proposals per instance. We measure latency on one RTX 3080Ti GPU (batch size = 1).

method in enhancing the reliability of pseudo boxes.

4.3.2 Analysis of Complementary Augmentation

In this part, we investigate the importance of each module in CA (Sec. 3.3). As shown in Table 3, removing PointRemove, BoxReplace, and weighted sampling all lead to a performance drop of 6.3%, 5.8%, and 8.7% (2.3%, 1.1%, and 2.6%) in terms of AP_{3D} (AP_{BEV}). The results demonstrate that PointRemove, BoxReplace, and weighted sampling all contribute to performance improvement. Particularly, without weight, random sampling achieves the lowest performance, highlighting the greater importance of weighted sampling compared to the other modules.

4.3.3 Analysis of Additional Proposal Generation

We present the accuracy-latency analysis of the additional proposal generation (Sec. 3.4). As shown in Fig. 6, without I&E, using the basic proposals results in poor performance. Both interpolation and extrapolation contribute to performance gains. RG-2 demonstrates similar latency with our I&E approach but exhibits inferior detection accuracy. Besides, RG-10 achieves the second-best accuracy, while costing $1.20 \times$ latency (compared to using the basic proposals). In contrast, our I&E requires around $1.05 \times$ latency while achieving superior performance. These findings indicate that our I&E can effectively strike a balance between model accuracy and computational burden.

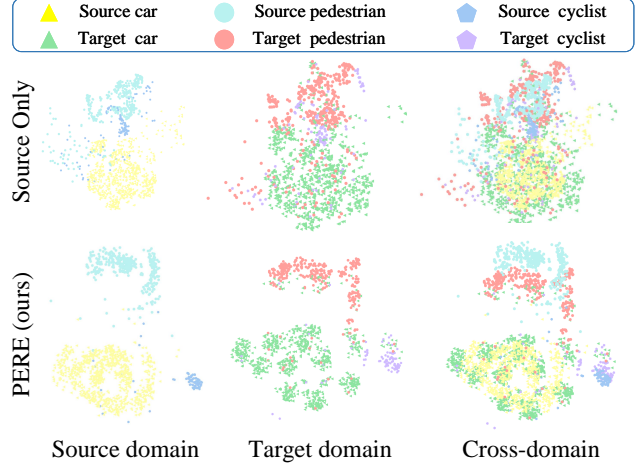


Figure 7. The T-SNE [36] visualization of RoI feature distribution of three categories. This visualization is conducted on the validation subset of the $N \rightarrow K$ task, utilizing the PVRCNN backbone.

4.3.4 T-SNE visualization

We employ T-SNE [36] to visualize the distribution of RoI features (Sec. 3.5). As shown in the upper part of Fig. 7, the source only model confuses RoI features of different categories in the target domain, which validates the negative impact caused by IPNI. In contrast, as shown in the lower part of Fig. 7, PERE achieves superior performance for RoI feature clustering and aligning. Compared with the source only model, PERE effectively mitigates the cross-domain RoI feature confusion by aligning RoI features of the same categories across different domains. By incorporating the findings presented in Table 2, this alignment effectively enhances the quality of pseudo labels.

5. Conclusion

In this paper, we have presented a framework named Pseudo Label Refinery (PERE) for 3D UDA. PERE contains a complementary augmentation, additional proposal generation, and cross-domain RoI feature alignment. These mechanisms all contribute to improving the quality of pseudo labels by improving the reliability of pseudo boxes and alleviating the adverse impact of the cross-dataset instance-level point number inconsistency. Extensive experiments along with comprehensive ablation analysis validate the effectiveness and the generalization ability of our PERE.

Acknowledgement. This work was supported in part by The National Nature Science Foundation of China (Grant Nos.: 62273302, 62036009, 61936006, 62273303, 62303406), in part by Yongjiang Talent Introduction Programme (Grant No.: 2023A-194-G, 2022A-240-G, 2023A-197-G), in part by Zhejiang Provincial Natural Science Foundation of China (Grant No.: LR23F020002).

References

- [1] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3722–3731, 2017. 2
- [2] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. 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. 1, 2, 6
- [3] L. Chen, H. Chen, Z. Wei, X. Jin, X. Tan, Y. Jin, and E. Chen. Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7181–7190, 2022. 2
- [4] Z. Chen, Y. Luo, Z. Wang, M. Baktashmotlagh, and Z. Huang. Revisiting domain-adaptive 3d object detection by reliable, diverse and class-balanced pseudo-labeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3714–3726, 2023. 1, 4
- [5] Y. Cho, W. J. Kim, S. Hong, and S.-E. Yoon. Part-based pseudo label refinement for unsupervised person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7308–7318, 2022. 2
- [6] L. Fan, X. Xiong, F. Wang, N. Wang, and Z. Zhang. Rangedet: In defense of range view for lidar-based 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2918–2927, 2021. 2
- [7] Y. Ge, D. Chen, and H. Li. Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification. In *International Conference on Learning Representations*, 2019. 2, 5
- [8] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 1, 2, 6
- [9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *NeurIPS*, 2014. 2
- [10] Z. Han, H. Sun, and Y. Yin. Learning transferable parameters for unsupervised domain adaptation. *IEEE Transactions on Image Processing*, 31:6424–6439, 2022. 2
- [11] Q. Hu, D. Liu, and W. Hu. Density-insensitive unsupervised domain adaption on 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17556–17566, 2023. 1, 2, 3, 7
- [12] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang. Acquisition of localization confidence for accurate object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 784–799, 2018. 4
- [13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6
- [14] H. Kuang, B. Wang, J. An, M. Zhang, and Z. Zhang. Voxel-fpn: Multi-scale voxel feature aggregation for 3d object detection from lidar point clouds. *Sensors*, 20(3):704, 2020. 2
- [15] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. 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. 2
- [16] D.-H. Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013. 2, 3
- [17] B. Li, W. Ouyang, L. Sheng, X. Zeng, and X. Wang. Gs3d: An efficient 3d object detection framework for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1019–1028, 2019. 4
- [18] R. Li, S. Li, C. He, Y. Zhang, X. Jia, and L. Zhang. Class-balanced pixel-level self-labeling for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11593–11603, 2022. 2
- [19] S. Li, J. Huang, X.-S. Hua, and L. Zhang. Category dictionary guided unsupervised domain adaptation for object detection. In *Proceedings of the AAAI conference on artificial intelligence*, pages 1949–1957, 2021. 2
- [20] Z. Li, J. Guo, T. Cao, L. Bingbing, and W. Yang. Gpa-3d: Geometry-aware prototype alignment for unsupervised domain adaptive 3d object detection from point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6394–6403, 2023. 1, 4
- [21] Z. Luo, Z. Cai, C. Zhou, G. Zhang, H. Zhao, S. Yi, S. Lu, H. Li, S. Zhang, and Z. 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. 3
- [22] X. Peng, X. Zhu, and Y. Ma. Cl3d: Unsupervised domain adaptation for cross-lidar 3d detection. In *Pro-*

- ceedings of the AAAI Conference on Artificial Intelligence*, pages 2047–2055, 2023. 1, 2, 3, 4
- [23] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2
- [24] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 1, 2
- [25] R. Ramamonjison, A. Banitalebi-Dehkordi, X. Kang, X. Bai, and Y. Zhang. Simrod: A simple adaptation method for robust object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3570–3579, 2021. 2
- [26] H. Rami, M. Ospici, and S. Lathuilière. On-line unsupervised domain adaptation for person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3830–3839, 2022. 2, 5
- [27] C.-X. Ren, Y.-H. Liu, X.-W. Zhang, and K.-K. Huang. Multi-source unsupervised domain adaptation via pseudo target domain. *IEEE Transactions on Image Processing*, 31:2122–2135, 2022. 2
- [28] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 2, 5
- [29] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. 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, 4, 7
- [30] S. Shi, Z. Wang, J. Shi, X. Wang, and H. 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. 2
- [31] W. Shi and R. Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1711–1719, 2020. 2
- [32] P. Sun, H. Kretschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. 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
- [33] O. D. Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. <https://github.com/open-mmlab/OpenPCDet>, 2020. 6
- [34] M. Thota and G. Leontidis. Contrastive domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2209–2218, 2021. 2
- [35] Z. Tian, X. Chu, X. Wang, X. Wei, and C. Shen. Fully convolutional one-stage 3d object detection on lidar range images. *NeurIPS*, 2022. 2
- [36] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 8
- [37] W. Wang, F. Zhao, S. Liao, and L. Shao. Attentive waveblock: Complementarity-enhanced mutual networks for unsupervised domain adaptation in person re-identification and beyond. *IEEE Transactions on Image Processing*, 31:1532–1544, 2022. 2
- [38] Y. Wang, X. Chen, Y. You, L. E. Li, B. Hariharan, M. Campbell, K. Q. Weinberger, and W.-L. 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
- [39] Y. Wei, Z. Wei, Y. Rao, J. Li, J. Zhou, and J. Lu. Lidar distillation: Bridging the beam-induced domain gap for 3d object detection. In *ECCV 2022: 17th European Conference*, page 179–195, 2022. 1, 2, 3, 6, 7
- [40] H. Xia, H. Zhao, and Z. Ding. Adaptive adversarial network for source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9010–9019, 2021. 2
- [41] Y. Yan, Y. Mao, and B. Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 1, 2, 4, 7
- [42] J. Yang, S. Shi, Z. Wang, H. Li, and X. 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
- [43] J. Yang, S. Shi, Z. Wang, H. Li, and X. Qi. St3d++: Denoised self-training for unsupervised domain adaptation on 3d object detection. *IEEE transactions on pattern analysis and machine intelligence*, 45(5):6354–6371, 2022. 1, 3, 7
- [44] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. 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, 3
- [45] Z. Yihan, C. Wang, Y. Wang, H. Xu, C. Ye, Z. Yang, and C. Ma. Learning transferable features for point cloud detection via 3d contrastive co-training. *Ad-*

vances in Neural Information Processing Systems, 34:21493–21504, 2021. 1, 3

- [46] T. Yin, X. Zhou, and P. 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. 2
- [47] Y. You, C. A. Diaz-Ruiz, Y. Wang, W.-L. Chao, B. Hariharan, M. Campbell, and K. Q. Weinbergert. Exploiting playbacks in unsupervised domain adaptation for 3d object detection in self-driving cars. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5070–5077. IEEE, 2022. 1, 3
- [48] Y. You, K. Luo, C. P. Phoo, W.-L. Chao, W. Sun, B. Hariharan, M. Campbell, and K. Q. Weinberger. Learning to detect mobile objects from lidar scans without labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1130–1140, 2022. 1, 3
- [49] Y. You, C. P. Phoo, K. Luo, T. Zhang, W.-L. Chao, B. Hariharan, M. Campbell, and K. Q. Weinberger. Unsupervised adaptation from repeated traversals for autonomous driving. *Advances in Neural Information Processing Systems*, 35:27716–27729, 2022. 1, 3
- [50] X. Yue, Z. Zheng, S. Zhang, Y. Gao, T. Darrell, K. Keutzer, and A. S. Vincentelli. Prototypical cross-domain self-supervised learning for few-shot unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13834–13844, 2021. 2
- [51] W. Zhang, W. Li, and D. 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. 3
- [52] Y. Zhou and O. 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