3D-UDA: Unsupervised Domain Adaptation for 3D Object Detection

Anonymous Author(s)
Affiliation
Address
email

Abstract

Unsupervised domain adaptation (UDA) enables a model learned from a source domain to generalize well to a target domain, without any access to target labels. The performance of UDA approaches hinges on effectively narrowing the distribution discrepancy between source and target domains. Despite the considerable progress in 2D vision problems, UDA for 3D object detection has been rarely discussed due to the highly different data structures and large distribution discrepancy of LiDAR point clouds in comparison with 2D images. In this paper, we address the UDA problem for 3D object detection by firstly introducing a conceptually simple yet effective distribution alignment scheme. On the one hand, we present a novel multi-input single-output (MISO) model architecture to handle the cross-sensor shift existing between 3D domains, which is also capable of retaining performance on the source domain and is extensible for multiple target domains. On the other hand, we propose a contrastive adaptation procedure with data augmentation, which mines natural hard samples together with simple samples to conduct category-wise sample-level feature alignment. Experiments on the adaptation tasks between three large-scale 3D datasets illustrate that our proposed distribution alignment approach is superior to previous methods on the adapted performance.

1 Introduction

Recently, driven by the cheaper laser sensors and large-scale human-annotated datasets [1, 5, 26], LiDAR-based 3D object detection for autonomous vehicles has witnessed remarkable advancements [15, 44, 40, 45]. Despite their success, 3D detectors developed on one specific domain, *i.e.*, source domain, usually performs poorly on a unseen applied domain, *i.e.*, target domain, due to the unavoidable domain-shifts arising from different settings of LiDAR sensors and physical environments, etc. This issue might significantly limits the generalization of 3D detectors in practical application, owing to the expensive and time-consuming 3D annotations. To address this issue, we tackle the unsupervised domain adaptation (UDA) problem for 3D detection, which aims to effectively adapt 3D detectors trained on labeled source domain to a novel unlabeled target domain.

Commonly adopted approaches of UDA can be roughly classified into two categories: 1) Self-training methods [46, 11, 13] that supervisedly re-train model on target domain with pseudo-labels. 2) Distribution alignment methods that learn domain-invariant feature representations on feature-level [12, 29, 9, 24] or pixel-level [33, 18, 10]. These strategies are comprehensively exploited in 2D scenes. However, studies on 3D scenes still fall behind compared to their 2D counterpart as evidently. In all existing UDA methods for 3D detection, self-training [37, 22, 41, 3] is still the dominant approach. Notably, those methods drop out the labeled source data in training process and thus the performance is sensitive to the quality of predicted pseudo-labels, which depends on adequate sophisticated assignment strategies. In contrast, distribution alignment methods take consideration

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

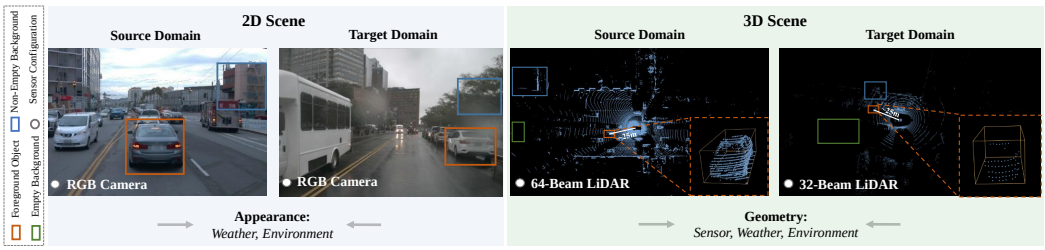


Figure 1: Comparison of the domain shift between 2D scenes and 3D scenes.

of the UDA nature that encourages domain-invariant features to narrow domain gap, which is more steady and faithful with the guidance of source labels but lack of exploration in 3D scenes.

The status that previous distribution alignment approaches for 2D scenes are not readily applicable to 3D scenes can be attributed to two main reasons as illustrated in Fig. 1: 1) Different data structures result in different genres of domain shift. LiDAR grasps depth point clouds in the form of sparse points while camera captures RGB images in the form of dense 2D pixels. Therefore the domain shift in 2D and 3D scenes are mainly reflected in appearance and geometry respectively. Furthermore, distinct from the compact features obtained from 2D images, features in 3D scenes contain large meaningless regions due to the large-scale empty zone unique to lidar points, which requires specific strategies for distribution alignment. 2) Different 3D sensor settings induce larger domain gaps in 3D scenes. Domain shift on 2D scenes chiefly derives from environment changes and weather conditions. While for 3D scenes, data variations additionally arise from 3D sensor configuration. Concretely, different 3D data collector settings, such as laser beams or perception distance, can render massive diversity of basic data distributions, such as density or range, which needs attention when dealing with 3D UDA problems. Though previous work [20] has first proposed to match 3D point distributions between domains via adversarial learning, they only considered the classification problem of isolated objects without capability of handling complete scenes. Based on the aforementioned observations, the purpose of this work is to move the needle by initializing the research on distribution alignment for UDA of 3D detection.

To this end, we presents a novel MISO-3D Network to achieve unsupervised domain adaptation that feasible for 3D scenes understanding. We highlight two main contributions in this work. Firstly, we propose a Multi-Input Single-Output (MISO) model architecture that utilizes separate 3D backbone networks but shared 2D backbone to parse point clouds. Compared to 2D scenarios, the cross-sensor gap exists in 3D domains results in additional distribution discrepancy, which is hard to handle within integrated network. We posit that our architecture adapts to the sensor-specific features by separate parts and inure to the domain-invariant features by integrated part. Furthermore, our architecture benefits the retention of model capability in source domain. Namely, the adapted model performs well both in source and target domain simultaneously. The proposed architecture can be extended to multiple target domains as well.

Secondly, we propose a novel contrastive distribution alignment procedure with data augmentation mechanisms. To the best of our knowledge, we are the first work that introduces the concept of distribution alignment to the UDA of 3D detection. The keys to the feasibility is that: 1) Assigning XXX sampling modules according to 3D data properties, which relieves the heavy imbalance proportion problem and massive feature discrepancy problems that exists between empty zone and non-empty zone of 3D background. 2) Proposing data augmentation mechanisms with support of the high editability of point clouds, which mine natural hard samples to prevent model from overfitting and bad local minimal. We have conducted ablation studies and in-depth discussions to show our alignment procedure effectively narrow domain gap compared to previous vanilla self-training strategy.

In the experiment, we show that the proposed MISO-3D Network outperforms previous approaches with a large margin on all adaptation directions, demonstrating its efficacy on UDA problem for 3D detection.

2 Related Work

LiDAR-based 3D Object Detection. LiDAR-based 3D detectors aims to localize and classify 3D objects from point clouds, which can be broadly grouped into two categories: *i.e.*, grid-based and point-based. The grid-based methods divide point clouds into regular 3D voxels [34, 44, 40, 7] or BEV maps [15, 36, 35]. In SECOND [34], a sparse convolution operation is proposed to parse the compact representation. CenterPoint [40] replaces the general anchor-based detector with a keypoint-based detector. For point-based approaches, PointRCNN [25] and STD [39] apply PointNet [19] to segment foreground points and generate proposals for each point. 3DSSD [38] disposes all upsampling layers and refinement modules for computational efficiency. Compared to grid-based methods, point-based approaches require high computation loads, resulting in time-consuming training on large-scale datasets such as the nuScenes [1] and Waymo [26] datasets. For computation efficiency, we adopt the grid-based methods [44, 15] based on the anchor-free detector [40] in this paper.

2D Unsupervised Domain Adaptation. The main challenge of UDA is that distribution shift, *i.e.*, domain gap, exists between the target and source domain. To bridge the domain gap, a variety of domain adaptation methods have been proposed for 2D vision tasks including classification [4, 12], detection [29, 14, 24, 2] and segmentation [46], which can be roughly divided into two categories: feature distribution alignment [4, 12, 14] and self-training [46, 11, 13]. For feature distribution alignment methods, a group of approaches focus on minimizing a specific domain discrepancy metric, *e.g.* Maximum Mean Discrepancy (MMD) [6, 28], Wasserstein Distance [16]. With the development of adversarial learning, additional works proposed to introduce domain classifier to perform alignment in the feature space [4, 27, 17, 9, 29, 14]. Furthermore, contrastive learning are employed in more fine-grained feature alignment such as category-level alignment [12, 32] and instance-level alignment [24, 42]. Besides, image translation methods have been adopted to perform alignment in the pixel-level [23, 33, 8, 18, 10]. For self-training methods [46, 11, 13, 2, 21], they assign pseudo-labels with filtering schemes by pre-trained models from labled source domain to guide the re-training process on target domain.

3D Unsupervised Domain Adaptation. On par with the developments on domain adaptation for image setting, some recent works aim to address the domain shift on point clouds for shape classification [20], semantic segmentation [31, 43]. In this paper, we focus on the domain adaptation task for 3D object detection, for which few approaches have been proposed. Statistical normalization(SN) [30] is firstly proposed to close the size-level domain gap by normalizing the object size of source domain. Though it points out the importance of aligning scale bias between 3D domains, the SN approach requires the target statistic to enable scale-transformation, which is generally unavailable for UDA tasks. So SF-UDA [22] utilize temporal coherency to estimates the target scales without target statistic, which transforms the scale of pseudo-labels on target data to re-train the model. Current works[37, 41, 3] further explore the generation mechanisms of pseudo-labels on target data as the supervision signals for self-training technique. Since the those self-training methods fully depends on the pseudo-labels, they rely on adequate sophisticated strategies to enhance the quality of original pseudo-labels generated from source modal. But the adaptation process without labeled source data can be easily misguided by the noisy supervision signals. Therefore we posit that they are sub-optimal to narrow domain shift compared to the our proposed distribution alignment methods, in which the labeled source data are utilized to provide stable supervision for the adaptation process.

3 MISO-3D Network for UDA of 3D Object Detection

3.1 Problem Formulation

Supervised 3D object detection: A 3D object detector M aims to generate predictions of category and localization for 3D objects Y from point clouds P:

$$M : P \rightarrow Y \tag{1}$$

which denotes point clouds consisting of n sparse points by a set $P = \{p_1, \cdots, p_n\}$, where $p_i \in \mathrm{R}^d$, for $i = 1, \cdots, n$. We set d to 4 to represent each LiDAR point including 3-dimensional coordinates (x, y, z) and additional intensity i. The 3D annotations for m objects are represented as $Y = \{y_1, \cdots, y_m\}$, where y_i contains category and 3D bounding box information for each object, and each box is parameterized by its size (l, w, h), center (c_x, c_y, c_z) and heading angle r. Usually, the detector M

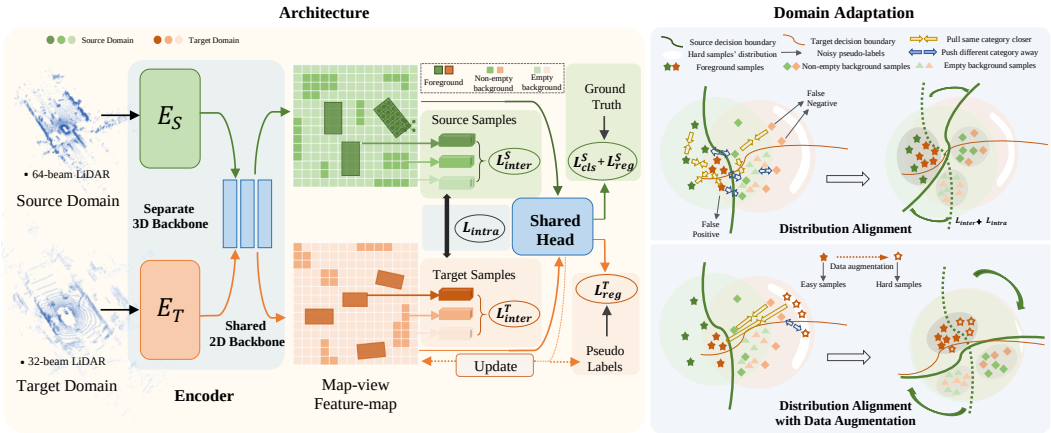


Figure 2: An illustration of the proposed MISO-3D framework.(a) The MISO model architecture that consists of domain-specific 3D backbone and domain-invariant 2D backbone. (b) The illustration of proposed contrastive distribution alignment procedure with data augmentation.

is trained in a supervised manner by minimizing the category and localization errors between the prediction $\hat{Y}$ and its groundtruth Y for every frame of point clouds.

UDA for 3D object detection: The UDA problem aims to adapt a 3D object detector trained on source labeled data $\{(P_i^S, Y_i^S)\}_{i-1}^{N_S}$ of N_S frames to unlabeled target domain given target unlabeled data $\{P_i^T\}_{i=1}^{N_T}$ of N_T frames:

$$M_{uda} : P^S \cup P^T \rightarrow Y^S \cup Y^T \tag{2}$$

where M_{uda} is trained on the labeled data (P^S, Y^S) in the source domain together with the unlabeled data P^T in the target domain.In addition to employing the detector model, M_{uda} includes additional carefully-designed modules for domain adaptation.

3.2 Multi-Input Single-Output Model Architecture

The network architecture of our proposed MISO-3D network is shown in Fig. 2. We rely on CenterPoint [40] as base detector model including an anchor-free head and two alternative kinds of encoders, namely VoxelNet [44] and PointPillars [15]. Based on the base detector, we devise a novel multi-input single-output architecture with partly-sharing constraint, which is feasible to reduce domain shift and is extensible for multiple domain adaptation.

Typical UDA models benefit from the parameter-sharing feature encoder to learn domain-invariant feature. Based on the observation that the 3D data suffers more serve discrepancy that arises from different 3D sensor configurations, we posit that the parameter-sharing constraint of model architecture are required for reconsideration. Compared to images from different domains that vary on appearance and yet consist of homogeneous dense pixels, sparse point clouds collected by different collector exist massive gap on basic data distribution according to point density, perception angle and even perception range. Therefore complete sharing encoder is sub-optimal to process 3D data from different domains, which prefers to be parsed by sensor-specific backbones with differentiated receptive field. However, totally separate encoders are adverse to domain-invariant feature learning. So we devise a novel model architecture including separate 3D backbone together with shared 2D backbone. Specifically, the 3D backbone learn different mapping functions to convert LiDAR points into map-view feature map for each domain, while the 2D backbone further process the feature map to learn domain-invariant feature for narrowing domain gap. Mathematically, this can be written as:

$$XXX \tag{3}$$

The two alternative kinds of encoders we employ, *i.e.*, VoxelNet and PointPillars, process LiDAR points in different ways. For VoxelNet, the LiDAR points are quantized into small 3D voxels using a process called voxelization, and then convert the 3D features within each voxel into a fixed size

representation. Then the 3D feature-map are processed by a 3D convolutional backbone network which collapses all voxels along the height dimension into the map-view representation and processed by 2D convolutional backbone network to produce the feature map M. For PointPillar, the LiDAR points are collapsed in the height dimension and quantized into tall voxels, called pillars, along a regular 2D grid. Then the 2D grids are processed by a linear transformation and per-pillar max-pooling into a fixed size representation. In general, PointPillars encoder is faster than VoxelNet but less accurate. We conduct our architecture based on both two kinds of encoders to validate the application generality. Namely, our model architecture based on VoxelNet has separate voxelization modules and 3D convolutional networks as 3D backbones, while the one based on PointPillar has separate pillar-processing networks as 3D backbones. Both architecture have shared 2D convolutional networks as 2D backbones.

It is worth noting that our architecture not only benefits the model performance retention on source domain, but also can be easily extended to the domain adaptation problem for multiple unlabeled target. The extended model can do well in multiple domains simultaneously.

3.3 Contrastive Distribution Alignment with Data Augmentation

To reduce domain gap, we introduce the contrastive learning concept to conduct distribution alignment on the features from map-view feature map M. In contrast to 2D vision tasks, the application to 3D detection is more sophisticated. On one hand, there exists heavy background-imbalance. Concretely, the background occupy largely in 3D scenes, which consists of empty zone and non-empty zone. The feature representation can vary greatly among different background part according to whether points are contained. On the other hand, another problem for 3D detection is that the 3D objects come in a wide range of sizes, shapes, and aspect ratios compared to axis-aligned 2D bounding boxes, which enhance the regression difficulty for detection head. Besides, we posit that hard examples are essential in training to prevent model from overfitting, but they are easily omitted in alignment process for being the target edging distribution far away from source distribution.

To address above issues, we devise a contrastive distribution alignment framework with data augmentation. In specific, domain adaptation is realized via aligning two domain's sample-level feature, in which the sample feature are represented by XXX modules. On the basis of this framework, we propose data augmentation mechanisms to mine hard samples for training. The details are as follows:

3.3.1 Sample-to-Sample Contrastive Adaptation

Since the anchor-free detection head predicts detections based on the map-view feature map M that converted by 3D encoder from LiDAR points, we expect the feature centroids of similar samples get closer between domains.

XXX sample representation modules. To extract sample-level feature, we consider the representation of both foreground points and background zones respectively. Based on the pseudo-labels predicted on target data by applying source model, we reserve them by a simple confidence score threshold. Then we convert those pseudo bounding boxes $Y^P = \{y_1^P, \cdots, y_n^P\}$ into map-view as the foreground proposals on feature map M. For each proposal y^P, we extract its feature using a rotated ROI feature extraction approach. In specific, we identify $K \times K \times 1$ equally distributed key points with respect to the proposal box. For each key point, we compute a feature by bilinear interpolation of its value in the feature map. Then we use average pooling across all the key points to obtain a single feature vector as the final representation of foreground sample S_F. For background zones beyond the proposal box on feature map, we extract features from each grid as the corresponding background samples S_B. It is worth noting that background zones contain larger proportion of empty zones without points and smaller proportion of non-empty zones with background objects, which vary greatly at feature distribution. So we breakdown those background samples into two categories as empty background samples S_B^E and non-empty background samples S_B^{NE} according to whether it is empty zone, which helps relieves the heavy unbalanced proportion problem and massive feature discrepancy problem .

Category-aware feature alignment. Category-aware domain alignment is comprehensively studied in recent literatures[12, 32]. The core idea of these methods is to align the samples from same category of two domains. As opposed to prior works that perform class-level alignment by designing categories' prototype and narrow the distance between same categories' prototype of two domains, we conduct

sample-level alignment to save computation from assigning complicated prototypes. However, the main challenge with sample-level approach is that the sample feature can be too inconstant to represent a category, especially for foreground samples. To overcome this, we propose a similarity-based technique to construct pairwise relations. Considering the distribution alignment is gradual onset, the convergence process is flattened when aligning nearside distribution but is sharp when aligning distant distributions. Based on this fact, we choose the closest foreground feature as a pair:

$$\underset{k\in C, 1\leq i\leq N_k^S}{\mathbf{P}^k(i)} = \underset{1\leq j\leq N_k^T}{\arg\min}\{\Phi(S_i^S, S_j^T)\} \tag{4}$$

where $\mathbf{P}^k$ represents the pairwise relation within class k, $\Phi(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}$ calculate the cosine distances between N_k^S source sample features and N_k^T target candidates samples. Also, we control the sample number of selected background samples to deal with the proportion imbalance. We set the sample number of both S_B^E and S_B^{NE} at 200 in this paper. In addition to achieve intra-class distance minimizing, we further constrain the distance between different categories within same domain, namely inter-class distance. The contrastive learning-based losses is defined:

$$L_{intra}(S,T) = \sum_{k\in C} \sum_{\substack{i\in N_k^S \\ j=\mathbf{P}^k(i)}} \log \frac{exp(S_i^S \cdot S_j^T/\tau)}{exp(S_i^S \cdot S_j^T/\tau) + \sum_{j\in N_k^T, j\neq i} exp(S_i^S \cdot S_j^T)} \tag{5}$$

$$L_{inter}(D) = \sum_{k\in C} \sum_{\substack{i\in N_k^D, j\in N_k^D \\ i\neq j}} \log \frac{exp(S_i^D \cdot S_j^D/\tau)}{exp(S_i^D \cdot S_j^D/\tau) + \sum_{j\in N_k^D, j\neq i} exp(S_i^D \cdot S_j^D)} \tag{6}$$

$$L_{da} = L_{intra}(S,T) + L_{inter}(S) + L_{inter}(T) \tag{7}$$

In the total domain adaptation loss L_{da}, all pairwise relations of samples between domains and within domains are considered, which enhance the intra-class compactness and inter-class separability.

3.3.2 Data Augmentation Mechanisms

We conduct analysis on the distribution of pseudo-labels, which is the base of sample generation. We highlight the facts that: First, patterns with dense point clouds tend to obtain higher confidence as reserved pseudo-labels as illustrated in Fig. **??**(a). Second, patterns with intact structures are easier to category and localize as demonstrated in Fig. **??**(b). Therefore, most positive pseudo annotations that generated from high-confident detections are easy examples with adequate points or complete geometry shape. However, we posit that those omitted hard examples, which are distributed at the edging area as shown in Fig. 2(b), make great difference in training process. They can further promoting distribution alignment and prevent the model from overfitting towards those easy examples by bad local minimal.

The key to creating synthetic hard samples lies in the the consistency with genuine hard samples. Based on the observation from above two factors, we devise two augmentation mechanisms to provide spurious samples from existing annotations. The first mechanism aims to transfer dense simple samples into sparse hard ones. Specifically, we uniformly sample and delete the LiDAR points that make up the point cloud cluster in both the horizontal and vertical directions, as shown in Fig. **??**(a). The other mechanism targets on breaking the complete geometries and generating incomplete structures. Concretely, we calculate the viewpoint of certain sample and randomly select part of angle as the final incomplete structure, as shown in Fig. **??**(b).

Consequently, our purposeful augmented samples efficiently supplement the absent hard distribution in terms of density and geometry shape. Compared to previous common augmentation strategies including translation, random flipping, shifting, scaling and rotation that applied in 3D detection training [40, 34] , we focus more on replenishing valid hard distribution rather than just enriching sample diversity by geometry transformations. Furthermore, those density-transformed and completeness-transformed samples are of assistance in creating unseen objects on target domain. Especially for the adaptation tasks from source domain of richer laser beams to target domain of insufficient laser beams, *e.g*. Waymo to nuScenes, pre-trained model is inductive to target patterns of comparable density with source domain, but is new to those over-few-point patterns that may be average distribution for target domain. Therefore our augmentation mechanisms can make those missing distributions available.

3.4 Training Procedure

Final objective functions: Our model predicts 3D bounding boxes and corresponding category label with an single-stage anchor-free detection head. When training the XXX model, we leverage the supervision signals from source domain to define a detection loss as follow:

$$L_{det} = L_{cls}^{S} + L_{loc}^{S} \tag{8}$$

where L_{cls} and L_{loc} calculate the classification and localization error respectively. We also propose domain adaptation loss L_{da} based on the source label and target pseudo-labels as defined in Eq. 7. Although only the pairwise samples are calculated to conduct alignment, the samples within each category are gradually clustered by minimizing the intra-category variances and maximizing the inter-category distances. The illustration is shown in Figure. 2(b). Meanwhile, since the 3D bounding boxes come in a wide range of shapes and aspect ratios, we exploit a regression regularization L_{reg} that calculates localization error on target domain for prompting the detection head adapt to the target domain simultaneously. The final overall objective is:

$$\arg\min_{M_\theta} L_{det} + \lambda_1 L_{da} + \lambda_2 L_{reg} \tag{9}$$

where M_θ represents the whole parameterized model, and λ_1 and λ_2 are the trade-off parameters between domain adaptation loss and regression regularization.

Stepwise training procedure: We tried to train the XXX model in a single step, where pseudo-labels on target domain are updated at each iteration. However, the training is strongly unstable for the frequently updated supervision signals. On one hand, the fraction of false labels is increased in the beginning training epochs as illustrated in Fig. **??**. On the other hand, model fluctuations can inconsistent pseudo-labels as demonstrated in Fig. **??**. These factors lead to noisy supervisory information and instability for training process. Therefore, we propose a stepwise training procedure with pseudo-labels update mechanism as following. First, we pretrain the detector on the source domain with the detection loss as Eq. 8. Then, we leverage the pretrained model to predict pseudo-labels on target data to warm up the training process as Eq. 9 for half the total epochs and obtain a well-initialized model. Next, we update the pseudo-labels for remaining epochs with a IoU-based ensembling update mechanism and continue the training process. In specific, we newly generate pseudo-labels every few epochs, then we calculate pairwise IoU scores of labels between two generations to ensemble the overlapped pairs with the higher confidence annotations. That ensembling mechanism is benefit to avoid ambiguous supervision from model fluctuations. Hence, the stepwise training procedure adapt the proposed model progressively in the whole training process.

4 Experiments

4.1 Experiments Setup

Datasets. We evaluate our model on three widely applied driving datasets: Waymo [26], nuScenes [1] and KITTI [5]. The three datasets have their own configurations for 3D data collectors. The Waymo dataset [26] captures data by five LiDAR sensors from Phoenix, Mountain View and San Francisco in multiple weathers at different times. The nuScenes dataset [1] collects data by a 32-beam roof LiDAR sensor from Boston, USA and Singapore in multiple weathers at different times. KITTI dataset [5] provides 64-beam Velodyne LiDAR point clouds around Karlsruhe, Germany in clear weather and day time. The label scale also differs among datasets, 122000 training frames and 30407 validation frames for Waymo, 28130 training frames and 6019 validation frames for nuScenes, 7481 training frames and 7518 validation frames for KITTI. We conduct adaptation experiments including Waymo→nuScenes, nuScenes→Waymo, Waymo→KITTI and nuScenes→KITTI.

Evaluation. For consistent cross-dataset analysis, we adopt the KITTI evaluation metric on the commmon category Car(named Vehicle in the Waymo). We report the average precision(AP) for both bird's eye view(BEV) IoUs and 3D IoUs under the IoU thresholds of 0.7 and 0.5. To further evaluate the adaptation performance, we also report the promotion ratio relative to the initial source model, which is formulated as **promotion ratio** $= \frac{AP_{model} - APsource}{APsource} \times 100\%$. Since the adaptation progress is bound with the initial status which generate the pseudo-annotations, the promotion ratio represents the degree of performance improvement and model robustness to noisy pseudo-labels.

Task	Method	VoxelNet		PointPillars	
		AP_{BEV}/AP_{3D}	Closed Gap	AP_{BEV}/AP_{3D}	Closed Gap
Nusc2Kitti	Source Only	46.7 / 11.1		22.8 / 0.5	
	Source Only (w/ ROS)	51.4 / 32.2		60.2 / 11.9	
	ST (w/ ROS)	56.4 / 38.1		46.0 / 9.5	
	Ours (w/ ROS)	77.1 / 56.6		78.3 / 36.9	
	Source Only (w/ SN)	45.2 / 32.6		45.9 / 6.1	
	ST (w/ SN)	52.4 / 38.9		57.6 / 23.1	
	Ours (w/ SN)	76.0 / 61.1		68.7 / 22.4	
	Oracle	84.2 / 71.9		84.8 / 71.6	
Waymo2Kitti	Source Only	55.1 / 18.9		47.8 / 11.5	
	Source Only (w/ ROS)	56.0 / 23.3		54.9 / 14.7	
	ST (w/ ROS)	63.8 / 30.1		46.5 / 23.1	
	Ours (w/ ROS)	72.5 / 39.0		65.2 / 26.2	
	Source Only (w/ SN)	63.9 / 45.2		60.5 / 34.5	
	ST (w/ SN)	54.9 / 38.9		36.6 / 17.2	
	Ours (w/ SN)	75.8 / 60.2		75.0 / 42.1	
	Oracle	84.2 / 71.9		84.8 / 71.6	
Waymo2Nusc	Source Only	32.4 / 21.1		27.8 / 12.1	
	ST	32.8 / 21.1		30.2 / 12.2	
	Ours	39.2 / 25.4		32.6 / 21.5	
	Oracle	49.5 / 32.7		49.0 / 31.3	
Nusc2Waymo	Source Only	40.2 / 22.9		28.1 / 8.5	
	ST	42.7 / 29.0		18.0 / 14.6	
	Ours	56.7 / 33.0		50.3 / 29.5	
	Oracle	78.9 / 74.7		72.9 / 50.4	

Table 1: Results 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.

Implementation. We build our proposed model based on CenterPoint [40] including two alternative kinds of 3D backbones. Specifically, we set voxel size as $(0.1, 0.1, 0.2)m$ for VoxelNet backbone and set grid as $(0.2, 0.2)m$ for PointPillar backbone. The detection range is set according to the dataset configuration, which is $[75.2, 75.2, 2]m$ in x, y, z axis for Waymo dataset, $[51.2, 51.2, 2]m$ for nuScenes dataset, $[70.4, 70.4, 2]m$ for nuScenes dataset. For the pre-training and training process, we use Adam [] with learning rate 1.5×10^{-3} and one cycle scheduler. We adopt the database sampling in SECOND [34] to increase the ratio of positive examples. Additionally, scale bias is severe from other datasets to KITTI dataset as proposed in [30] So we append scaling augmentation strategy in training process including SN [30] and ROS [37], where SN requires target statics while ROS doesn't.

4.2 Main Results

4.3 Ablation Studies

In this section, we conduct extensive ablation experiments to investigate the individual components of our models. All experiments are conducted with the 3D detector VoxelNet on the task of Nusc2KITTI.

Architecture.

Augmentation.

Contrastive Learning.

	Method	Target		Source		
		Kitti-0.7	Kitti-0.5	Kitti-0.7	Kitti-0.5	Nusc
w/ ROS	Source only	51.4 / 32.2	59.6 / 57.0	44.0 / 28.0	55.4 / 49.4	72.6
	Ours (w/ 1E)	73.5 / 53.7	83.9 / 83.1	39.1 / 25.5	50.2 / 43.8	56.0
	Ours (w/ 2E)	77.1 / 55.6	87.4 / 86.8	44.6 / 30.0	56.8 / 50.8	67.6
w/ SN	Source only	45.2 / 32.6	50.1 / 49.1	32.2 / 19.4	51.3 / 41.8	60.9
	Ours (w/ 1E)	74.1 / 60.8	81.4 / 80.7	27.3 / 16.1	42.3 / 34.0	45.0
	Ours (w/ 2E)	76.0 / 61.1	79.1 / 78.6	29.2 / 17.1	46.1 / 37.1	51.8

Table 2: Comparison of architecture.

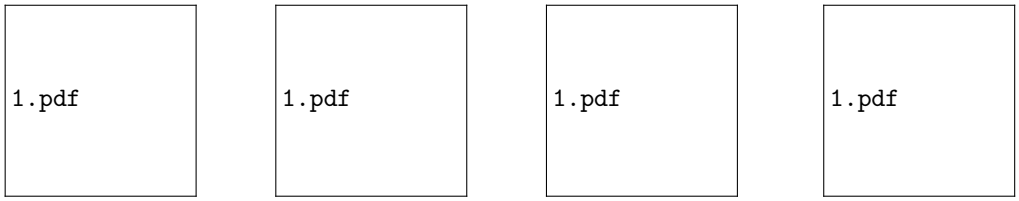
Table 3: 1

	Random	Line	Occlude	AP_{BEV}/AP_{3D}
(a)				71.8 / 57.9
(b)	✓			
(c)		✓		75.3 / 60.9
(d)			✓	68.4 / 60.5
(e)		✓	✓	76.0 / 61.1

Table 4: 2

	Bg	Grid	Simlar	AP_{BEV}/AP_{3D}
(a)				
(b)	✓			
(c)	✓	✓		
(d)	✓	✓	✓	76.0 / 61.1

Comparison with Self-Training



(a) pic1. (b) pic2. (c) pic3. (d) pic4.

Figure 3: Comparison of self-training and feature alignment.

5 Conclusion

References

[1] 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 *CVPR*, 2020.

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

[3] Benjamin Caine, Rebecca Roelofs, Vijay Vasudevan, Jiquan Ngiam, Yuning Chai, Zhifeng Chen, and Jonathon Shlens. Pseudo-labeling for scalable 3d object detection. *arXiv preprint arXiv:2103.02093*, 2021.

[4] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[5] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012.

[6] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[7] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11873–11882, 2020.

[8] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR, 2018.

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

[10] Han-Kai Hsu, Chun-Han Yao, Yi-Hsuan Tsai, Wei-Chih Hung, Hung-Yu Tseng, Maneesh Singh, and Ming-Hsuan Yang. Progressive domain adaptation for object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 749–757, 2020.

[11] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5001–5009, 2018.

[12] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019.

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

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

[15] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019.

[16] 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. PMLR, 2015.

[17] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *arXiv preprint arXiv:1705.10667*, 2017.

[18] Xinhong Ma, Tianzhu Zhang, and Changsheng Xu. Gcan: Graph convolutional adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8266–8276, 2019.

[19] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas 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 (CVPR)*, pages 652–660, 2017.

[20] 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. *arXiv preprint arXiv:1911.02744*, 2019.

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

[22] Cristiano Saltori, Stéphane Lathuiliére, Nicu Sebe, Elisa Ricci, and Fabio Galasso. Sf-uda^{3D}: Source-free unsupervised domain adaptation for lidar-based 3d object detection. *arXiv preprint arXiv:2010.08243*, 2020.

[23] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8503–8512, 2018.

[24] Astuti Sharma, Tarun Kalluri, and Manmohan Chandraker. Instance level affinity-based transfer for unsupervised domain adaptation. *arXiv preprint arXiv:2104.01286*, 2021.

[25] 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 (CVPR)*, pages 770–779, 2019.

[26] 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*, 2020.

[27] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[28] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[29] Vibashan VS, Poojan Oza, Vishwanath A Sindagi, Vikram Gupta, and Vishal M Patel. Mega-cda: Memory guided attention for category-aware unsupervised domain adaptive object detection. *arXiv preprint arXiv:2103.04224*, 2021.

[30] 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.

[31] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4376–4382. IEEE, 2019.

[32] Minghao Xu, Hang Wang, Bingbing Ni, Qi Tian, and Wenjun Zhang. Cross-domain detection via graph-induced prototype alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12355–12364, 2020.

[33] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6502–6509, 2020.

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

[35] Bin Yang, Ming Liang, and Raquel Urtasun. Hdnet: Exploiting hd maps for 3d object detection. In *Conference on Robot Learning*, pages 146–155, 2018.

[36] 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 (CVPR)*, pages 7652–7660, 2018.

[37] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d: Self-training for unsupervised domain adaptation on 3d objectdetection. *arXiv preprint arXiv:2103.05346*, 2021.

[38] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11040–11048, 2020.

[39] 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 (ICCV)*, pages 1951–1960, 2019.

[40] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *CoRR*, 2020.

[41] Yurong You, Carlos Andres Diaz-Ruiz, Yan Wang, Wei-Lun Chao, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Exploiting playbacks in unsupervised domain adaptation for 3d object detection. *arXiv preprint arXiv:2103.14198*, 2021.

[42] Xiangyu Yue, Zangwei Zheng, Shanghang Zhang, Yang Gao, Trevor Darrell, Kurt Keutzer, and Alberto Sangiovanni Vincentelli. Prototypical cross-domain self-supervised learning for few-shot unsupervised domain adaptation. *arXiv preprint arXiv:2103.16765*, 2021.

[43] Sicheng Zhao, Yezhen Wang, Bo Li, Bichen Wu, Yang Gao, Pengfei Xu, Trevor Darrell, and Kurt Keutzer. epointda: An end-to-end simulation-to-real domain adaptation framework for lidar point cloud segmentation. *arXiv preprint arXiv:2009.03456*, 2020.

[44] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, 2018.

[45] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *CoRR*, 2019.

[46] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018.