DiffRefine: Diffusion-based Proposal Specific Point Cloud Densification for Cross-Domain Object Detection

Sangyun Shin[1] Yuhang He[2] Xinyu Hou[1] Samuel Hodgson[1] Andrew Markham[1] Niki Trigoni[1]

[1] Department of Computer Science, University of Oxford, United Kingdom
[2] Microsoft Research

Project Page: https://yunshin.github.io/DiffRefine/

Abstract

The robustness of 3D object detection in large-scale outdoor point clouds degrades significantly when deployed in an unseen environment due to domain shifts. To minimize the domain gap, existing works on domain adaptive detection focuses on several factors, including point density, object shape and sizes, to reduce the false negative detections. However, the adaptation results indicate that there are still remaining challenges. We argue that this is due to the challenge in recognizing comparably less distinctive region on object surface due to sparsity, occlusion, etc. In this work, we aim to reinforce those features by generating points on object surface to make them straightforwardly recognizable. We draw our motivation from a common observation that detection proposals already contain the accurate bounding boxes, but with relatively low objectness score predictions, which lead to false negatives. Given these box proposals, we densify sparse object points with a diffusion approach. As a result, our model DiffRefine can act as a simple additional module before second-stage refinement, where most existing detection models for two-stage detection can use. Experimental results on domain adaptive detection show competitive performance, especially on vanishing points due to distance on various detection architectures.

1. Introduction

Point cloud based object detection is a popular research topic in computer vision due to the wide range of applications. Despite recent progress, it still faces challenges caused by domain gaps, due to factors such as variations in object sizes and sparsity caused by different sensors. Although different types of sensor specification can lead to different point cloud densities, such as 32 or 64-beam LiDAR, a key challenge that most sensors for point cloud share is diminishing point density on object surface due to the distance from the sensor, leaving object surface featureless (see Fig. 1). The problem becomes even worse when the domain changes, as sparse points on distant objects can be easily confused when the surrounding structure, such as roads or buildings, look different. Existing domain adaptive detection works addressing this issue can be broadly divided into two categories: 1) data augmentation-based techniques and 2) prototype-based techniques.

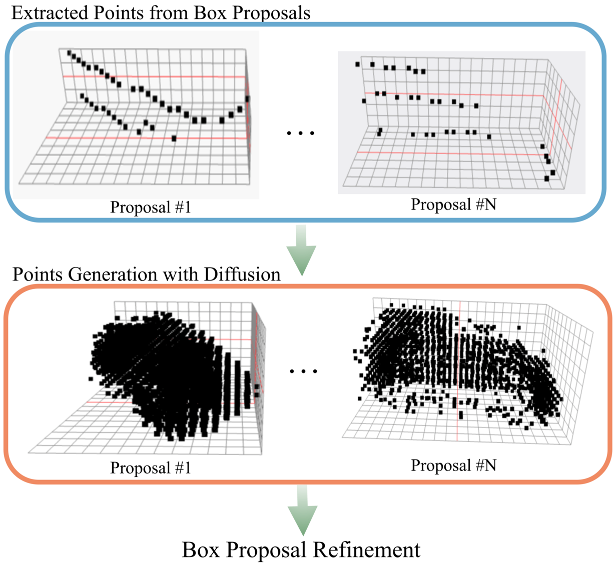


Figure 1. *DiffRefine* performs proposal specific generation to boost the detection performance.

Augmentation based techniques [9, 19, 32] aim to make the detection model insensitive to variation in sparsity caused by different sensors across domains. More specifically, during the training, they strategically sample the object points while providing the consistent targets for bounding box regression and confidence scores. During this process, the detection model learns to be insensitive towards the density of objects. However, the augmentation is usually only effective on already dense objects, which are usually close to the sensors, as their subsampling could help model the sparse objects. Since the upsampling is based on the interpolation of existing points, the distant objects, which are usually partially visible, do not get the maximum benefit from the techniques. Another approach to deal with this problem is using learnable prototypes [15]. In the work, prototypes are learned from the multiple viewing angles of the objects of the same category. Since each prototype learns each partial view of the object, they can be used to improve false negative detections on partial view. However, the prototypes also struggle to reduce the false negative detections when the objects are featureless due to sparsity.

In this paper, we address these two issues with target-specific object points generation, which focuses on the generation of specific areas where objects of interest are likely to exist instead of looking at the entire point cloud. We draw our motivation from a common observation of recent works [18, 34], where they found that the predicted boxes actually have sufficient overlaps with ground-truth, but the predicted objectness scores paired with the boxes are comparably low if the Intersection over Unions (IoUs) are not large enough ($\leq$0.7), leading to false negative detections. We argue that this is because of the absence of necessary features in the false negative region, caused by variations in sparsity, etc. Our aim is to take these potential boxes and reinforce the features of the corresponding region of boxes by point densification.

We adopt a diffusion mechanism to generate the target specific point cloud to address the sparsity challenge (see Fig. 1). Our core intuition for adopting diffusion is two-fold: 1) sparse points on object surfaces can be considered to be a noisy representation of dense object points, which can be recovered by iterative denoising, and 2) diffusion with multiple steps allows effective learning in challenging point cloud generation, rather than conventional densification in one step. Nevertheless, we notice that using generation models introduces extra challenges. For instance, an erroneously generated car instance appears despite not being warranted. To address this issue, we propose an approach to incorporate the spatial context into generation. The generated points can then be considered with the surrounding region to reject the false positive generation. Our contributions can be summarized as follows:

1. We introduce a proposal specific point generation method, which focuses on objects rather than entire point cloud, improving the problem of sparse points on object surfaces.
2. In order to maximize the generation ability for detection, we propose a differentiable 3D generation with voxel grids, making a solid second stage refinement widely available to existing detection models.
3. To avoid hallucination (false positive) generation problem, we introduce a conditioning on spatial features, which also incorporates spatial context.
4. Extensive adaptation experiments on KITTI [7], NuScenes [1], and Waymo [28], and NuScenes datasets show the effectiveness of our approach (see Fig. 1 (b)), particularly for sparse object points caused by various factors, including different sensors and distances for bridging domain gaps.

2. Related Work

Object Detection in Point Cloud There are two main category methodologies for object detection in point cloud: voxel-based methods and point-based methods. Voxel-based methods usually voxelize irregular point cloud into regular voxel grids before feeding the point cloud to the encoders to generate Bird's Eye View (BEV) feature. The encoder can be either Transformers architectures [6, 13, 14, 29, 33] or sparse convolution architectures [3, 10, 12, 22, 24, 38, 43, 47]. After obtaining the BEV feature, a Region Proposal Network (RPN) is used for final object detection. More recent works [23, 25, 35] combine Transformer based architectures and sparse convolution based architectures for better performance. Based on these observations, in this work, we build the base detectors on top of the widely used Second-IoU [38] and PointPillars [12] for the extendability.

Domain Adaptive Detection Domain adaptation aims to mitigate the domain gap discrepancy between the source and target domain. Factors such as encoder strategy [41], object size difference [21, 34] and deterioration in point cloud [37] stays as the main reasons for domain gap. Self-training strategy has been extensively leveraged to address the domain gap. It either adopts prototype learning [15, 20], knowledge distillation [36], beam augmentation [9] or pseudo-label consistency [4] to mitigate the domain gap effects. Although these attempts, sparsity issue still exists and has not been sufficiently addressed. In this work, we specifically focus on resolving the sparsity issue for distant objects in the point cloud.

Diffusion for Point Cloud Diffusion models [8] are a class of latent variable models that use Markov chains to convert noise distributions to data distributions. They have been successfully applied across a range of generative modeling tasks. For point cloud data, shape generation methods have been proposed that use diffusion for point cloud completion [45] or to act over the latent spaces of a hierarchical VAE [30]. Diffusion has also been used for object detection, refining randomly-proposed bounding boxes [46], serving as the basis for a proposal refinement stage in a two-stage detector [11], or being used to refine bounding boxes under domain shift [2]. Although these promising results, no prior has adopted Diffusion model for proposal-specific point generation.

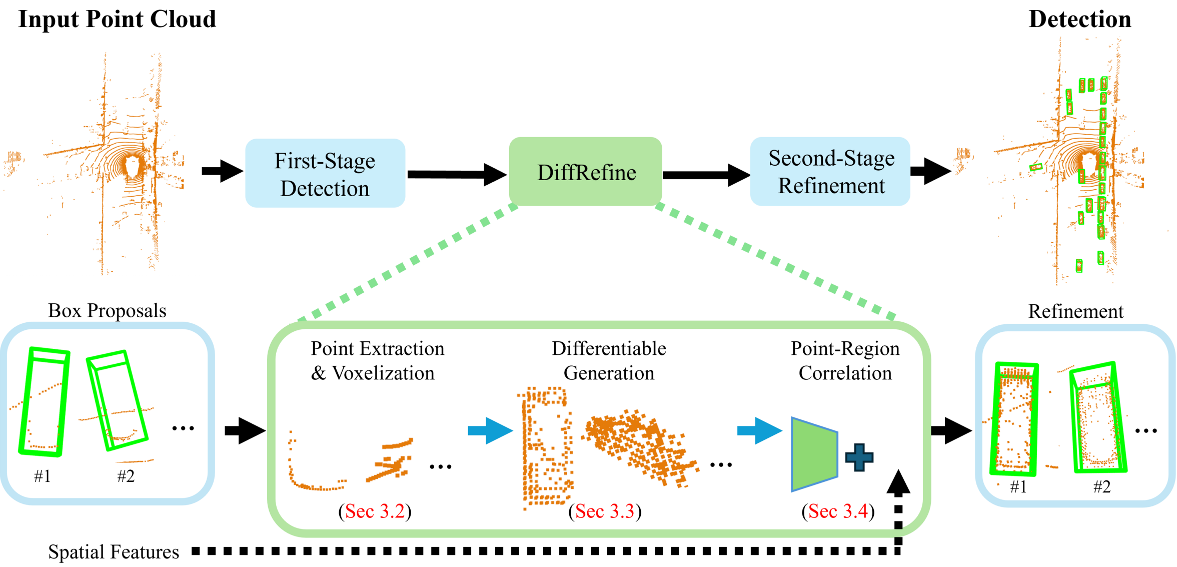


Figure 2. Overall pipeline of our proposed method for the second stage refinement: Given the box proposals from the first-stage detection, the object points inside each proposal are extracted and voxelized. The points are then densified following the denoising process of Diffusion with differentiable warping. The final refinement head takes the densified points and spatial context feature as input and predicts the refined boxes and class of objects.

3. DiffRefine Framework Introduction

3.1. Overview

Following the self-training-based unsupervised domain adaptation scheme [39, 40] for 3D detection, we are given point clouds $X = X_s \cup X_t$ and labels $Y = Y_s \cup Y_t$ as the initial set. Here, s denotes the source and t the target domain, with X_s and Y_s being point cloud and box labels of the known source domain. X_t and Y_t are the point cloud and initial pseudo label set in the unlabeled target domain. Y_t is collected by the detector trained only on the source domain using X_s and Y_s. A label in Y consists of seven parameters defining a 3D box with three parameters for center (x, y, z), three parameters for size (l, w, h), and one parameter for vertical rotation θ. Our goal is to improve the detector's domain adaptation by focusing on the pseudo-label collection. In particular, we focus on improving false negatives, caused by featureless area with point sparsity.

Specifically, given a set of box proposals and their corresponding spatial features from the first stage detection, we first extract points (Sec. 3.2) before processing them into voxel grids for fixed input to our generation module, where the points are densified with differentiable warping (Sec. 3.3). The generated points are then fed into a final box prediction module to refined the box proposals (Sec. 3.4). The overall pipeline is shown in Fig. 2.

3.2. Point Extraction and Size Agnostic Voxelization

As reported in [34], the variance in object sizes across domains negatively affects the performance of detection models. To avoid the impact led by object size variance, we follow [2, 17] to process points in normalized coordinates to achieve size-agnostic detection.

Given N_1 predicted box proposals, $B_1 \in \mathbb{R}^{N_1 \times 7}$, and their corresponding spatial features, $F_{bev} \in \mathbb{R}^{N_1 \times K}$, and corresponding confidence scores, $C_1 \in \mathbb{R}^{N_1 \times 1}$, in the first stage detection, we extract the points inside each box from a point cloud P and process them using Normalized Box View (NBV) [2]. The extracted points $p \subset P$ from a box $b \subset B_1$ are transformed into normalized space, p_{norm} as:

$$\begin{bmatrix} p_{\text{norm}} \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{l} & 0 & 0 & 0 \\ 0 & \frac{1}{w} & 0 & 0 \\ 0 & 0 & \frac{1}{h} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & -x \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & -z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} \tag{1}$$

p_{norm} is then voxelized into $v \in \mathbb{R}^{W \times H \times D}$ as follows:

$$v(w, h, d) = \begin{cases} 1, & \text{if } (x, y, z) \in p_{\text{norm}} \text{ falls in the grid} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where (w, h, d) refer to the indices of v for (W, H, D).

3.3. Proposal-specific Point Generation with Diffusion

Our core intuition behind point generation is the assumption that despite differences in density, objects of the same category, such as car and pedestrian, share similar physical features. For example, a car has four wheels below the body frame, which can be recovered even from a partial view.

Diffusion Formulation Following the intuition from point diffusion [44], the forward diffusion process continuously adds random noise to the input through a Markov chain, assuming that there is a point-to-point mapping relationship between adjacent timestamps. Given a noisy sample at timestamp $t = T$, the task of diffusion model is to continuously denoise the sample in reverse order to recover the clean data at timestamp $t = 0$, which is called generation. We apply this principle to a voxel grid. More specifically, we consider $v = v_{t=T}$ as a noisy sample (sparse object points) at time T and attempt to denoise it to obtain a clean sample (dense object points), $v_{t=0}$ that provides enough features to reduce false negatives. In the next section, we discuss how we create the generation target for densification now that the noisy sample can be provided from v.

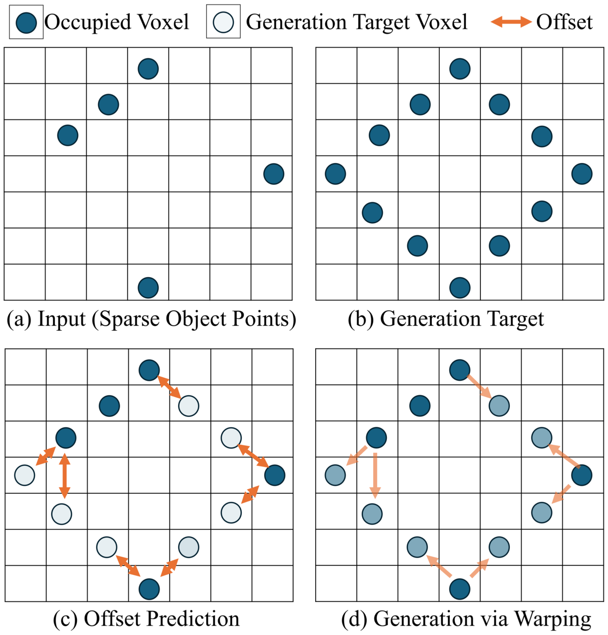
Generation Target The challenge of making a generation target is the fact that the complete shape of the object with dense points is unknown and not trivial to infer from a given sparse set of object points. An alternative way to create dense points is to accumulate object points inside ground truth bounding boxes in NBV as described in Sec 3.2. Formally, we create the generation target v_0 as:

$$v_{t=0} = v^1_{t=T} \cup v^2_{t=T}, ..., \cup v^{N_{gt}}_{t=T}, \tag{3}$$

where N_{gt} stands for the number of ground truth boxes in a batch for the same class, and $v^i_{t=T}$ is voxels extracted from i_{th} ground-truth bounding box and corresponding P. The generation task then aims to generate voxels for empty grids in $v_{t=T}$, which are occupied in $v_{t=0}$ by comparing them.

Differentiable Generation One straightforward way to generate points is to formulate it as dense grid classification problem as in [37]. However, it is not straightforward to use the output of classification during multiple reverse steps of denoising as the classification output is discrete and non-differentiable. Also, dense classification for generation gives an equal target, 1, for each grid. This does not fit well with denoising, which iteratively refines the noisy samples step by step. We propose to learn offsets from the occupied voxel grid to the input. We do this for two reasons: firstly, the generation target of grids close to occupied voxels have relatively small offsets, whereas grids further away could be considered harder as the offsets become larger.

Diffusion Training As mentioned above, we consider the sparse object points as a nosier representation of the dense object points. Therefore, we first start by introducing random noise to the clean data, $v_{t=0}$, which can be learned by our generation network. In order to formulate the sparse voxel as noisy sample, we introduce the random voxel subsampling strategy utilizing Gaussian randomness.



(a) Input (Sparse Object Points)

(b) Generation Target

(c) Offset Prediction

(d) Generation via Warping

Figure 3. Conceptual diagram of differentiable object points generation. Given given the sparse input $v_{t=T}$ (a), the process aims to generate target $v_{t=0}$ (b). During the generation process, we predict 3D dense offset. The predicted offsets from grids to be occupied according to the target, point at the closest occupied voxel from input voxel grid (c). Using the predicted offsets, the generation is performed by differentiable warping operation (d). (c) and (d) are iterative process in the formulation of denoising from Diffusion.

More specifically, we define the probability of a voxel grid for remaining occupied as:

$$m(w,h,d)_{t=i} = \begin{cases} \prod_{t=0}^{i} \exp(-\frac{|whd - \mu_{t=T}|_2}{2\sigma^2_{t=T}}), & \text{if empty,} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $whd \in \mathbb{R}^3$ refers to voxel coordinate (w,h,d). Here, the mean $\mu_{t=T} \in \mathbb{R}^3$ and the covariance matrix $\sigma_{t=T} \in \mathbb{R}^{3\times3}$ are measured from the noisy input $v_{t=T}$ to encourage the points existing in $v_{t=T}$ to survive while taking advantage of randomness for learning to denoise. In other words, the voxels close to the mean of $v_{t=T}$ have greater chance to remain occupied, encouraging the similar shape similar to $v_{t=T}$ during random sampling. For i_{th} forward process, we use $m_{t=i}$ to remove voxels from $v_{t=i-1}$. The diffusion model then learns to predict the offset to the closest occupied voxels from the removed voxel at $t = i$. Formally, the $v_{t=i+1}$ is given as:

$$v(w,h,d)_{t=i+1} = \begin{cases} 1, & \text{if } m(w,h,d)_{t=i} < \epsilon(w,h,d) \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $\epsilon(w,h,d)$ is a random threshold following uniform distribution. The loss for training the denoising process is

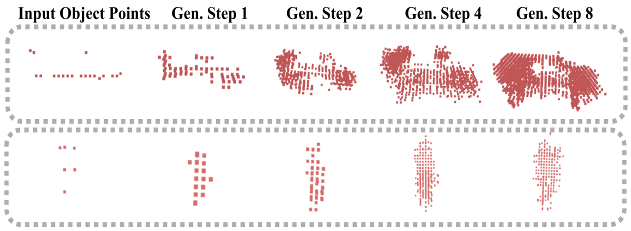


Figure 4. Generated object points of *Car* (first row) and *Pedestrian* (second row) classes with respect to denoising steps.

then given as:

$$L_{\text{diff}} = \sum_{i=1}^{T-1} \sum_{j \in S} |\hat{v}^{j}_{t=i} - v^{j*}_{t=i}|_2 \tag{6}$$

where S is a set indices for removed voxels at i_{th} step, and $j*$ is the closest occupied voxels to j.

Denoising Process During inference, as shown in Figure 3 (c) and (d), the diffusion model takes $v_{t=T}$ as input and predict offsets $o_{t=T} \in \mathbb{R}^{W \times H \times D \times 3}$ from dense 3D grid for warping as:

$$v_{t=T-1} = warp(v_{t=T}, o_{t=T}), \tag{7}$$

where $warp(,)$ refers to differentiable warping operator. As the denoising step iterates, the input voxel is densified as illustrated in Fig. 4.

3.4. Points-Region Correlation and Box Refinement

A potential risk of point denoising is false positive generation, where the diffusion model could hallucinate non-existing objects, which could lead to false positive detection. One way to avoid is to fuse a spatial context to the generated output, so that the generated output can be considered in accordance with the spatial context. For instance, a car is likely to be located near a road. Based on this motivation, we propose to fuse spatial feature and generated object points. To achieve this, we first transform $v_{t=0}$ with a 3D convolution based encoder to get the encoded feature $f_v \in \mathbb{R}^{1 \times K}$. Our Points-Region correlation then performs cross-attention to make fused feature as:

$$f_{\text{gen}} = cAttn(f_v, f_{\text{bev}}, f_{\text{bev}}) \tag{8}$$

where $cAttn(.)$ is cross attention [31] that takes query, key, and value as input and outputs the cross-attended feature. Here, $f_{\text{bev}} \in F_{\text{bev}}$ refers to a spatial feature corresponding to the box that $v_{t=T}$ is made from.

The Points-Region Correlation Features f_{gen} can then be fed into any existing second-stage refinement structures [5, 12, 38] following their formulation to refine the box from the first-stage detection b_1.

3.5. Overall Training

Our overall training losses are defined the same as the general RPN learning for detection, as the proposed modules do not need to change the formulation of the existing works. In general, the first stage detection is learned with loss L_{1st} as:

$$L_{\text{1st}} = L_{\text{obj}} + L_{\text{reg}} \tag{9}$$

where L_{obj} learns the objectness score of proposals using Focal Loss [16] and L_{reg} learns the box regression of the proposals. The second refinement modules generally uses loss L_{ref} as:

$$L_{\text{2nd}} = L_{\text{cls}} + L_{\text{ref}} \tag{10}$$

where L_{cls} learns the classification of the object type and L_{ref} learns to refine the box from the first-stage prediction, which is B_1.

Overall, our training loss L is defined as:

$$L = L_{\text{diff}} + L_{\text{1st}} + L_{\text{2nd}} \tag{11}$$

4. Experiment

4.1. Datasets

We conduct a comprehensive evaluation of our proposed methods against multiple baseline approaches across three widely used benchmark datasets: KITTI [7], NuScenes [1], and Waymo [28]. The KITTI dataset comprises 7,481 LiDAR point cloud frames for training and validation, all captured using a 64-beam Velodyne LiDAR. The NuScenes dataset consists of 28,130 training frames and 6,019 validation frames, collected with a 32-beam LiDAR. The Waymo dataset provides a significantly larger-scale collection, including 122,000 training frames and 30,407 validation frames, captured using a multi-LiDAR setup comprising one 64-beam LiDAR and four 200-beam LiDARs.

4.2. Implementation Details

Our diffusion model consists of three Transformer blocks followed by a 3D convolutional layer to output the 3D offsets of dense voxel grid, $o_{t=T}$, given the input $v_{t=T}$. Here, the dense grid parameters are set as $W = 32$, $H = 32$, $D = 32$. The 3D encoder to produce f_v consists of three 3D convolutional layers followed by MLPs to output the embedding with $K = 512$ dimension for the cross attention with f_{bev}. Following existing works on domain adaptive detection in point cloud [9, 15, 26, 39, 40], we test our model on two base detectors, Second IoU [38] and PointPillars [12]. They are widely used and applicable to most recent detectors. Network parameters used for the experiments are from ST3D [40]. Similar to [15, 26], we first train each detector for 50 epochs with batch-size 4 as a pretraining step using two NVIDIA A10 GPUs. Following the same parameters as [26, 40], the self-training stage trains 30 more epochs

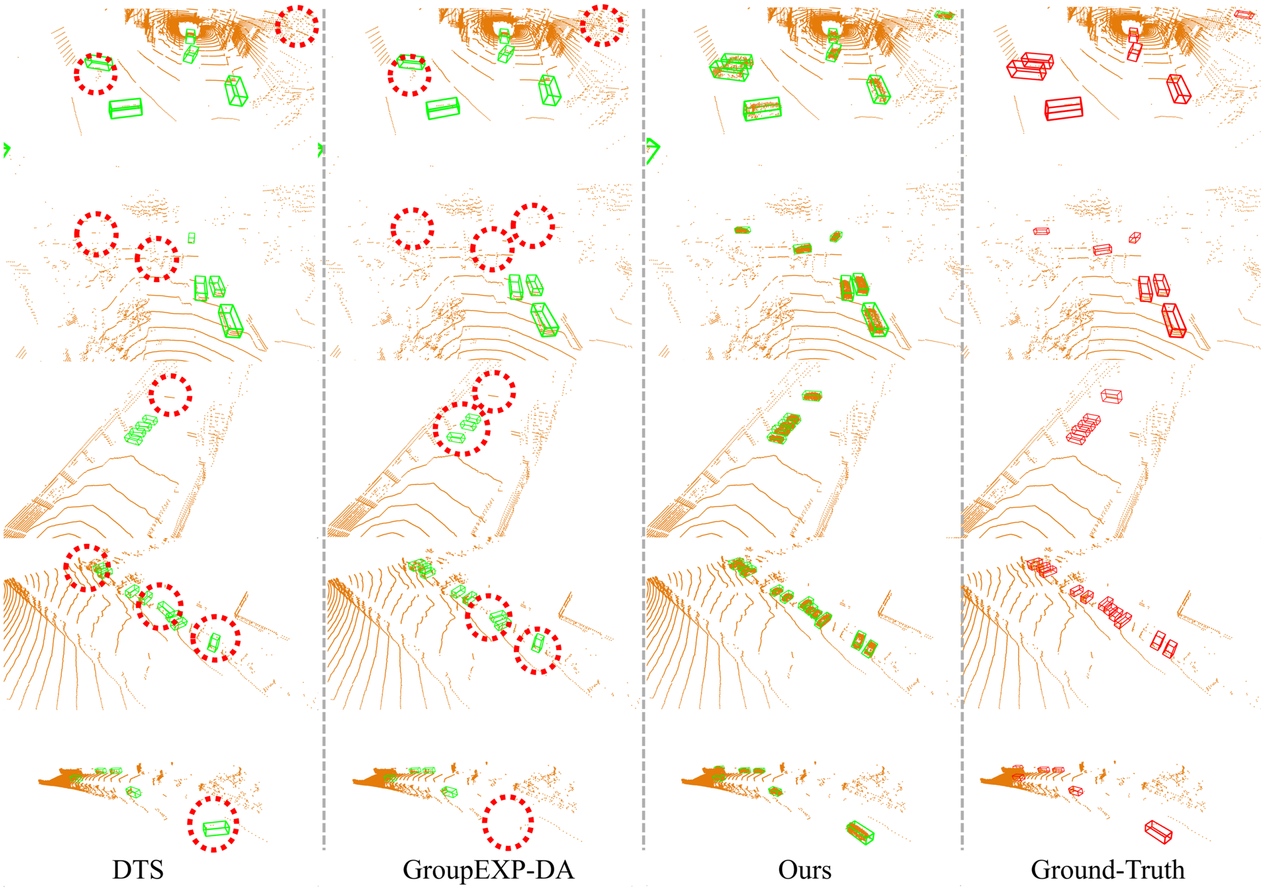


Figure 5. Qualitative comparison of Baseline DTS [9], GroupEXP-DA [26] and ours on Waymo [28] to NuScenes [1] adaptation scenario (rows 1-3) and NuScenes [1] to KITTI (rows 4-5) adaptation scenarios.

to adapt to the target domain. We utilize Adam optimizer with Cosine annealing [42] for scheduling the learning rate, which is set to 0.0015.

4.3. Comparing Methods

We compare recent existing 3D domain adaptive detection methods, such as SN [34], 3D-CoCo [41], ST3D [39, 40], GPA-3D [15], DTS [9], and GroupEXP-DA [26] with our proposed method. As our method is based on the self-training, we set ST3D [40] as our baseline and show experimental results by comparing with more recent methods. Additionally, we also illustrate the performance of the **oracle** models, which refer to a fully-supervised model on the target domain directly as an upper bound. Following the most recent works [9, 15], all methods are compared in three adaptation scenarios focusing on "car" class: (1) Waymo → NuScenes (2) NuScenes → KITTI (3) Waymo → KITTI.

4.4. Evaluation Metric

Following [9, 15, 26, 39, 40], we evaluate the performance using the metrics AP_{BEV}, AP_{3D}, and closed gap, where Closed Gap $= \frac{AP_{\text{model}} - AP_{\text{source}}}{AP_{\text{oracle}} - AP_{\text{source}}} \times 100$ [40].

4.5. Quantitative Results

Waymo → KITTI Table 1 (first task) shows the quantitative results of 3D detection in AP_{BEV} and AP_{3D}. When using Second-IoU as detector [38], our proposed method outperforms the baseline ST3D series for 7.03/9.61 in $AP_{\text{BEV}}/AP_{\text{3D}}$, respectively. Compared with the SOTA method, GroupExp-DA [26], 0.87/1.54 improvements are achieved. When using PointPillars as the base detector [12], our approach outperforms the previous best-performing method by 6.29/1.73.

NuScenes → KITTI As shown in Table 1 (second task), our approach shows 2.28/2.92 gains over SOTA in terms of $AP_{\text{BEV}}/AP_{\text{3D}}$ with Second-IoU [38] as the base detector. With PointPillars [12] as the base detector, our approach exceeds the baseline and GroupExp-DA by 26.23/43.06 and 4.74/1.32, respectively. Furthermore, Table 2 shows the adaptation performance of the categories *Pedstrian* and *Cyclist*. *DiffRefine* exceeds the best-performing method by 1.97/1.43 for *Pedestrian* and 1.35/0.81 for *Cyclist*.

Waymo → NuScenes Table 1 (third task) illustrates the adaptation results. Our approach outperforms the baseline and the best-performing method [27] by 8.97/5.06 and 1.05/0.83

Task	Methods	SECOND-IOU		PointPillars	
		AP$_{\text{BEV}}$ (↑)/AP$_{\text{3D}}$ (↑)	Closed Gap (↑)	AP$_{\text{BEV}}$ (↑)/AP$_{\text{3D}}$ (↑)	Closed Gap (↑)
Waymo → KITTI	Source Only	67.64/27.48	-	47.8/11.5	-
	SN [34]	78.96/59.20	72.33/69.00	27.4/6.4	55.14/8.49
	3D-CoCo [41]	-	-	76.1/42.9	76.49/52.25
	ST3D [40]	82.19/61.83	92.97/74.72	58.1/23.2	27.84/19.47
	ST3D++ [39]	80.78/65.64	83.96/83.01	-	-
	GPA-3D [15]	83.79/70.88	103.19/94.41	77.29/50.84	79.70/65.46
	DTS [9]	85.80/71.50	115.9/95.7	76.1/50.2	76.50/64.4
	GroupExp-DA [26]	86.94/73.70	123.2/100.4	78.44/54.11	82.81/71.0
	DiffRefine (Ours)	87.81/75.25	128.79/103.80	81.73/55.84	91.53/73.77
	Oracle	83.3/73.5	-	84.8/71.6	-
NuScenes → KITTI	Source Only	51.8/17.9	-	22.8/0.5	-
	SN [34]	59.7/37.6	25.1/35.4	39.3/2.0	26.6/2.1
	3D-CoCo [41]	-	-	77.0/47.2	87.4/65.7
	ST3D [40]	75.9/54.1	76.6/59.5	60.4/11.1	60.6/14.9
	ST3D++ [39]	80.5/62.4	91.1/80.0	-	-
	DTS [9]	81.4/66.6	94.0/87.6	79.5/51.8	91.5/72.2
	GroupExp-DA [26]	81.47/68.2	98.3/90.0	81.89/52.84	95.3/73.6
	DiffRefine (Ours)	83.751/71.13	101.43/95.74	86.63/54.16	102.83/75.47
	Oracle	83.3/73.5	-	84.8/71.6	-
Waymo → NuScenes	Source Only	32.91/17.24	-	27.8/12.1	-
	SN [34]	33.23/18.57	1.69/7.54	28.1/12.98	2.41/4.58
	3D-CoCo [41]	-	-	33.1/20.7	25.00/44.79
	ST3D [40]	35.92/20.19	15.87/16.73	30.6/15.6	13.21/18.23
	ST3D++ [39]	35.73/20.90	14.87/20.76	-	-
	GPA-3D [15]	37.25/22.54	22.88/30.06	35.47/21.01	36.18/46.41
	DTS [9]	41.2/23.0	43.7/32.80	42.2/21.5	67.9/49.0
	GroupExp-DA [26]	43.84/24.42	57.56/40.66	44.31/22.15	77.88/52.34
	DiffRefine (Ours)	44.89/25.25	64.09/45.37	45.51/22.79	83.54/55.68
	Oracle	51.9/34.9	-	49.0/31.3	-

Table 1. Quantitative comparisons of the recent domain adaptive 3D detection methods on three adaptation scenarios. The top-, second- and third- performing methods are labeled in different colors.

Method	Pedestrian	Cyclist
	AP$_{\text{BEV}}$/AP$_{\text{3D}}$	AP$_{\text{BEV}}$/AP$_{\text{3D}}$
Source Only	39.95/34.57	17.70/11.08
ST3D [40]	44.00/42.60	29.58/21.21
DTS [9]	48.65/45.87	30.76/21.93
GroupExpDA [26]	49.23/46.56	32.17/23.48
DiffRefine (Ours)	51.20/48.0	33.52/24.29
Oracle	46.64/41.33	62.92/60.32

Table 2. Quantitative comparison of recent adaptation methods for *pedestrian* and *cyclist* categories in NuScenes → KITTI adaptation scenario with Second-IoU as a base detector.

in $AP_{\text{BEV}}/AP_{\text{3D}}$, respectively, with Second-IoU as the detector. Similarly, with PointPillars as the detector, 14.90/7.19 and 32.8/1.94 improvements are gained compared with the baseline and SOTA in terms of $AP_{\text{BEV}}/AP_{\text{3D}}$.

4.6. Qualitative Results

Figure 5 visually compares SOTA methods, such as DTS [9], GroupEXP-DA [26] and ours. Notably, for each existing method, most of the false negatives are caused by distant objects with sparse points, as they do not contain distinctive features. Interestingly, due to the density insensitive nature, DTS handles objects with sparse points better than GroupEXP-DA (row 2, row 3). However, DTS fails to accurately infer the rotation (row 4, row 5) when the object points are sparse due to distance. Despite the fact that GroupEXP-DA has more false negatives for objects with sparse points, it handles the different shape better than DTS (row 4) due to its diverse group-based detection. On the other hand, our proposed method is able to densify the sparse points of objects regardless of distance, demonstrating its effectiveness, as can be additionally seen in Fig. 8.

	Differentiable	f_{gen}	AP_{BEV}	AP_{3D}
(a)			42.64	23.73
(b)	✓		43.31	24.19
(c)		✓	44.30	24.82
(d)	✓	✓	44.89	25.25

Table 3. Impact of spatial context feature and differentiability in AP_{BEV} and AP_{3D} on Waymo → NuScenes adaptation.

4.7. Ablations

Impact of Differentiable Warping and f_{gen} is illustrated in Table 3 in Waymo→NuScenes adaptation using Second IoU [38] in terms of $AP_{\text{BEV}}/AP_{\text{3D}}$.

For the experiment on differentiable warping, we manually cut the gradient on $v_{t=0}$ after denoising and feed it as input to the refinement module to see the impact. For the experiment on f_{gen}, we directly feed $v_{t=0}$ as input to the

Voxel Grid Size	Voxel Cls.		Diffusion	
	AP_{BEV}	AP_{3D}	AP_{BEV}	AP_{3D}
(W, H, Z)=(8,8,8)	36.27	21.58	42.87	23.74
(W, H, Z)=(16,16,16)	36.62	21.87	44.14	24.74
(W, H, Z)=(32,32,32)	35.91	20.44	44.89	25.25

Table 4. Quantitative comparisons of generation methods w.r.t. different voxel grid sizes in Waymo → NuScenes adaptation.

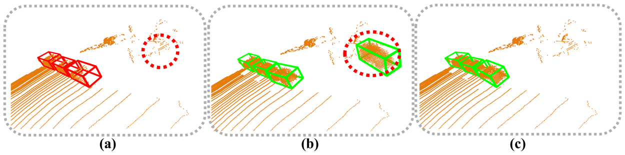


Figure 6. Illustration of (a) ground-truth, (b) detection output without spatial context f_{gen}, (c) detection output with f_{gen}. f_{gen} improves the false positive generation problem indicated by red dotted circles in (a) and (b) (i.e. hallucination) by providing the context of a surrounding area, avoiding the generation of a car generated where there should not be a car.

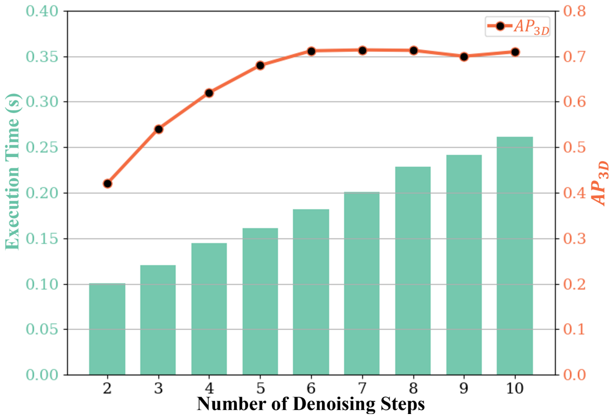
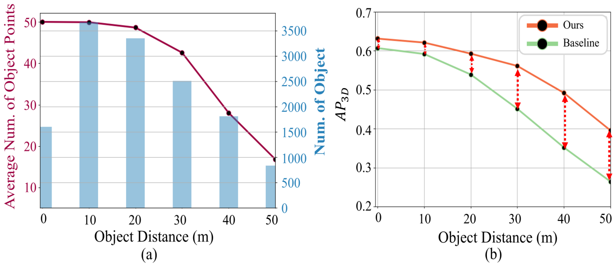


Figure 7. Runtime analysis with the execution time (***bar***) and AP_{3D} (***line***) with respect to number of denoising steps used for the generation in NuScenes → KITTI adaptation.

refinement module without fusing the spatial context feature f_{bev} to see how it affects performance. Notably, excluding f_{gen} (Table 3 (b)) leads to the biggest performance drop of 1.58/1.06. This is expected as f_{gen} encodes spatial context, which can provide additional signals for refinement to avoid false positive detection, as shown in Fig. 6.

Cutting the gradient flow from object point generation to second stage module (Table 3 (c)) also deteriorates the performance of 0.59/0.43 as it prevents the joint optimization of two modules, where each module could learn helpful feature for each other. When both differentiable $v_{t=0}$ and f_{gen} are used, the performance reaches the best, making 2.25/1.52 improvement compared to the setting where none of them are used (Table 3(a)).

Impact of Generation Steps is illustrated in Fig. 4 and Fig. 7. Increasing generation steps from 2 to 6 shows gradual improvement in AP_{3D} of 0.12, 0.08, 0.06, and 0.03 between individual steps. However, the performance reaches the best when using 6 steps for denoising. This is in accordance with visual comparison in Fig. 4, where the generated object points proceed to contain more clear features as the steps increase. The overall execution time monotonously increases around 0.02 per each denoising step.



(a)

(b)

Figure 8. Average number of points on object surface (a)-***line***, number of existing objects (a)-***bar***, and AP_{3D} (b) with respect to distance between objects and LiDAR, showing negative correlation between object distance and detection performance due to sparsity for a substantial number of objects. The resulting domain adaptation from NuScenes→KITTI in (b) shows a growing gap in performance between the baseline [40] and ours as the object distance increases.

Impact of Diffusion based Generation is shown in Table 4. The aim of this experiment is to see how effective Diffusion based generation is compared to existing technique [37]. Specifically, we formulate the generation as a dense binary classification of $v_{t=T}$ problem, where each voxel grid is considered occupied if the predicted output for the grid is higher than 0.5, following [37]. Except for changing the learning task, all the network configurations stay the same as Diffusion-based generation for each grid size. As can be seen, the performance of detection when using diffusion-based generation outperforms the voxel classification method (Voxel Cls.) in all the configurations for at least 6.59/2.16 in $AP_{\text{BEV}}/AP_{\text{3D}}$. Interestingly, the classification method shows better performance when the grid sizes are smaller than (32,32,32), probably because one-step classification is unable to learn the complexity from the larger grid. In contrast, the performance of Diffusion-based generation deteriorates as the grid size decreases, suggesting that the multiple steps of denoising enable the learning of more complex shapes of object points in larger grid sizes.

5. Conclusion

In this paper, we present *DiffRefine*, which learns to generate proposal-specific object points for domain adaptive detection. We improve the box refinement process by densifying the object points to be more distinctive. By formulating the challenging points generation problem as denoising from Diffusion process, *DiffRefine* shows significant improvement, particularly for distant objects, where the object point naturally becomes sparse, leading to false negative detection.

Acknowledgment

This work was supported by Innovate UK Grant No: 10107189 “ScanSpot: 3D-Modelling “Digital Twin” Data For New Insurance Products”.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2, 5, 6

[2] Xiangyu Chen, Zhenzhen Liu, Katie Luo, Siddhartha Datta, Adhitya Polavaram, Yan Wang, Yurong You, Boyi Li, Marco Pavone, Wei-Lun Harry Chao, et al. Diffubox: Refining 3d object detection with point diffusion. *Advances in Neural Information Processing Systems*, 37:103681–103705, 2025. 2, 3

[3] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Largekernel3d: Scaling up kernels in 3d sparse cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13488–13498, 2023. 2

[4] Zhuoxiao Chen, Yadan Luo, Zheng Wang, Mahsa Baktashmotlagh, and Zi 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. 2

[5] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. CenterNet: Keypoint Triplets for Object Detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 5

[6] Lue Fan, Ziqi Pang, Tianyuan Zhang, Yu-Xiong Wang, Hang Zhao, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Embracing single stride 3d object detector with sparse transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8458–8468, 2022. 2

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *CVPR*, 2012. 2, 5

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[9] Q. Hu, D. Liu, and W. Hu. Density-Insensitive Unsupervised Domain Adaption on 3D Object Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2, 5, 6, 7

[10] Xin Jin, Kai Liu, Cong Ma, Ruining Yang, Fei Hui, and Wei Wu. Swiftpillars: High-efficiency pillar encoder for lidar-based 3d detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2625–2633, 2024. 2

[11] Se-Ho Kim, Inyong Koo, Inyoung Lee, Byeongjun Park, and Changick Kim. Diffref3d: A diffusion-based proposal refinement framework for 3d object detection. *arXiv preprint arXiv:2310.16349*, 2023. 2

[12] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019. 2, 5, 6

[13] Jianan Li, Shaocong Dong, Lihe Ding, and Tingfa Xu. Mssvt++: Mixed-scale sparse voxel transformer with center voting for 3d object detection. *IEEE transactions on pattern analysis and machine intelligence*, 2023. 2

[14] Yanwei Li, Yilun Chen, Xiaojuan Qi, Zeming Li, Jian Sun, and Jiaya Jia. Unifying voxel-based representation with transformer for 3d object detection. *Advances in Neural Information Processing Systems*, 35:18442–18455, 2022. 2

[15] Ziyu Li, Jingming Guo, Tongtong Cao, Bingbing Liu, and Wankou Yang. GPA-3D: Geometry-aware Prototype Alignment for Unsupervised Domain Adaptive 3D Object Detection from Point Clouds. In *International Conference on Computer Vision (ICCV)*, 2023. 2, 5, 6, 7

[16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 5

[17] Katie Luo, Zhenzhen Liu, Xiangyu Chen, Yurong You, Sagie Benaim, Cheng Perng Phoo, Mark Campbell, Wen Sun, Bharath Hariharan, and Kilian Q Weinberger. Reward finetuning for faster and more accurate unsupervised object discovery. *Advances in Neural Information Processing Systems*, 36:13250–13266, 2023. 3

[18] Shitong Luo and Wei Hu. Diffusion Probabilistic Models for 3D Point Cloud Generation. In *CVPR*, 2021. 2

[19] Wenxin Ma, Jian Chen, Qing Du, and Wei Jia. PointDrop: Improving Object Detection from Sparse Point Clouds via Adversarial Data Augmentation. In *International Conference on Pattern Recognition (ICPR)*, 2021. 1

[20] Xidong Peng, Xinge Zhu, and Yuexin Ma. CL3D: Unsupervised Domain Adaptation for Cross-Lidar 3D Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2023. 2

[21] Cristiano Saltori, Stéphane Lathuiliére, Nicu Sebe, Elisa Ricci, and Fabio Galasso. Sf-uda 3d: Source-free unsupervised domain adaptation for lidar-based 3d object detection. In *2020 International Conference on 3D Vision (3DV)*, pages 771–780. IEEE, 2020. 2

[22] Guangsheng Shi, Ruifeng Li, and Chao Ma. Pillarnet: Real-time and high-performance pillar-based 3d object detection. In *European Conference on Computer Vision*, pages 35–52. Springer, 2022. 2

[23] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10529–10538, 2020. 2

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

[25] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN++: Point-Voxel Feature Set Abstraction With Local Vector Representation for 3D Object Detection. *arXiv:2102.00463*, 2021. 2

[26] Sangyun Shin, Yuhang He, Madhu Vankadari, Ta-Ying Cheng, Qian Xie, Andrew Markham, and Niki Trigoni. Towards learning group-equivariant features for domain adaptive 3d detection. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 5, 6, 7

[27] Sangyun Shin, Kaichen Zhou, Madhu Vankadari, Andrew Markham, and Niki Trigoni. Spherical mask: Coarse-to-fine 3d point cloud instance segmentation with spherical representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 6

[28] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 5, 6

[29] Pei Sun, Mingxing Tan, Weiyue Wang, Chenxi Liu, Fei Xia, Zhaoqi Leng, and Dragomir Anguelov. Swformer: Sparse window transformer for 3d object detection in point clouds. In *European Conference on Computer Vision*, pages 426–442. Springer, 2022. 2

[30] Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, Karsten Kreis, et al. Lion: Latent point diffusion models for 3d shape generation. *Advances in Neural Information Processing Systems*, 35:10021–10039, 2022. 2

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 5

[32] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. PointAugmenting: Cross-Modal Augmentation for 3D Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11794–11803, 2021. 1

[33] Haiyang Wang, Chen Shi, Shaoshuai Shi, Meng Lei, Sen Wang, Di He, Bernt Schiele, and Liwei Wang. Dsvt: Dynamic sparse voxel transformer with rotated sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13520–13529, 2023. 2

[34] 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. 2, 3, 6, 7

[35] Zhenyu Wang, Ya-Li Li, Xi Chen, Hengshuang Zhao, and Shengjin Wang. Uni3detr: Unified 3d detection transformer. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[36] Yi Wei, Zibu Wei, Yongming Rao, Jiaxin Li, Jie Zhou, and Jiwen Lu. Lidar distillation: Bridging the beam-induced domain gap for 3d object detection. In *European Conference on Computer Vision*, pages 179–195. Springer, 2022. 2

[37] Qiangeng Xu, Yin Zhou, Weiyue Wang, Charles R Qi, and Dragomir Anguelov. Spg: Unsupervised domain adaptation for 3d object detection via semantic point generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15446–15456, 2021. 2, 4, 8

[38] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 2, 5, 6, 7

[39] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d++: Denoised self-training for unsupervised domain adaptation on 3d object detection. *arXiv preprint arXiv:2108.06682*, 2021. 3, 5, 6, 7

[40] Jihan Yang, Shaoshuai Shi, Zhe Wang, and Xiaojuan Qi. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 3, 5, 6, 7, 8

[41] Zeng Yihan, Chunwei Wang, Yunbo Wang, Hang Xu, Chaoqiang Ye, Zhen Yang, and Chao Ma. Learning transferable features for point cloud detection via 3d contrastive co-training. *Advances in Neural Information Processing Systems*, 34:21493–21504, 2021. 2, 6, 7

[42] Biao Zhang and Peter Wonka. Point cloud instance segmentation using probabilistic embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8883–8892, 2021. 6

[43] Gang Zhang, Chen Junnan, Guohuan Gao, Jianmin Li, and Xiaolin Hu. Hednet: A hierarchical encoder-decoder network for 3d object detection in point clouds. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[44] Xiao Zheng, Xiaoshui Huang, Guofeng Mei, Yuenan Hou, Zhaoyang Lyu, Bo Dai, Wanli Ouyang, and Yongshun Gong. Point cloud pre-training with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22935–22945, 2024. 4

[45] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5826–5835, 2021. 2

[46] Xin Zhou, Jinghua Hou, Tingting Yao, Dingkang Liang, Zhe Liu, Zhikang Zou, Xiaoqing Ye, Jianwei Cheng, and Xiang Bai. Diffusion-based 3d object detection with random boxes. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 28–40. Springer, 2023. 2

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