Domain Generalization of 3D Object Detection by Density-Resampling

Shuangzhi Li[1], Lei Ma[2,1], and Xingyu Li[1] *

[1]University of Alberta, Canada [2]The University of Tokyo, Japan
{shuangzh, xingyu}@ualberta.ca, ma.lei@acm.org

Abstract. Point-cloud-based 3D object detection suffers from performance degradation when encountering data with novel domain gaps. To tackle it, single-domain generalization (SDG) aims to generalize the detection model trained in a limited single source domain to perform robustly on unexplored domains. Through analysis of errors and missed detections in 3D point clouds, it has become evident that challenges predominantly arise from variations in point cloud density, especially the sparsity of point cloud data. Thus, in this paper, we propose an SDG method centered around the theme of point cloud density resampling, which involves using data augmentation to simulate point clouds of different densities and developing a novel point cloud densification algorithm to enhance the detection accuracy of low-density point clouds. Specifically, our physical-aware density-resampling data augmentation (PDDA) is the first to consider the physical constraints on point density distribution in data augmentation, leading to a more realistic simulation of variation in cloud density. In systematic design, an auxiliary self-supervised point cloud densification task is incorporated into the detection framework, forming a basis for test-time model update. By manipulating point cloud density, our method not only increases the model's adaptability to point clouds of different densities but also allows the self-supervised densification algorithm to serve as a metric for assessing the model's understanding of the environment and semantic information. This, in turn, enables a test-time adjustment of the model to better adapt to varying domains. Extensive cross-dataset experiments covering "Car", "Pedestrian", and "Cyclist" detections demonstrate our method outperforms state-of-the-art SDG methods and even overpass unsupervised domain adaptation methods under some circumstances. The code is released at https://github.com/xingyu-group/3D-Density-Resampling-SDG.

Keywords: Domain generalization, 3D object detection, Density resampling, Data Augmentation

1 Introduction

LiDAR-based 3D object detection is a crucial task in real-world applications, such as autonomous driving [1,32] and UAV sensing [15,55]. Deep learning-based

* Corresponding Author.

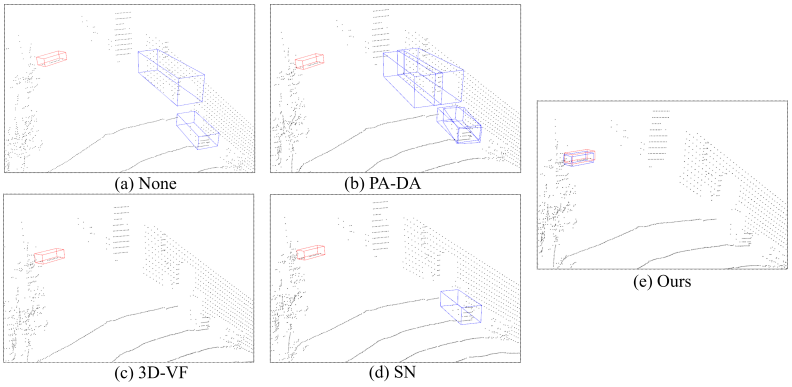


Fig. 1: Detection performance comparison w.r.t. Waymo → NuScenes, where the red boxes are ground-truth 3D boxes and blue ones are detected 3D boxes. Our method achieves better performance than other SDG methods, *PA-DA* [8] and *3D-VF* [21], and even UDA method *SN* [45] (refers to Table 1 for statistical details).

models [1,10,32,35,51,55] are the mainstream for 3D object detection. However, when these models encounter data with domain gaps (*e.g.*, adverse weather [13, 14, 19], diverse laser scanning densities [17], and shifted object sizes [45]), their performance declines significantly. To tackle it, unsupervised domain adaptation (UDA) [17, 45, 52] focuses on adapting the model to a new domain distribution during training with the help of unlabeled target data. Targeting a more general case, domain generalization (DG) [18, 29, 42] endeavors to enhance the model's generalizability with no access to target data during model training.

The approaches of domain generalization in 3D object detection can be primarily categorized into multi-domain generalization and single-domain generalization. Multi-domain generalization (MDG) [38, 48] utilizes diverse source domain knowledge to bridge the unseen target domain gaps. Given the high diversity of real-world domains, 3D data collection and label annotation for multiple source domains is costly. In this paper, we focus on the more universal yet more challenging single-domain generalization (SDG) setting, where the model learns from a single source domain only. The primary challenge of SDG lies in how to rely on limited single-domain information to enable a model to achieve domain-invariant learning and therefore perform robustly on diverse unseen domains. Though single-domain generalization [58] has gathered significant attention in various 2D image tasks (*e.g.*, classification [46], detection [41], and segmentation [30]), it's still under-explored in the context of the 3D point cloud, especially for object detection. Prior works in SDG of 3D object detection like *PA-DA* [8] and *3D-VF* [21] usually utilize data augmentation to synthesize more 3D data, aiming to eliminate domain-dependent information in model learning. Yet, no effort has been reported from the learning methodology perspective.

When examining the problem of 3D point cloud detection, we observe that *the low point density is highly correlated to object miss-detection*. As demonstrated

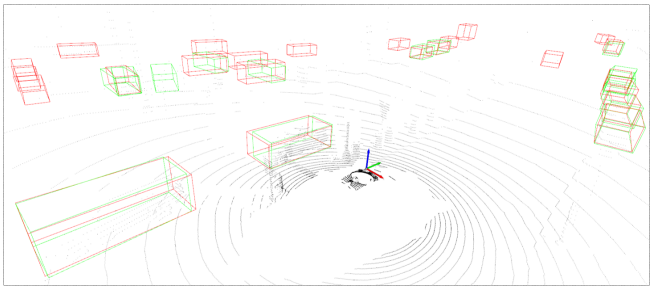


Fig. 2: Sparse point clouds poses challenges to 3D object detection. The visualizations are generated by VoxelRCNN [9] with a voxel-based backbone on NuScenes. Due to the blockage by front objects and various distances, cars with sparse scanning are hard to detect. (red boxes are ground-truth 3D boxes and green ones are detected 3D boxes).

in Figure 2, object occlusion [50] causes significant local point density reduction, and various distances between objects and the laser sensor [16] also cause variations in the point density on imaged objects. In inter-domain cases, further considering potential adverse weather (*e.g.*, rain [19], snow [13], and fog [14]) that may cause the reduced intensity of reflected light resulting in sparse laser scans and the diverse 3D sensors with different scanning beam layers, we hypothesize that the issue introduced by point density variations becomes more pronounced in inter-domain object detection. In this regard, we design a novel SDG framework with density resampling as the central pivot, encapsulating both a universal physical-aware density-resampling data augmentation (PDDA) and a novel self-supervised point cloud densification algorithm. Specifically, PDDA bridges the domain gap mainly introduced by various point densities. Unlike prior 3D DG data augmentation methods like *PA-DA* [8] and *3D-VF* [21] which modify point density in a localized and random manner, PDDA innovatively aligns with real-world physical constraints to reshape point density, ensuring the accurate and reliable representation of diverse density patterns for augmentation. Furthermore, we innovatively incorporate a sparse point cloud densification algorithm into the detection model and thus construct a self-supervised manner for assessing the model's understanding of the environment and semantic information. During model deployment, this auxiliary self-supervised algorithm densifies the down-sampled query point clouds and leverages the discrepancy between its output and the original data to enable model updates to better adapt to the target domains with lightweight optimization. Noteworthy, as the first exploration in the literature to use test-time model adaptation to address DG for 3D object detection, this study provides another viable technique beyond data augmentation, offering inspiration for future research.

In summary, this work makes the following contributions:

– We design a straightforward yet effective universal physical-aware density-resampling data augmentation method for source training. Extensive experi-

ments show that such a simply physically-aware design surpasses all existing DG augmentation methods and even some UDA methods.
– We introduce a novel point cloud densification strategy to enhance model's scene comprehension. This density resampling strategy enables lightweight updates to adapt the model to target domains during testing.
– Extensive cross-dataset experiments covering "Car", "Pedestrian", and "Cyclist" detections demonstrate our model outperforms the state-of-the-art DG methods and even overpass UDA methods under some circumstances.

2 Relate Work

Point-cloud-based 3D object detection aims to locate and classify objects of interest in point clouds [1, 9, 32, 33, 35, 51]. Existing detection methods can be categorized into voxel-based and point-based. Voxel-based methods [9,10,20,26,51] voxelize point clouds into grid-like 3D images and use sparse convolution to extract abstract features for object detection, which are computationally efficient but may lose fine-grained details due to voxelization. The point-based methods [35, 36] extract abstract features directly from raw points, preserving fine-grained details but possessing relatively heavy computation. As a trade-off, some point-voxel-mixed methods [33, 34] combine raw points and voxels to balance representation learning and computation efficiency. In this article, we mainly focus on voxel-related object detection.

Domain Generalization on Object Detection aims to bridge domain shifts unseen during the source training but may encountered in the target testing [18, 29, 42]. In 2D image tasks, DG has been well explored by data augmentation [37,42,43], domain alignment learning [28,29], meta-learning [2,22], test-time adaptation [6,18], *etc*. In the realm of 3D object detection, existing works [38,48] mainly focus on multi-domain generalization. Wu *et al*. [48] propose to align extracted features of detectors individually trained on multiple sources to learn domain-invariant features. Soum-Fontez *et al*. [38] utilize label re-annotation and cross-dataset instance injection to mitigate the performance degradation. However, regarding single-domain cases, although well-explored in 2D image detection [41,47,58], there has been relatively rare research in 3D object detection. Lehner *et al*. [21] propose an adversarial augmentation to deal with rare-shape or broken cars. Choi *et al*. [8] combine five basic augmentations to build the part-aware data augmentation to handle noise and locally missing points. Unlike prior works solely focusing on data augmentation, our work not only introduces a novel physical-aware data augmentation method but also contributes a new point cloud densification strategy for test-time model update in methodology.

Test-Time Adaptation against Domain Shift refers to conducting adjustments or refinements to a model's parameters or inferences during the inference phase [5, 23]. Current methods include pseudo-labeling [7, 24], consistency training [44], self-supervised learning [5, 27], *etc*. Recently, the increasing attention

has landed on using test-time adaptation to tackle the domain generalization problem in 2D tasks [5–7, 18, 24, 25] and 3D classification [27, 44]. To the best of our knowledge, our work is the first to apply the test-time adaptation to the SDG for 3D object detection.

3 Problem Formulation

3D object detection aims to detect objects of interest in 3D point clouds. A frame of point cloud $\mathbf{X}$ is a set of points $p = [x^p, y^p, z^p]$, where $[x^p, y^p, z^p]$ denotes the 3D Cartesian coordinates of points. The object detection can be formulated as $f(\mathbf{X}) = \{\mathbf{b}_i\}_{i=1}^{N_{obj}}$, where $f(\cdot)$ denotes the detection model and N_{obj} is the number of detected 3D boxes in $\mathbf{X}$. The $\mathbf{b}_i$ denotes the i^{th} detected box, which contains the location $[x_i, y_i, z_i]$, the dimension $[w_i, h_i, l_i]$, the heading angle θ_i, the classification label c_i, and the confidence score s_i.

Domain generalization for 3D object detection aims to generalize the model $f(\cdot)$ trained on a well-labeled source-domain dataset $\{\mathbf{X}_j^s, \mathbf{B}_j^s\}_{j=1}^{N^s}$ to a unseen target domain $\{\mathbf{X}_j^t, \mathbf{B}_j^t\}_{j=1}^{N^t}$, where the superscript s and t denote the source and target domain respectively. Correspondingly, N^s and N^t denote the number of point clouds, and $\mathbf{B}_j^s$ and $\mathbf{B}_j^t$ denote the sets of 3D boxes. Note that in domain generalization, only the source-domain data is available for model training. The target-domain data is accessible during model evaluation/deployment only.

4 Methodology

Point density plays a crucial role in 3D object detection. It is affected by several factors, including the distances between objects in relation to LiDAR sensors, inter-object occlusion, and different weather conditions. Furthermore, various types of LiDAR sensors can introduce variations in point density, which can negatively impact the model's ability to generalize to unseen environments.

To tackle these challenges, we introduce a density-resampling-centered solution for DG on 3D point cloud data. Figure 3 presents the system diagram of the proposed framework. During training on the source domain, the point-cloud data is augmented by our PDDA method. Then, the augmented sample is fed into the multi-task learning scheme, where the auxiliary self-supervised task (i.e., 3D point cloud densification) promotes the encoder to extract features for scene comprehension. During testing on the target domain, we utilize the auxiliary point cloud densification loss to update parameters in the encoder-decoder path to adapt the encoder to unseen target domains, further bridging the domain gap. The test-time adapted encoder, together with the frozen detection head, is then exploited for the final object detection.

4.1 Physical-Aware Density-Resampling Data Augmentation

When designing data augmentation, we are aware that the variations in point density due to various factors may follow distinct physical constraints. For ex-

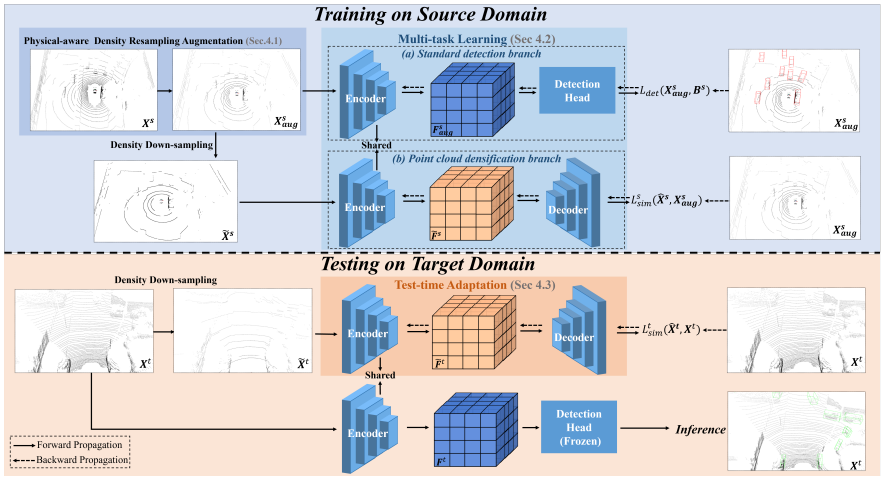


Fig. 3: Pipeline of our proposed DG method. During training on the source domain, the training sample is augmented with density re-sampling, which is then used to train the multi-task model for (a) standard detection and (b) point cloud densification from its down-sampled version. During testing on the target domain, given a query sample, self-supervised point cloud densification is conducted on the corresponding density-downsampled version for lightweight model update. Then, the updated encoder works together with the frozen detection head for the final prediction. In this figure, source and target samples are from NuScenes [4] and KITTI [11], respectively.

ample, point density is inversely proportional to the imaging distance. Furthermore, for spinning sensors, different beam layers of the sensor can lead to varying uniform-distributed densities in the vertical direction. To address the domain-specific bias arising from variations in point densities, Hu *et al*. [17] proposed a random beam re-sampling (RBRS) method. Nonetheless, this solely randomly layer-sampling strategy does not adhere to the previously mentioned physical constraints, resulting in suboptimal outcomes. In this study, we design a physical-aware density-resampling data augmentation (PDDA) method to better simulate the real-world point cloud distribution, accounting for different density patterns.

Given the point clouds of the source dataset with M beam layers of LiDAR scanning, we first convert the Cartesian coordinates $[x, y, z]$ into spherical coordinates $[\gamma, \Theta, \Phi]$:

$$\gamma = \sqrt{x^2 + y^2 + z^2}, \Theta = \arctan \frac{y}{x}, \Phi = \arccos \frac{z}{\gamma}. \tag{1}$$

Considering the spinning beams commonly used in autonomous driving, we gather vertical angles $\{\Phi\}$ of points. After removing outliers of $\{\Phi\}$ (out of $3.1\times$ standard deviation), we obtain the range of vertical angles $[\min\{\Phi\}, \max\{\Phi\}]$. Then we uniformly divide the range into M bins $[\Phi_k, \Phi_{k+1}]$ ($k \in \{1, 2, \cdots, M\}$) and label points according to which bin the points land on. Then we perform

our density-resampling data augmentation on source data by comprehensively considering physical-aware down-sampling and up-sampling.

To uniformly *down-sample* the density of the point cloud into a lower density, we keep one out of every C bins of points. A higher value of C means a more severe reduction in density. Additionally, we remove points with a probability of P to simulate the global loss of laser reflections. To *up-sample* the density of the point cloud into a higher density, we adopt efficient linear interpolation to obtain new points. Specifically, new points will be interpolated by original points within two neighboring bins:

$$\eta_s^{new} = \lambda\eta_k + (1-\lambda)\eta_{k+1} \text{ s.t. } s \in \{1, 2, \cdots, S-1\}, \lambda = \frac{s}{S}, \tag{2}$$

where $\eta \in \{\gamma, \Theta, \Phi\}$ represents the spherical coordinate of points; k refers to points with Φ belonging to the k_{th} bin; s indicates the new points of s_{th} interpolated beam layer; λ is the interpolation ratio; the integer S ($S > 2$) is the interpolation factor indicating the number of newly interpolated beam layers.

Considering low-density is relatively more detrimental to object detection than high-density, we select two down-sampling operations with $C \in \{2, 3\}$ (*i.e.*, $2\times$*down-sampling* and $3\times$*down-sampling*) and one up-sampling operation with $S = 2$ ($2\times$*up-sampling*). At last, we design the universal density-resampling augmentation method operating on source training data, which randomly conducts one of {$2\times$*down-sampling,* $3\times$*down-sampling, no-sampling,* $2\times$*up-sampling*} on each source point cloud to simulate potential various densities on target domains. More visualizations are shown in the Supplementary.

4.2 Multi-Task Learning with Density-Resampling

As shown in Figure 3, our multi-task learning scheme consists of the main standard detection task and the auxiliary point cloud densification task. The latter self-supervised task targets to restore the density-downsampled point clouds into their original version. Recent studies [3, 57] have shown that background information is also crucial in object recognition and detection. Through this self-supervised task, the restoration of the 3D scene by point cloud densification helps the encoder to better comprehend the background and foreground details of the scene, which benefits object recognition and thereby object detection.

Main standard detection task. By the PDDA in Section 4.1, we obtain the augmented point cloud $\mathbf{X}_{aug}^s$ from the original point cloud $\mathbf{X}^s$. Then, we feed $\mathbf{X}_{aug}^s$ into the encoder to obtain abstract features F_{aug} by $F_{aug} = f_{\theta_E}(\mathbf{X}_{aug}^s)$, where θ_E is the parameter of the encoder. Following the setting in [9], we calculate the standard detection loss $\mathcal{L}_{det}$:

$$\mathcal{L}_{det}(\mathbf{X}_{aug}^s, \mathbf{B}^s; \theta_E, \theta_H) = f_{\theta_H}(F_{aug}), \tag{3}$$

where θ_H is the parameter of the detection head.

Auxiliary self-supervised task. By density down-sampling in Section 4.1, we further down-sample $\mathbf{X}_{aug}^s$ into $\tilde{\mathbf{X}}^s$. Then, we feed $\tilde{\mathbf{X}}^s$ into the encoder to obtain

the abstract feature $\tilde{F}$ by $\tilde{F} = f_{\theta_E}(\tilde{\mathbf{X}}^s)$. For 3D point cloud densification, we use the decoder (symmetrical to the encoder) to restore the point cloud $\hat{\mathbf{X}}^s$ from $\tilde{F}$, namely $\hat{\mathbf{X}}^s = f_{\theta_D}(\tilde{F})$ where θ_D is the parameter of the decoder. To align $\hat{\mathbf{X}}^s$ with $\mathbf{X}^s_{aug}$, we first adopt the mean squared error (MSE) to calculate the $\mathcal{L}^s_{mse}$.

$$\mathcal{L}^s_{mse}(\tilde{\mathbf{X}}^s, \mathbf{X}^s_{aug}; \theta_E, \theta_D) = \|\hat{\mathbf{X}}^s - \mathbf{X}^s_{aug}\|_2^2. \tag{4}$$

To improve semantic similarity, we also leverage pre-trained PointNet++ [31] to acquire multi-scale features of $\hat{\mathbf{X}}^s$ and $\mathbf{X}^s_{aug}$ for perceptual loss [54] calculation:

$$\mathcal{L}^s_{pcp}(\tilde{\mathbf{X}}^s, \mathbf{X}^s_{aug}; \theta_E, \theta_D) = \sum_{i=1}^{3} \mathcal{L}_{cos}(\mathrm{PNet}_i(\hat{\mathbf{X}}^s), \mathrm{PNet}_i(\mathbf{X}^s_{aug})), \tag{5}$$

where the cosine similarity loss $\mathcal{L}_{cos}(\mathbf{A}, \mathbf{B}) = 1 - \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$ and $\mathrm{PNet}_i(*)(i \in \{1, 2, 3\})$ outputs the flattened feature of the i_{th} block of the PointNet++ encoder. Thus, the compound similarity loss of the self-supervised training $\mathcal{L}^s_{sim}$,

$$\mathcal{L}^s_{sim} = \mathcal{L}^s_{mse} + \lambda_1 \mathcal{L}^s_{pcp}, \tag{6}$$

is used to restore the sparse point cloud for better 3D scene comprehension.

Regarding the multi-task learning scheme, we adopt the two-stage training as in Figure 3: first, we conduct standard detection training to update the encoder and the detection head by $\mathcal{L}_{det}$ and then utilize the self-supervised training to update the parameters of the decoder by $\mathcal{L}^s_{sim}$. During the self-supervised training, we freeze the major trainable parameters of the encoder to ensure the output features aligned with the detection head and only update statistical parameters (*i.e.*, the mean and the standard deviation) of BatchNorm layers to adapt the encoder to diverse feature styles given various densities.

4.3 Test-Time Adaptation with Point Cloud Densification

Prior research in the field of 2D domain generalization has shown that test-time adaptation can be a highly effective approach to mitigate the disparities between source and target domains [6, 18, 25]. It utilizes lightweight test-time optimization to adjust the model's parameters during the testing. Nevertheless, this strategy has yet to be explored in the context of domain generalization of 3D point-cloud-based object detection.

Given the query data from an unseen domain, the effectiveness of test-time adaptation hinges on the proper construction of self-supervised learning. In this study, we make use of the auxiliary 3D point cloud densification task proposed in our multi-task learning to adapt the parameters of the encoder to the target domain. Specifically, given a point cloud $\mathbf{X}^t$ in the target domain, we first down-sample $\mathbf{X}^t$ through the PDDA augmentation, obtaining $\tilde{\mathbf{X}}^t$. Then $\tilde{\mathbf{X}}^t$ is interpolated by the encoder and decoder trained on the source data and the corresponding similarity loss

$$\mathcal{L}^t_{sim}(\tilde{\mathbf{X}}^t, \mathbf{X}^t; \theta_E, \theta_D) = \mathcal{L}^t_{mse}(\tilde{\mathbf{X}}^t, \mathbf{X}^t; \theta_E, \theta_D) = \|f_{\theta_D}(f_{\theta_E}(\tilde{\mathbf{X}}^t)) - \mathbf{X}^t\|_2^2 \tag{7}$$

is used to update θ_E, θ_D, improving the encoder's comprehension of the query scene. One may notice that the perceptual loss in (6) is omitted in our test-time adaptation, even though it leads to a marginal improvement in test-time optimization. This decision was made due to the significant computational latency introduced by the perceptual loss. We show this trade-off in Table S8 in the Supplementary.

During testing time, for each query data, we reset the parameters θ_E and θ_D to the initial state produced in the source training phase. We then iteratively update them a specific number of times, denoted as N_{iter}, to minimize the test-time similarity loss $\mathcal{L}^t_{sim}$. This approach ensures both high computational efficiency and improvement in detection performance. After undergoing N_{iter} updates using the data from point cloud $\mathbf{X}^t$, the encoder is combined with the detection head to produce the final detection result.

5 Experiment

5.1 Experiment Settings

Datasets and metrics. We select select three widely-recognized datasets of autonomous driving: KITTI [11], Waymo [39], NuScenes [4]. KITTI contains ∼15K frames of point clouds collected by a spinning 64-beam LiDAR. Waymo comprises ∼200K frames collected by one 64-beam spinning LiDAR and four 200-beam forward-fixed LiDARs which shapes the regionally (rather than globally) uniform point distribution. For Waymo samples collected continuously over time, we evenly sample 20% of them for experiments. NuScenes contains ∼40K labeled key-frames. Three datasets present shifted distributions caused by different LiDAR types, geographic collection locations, inconsistent object sizes, *etc*. Following conventional settings in [17, 52, 53], we evaluate models under cross-dataset settings of NuScenes → KITTI, Waymo → NuScenes, and Waymo → KITTI. We select "Car" (also "Vehicle" in Waymo), "Pedestrian" and "Cyclist" (also "bicycle" in NuScenes) for detection. For the fair evaluation across datasets, we take the average precision (AP) of 40 recalls and the mean AP (mAP) averaging on all object classes on both 3D and BEV views as the evaluation metrics. The IoU thresholds are 0.7 for "Car" and 0.5 for "Pedestrian" and "Cyclist". Note that, for KITTI, we record the average AP at all difficulty levels (*i.e.*, Easy, Moderate, and Hard).

Implementation details. We evaluate all DG and UDA methods based on the powerful detector VoxelRCNN [9] which uses voxel-based features for both representation extraction and box refinement. For a fair comparison, we use the single unified VoxelRCNN model simultaneously detecting "Car", "Pedestrian", and "Cyclist". We conduct experiments on the popular codebase openpcdet [40] and openpcdet-based Uni3D [56]. In PDDA, we adopt the point dropout $P = 0.95$, and M is determined by the source dataset, *i.e.* 32 for NuScenes and 64 for KITTI and Waymo (during testing, M is 64 by default). For source training, we adopt the down-sampling with $C \in \{3, 4, 6\}$ and the one-cycle Adam optimizer with a learning rate of 0.01 during 30 training epochs. Due to unreachable labeled

target data, we select the best checkpoint through the validation by labeled source *val* data. For self-supervised learning, we finetune the model for 5 epochs with $\lambda_1 = 1.0$. We also adopt common data augmentations for all detections, *i.e.* random world flipping, random world scaling, random world rotation, and ground-truth object sampling. For the test-time training, we adopt the down-sampling with $C \in \{6, 8\}$, the Adam optimizer with a learning rate of 0.001 and set $N_{iter} = 5$ for each sample. Using 2× GeForce RTX-3090/A100, we set the batch size to 4 in source training and 1 in test-time training following the real-world online frame-flow input. For the consistent format of point clouds across datasets, we unify the LiDAR coordinate system with the origin on the ground within the range of $[-75.2m, -75.2m, -2m, 75.2m, 75.2m, 4m]$ and the voxel size of $[0.1m, 0.1m, 0.15m]$.

Compared methods. (a) DG methods: 3D-VF [21] leverages the adversarial vector-field-based augmentation to generalize the detector to the rare-shape or broken objects. PA-DA [8] utilizes the part-aware data augmentation combining 5 basic augmentation methods (*i.e.*, {*dropout, swap, mix, sparse, noise*}) to enable the detector robust to noise and dropout points. (b) UDA methods: SN [45] leverages data augmentation to de-bias the impact of different object sizes on model generalization. Incorporating random object scaling (ROS), ST3D++ [52] designs a self-training pipeline to improve the quality of pseudo-labels of unlabeled target-domain data for cross-domain adaptation. The implementation details can be referred to the Supplementary.

5.2 Comparison with SOTA Methods

Comparison among DG methods. For DG settings, our method outperforms the compared methods, as shown in Table 1. Specifically, for NuScenes → KITTI and Waymo → NuScenes where significant low-to-high-density and high-to-low-density domain gaps exist, compared with the second-best performance, the mAPs of our method have increased by ratios of 12.60%/8.73% and 21.61%/20.30% (with mAP increases of 4.72%/2.02% and 3.06%/1.90%). It demonstrates our method's generalizability to various density-related domain gaps. 3D-VF and PA-DA aim to overcome the bias caused by the rare-shape objects and noisy and locally missing points. However, due to the ignorance of the point layer variation by different sensors, they lead to limited detection performance. For Waymo → KITTI with no significant density-related domain gap, our method still improves the detection accuracy on "Car" objects where the bias of object size exists considering "Car" in KITTI v.s. "Vehicle" (including trunks, vans, etc.) in Waymo, which indicates our method's improvement on the model's generalizability to unseen object sizes. Further, besides VoxelRCNN, we also conduct DG experiments on the point-voxel-based detector PVRCNN++ [34]. Its results are specified in Table S1 in the Supplementary.

To verify the effectiveness of our proposed DG method w.r.t. various types of LiDARs, we conduct experiments on the generalization to detection datasets A2D2 [12] with a 16-beam spinning LiDAR and PandaSet [49] with a 64-beam spinning LiDAR and a forward-facing LiDAR. Table S2 in the Supplementary

Table 1: Performance comparison of different methods on DG and UDA tasks. The values on both sides respectively represent the APs on BEV and 3D views (*i.e.*, AP_{BEV}/AP_{3D}). The bold **values** represent the best performance in DG tasks and the underlined values for the best performance in DG + UDA tasks. "Source-only" represents the standard detection model trained by source data without DG, UDA, or additional augmentation methods.

Tasks		Methods	Car	Pedestrian	Cyclist	mAP
NuScenes → KITTI	DG	Source-only	62.69/19.02	22.72/18.37	20.61/18.13	35.34/18.5
		PA-DA [8]	65.09/32.44	18.73/14.94	18.66/15.91	34.16/21.1
		3D-VF [21]	65.36/29.21	24.85/20.87	22.13/**19.31**	37.45/23.13
		Ours	**73.58**/**33.11**	**30.01**/**23.73**	**22.93**/18.62	**42.17**/**25.15**
	UDA	SN [45]	70.5/54.78	19.71/15.42	13.36/10.83	34.52/27.01
		ST3D++ [52]	83.57/64.17	29.27/25.07	16.61/16.12	43.15/35.12
Waymo → NuScenes	DG	Source-only	31.2/19.13	10.52/8.39	0.75/0.55	14.16/9.36
		PA-DA [8]	29.43/18.06	10.84/8.43	0.82/0.43	13.7/8.97
		3D-VF [21]	30.17/18.91	10.54/7.23	0.76/0.78	13.82/8.97
		Ours	**36.04**/**22.25**	**14.48**/**10.56**	**1.15**/**0.95**	**17.22**/**11.26**
	UDA	SN [45]	29.32/18.84	12.72/10.57	0.72/0.45	14.25/9.96
		ST3D++ [52]	27.58/20.25	11.88/9.44	0.01/0.01	13.15/9.9
Waymo → KITTI	DG	Source-only	66.65/19.27	**66.55**/**64.00**	63.04/57.11	65.41/**46.79**
		PA-DA [8]	65.82/17.61	66.40/63.88	61.30/56.23	64.51/45.91
		3D-VF [21]	66.72/19.37	66.21/63.12	62.74/56.44	65.22/46.31
		Ours	**69.9**/**20.21**	63.24/62.59	**63.27**/**57.21**	**65.47**/46.67
	UDA	SN [45]	72.43/49.34	71.08/69.35	56.00/53.02	66.51/57.23
		ST3D++ [52]	83.59/60.63	50.18/48.4	52.61/47.38	62.13/52.14

illustrates that for various domain gaps introduced by diverse LIDAR types, our method is effective and outperforms other DG methods.

Comparison with UDA methods. The UDA methods leverage reachable unlabeled data and relevant knowledge on the target domain to improve the detection of target domain data. SN aims to normalize object sizes to the target domain, and ST3D++ utilizes ROS augmentation to make the model adaptable to various object sizes. Hence, as shown in Table 1, these two methods perform best on NuScenes → KITTI and Waymo → KITTI with noticeable domain gaps related to object size, especially for "Car" objects. However, on Waymo → NuScenes where there is no significant difference in object size, density-related domain gaps dominate. For Waymo → NuScenes, our method outperforms SN and ST3D++ and reaches the best performance.

5.3 Ablation Study

In this section, we conduct extensive ablation experiments to investigate the individual components of our proposed DG method. Ablation studies are mainly conducted on NuScenes → KITTI and Waymo → NuScenes and use the VoxelRCNN as the basic detection model. More detailed results are shown in the Supplementary.

Table 2: Component ablations in mAP(%). **Source** represents the conventional training procedure. **MTL** and **TTA** represent our multi-task learning and test-time adaptation, respectively.

	Source	PDDA	MTL	TTA	N→K	W→N
(a)	✓				35.34/18.50	14.16/9.36
(b)	✓		✓		35.40/18.57	14.44/9.47
(c)	✓		✓	✓	38.63/22.62	16.04/10.90
(d)	✓	✓			41.07/24.08	16.83/11.04
(e)	✓	✓	✓		41.35/24.12	17.09/11.17
(f)	✓	✓	✓	✓	**42.17**/**25.15**	**17.22**/**11.26**

Table 3: Data augmentation ablations in mAP(%). Bold **values** represent the best performance in DG tasks and underlined values for the best performance in DG + UDA tasks.

Tasks	Methods	N→K	W→N
DG	Source-only	35.34/18.50	14.16/9.36
	PA-DA [8]	34.16/21.10	13.70/8.97
	3D-VF [21]	37.45/23.13	13.82/8.97
	Fog [14]	38.99/23.01	14.59/9.43
	Rain [19]	37.04/23.12	15.21/9.86
	PDDA	**41.07**/**24.08**	**16.83**/**11.04**
UDA	RBRS [17]	39.21/20.82	17.33/11.20
	SN [45]	34.52/27.01	14.25/9.96
	ROS [52]	38.18/28.29	13.67/8.86

Component ablation. Table 2 investigates the effectiveness of the components in our density-resampling-based solution. Note that though MTL makes no direct boosts on detection, as shown in rows (a) → (b) or (d) → (e), it serves as a solid framework base for cloud densification and provides a close-loop for TTA model update. Compared with (a) the detection model trained with source domain data only, both (c) applying TTA during testing and (d) applying PDDA augmentation during source training bring significant improvement. With (f) employing both TTA and PDDA augmentation, our proposed method further boosts the detection to reach the best performance. It should be noted that since both innovations, PDDA and TTA, are essentially centered around the idea of density resampling to improve the model's capability of 3D scene understanding, the final performance of integrating them into one framework is not simply additive, and the performance improvement, e.g. (c) → (f) and (d) → (f), reflects their complementarity to each other.

Comparison on data augmentation. Besides the DG augmentation methods (*i.e.*, PA-DA [8] and 3D-VF [21]), we also include the weather-simulating augmentation methods Rain [19] and Fog [14] which simulate the adverse rain and fog in point clouds. As shown in Table 3, our PDDA method outperforms all compared DG augmentation methods. In this ablation, we also include UDA augmentation methods that require target-domain information: SN [45], RBRS [17], and ROS [52, 53]. Particularly, for density-related domain shifts, RBRS [17] employs random upsampling on the low-to-high-density NuScenes → KITTI and downsampling on the high-to-low-density Waymo → NuScenes. For comparison, without relying on target domain information, our PDDA outperforms RBRS on NuScenes → KITTI and closely approximates RBRS on Waymo → NuScenes (with a slight 0.5%/0.16% lag), which indicates the superiority of PDDA's capturing the physical-aware characteristics of real-world beam layer scanning. Further, considering the domain-specific density re-sampling circumstance (*i.e.* with density prior of the target domain), Table S5 in the Supplementary indicates that PDDA can be further improved with prior density knowledge of the target domain and outperforms the RBRS which also requires target density prior.

Table 4: Comparison on test-time training. KNN-TTT stands for applying the KNN-based masking strategy during self-supervised training and test-time training, instead of density-downsampling as in our proposed method. FPS stands for the number of processed frames per second.

Methods	**N→K**		**W→N**	
	mAP (%)	computation efficiency (FPS)	mAP (%)	computation efficiency (FPS)
KNN-TTT [27]	40.51/24.92	0.52	16.67/11.03	0.41
Our TTT	**42.17/25.15**	**4.23**	**17.22/11.26**	**3.52**

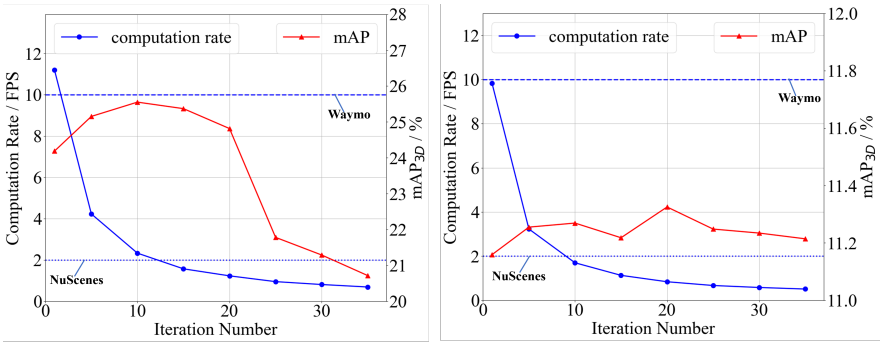
Table 5: Performance comparison in mAP(%) on Waymo+KITTI→NuScenes

Tasks	Methods	Car	Pedestrian	Cyclist	Average
W+K→N	Source-only	27.49/18.24	8.62/7.63	1.00/0.38	12.37/8.75
	PA-DA	29.24/20.05	9.12/7.02	0.50/0.44	12.95/9.17
	3D-VF	30.19/20.07	9.85/8.94	0.37/0.36	13.47/9.79
	Ours	**35.30/21.63**	**16.72/12.69**	**1.31/1.16**	**17.78/11.83**

Comparison on test-time training. To further explore the performance of our proposed test-time training, following the Point-MATE [27], we adopt the masking strategy based on K-nearest-neighbors (KNN) with the masking rate of 90% during self-supervised training and test-time training (refer to implementation details in the Supplementary). As shown in Table 4, while the KNN-masking-based test-time training reaches approximate accuracy to our density-downsampling-based method (with a slight lag), the computation of farthest point sampling and KNN clustering on all points cause the severe latency. In contrast, our density-sampling operations on points perform more accurately and more efficiently on object detection.

Comparison on multi-domain generalization We also explore the performance of our proposed single-domain generalization method in multi-domain generalization. Specifically, we train the model with source Waymo + KITTI *train* data and evaluate it on NuScenes *val* set. As shown in Table 5, our method still outperforms prior arts in the detection of all object classes, which illustrates the significant boost of our SDG method on the multi-domain generalization for 3D object detection.

Computation efficiency. We explore the computation efficiency w.r.t. processing of a single GeForce RTX-3090 GPU. As shown in Figure 4, the optimal N_{iter} reaching the best performance are 10 for NuScenes → KITTI and 20 for Waymo → NuScenes. After that, the detection performance gets worse due to the model overfitting to the auxiliary task. Given such N_{iter} settings, the real-time processing requires parallel computation with multiple GPUs. For the single-GPU setting, the computation speed with $N_{iter} = 5$ still overpasses the 2 FPS labeled keyframe rate of NuScenes. By a minor performance penalty, the computation speed with $N_{iter} = 1$ meets the 10 FPS real-time running requirement of Waymo,



(a) NuScenes → KITTI **(b) Waymo → NuScenes**

Fig. 4: Computation efficiency on (a) NuScenes → KITTI and (b) Waymo → NuScenes. We indicate Waymo's frame rate of 10 FPS and NuScenes's keyframe rate of 2 FPS by dash lines.

and the accuracy (41.41%/24.19% on NuScenes → KITTI and 17.00%/11.16%) still overpass other DG methods in Table 1.

5.4 Limitation

As shown in Table 1, our proposed method improves the model's generalizability on density-variant settings, e.g. NuScenes → KITTI, Waymo → NuScenes. However, regarding density-invariant settings on Waymo → KITTI, our method brings marginal or no improvement in detection accuracy, which is a topic we plan to solve in the future.

6 Conclusion

Point-cloud-based 3D object detection suffers from performance degradation when encountering data with unexplored domain gaps. To tackle this problem, we propose an SDG method centered around the theme of point cloud density resampling to improve the model's generalizability. Specifically, we first design a physical-aware density-resampling data augmentation that aligns with real-world physical constraints to reshape point density, to mitigate the performance loss stemming from diverse point densities. In systematic learning design, an auxiliary self-supervised point cloud densification task is designed to assess the model's understanding of the environment and semantic information. The self-supervised task enables the first test-time adaptation solution on domain generalization of 3D point-cloud-based object detection, to further adapt the model to varying domains. Extensive cross-dataset experiments demonstrate our method outperforms state-of-the-art SDG methods and even overpass UDA methods under some circumstances.

Acknowledgments: This work is supported in part by Canada CIFAR AI Chairs Program, the Natural Sciences and Engineering Research Council of Canada, Alberta Innovates, as well as JST-Mirai Program Grant No.JPMJMI20B8, JSPS KAKENHI Grant No.JP21H04877, No.JP23H03372, and No.JP24K02920.

References

1. Arnold, E., Al-Jarrah, O.Y., Dianati, M., Fallah, S., Oxtoby, D., Mouzakitis, A.: A survey on 3d object detection methods for autonomous driving applications. IEEE Transactions on Intelligent Transportation Systems **20**(10), 3782–3795 (2019)
2. Balaji, Y., Sankaranarayanan, S., Chellappa, R.: Metareg: Towards domain generalization using meta-regularization. Advances in neural information processing systems **31** (2018)
3. Boyce, S.J., Pollatsek, A., Rayner, K.: Effect of background information on object identification. Journal of Experimental Psychology: Human Perception and Performance **15**(3), 556 (1989)
4. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)
5. Chen, D., Wang, D., Darrell, T., Ebrahimi, S.: Contrastive test-time adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 295–305 (2022)
6. Chen, L., Zhang, Y., Song, Y., Shan, Y., Liu, L.: Improved test-time adaptation for domain generalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24172–24182 (2023)
7. Chen, W., Lin, L., Yang, S., Xie, D., Pu, S., Zhuang, Y.: Self-supervised noisy label learning for source-free unsupervised domain adaptation. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 10185–10192. IEEE (2022)
8. Choi, J., Song, Y., Kwak, N.: Part-aware data augmentation for 3d object detection in point cloud. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 3391–3397. IEEE (2021)
9. Deng, J., Shi, S., Li, P., Zhou, W., Zhang, Y., Li, H.: Voxel r-cnn: Towards high performance voxel-based 3d object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 1201–1209 (2021)
10. Du, L., Ye, X., Tan, X., Feng, J., Xu, Z., Ding, E., Wen, S.: Associate-3ddet: Perceptual-to-conceptual association for 3d point cloud object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13329–13338 (2020)
11. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013)
12. Geyer, J., Kassahun, Y., Mahmudi, M., Ricou, X., Durgesh, R., Chung, A.S., Hauswald, L., Pham, V.H., Mühlegg, M., Dorn, S., et al.: A2d2: Audi autonomous driving dataset. arXiv preprint arXiv:2004.06320 (2020)
13. Hahner, M., Sakaridis, C., Bijelic, M., Heide, F., Yu, F., Dai, D., Van Gool, L.: Lidar snowfall simulation for robust 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16364–16374 (2022)

14. Hahner, M., Sakaridis, C., Dai, D., Van Gool, L.: Fog simulation on real lidar point clouds for 3d object detection in adverse weather. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15283–15292 (2021)
15. Hammer, M., Hebel, M., Laurenzis, M., Arens, M.: Lidar-based detection and tracking of small uavs. In: Emerging Imaging and Sensing Technologies for Security and Defence III; and Unmanned Sensors, Systems, and Countermeasures. vol. 10799, pp. 177–185. SPIE (2018)
16. Hu, J.S., Kuai, T., Waslander, S.L.: Point density-aware voxels for lidar 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8469–8478 (2022)
17. Hu, Q., Liu, D., Hu, W.: Density-insensitive unsupervised domain adaption on 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17556–17566 (2023)
18. Iwasawa, Y., Matsuo, Y.: Test-time classifier adjustment module for model-agnostic domain generalization. Advances in Neural Information Processing Systems **34**, 2427–2440 (2021)
19. Kilic, V., Hegde, D., Sindagi, V., Cooper, A.B., Foster, M.A., Patel, V.M.: Lidar light scattering augmentation (lisa): Physics-based simulation of adverse weather conditions for 3d object detection. arXiv preprint arXiv:2107.07004 (2021)
20. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12697–12705 (2019)
21. Lehner, A., Gasperini, S., Marcos-Ramiro, A., Schmidt, M., Mahani, M.A.N., Navab, N., Busam, B., Tombari, F.: 3d-vfield: Adversarial augmentation of point clouds for domain generalization in 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17295–17304 (2022)
22. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.: Learning to generalize: Meta-learning for domain generalization. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)
23. Liang, J., He, R., Tan, T.: A comprehensive survey on test-time adaptation under distribution shifts. arXiv preprint arXiv:2303.15361 (2023)
24. Liang, J., Hu, D., Feng, J.: Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In: International conference on machine learning. pp. 6028–6039. PMLR (2020)
25. Liu, Q., Chen, C., Dou, Q., Heng, P.A.: Single-domain generalization in medical image segmentation via test-time adaptation from shape dictionary. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 1756–1764 (2022)
26. Mao, J., Xue, Y., Niu, M., Bai, H., Feng, J., Liang, X., Xu, H., Xu, C.: Voxel transformer for 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3164–3173 (2021)
27. Mirza, M.J., Shin, I., Lin, W., Schriebl, A., Sun, K., Choe, J., Kozinski, M., Possegger, H., Kweon, I.S., Yoon, K.J., et al.: Mate: Masked autoencoders are online 3d test-time learners. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16709–16718 (2023)
28. Motiian, S., Piccirilli, M., Adjeroh, D.A., Doretto, G.: Unified deep supervised domain adaptation and generalization. In: Proceedings of the IEEE international conference on computer vision. pp. 5715–5725 (2017)
29. Muandet, K., Balduzzi, D., Schölkopf, B.: Domain generalization via invariant feature representation. In: International conference on machine learning. pp. 10–18. PMLR (2013)

30. Ouyang, C., Chen, C., Li, S., Li, Z., Qin, C., Bai, W., Rueckert, D.: Causality-inspired single-source domain generalization for medical image segmentation. IEEE Transactions on Medical Imaging **42**(4), 1095–1106 (2022)
31. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems **30** (2017)
32. Qian, R., Lai, X., Li, X.: 3d object detection for autonomous driving: A survey. Pattern Recognition **130**, 108796 (2022)
33. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H.: Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10529–10538 (2020)
34. Shi, S., Jiang, L., Deng, J., Wang, Z., Guo, C., Shi, J., Wang, X., Li, H.: Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. International Journal of Computer Vision **131**(2), 531–551 (2023)
35. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 770–779 (2019)
36. Shi, W., Rajkumar, R.: 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. pp. 1711–1719 (2020)
37. Shi, Z., Chen, Z., Xu, Z., Yang, W., Yu, Z., Huang, L.: Shape prior guided attack: Sparser perturbations on 3d point clouds. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 8277–8285 (2022)
38. Soum-Fontez, L., Deschaud, J.E., Goulette, F.: Mdt3d: Multi-dataset training for lidar 3d object detection generalization. arXiv preprint arXiv:2308.01000 (2023)
39. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2446–2454 (2020)
40. Team, O.D.: Openpcdet: An open-source toolbox for 3d object detection from point clouds. `https://github.com/open-mmlab/OpenPCDet` (2020)
41. Vidit, V., Engilberge, M., Salzmann, M.: Clip the gap: A single domain generalization approach for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3219–3229 (2023)
42. Volpi, R., Murino, V.: Addressing model vulnerability to distributional shifts over image transformation sets. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7980–7989 (2019)
43. Volpi, R., Namkoong, H., Sener, O., Duchi, J.C., Murino, V., Savarese, S.: Generalizing to unseen domains via adversarial data augmentation. Advances in neural information processing systems **31** (2018)
44. Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization. arXiv preprint arXiv:2006.10726 (2020)
45. Wang, Y., Chen, X., You, Y., Li, L.E., Hariharan, B., Campbell, M., Weinberger, K.Q., Chao, W.L.: 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. pp. 11713–11723 (2020)
46. Wang, Z., Luo, Y., Qiu, R., Huang, Z., Baktashmotlagh, M.: Learning to diversify for single domain generalization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 834–843 (2021)

47. Wu, A., Deng, C.: Single-domain generalized object detection in urban scene via cyclic-disentangled self-distillation. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition. pp. 847–856 (2022)
48. Wu, G., Cao, T., Liu, B., Chen, X., Ren, Y.: Towards universal lidar-based 3d object detection by multi-domain knowledge transfer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8669–8678 (2023)
49. Xiao, P., Shao, Z., Hao, S., Zhang, Z., Chai, X., Jiao, J., Li, Z., Wu, J., Sun, K., Jiang, K., et al.: Pandaset: Advanced sensor suite dataset for autonomous driving. In: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). pp. 3095–3101. IEEE (2021)
50. Xu, Q., Zhong, Y., Neumann, U.: Behind the curtain: Learning occluded shapes for 3d object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 2893–2901 (2022)
51. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
52. Yang, J., Shi, S., Wang, Z., Li, H., Qi, X.: St3d++: denoised self-training for unsupervised domain adaptation on 3d object detection. arXiv preprint arXiv:2108.06682 (2021)
53. Yang, J., Shi, S., Wang, Z., Li, H., Qi, X.: St3d: Self-training for unsupervised domain adaptation on 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10368–10378 (2021)
54. Yang, Q., Yan, P., Zhang, Y., Yu, H., Shi, Y., Mou, X., Kalra, M.K., Zhang, Y., Sun, L., Wang, G.: Low-dose ct image denoising using a generative adversarial network with wasserstein distance and perceptual loss. IEEE transactions on medical imaging **37**(6), 1348–1357 (2018)
55. Ye, Y., Chen, H., Zhang, C., Hao, X., Zhang, Z.: Sarpnet: Shape attention regional proposal network for lidar-based 3d object detection. Neurocomputing **379**, 53–63 (2020)
56. Zhang, B., Yuan, J., Shi, B., Chen, T., Li, Y., Qiao, Y.: Uni3d: A unified baseline for multi-dataset 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9253–9262 (2023)
57. Zheng, Y., Duan, Y., Lu, J., Zhou, J., Tian, Q.: Hyperdet3d: Learning a scene-conditioned 3d object detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5585–5594 (2022)
58. Zhou, K., Liu, Z., Qiao, Y., Xiang, T., Loy, C.C.: Domain generalization: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)