An Empirical Study of the Generalization Ability of Lidar 3D Object Detectors to Unseen Domains

George Eskandar*
University of Stuttgart, Germany

Abstract

3D Object Detectors (3D-OD) are crucial for understanding the environment in many robotic tasks, especially autonomous driving. Including 3D information via Lidar sensors improves accuracy greatly. However, such detectors perform poorly on domains they were not trained on, i.e. different locations, sensors, weather, etc., limiting their reliability in safety-critical applications. There exist methods to adapt 3D-ODs to these domains; however, these methods treat 3D-ODs as a black box, neglecting underlying architectural decisions and source-domain training strategies. Instead, we dive deep into the details of 3D-ODs, focusing our efforts on fundamental factors that influence robustness prior to domain adaptation.

We systematically investigate four design choices (and the interplay between them) often overlooked in 3D-OD robustness and domain adaptation: architecture, voxel encoding, data augmentations, and anchor strategies. We assess their impact on the robustness of nine state-of-the-art 3D-ODs across six benchmarks encompassing three types of domain gaps - sensor type, weather, and location.

Our main findings are: (1) transformer backbones with local point features are more robust than 3D CNNs, (2) test-time anchor size adjustment is crucial for adaptation across geographical locations, significantly boosting scores without retraining, (3) source-domain augmentations allow the model to generalize to low-resolution sensors, and (4) surprisingly, robustness to bad weather is improved when training directly on more clean weather data than on training with bad weather data. We outline our main conclusions and findings to provide practical guidance on developing more robust 3D-ODs.

1. Introduction

The key objective of 3D object detection (3D-OD) is to accurately localize and identify objects of different classes in the 3D environment using sensor data such as point clouds [35, 36, 45] or images [25, 40, 46]. Lidar sensors have played a pivotal role in this endeavor, as they can provide dense and highly precise point clouds using a long scanning range. While 3D object detection models reliant on Lidar point clouds have achieved remarkable results on demanding benchmarks such as KITTI [13], Waymo [37], and NuScenes [3], they face a significant performance drop when deployed in unfamiliar settings. This decline in performance might arise from migrating to a different sensor, owing to changes in point sparsity levels and scanning patterns, or from changes in the perceived 3D shapes of objects due to adverse weather conditions like rain, fog, or snow. Other changes can stem from deployment in a new geographical location that exhibits different statistical properties like smaller or bigger object sizes. All these scenarios pose a considerable threat to the safety and reliability of autonomous driving.

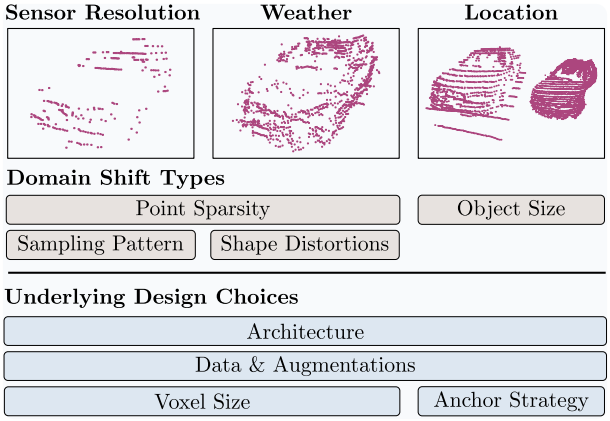


Figure 1. An overview of the three main domain gaps under study. We show the main shift types for each domain gap along with the underlying design choices that influence model robustness.

Domain Adaptation (DA) promises to address these issues by adapting models trained on a source domain to a target domain, achieving strong results on some domain gaps [9, 11, 16, 17, 26, 27, 41–44, 47–49, 53]. However, these methods often treat detection models as black-boxes relying heavily on the availability of target domain data. This leaves a gap in understanding: *how do architectural choices and training strategies affect generalization to target domains?* Furthermore, not all DA methods develop their techniques using the same architectures, making it

*We extend our profound gratitude to Chongzhe Zhang and Abhishek Kaushik, from the University of Stuttgart, Karim Guirguis from KIT, Mohamed Sayed, from Niantic, and Bin Yang, from the University of Stuttgart for their invaluable contributions to this work.

even harder to evaluate these models. Unlike images, point clouds are unordered 3D structures, allowing for a broader range of Lidar-based detector design options - operating on points, voxels, or a combination of both.

Understanding the robustness of 3D Lidar detectors thus holds significant importance, given that data distribution shifts are inevitable and more likely to occur with the growing deployment of autonomous vehicles. Recently, there have been studies in this field [12, 29, 51, 55], focusing primarily on corruptions benchmarking and identifying sensor representations (RGB, Lidar, or fusion) exhibiting greater resilience. Notably, these works highlight that fusion models suffer greater when the Lidar input, as opposed to RGB, is corrupted. Following this, our particular focus lies on Lidar-only 3D-ODs and the impact of design choices on detection robustness, ultimately providing recommendations on standard practice and solutions. Furthermore, unlike previous works [12, 55], we study domain gaps that are not necessarily caused by environmental or sensor corruptions but are still harmful to the detection performance, such as different geo-locations and sensor resolution. We methodically isolate and study each domain gap when benchmarking, giving careful attention to the differences in sensor differences, location, and weather.

Contribution. In this work, we methodically study the effect of several design choices in the architecture and training strategy on the generalization of Lidar-based detectors to unseen domains. Our aim is to preserve as much accuracy as we can before relying on domain adaptation. We present a taxonomy of various domain shift types in each domain gap (Fig. 1) and four underlying design choices (architecture, augmentation, voxel size, anchor size) that can potentially address these domain shifts. Then, in Sec. 3, we evaluate nine state-of-the-art 3D detectors on six DA benchmarks featuring three domain gaps (weather, location, and resolution), analyzing the impact of many common architectural designs. A large number of controlled experiments and apple-to-apple comparisons are subsequently conducted to disentangle the best practices for each domain gap in sections 4, 5 and 6. Similar to previous devil-in-the-detail investigative works for CNNs on images [4, 5, 33, 52], we show that often simple and overlooked details can significantly influence out-of-domain (OOD) performance. The main focus of this work is to provide an empirical study to understand the robustness of different design choices in Lidar 3D-OD. We also provide solutions to some of the problems based on our findings and point out potential research areas. We present a summary of the novel findings and practical recommendations in Sec. 7.

2. Related Works

Robustness Benchmarks for 3D-OD is a relatively new research area that seeks to study the robustness of 3D detectors under different unseen corruptions. Previous studies [12, 51, 55] primarily concentrated on categorizing these corruptions and establishing benchmarks to evaluate the OOD performance of different sensor representations (camera, Lidar, fusion of both). [51] introduced a benchmark for camera-lidar fusion models, revealing a heavy reliance on Lidar as these models fail worse when only Lidar data is corrupted. [12] established more benchmarks showing that fusion models are the most robust, while the camera-only models are the most vulnerable. Both studies underscore the importance of Lidar. Finally, [55] studied the robustness of bird-eye view representations in camera-only and fusion models when subjected to environmental corruption and adversarial attacks. Motivated by prior research, our study focuses exclusively on Lidar-only models and delves into the intricate relationship between design choices in the 3D-OD pipeline and robustness in unseen domains. While this has been extensively studied in images [18, 20, 31, 34], it is still lacking in Lidar.

Domain Adaptation for Lidar 3D-OD has started to gain more attention in the past few years but is still not as mature as DA on images [6–8, 19, 24]. SN [41] has identified object size as the major shift type across different locations and introduced an approach to resize the source domain labels based on the mean object size (MOS) of the target domain. Many approaches have then followed, leveraging self-training methods [11, 16, 17, 26, 42, 43, 47, 48], contrastive instance-level feature alignment [49], adversarial learning [9], data augmentations [42, 43, 47, 48, 53] or anchor scaling [11, 27]. Lidar beam distillation [43] specifically addresses the cross-resolution domain gap by learning progressively downsampled versions of the source domain with a teacher-student model while employing ST3D [47] on the target domain data. 3D-VField [22] introduces a data augmentation for synthetic-to-real domain generalization, where foreground objects are deformed using vector fields. To improve the generalization to adverse weather conditions, SPG [44] restores missing points through a self-supervised method. By simulating the characteristics of lasers in foggy settings, [15] transforms pointclouds collected in sunny scenes into pointclouds in fog and use them as data augmentation.

3. Benchmarking

It has been shown in several works on image perception [1, 14, 18, 20, 30] that the network's architecture has a significant impact on its generalization ability. Here, we seek to answer this question in the context of Lidar 3D-OD.

3.1. Architectures

We take a closer look at nine state-of-the-art 3D detectors (using their official code based on OpenPCDet [38]) and present a taxonomy of their architectural components in

Representation	Model	3D Feature Encoder		BEV Backbone	DenseHead	RCNN
		Voxels	Points			
Points	PointRCNN [35]	✗	PointNet	✗	PointHead	PointRCNN
Voxels	PointPillars [21]	✗	✗	✓	AnchorHead	✗
	Second [45]	3DCNN×8	✗	✓	AnchorHead	✗
	Voxel-RCNN [10]	3DCNN×8	✗	✓	AnchorHead	Voxel
	VOTR-VoxelRCNN	VOTR	✗	✓	AnchorHead	Voxel
Hybrid (PV)	CenterPoint [50]	3DCNN×8	✗	✓	Centerhead	Point
	PV-RCNN-Centerhead [38]	3DCNN×8	VSA	✓	CenterHead	PV
	PV-RCNN [36]	3DCNN×8	VSA	✓	AnchorHead	PV
	VOTR-TSD [28]	VOTR	VSA	✓	AnchorHead	PV

Table 1. Taxonomy of state-of-the-art Lidar 3D detectors based on their common and specific architectural components.

Tab. 1. We break down the architecture of a 3D detector into four main parts: a 3D feature encoder, a 2D BEV backbone, a dense head, and an RCNN head (in the case of a two-stage model). We find that these nine frameworks can be further categorized into three main approaches based on the point-cloud representation: point-based, voxel-based, and hybrid-based (points and voxels (PV)).

Point-based approaches like PointRCNN [35] employ a PointNet [32]-like feature extractor and an anchor-based dense head (PointHead) processing point features. On the other hand, voxel-based approaches discretize the point-cloud into pillars and process the resulting BEV features with a 2D backbone (Pillars) or discretize into voxels and employ a 3DCNN with $8\times$ downsampling (SECOND). Two-stage voxel detectors (VoxelRCNN) employ an additional voxel-only RCNN head. We also experiment by replacing the 3DCNN in VoxelRCNN with the VOTR backbone in VOTR-TSD [28] and name this model VOTR-VoxelRCNN. Lastly, in hybrid approaches like PVRCNN, PVRCNN-Centerhead and VOTR-TSD, point features are extracted alongside voxel features via the voxel set abstraction (VSA) module [36], which are then used in the RCNN head. Note that while CenterPoint uses voxels only in the feature extractor, its CenterHead also uses point features so we classify it as a hybrid approach. Tab. 1 can be viewed as an ablation study on a single abstract 3D object detector, allowing us to perform apple-to-apple comparisons and pinpointing the effect of single architectural components.

3.2. Experimental Setup

We choose six common benchmarks formed by four datasets to evaluate the robustness of the studied models. An overview of the utilized datasets and the existing domain gaps between them are shown in Tab. 2. To study the resolution domain gap in more details, we follow [43] and downsample the KITTI dataset two times (KITTI-32) and four times (KITTI-16). We report the 3D Average Precision (3D AP) with 40-points Recall on the classes car, pedestrian and cyclist using the official evaluation metrics for each benchmark: KITTI metrics for Waymo-to-Kitti (W→K), Waymo-to-NuScenes (W→N), NuScenes-to-KITTI (N→K), KITTI64-to-KITTI32 (K64→K32) and KITTI64-to-KITTI16 (K64→K16), and Waymo metrics for Waymo-to-Kirkland (W→Kr). Note that we train on the 20% training split of the Waymo dataset to achieve a high number of experiments. In all experiments, we choose the best-performing checkpoint on the validation set of the source domain and evaluate it on the target domain validation split, following DA works [47].

Dataset	VFOV	Lines	#Samples	Location	Weather
KITTI	$[-23.6°, 3.2°]$	64	14k/2k/734	Germany	Clear
Waymo	$[-17.6°, 2.4°]$	64	4.7M/2.2M/53k	USA	Clear
Kirkland	$[-17.6°, 2.4°]$	64	312k/21k/0	Kirkland	Rainy
nuScenes	$[-30.0°, 10.0°]$	32	196k/100k/10k	Boston & Singapore	Clear

Table 2. A summary of the datasets used in this study. # Training samples is reported for Car/Pedestrian/Cyclist.

3.3. Results

We report the results in Tab. 3 and the key findings.

Finding 1: *VOTRs outperform 3D CNNs in the mean AP across all domain gaps when coupled with point features (VOTR-TSD).* This finding only partially aligns with observations in image perception literature [1, 14, 18], where it has been shown that transformers are more robust than 3D CNNs. We observe that 3D CNNs can be more robust than VOTRs in voxel-only models (VoxelRCNN versus VOTR-VoxelRCNN). The addition of point features enhances the performance of VOTR, as it adds a much-needed spatial local context to the transformer's large receptive field of view. This is more important in Lidar than images, because the relative size of some classes, like Pedestrians, to the input size is much smaller in pointclouds than in images.

Finding 2: *Anchorless detectors are robust in the weather domain gap.* This is more evident on the Pedestrian class, where CenterPoint and PVRCNN-Centerhead outperform other models by 4-5 AP.

Finding 3: *Adding point features in the backbone increases robustness, especially for transformers.* VOTR-TSD outperforms VOTR-VoxelRCNN (Tab. 3), PVRCNN-Centerhead outperforms Centerpoint but PVRCNN and VoxelRCNN have similar performance. The key distinction between these pairs is point features in the backbone.

Features	Architecture	Method	K64→K32 (R)			K64→K16 (R)			W→K (G)			W→N (G+R)			W→Kr (W)		N→K (G+R)			Mean		
			Car	Ped	Cyc	Car	Ped	Cyc	Car	Ped	Cyc	Car	Ped	Cyc	Car	Ped	Car	Ped	Cyc	Car	Ped	Cyc
Point	MLP	PointRCNN	74.7	51.86	**61.52**	51.74	21.03	24.71	5.04	28.22	0.0	12	3.92	0.0	19.24	5.86	13.81	25.31	0.0	29.42	22.7	18.25
Voxel	Conv	PointPillars	70.48	35.81	25.93	53.16	20.5	9.28	12.75	48.34	34.9	20.9	5.45	0.04	46.77	13.79	0.0	0.0	0.0	34.01	20.65	10.93
	Conv	Second	73.5	41.11	39.1	50.71	16.63	17.58	9.91	41.39	22.74	17.84	4.44	0.22	46.04	14.97	6.36	13.43	0.0	34.06	22	15.17
	Conv	VoxelRCNN	**76.96**	**56.71**	50.18	**56.11**	**27.24**	24.64	**20.09**	**55.33**	34.81	19.71	0.05	0.0	52.18	20.7	7.48	20.96	0.0	38.75	**30.16**	19.47
	ViT	VOTR-VoxelRCNN	76.05	51.62	46.04	55.7	24.55	20.79	21.34	29.96	1.63	15.34	0.02	0.0	49.21	15.81	19.26	**26.82**	2.04	39.48	24.8	22.48
Hybrid	Conv	CenterPoint	71.18	42.71	44.19	52.8	15.11	14.27	13.78	53.39	**42.43**	19.05	5.63	0.5	50.23	**30.72**	8.96	19.94	0.04	36	27.92	18.57
	Conv	PVRCNN Centerhead	71.39	44.64	34.2	44.24	15.39	12.02	15.06	52.47	40.13	20.59	6.84	0.42	52.79	29.02	**28.52**	21.67	0.03	38.76	28.34	23.46
	Conv	PVRCNN	77.62	53.96	50.3	54.07	27.13	**26.36**	16.61	50.22	34.09	20.36	5.79	0.53	**54.34**	25.79	10.46	17.46	0.0	38.91	30.06	19.38
	ViT	VOTR-TSD	76.29	49.78	49.76	55.61	21.71	24.38	15.75	46.19	39.28	**21.32**	**7.0**	**3.48**	52.52	26.56	26.26	26.56	**4.7**	**41.29**	29.63	**25.69**

Table 3. Evaluation of different architectures on six DA benchmarks. Beside each benchmark, we denote whether it is mainly caused by a resolution (R), weather (W), or geographical location (G) discrepancy. Voxel Transformers and hybrid representations are found to be more robust, on average, across the considered domains. Note that the Kirkland (Kr) dataset does not have the class cyclist.

4. Location Domain Gap

It has been shown that discrepancies in object sizes across geographical locations are the main cause of bad generalization [41, 47, 48]. Explanations and remedies to this problem are controversial: while some works address this problem by varying the label size on the source domain [41, 47], others [11, 27] change the anchor size without a labeled target set with mixed results. As both strategies were explored in different settings, it is not clear which is the most beneficial. In this section, we perform experiments on the W→K, W→N and N→K benchmarks to highlight the most influential design choices. We train exclusively on the class car as it exhibits the most variations across different locations, following [41, 47] (see Appendix for Multiclass experiments).

4.1. Anchor Size

The anchor size in 3D-OD is a carefully tuned hyperparameter and is usually close to the mean object size (MOS) of the training dataset. In Fig. 2, we seek to answer two questions: Is there a connection between anchor size and performance in OOD scenarios? And, when considering training and test phases, which holds greater significance: the anchor size during training or testing? We perform our experiments using the SECOND model [45] due to its good performance in DA benchmarks [47] and its low computation requirements. We train three models, each with a different anchor. The first uses the default training anchor on Waymo dataset, while the second and third use a smaller and a larger one, respectively. We notice the following: 1) *There is no correlation between the anchor size at training time and the OOD performance*. All three models have low AP when they use the same anchor size they were trained on. 2) However, *changing anchor size at test time leads to strong variation in the performance*. 3) *The best-performing anchors are found to be always smaller than the training anchor*. This is also true when the target MOS is bigger than the source MOS (see Appendix). 4) There exists indeed a *test-time anchor size that yields a very high performance* on the target domain without retraining. This is heuristically determined with a simple greedy algorithm: we change one dimension at a time and select the value that

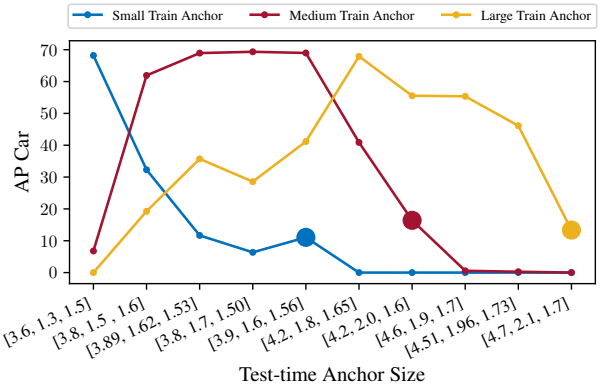


Figure 2. Evaluation of different anchor sizes (in the order of increasing volumes) at test-time on W→K car benchmark. Three training experiments using SECOND are performed, each with a different anchor size. Using the same size at test-time results in a poor performance (○), but a very high peak can be obtained by going for lower anchor sizes.

achieves the best AP on the validation set. Then, we fix this value and change the subsequent dimensions following the same procedure (see Appendix). While this is the best anchor we find using this optimization procedure, there might be a more optimal solution. Finally, note that the described technique of changing anchor size at test-time is *not meant* to be a proposed domain generalization approach (since we directly tune the anchor on the target dataset). Instead, we reveal that this simple trick can significantly enhance the performance in a semi-supervised, weak, or federated domain adaptation setting.

4.2. Anchor vs. Label Strategy

In Tab. 4, we contrast the two strategies against each other at training time. Specifically, we train SECOND using the random object scaling (ROS) proposed in ST3D [47], where the foreground objects and their labels are randomly resized. Moreover, we train SECOND with multiple anchors (three) and include an anchorless detector in the analysis (CenterPoint). While training with anchors of different sizes is very common in 2D-OD, it is surprisingly not used in 3D-OD [38], as models are usually trained with two anchors per class of the same size with different ori-

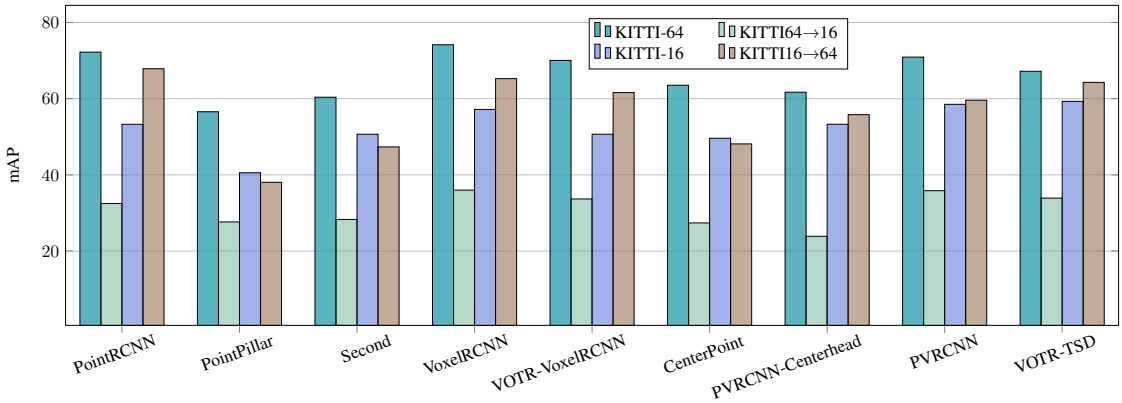


Figure 3. Evaluation of all models on K64, K16, K64→K16 and K16→K64 benchmarks. The mAP for all classes is reported. The performance drops significantly from high-to-low resolution while it increases from low-to-high at test-time.

Model	Waymo	W→K	W→N
Second	57.42	16.41	18.69
Second w/ ROS	49.01	**41.91**	14.09
Second 3 Anchors	**59.86**	14.58	**21.22**
Second 3 Anchors w/ ROS	51.83	33.74	16.01
CenterPoint	58.48	13.86	20.08
CenterPoint w/ ROS	53.30	34.46	16.61

Table 4. Investigating the effect of ROS, using multiple anchors and anchorless detectors on the OOD performance of the class car. The source AP (Waymo) is also considered.

Model	W→K	W→N	N→K
PointRCNN	5.38 / 35.55	1.31 / 14.29	15.0 / 27.11
PointPillar	10.48 / 65.14	21.01 / **23.84**	0.04 / 0.01
SECOND	16.41 / **66.81**	18.69 / 21.31	4.97 / 41.64
VoxelRCNN	18.28 / 59.49	19.47 / 21.34	8.74 / 29.0
VOTR-VORCNN	**18.15** / 65.14	19.38 / 21.18	18.47 / **46.96**
PVRCNN	9.69 / 40.86	20.08 / 22.31	14.84 / 26.67
VOTR-TSD	14.47 / 52.64	**21.66** / 23.74	**24.75** / 35.0

Table 5. Benchmarking the effect of changing the anchor size on different anchor-based models: we report the 3D AP on the class car before/after tuning the anchor at test time.

entations. We find that: 1) *Anchorless detectors exhibit only a marginal improvement in generalization compared to anchor-based detectors* when dealing with objects of various sizes. Their performance is notably suboptimal. 2) *ROS enhances robustness on KITTI but decreases performance on NuScenes and the source domain.* 3) *The utilization of multiple anchor sizes adds positive gains on Waymo and W→N.* In summary, there is no universally generalizable technique at training time across all locations. From a continual learning perspective, we argue it is more beneficial to fix the original training labels and change the anchor at test time using a small set of labeled target data.

4.3. Effects on different Architectures

In Tab. 5, we evaluate the effect of tuning the anchor size at test time on seven state-of-the-art anchor-based models. Results are reported on the three DA benchmarks exhibiting different object sizes: W→K, W→N, and N→K. We find that: 1) *The performance of all models increases on all 3 benchmarks.* 2) *The effect is more pronounced when the discrepancy between object sizes in the source and target datasets is great.* For instance, the effect is stronger on W→K than on W→N. 3) *This technique more positively influences voxel-only models than point-only or hybrid models.* The AP of VOTR-TSD and PVRCNN post-tuning is lower than voxel-only models.

5. Sensor Domain Gap

We commonly notice an underlying domain gap on the sensor level, which can sometimes lead to very different representations of the 3D environment. It is hard to isolate this domain gap since acquiring the same data with different sensors is practically troublesome. Moreover, common Lidar DA benchmarks are complicated by the fact that the gaps between source and target domains stem from multiple factors at once. For instance, the benchmarks W→N and N→K feature domain gaps on the sensor level and the object size. For this reason, we follow [43] and study the benchmarks which predominantly exhibit a discrepancy on the sensor level, namely K64→K32, K64→K16, K16→K64 and W→N (since the average object sizes between Waymo and NuScenes are close). We make the distinction between two cases: (1) same Vertical-FOV and different number of beams, and (2) different V-FOV. The second case is represented by W→N, resulting in a different sampling pattern and is considerably harder to solve.

5.1. Low-to-High vs. High-to-Low

In Fig. 3, we compare the performance of all models on the benchmarks K64→K16 and add K16→K64, presenting a high-to-low and low-to-high domain gaps respectively. We report the mean AP for all 3 classes (car, pedestrian and cyclist) and add oracles on K64 and K16 as well. The results consistently show that the *high-to-low domain gap is much harder to tackle than the low-to-high gap*, as all mod-

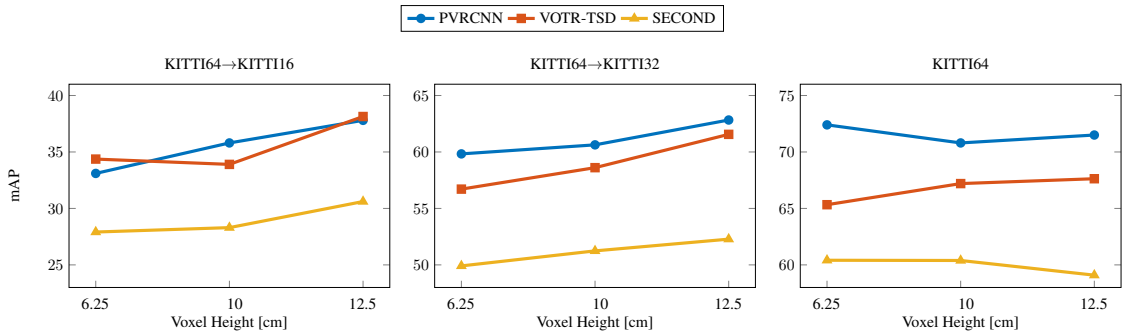


Figure 4. We train each model of (VOTR-TSD, PVRCNN, and SECOND) on K64 with three different voxel heights. We evaluate these nine training experiments on K64, and the lower resolutions K32, and K16. mAP for all classes is reported. Larger heights increase performance on the unseen target domains (2-7 mAP points), while keeping the source performance relatively stable (2-3 mAP points).

Source Domain - 64 Lines Target Domain - 32 Lines

Figure 5. **Left:** Source domain with 64 lines. **Right:** In a target domain with 32 lines, a large number of voxels will be empty if the voxel height is too small (blue voxel), but this number will be reduced if the voxel height is increased (red voxel).

els trained on K16 achieve a higher mAP on K64 (out-of-domain) than K16 (in-domain). While the performance of K16→K64 is still lower than the Oracle on K64, the difference is small for most models. In the rest of this section, we focus on the High-to-Low discrepancy.

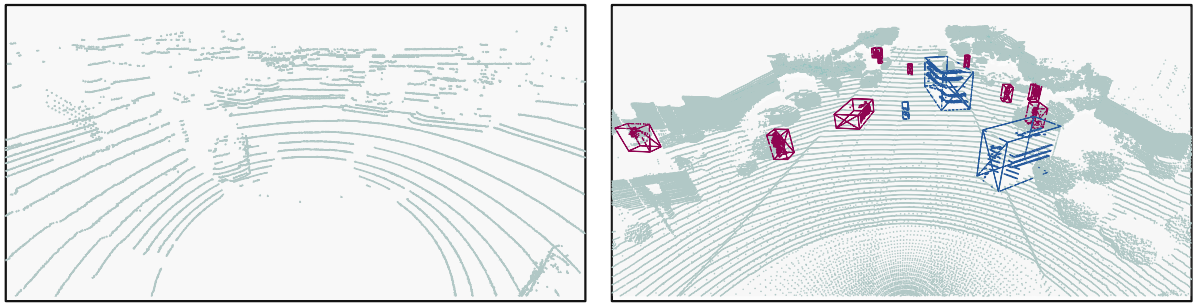
5.2. Voxel Encoding

In [36, 45], it is observed that reducing the voxel size improves source domain performance. The impact of this change on the model's ability to handle sparser pointclouds at test time remains unclear. A common view in the field [43] is that a smaller number of points per foreground object at test time is behind the performance drop. We reframe this explanation on the voxel level: a smaller number of beams results in more empty voxels vertically, which will not be used for detection. The smaller the voxel height, the stronger this effect becomes. On the other hand, a larger height would reduce the difference in the number of empty voxels between source and target domains. In our investigation, we train models with different voxel heights in Fig. 4. We validate this hypothesis on three models (SECOND, PVRCNN and VOTR-TSD), by changing the voxel height from $0.1m$ on KITTI to $0.0625m$ (small) and $0.125m$ (high). Results confirm *that there is a positive correlation between voxel height and robustness on low-resolution sensors* as the 3D mAP increases in most experiments on the target domain (K32 and K16) while remaining relatively stable on the source domain (K64). The improvement is especially pronounced in the sparsest domain (K16), where models exhibit an increase of 3 to 7 mAP points. A simple conceptual illustration is drawn in Fig. 5.

5.3. Data Augmentations

In Tab. 6, we investigate the influence of data augmentations on model robustness across the three benchmarks in the High-to-Low domain gap. In particular, Groundtruth-Sampling (GT-sampling) is a widely used technique that augments the number of foregrounds by drawing samples from a database of source domain objects. We also evaluate shape augmentation (SA) proposed in [54], which removes certain parts of the object and/or downsamples its points. For more investigations, we introduce two other augmentations (see Fig. 6). (1) Line downsampling (LD), inspired by [43], consists in downsampling Lidar beams of the source scans in the vertical direction by a factor of 2 with a probability $p_{d=2} = 0.3$ and by a factor of 4 with a probability $p_{d=4} = 0.2$. Unlike [43], no teacher-student training is needed. (2) A variant of GT-sampling, which we term Mixed GT-Sampling, consists in adding samples extracted from another dataset with a different V-FOV compared to the source dataset. In this case, we add samples from KITTI for W→N and samples from NuScenes for the KITTI benchmarks. This technique can also be used with pointclouds synthesized by generative models like [2, 23].

We find that: 1) *GT-Sampling has negative effects on some classes*, as the detection accuracy deteriorates on pedestrians in K64→K32-16 and cars in W→N. We hypothesize this is due to overfitting on source domain shapes. 2) *Mixed GT-Sampling can restore the performance drop of GT-Sampling* and adds small gains, reducing the overfitting of the original method. 3) *The impact of point sparsity augmentations on robustness is the most significant*. SA and LD serve as potent domain randomization tools that enhance performance on the sparser target. The mAP can even exceed the PVRCNN oracle on K32. It is worth noting that in the K64→K32-16 benchmarks, LD is employed to perfectly mimic the target domain, which accounts for the out-



(a) Line Downsampling (b) Mixed Groundtruth Sampling

Figure 6. Visualizattion of the two introduced augmentations on the Waymo dataset. In (b), objects with different colors denote the augmented groundtruth samples from the target domain (KITTI).

Model	Augmentation	W→N			K64 → K32			K64 → K16		
		Car	Ped	Cyc	Car	Ped	Cyc	Car	Ped	Cyc
PVRCNN	No Aug	20.36	5.79	**0.53**	77.62	53.96	50.3	54.07	27.13	26.36
	GT-Sampling	15.86	5.29	0.0	77.97	47.83	60.54	57.61	14.45	25.44
	Mixed GT-Sampling	20.57	7.62	0.0	78.79	55.37	50.74	55.92	23.93	27.23
	Shape Augmentation (SA)	20.16	5.73	0.0	78.46	54.98	56.11	59.08	33.74	30.45
	Line Downsampling (LD)	**23.97**	**10.57**	0.0	**82.72**	**61.28**	**64.24**	**71.57**	**51.64**	**46.33**
	Oracle	37.85	24.56	1.67	81.45	51.75	61.55	72.71	53.05	49.8

Table 6. Impact of common and introduced data augmentations on the OOD performance in high-to-low resolution domain gaps. SA and LD are found to consistently improve the AP on target domains, while the widely used GT-Sampling causes overfitting on some classes.

standing performance. However, this controlled experiment shows that if point sparsity arises from a different number of beams while maintaining the same V-FOV, this straightforward technique will enhance the model's robustness. It is also the highest performing on the more challenging W→N.

6. Weather Domain Gap

The weather domain gap in Lidar is particularly hard to address: it introduces artifacts such as missing points or clutters resulting from light scattering. Sometimes a significant part of the pointcloud can be missing [44]. We choose to perform experiments on the Waymo to Kirkland [37] benchmark similar to SPG [44] instead of introducing simulated weather effects on existing datasets.

6.1. Transferability versus Discriminability

Previous research [44] has demonstrated a significant decrease in performance on bad weather data. This has often been formulated as a transferability problem: how can the models trained on good weather data transfer to bad weather data? However, we notice that a simple *in-domain* evaluation of common 3D detectors on bad weather data is missing. In Figure 7, we assess different models using the Kirkland validation split. Each model is trained two times, once on Waymo data (W→Kr) and once on the Kirkland training split, which is regarded as the oracle performance or upper bound. Interestingly, *we find that models trained on clean weather data exhibit comparable or even superior generalization to poor weather conditions compared to those trained on bad weather data*. This is especially true for the class Pedestrian, which exhibits the most deterioration in adverse weather. This finding aligns with a previous work [39] but takes it a step further: while [39] demonstrates that training on abundant clear weather data (source) is better than training on mixed weather samples (source + target), we show that it is even better than training on a large dataset of target domain only samples. Note that we use a much larger dataset of bad weather conditions than [39]. This reveals that detection in bad weather conditions is not merely a transferability problem; rather, it is a discriminability problem where simple supervised learning is hard. We believe more research should focus on building better models for this domain or on integrating other modalities (camera, radar) to improve detection in adverse conditions.

6.2. Data Augmentations and Voxel Size

In Tab. 7, we report the result of different design choices using PVRCNN. We find that: 1) *The weather domain gap is not significantly affected by augmentations*. GT-Sampling shows modest gains on the pedestrian class, while the other augmentations slightly increase the vehicle class. 2) *PVRCNN with a smaller voxel size trained only on* 20% *of Waymo dataset is able to outperform SPG [44] on the Vehicle class*, as SPG uses a larger voxel size. This highlights again the importance of choosing the right voxel size as it can increase or decrease the model's robustness. 3) *Training on source and target does not improve the performance.*

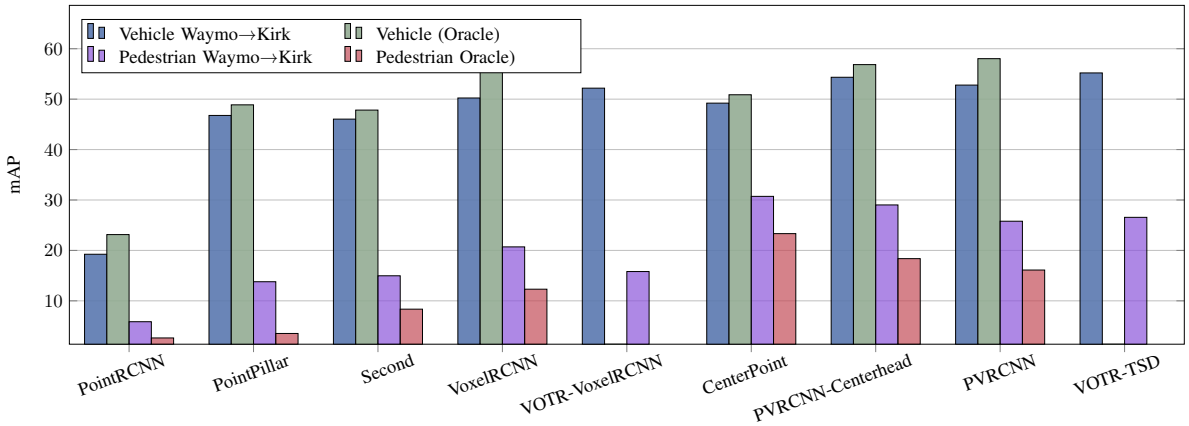


Figure 7. Evaluation of all models on the Waymo→Kirk benchmark, with Oracle included. The mAP is reported for cars and pedestrians since Kirk has no cyclist class. Surprisingly, the source-only model performs better than the oracle itself, especially on the pedestrian class, revealing the challenge of simple supervised learning on bad weather data.

Method	Waymo		Waymo → Kirk	
	Veh.	Ped.	Veh.	Ped.
No Augmentation w/ Voxel (0.1, 0.1, 0.15)	71.22	67.72	55.48	22.20
GT-Sampling	71.01	**68.23**	55.02	23.85
GT-Sampling + LD	71.22	67.62	**56.07**	22.04
GT-Sampling + SA	**71.66**	68.11	55.72	21.48
Mixed Domain training (source + target)	70.58	65.91	54.24	21.34
SPG* w/ Voxel (0.2, 0.2, 0.3) [44]	70.63	62.31	53.51	**26.44**
No Augmentation w/ Voxel (0.1, 0.1, 0.1)	71.27	68.21	55.78	23.73

Table 7. Results of different augmentation and voxel size setttings on Waymo → Kirk using PVRCNN. * trains on the whole Waymo dataset. No setting is optimal, but a small voxel size can outperform SPG on the class Vehicle, even though no DA method is used.

7. Discussion and Conclusion

We explored the robustness of Lidar-based 3D detectors across three main domain gaps: sensor, weather, and location. We evaluated nine state-of-the-art models on six common Domain Adaptation benchmarks. By conducting large-scale experiments and observing various design choices in the 3D-OD pipeline - architecture, data augmentation, anchor size, and voxel size - we draw several important findings across this pipeline.

Anchor Size

1. Changing anchor size at test-time can notably enhance model performance across various locations, indicating that contrary to common belief, models have learned a generalizable representation, albeit at different scales.
2. Training with larger anchors is recommended for practical applications, as the optimal anchor is always found to be smaller than the training anchor. This can enable semi-supervised, federated, and test-time DA without model retraining, allowing a seamless transition between locations by only changing the anchor.

Architecture and Voxel Size

1. Point-voxel representations in the backbone show better robustness across most domain pairings.
2. Voxel transformer backbones are more robust than 3D CNNs when coupled with point features.
3. Anchorless detectors are the most robust in adverse weather.
4. While shorter voxel heights generally improve performance on the source domain, increasing voxel height improves the robustness when training on high-resolution sensors and inferring on low-resolution sensors.

Data and Augmentations

1. While ground-truth sampling is known to enhance performance on the source domain, we find it has negative effects on the robustness on lower-resolution sensors, likely due to overfitting on the source domain shapes. Adding groundtruth samples from another dataset with a different V-FOV reduces the overfitting.
2. Point sparsity augmentations, like shape and line downsampling augmentations, prove effective on high-to-low resolution domain gaps. Differences in the V-FOV remain the most challenging.
3. Going from high-to-low resolution is more challenging than the reverse due to point sparsity at test-time, with models performing better on higher-resolution test data.
4. Surprisingly, training on clean weather samples leads to more robustness on bad weather than direct training on bad weather or mixed samples. This highlights the importance of choosing the right training data but also points out that the problem with bad weather data is about discriminability rather than transferability.

While our study is exhaustive, there is further work to be done. We have shown that simple architecture and data tricks can improve robustness. Yet, a universally adaptive model remains elusive. We hope our benchmarks and in-depth analysis can benefit the 3D-OD and DA communities.

References

[1] Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10231–10241, 2021. 2, 3

[2] Lucas Caccia, Herke van Hoof, Aaron C. Courville, and Joelle Pineau. Deep generative modeling of lidar data. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5034–5040, 2019. 6

[3] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In CVPR, 2020. 1

[4] Ken Chatfield, Victor S Lempitsky, Andrea Vedaldi, and Andrew Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In BMVC, page 8, 2011. 2

[5] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. arXiv preprint arXiv:1405.3531, 2014. 2

[6] Chaoqi Chen, Zebiao Zheng, Xinghao Ding, Yue Huang, and Qi Dou. Harmonizing transferability and discriminability for adapting object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8869–8878, 2020. 2

[7] Chaoqi Chen, Zebiao Zheng, Yue Huang, Xinghao Ding, and Yizhou Yu. I3net: Implicit instance-invariant network for adapting one-stage object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12576–12585, 2021.

[8] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3339–3348, 2018. 2

[9] Robert DeBortoli, Li Fuxin, Ashish Kapoor, and Geoffrey A Hollinger. Adversarial training on point clouds for sim-to-real 3d object detection. IEEE Robotics and Automation Letters, 6(4):6662–6669, 2021. 1, 2

[10] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, pages 1201–1209, 2021. 3

[11] Guangyao Ding, Meiying Zhang, E Li, and Qi Hao. Jst: Joint self-training for unsupervised domain adaptation on 2d&3d object detection. In 2022 International Conference on Robotics and Automation (ICRA), pages 477–483. IEEE, 2022. 1, 2, 4

[12] Yinpeng Dong, Caixin Kang, Jinlai Zhang, Zijian Zhu, Yikai Wang, Xiao Yang, Hang Su, Xingxing Wei, and Jun Zhu. Benchmarking robustness of 3d object detection to common corruptions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1022–1032, 2023. 2

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

[14] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, M. Bethge, Felix Wichmann, and W. Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. ArXiv, abs/1811.12231, 2019. 2, 3

[15] Martin Hahner, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. 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, pages 15283–15292, 2021. 2

[16] Deepti Hegde and Vishal Patel. Attentive prototypes for source-free unsupervised domain adaptive 3d object detection. arXiv preprint arXiv:2111.15656, 2021. 1, 2

[17] Deepti Hegde, Velat Kilic, Vishwanath Sindagi, A Brinton Cooper, Mark Foster, and Vishal M Patel. Source-free unsupervised domain adaptation for 3d object detection in adverse weather. In 2023 IEEE International Conference on Robotics and Automation (ICRA), pages 6973–6980. IEEE, 2023. 1, 2

[18] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9924–9935, 2022. 2, 3

[19] Cheng-Chun Hsu, Yi-Hsuan Tsai, Yen-Yu Lin, and Ming-Hsuan Yang. Every pixel matters: Center-aware feature alignment for domain adaptive object detector. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16, pages 733–748. Springer, 2020. 2

[20] Christoph Kamann and Carsten Rother. Benchmarking the robustness of semantic segmentation models with respect to common corruptions. International journal of computer vision, 129:462–483, 2021. 2

[21] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In CVPR, 2019. 3

[22] Alexander Lehner, Stefano Gasperini, Alvaro Marcos-Ramiro, Michael Schmidt, Mohammad-Ali Nikouei Mahani, Nassir Navab, Benjamin Busam, and Federico Tombari. 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, pages 17295–17304, 2022. 2

[23] Ruihui Li, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-gan: A point cloud upsampling adversarial network. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 7202–7211, 2019. 6

[24] Yu-Jhe Li, Xiaoliang Dai, Chih-Yao Ma, Yen-Cheng Liu, Kan Chen, Bichen Wu, Zijian He, Kris Kitani, and Peter Vajda. Cross-domain adaptive teacher for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7581–7590, 2022. 2

[25] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In European conference on computer vision, pages 1–18. Springer, 2022. 1

[26] Zhipeng Luo, Zhongang Cai, Changqing Zhou, Gongjie Zhang, Haiyu Zhao, Shuai Yi, Shijian Lu, Hongsheng Li, Shanghang Zhang, and Ziwei Liu. Unsupervised domain adaptive 3d detection with multi-level consistency. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 8866–8875, 2021. 1, 2

[27] Dušan Malić, Christian Fruhwirth-Reisinger, Horst Possegger, and Horst Bischof. Sailor: Scaling anchors via insights into latent object representation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 623–632, 2023. 1, 2, 4

[28] Jiageng Mao, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiashi Feng, Xiaodan Liang, Hang Xu, and Chunjing Xu. Voxel transformer for 3d object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 3164–3173, 2021. 3

[29] Muhammad Jehanzeb Mirza, Cornelius Buerkle, Julio Jarquin, Michael Opitz, Fabian Oboril, Kay-Ulrich Scholl, and Horst Bischof. Robustness of object detectors in degrading weather conditions. In 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), pages 2719–2724. IEEE, 2021. 2

[30] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. Advances in Neural Information Processing Systems, 34:23296–23308, 2021. 2

[31] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In Proceedings of the European Conference on Computer Vision (ECCV), pages 464–479, 2018. 2

[32] C. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 77–85, 2017. 3

[33] Mohamed Sayed and Gabriel Brostow. Improved handling of motion blur in online object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 1706–1716, 2021. 2

[34] Seonguk Seo, Yumin Suh, Dongwan Kim, Geeho Kim, Jongwoo Han, and Bohyung Han. Learning to optimize domain specific normalization for domain generalization. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16, pages 68–83. Springer, 2020. 2

[35] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 770–779, 2019. 1, 3

[36] 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 Conference on Computer Vision and Pattern Recognition, 2020. 1, 3, 6

[37] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 2446–2454, 2020. 1, 7

[38] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020. 2, 3, 4

[39] Teja Vattem, George Sebastian, and Luka Lukic. Rethinking lidar object detection in adverse weather conditions. In 2022 International Conference on Robotics and Automation (ICRA), pages 5093–5099. IEEE, 2022. 7

[40] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 913–922, 2021. 1

[41] Yan Wang, Xiangyu Chen, Yurong You, Li Erran, Bharath Hariharan, Mark E. Campbell, Kilian Q. Weinberger, and Wei-Lun Chao. Train in germany, test in the usa: Making 3d object detectors generalize. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 11710–11720, 2020. 1, 2, 4

[42] Yan Wang, Junbo Yin, Wei Li, Pascal Frossard, Ruigang Yang, and Jianbing Shen. Ssda3d: Semi-supervised domain adaptation for 3d object detection from point cloud. In Proceedings of the AAAI Conference on Artificial Intelligence, 2023. 2

[43] 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 Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX, pages 179–195. Springer, 2022. 2, 3, 5, 6

[44] 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. 1, 2, 7, 8

[45] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. Sensors, 18(10):3337, 2018. 1, 3, 4, 6

[46] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 10368–10378, 2021. 1

[47] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 10368–10378, 2021. 1, 2, 3, 4

[48] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d++: denoised self-training for unsu-

pervised domain adaptation on 3d object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022. 2, 4

[49] 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. 1, 2

[50] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 11784–11793, 2021. 3

[51] Kaicheng Yu, Tang Tao, Hongwei Xie, Zhiwei Lin, Tingting Liang, Bing Wang, Peng Chen, Dayang Hao, Yongtao Wang, and Xiaodan Liang. Benchmarking the robustness of lidar-camera fusion for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3187–3197, 2023. 2

[52] Chongzhi Zhang, Mingyuan Zhang, Shanghang Zhang, Daisheng Jin, Qiang Zhou, Zhongang Cai, Haiyu Zhao, Xianglong Liu, and Ziwei Liu. Delving deep into the generalization of vision transformers under distribution shifts. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7277–7286, 2022. 2

[53] Diankun Zhang, Xueqing Wang, Zhijie Zheng, and Xiaojun Liu. Unsupervised domain adaptive 3-d detection with data adaption from lidar point cloud. IEEE Transactions on Geoscience and Remote Sensing, 60:1–14, 2022. 1, 2

[54] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu. Se-ssd: Self-ensembling single-stage object detector from point cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14494–14503, 2021. 6

[55] Zijian Zhu, Yichi Zhang, Hai Chen, Yinpeng Dong, Shu Zhao, Wenbo Ding, Jiachen Zhong, and Shibao Zheng. Understanding the robustness of 3d object detection with bird's-eye-view representations in autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 21600–21610, 2023. 2