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

In the supplementary material, we provide more details and experiments.

- Sec. 8: implementation details for the conducted experiments.
- Sec. 9: related works for lidar-based 3D-OD.
- Sec. 10: further experiments concerning the location domain gap. This includes the effects of the voxel size, the impact of various anchor sizes, the optimization procedure, and single-model-multiclass experiments.
- Sec. 11: further experiments on the effect of augmentations on VOTR-TSD and SECOND.
- Sec. 12: Analysis of True Positives, False Positives and False Negatives per model and dataset
- Sec. 13 We report the results of multiple runs on two benchmarks.
- Sec. 14: Analysis of Recall at various overlaps and on different benchmarks.

8. Implementation details

For all experiments, we have used the OpenPCDet repository [49] and the official implementation of VOTR-TSD [33]. For all models, we use a batchsize of 8 to fit on our GPUs. To study the isolated effects of the architectural design choices, voxel encoding, and anchor size, we train the models without GT-Sampling.

9. Related Works: 3D Object Detection

Lidar-based 3D Object Detection can be divided into three groups: point-based, range image-based, and voxel-based. Point-based methods [41, 44, 61] extract 3D structural features from the raw points directly using permutation invariant feature extractors like Pointnet [39, 40]. Range image detectors project the pointcloud onto a 2D plane using spherical projection and employ 2D CNNs for detection [1, 15, 48]. Voxel-based methods divide the pointcloud into regular voxels and encode the points inside the voxels using point operations; then, they employ 3D and 2D backbones to generate 3D boxes. VoxelNet [69] is a seminal work that implements 3D convolutions on the pointclouds. SECOND [57] is a first-stage detector that introduces 3D sparse convolutions to boost the efficiency of 3D backbones. CIA-SSD [67] and SE-SSD [68] both build upon SECOND: CIA-SSD adds IoU prediction to the total loss and uses the predicted IoU values to correct classification scores prior to Non-Maximum Suppression (NMS), while SE-SSD employs a teacher-student framework with diverse shape augmentation strategies to boost the network's capacity of detecting different object shapes. PointPillars [26] encodes the pointcloud into bird's eye view (BEV) features and uses a 2D network on this representation. Centerformer [70] is a one-stage anchorless detector that leverages a DETR [5] transformer in its detection head. There have also been many two-stage approaches that exploit voxel operations using 3D convolutions in the backbone like [12, 34, 45, 46] or 3D transformers like [16, 35]. PV-RCNN [45], Pyramid-RCNN [34] and VOTR-TSD [35] use point features in the detector to refine the proposals, while VoxelRCNN uses pure voxel-level features eliminating the substantial computational overhead of point-level features. CenterPoint [63] presents an anchorless two-stage detector. BtcDet [56] predicts objects' occupancy in the occluded areas and leverages the occupancy to refine the proposals. M3DETR [19] extracts features from different representations (voxels, points, BEV) and computes the relationships between these features using transformers before using standard detection heads. In this work, we focus on voxel-based methods since they are getting increasingly more attention and have demonstrated high performance on standard detection benchmarks like KITTI [17] and Waymo [47]. Moreover, works on domain adaptation typically employ voxel-based methods, which motivates a closer inspection of these frameworks.

Model	Voxel Size	Waymo	W → K	W → N
PVRCNN	[0.1, 0.1, 0.125]	**62.95**	16.25	18.81
	[0.1, 0.1, 0.15]	62.60	**16.41**	18.69
	[0.1, 0.1, 0.2]	62.41	16.33	19.51
	[0.1, 0.1, 0.25]	62.32	15.94	**19.83**
VOTR-TSD	[0.1, 0.1, 0.125]	64.69	**16.97**	19.85
	[0.1, 0.1, 0.15]	**65.20**	14.46	**21.66**
	[0.1, 0.1, 0.2]	64.09	15.35	20.59
	[0.1, 0.1, 0.25]	64.43	16.86	20.43

Table 8. Impact of the voxel size on the out-of-domain performance across different locations. The Car 3D-AP is reported for PVRCNN and VOTR-TSD.

10. Location Domain Gap: Further Experiments

Voxel Size. In Tab. 8, we explore the effect of the voxel size on performance across multiple geographical locations. No significant effect is observed on W→K. This can be attributed to the primary domain shift type being the object size rather than point sparsity, which renders the anchor size the primary performance driver at test time. However, on W→N, some performance gain is observed when choosing a higher voxel height, as NuScenes has a lower resolution than Waymo.

Models	Car	Ped	Cyc
PointRCNN	5.04 / 38.15	28.22 / 31.76	0 / 0
PointPillars	12.75 / 66.27	48.34 / 49.47	34.9 / 38.51
Second	9.91 / 65.71	41.39 / 46.09	22.74 / 25.37
VoxelRCNN	20.09 / 59.70	55.33 / 59.82	34.81 / 40.79
VOTR-VoxelRCNNhead	21.34 / 55.91	29.96 / 28.09	1.63 / 3.55
PVRCNN	16.61 / 55.99	50.22 / 56.31	34.09 / 34.02
VOTR-TSD	15.75 / 59.39	46.19 / 48.11	39.28 / 46.82

Table 9. Benchmarking anchor-based architectures on W→K. We report the performance before and after anchor optimization for all classes and denote a consistent improvement for all models.

Anchor Size			BEV / 3D AP
3.9	1.6	1.56	52.80 / 11.05
3.8	1.6	1.56	61.61 / 16.24
3.7	1.6	1.56	66.17 / 21.58
3.6	1.6	1.56	68.76 / 24.63
3.5	1.6	1.56	68.90 / 23.70
3.4	1.6	1.56	68.02 / 20.55
3.6	1.5	1.56	79.26 / 49.86
3.6	1.4	1.56	82.11 / 63.49
3.6	**1.3**	1.56	**82.50** / 65.03
3.6	1.2	1.56	80.38 / 60.18
3.6	**1.3**	**1.5**	81.38 / **68.19**
3.6	1.3	1.4	80.45 / 58.01
3.6	1.3	1.6	82.53 / 60.88

Table 10. Illustration of the anchor optimization procedure on the SECOND model on Waymo→KITTI. The first row denotes the default training and testing anchor size.

Multiclass Experiments. In Tab. 9, we benchmark all anchor-based architectures on the W→K benchmark, reporting the 3D-AP for all classes. The results show an increase in performance for all models across all classes when tuning the test-time anchor size. This is more pronounced in cars and cyclists than pedestrians, as these classes exhibit a more significant change in size across datasets than pedestrians. Note the results of the class car are slightly different from Tab. 5, where all models were trained only on the class car.

Anchor Size Optimization Procedure. The anchor size optimization procedure is shown in Tab. 10 on the SECOND model for Waymo → KITTI. We adopt a greedy-like optimization approach, which involves altering a single size dimension while maintaining the others constant. Subsequently, the dimension that improves the BEV/3D-AP the most is fixed, and the process is repeated for the next dimension until all three dimensions have been optimized. Our findings reveal a notable relationship between the length and width of the anchors and the overall detection efficacy in this specific benchmark. Notably, adjustments to the width dimension have a pronounced effect on the OOD performance. This observation suggests that objects in the KITTI dataset are generally narrower compared to those in Waymo, implying that smaller anchor widths are more suitable for the KITTI objects. However, the determination of which dimension is most crucial varies depending on the size characteristics of objects in the source and target datasets. For test-time applications, we select the optimal anchor size, which is emphasized in bold in the table.

Test-time Anchor	NuScenes→Waymo
Training [3.9, 1.6, 1.56]	17.31 / 14.94
Best performing [3.8, 1.6 , 1.4]	18.13 / 15.49
Training [4.2, 2.0, 1.6]	18.03 / 15.39
Best performing [4.2, 2.1, 1.50]	22.07 / 18.86
Training [4.80, 2.11, 1.79]	21.01 / 17.96
Best performing [4.60, 2.11 , 1.70]	22.15 / 18.93

Table 11. Anchor Study on NuScenes→Waymo. LEVEL_1 and LEVEL_2 AP are reported for the class Vehicle.

Tuning the Anchor Size on Target Datasets with Larger Objects. In Tab. 11, we report the effect of changing the anchor size of SECOND on N→W, where the target objects in this setting are larger than the source objects. We find that the best-performing anchors are still smaller than the training anchors. A larger training anchor can provide better source-only performance than smaller training anchors and allows for an even higher performance at test-time after tuning the anchor size. This shows the importance of choosing a large training anchor size, which sets a wide enough spatial prior for object detection at different sizes. Nevertheless, the enhancements in this domain gap are not as significant as those observed in the transition from Waymo to KITTI. This is primarily because the average object sizes in both datasets are close, and the primary difference in the domain gap is attributed to the resolution and vertical field-of-view.

11. Sensor Domain Gap: Further Experiments

In Tab. 12, we investigate the effect of augmentations on VOTR-TSD and SECOND. We add PVRCNN from Tab. 6 for comparison purposes. We include the oracle and two domain adaptation models ST3D [59] and Beam-Distillation [54]. We notice that foreground augmentations (SA, GT-Sampling, Mixed GT-Sampling) can provide substantial improvements in some cases but are model-dependent and class-dependent. On the other hand, line downsampling proves to provide the most consistent and highest improvement on this domain gap on all benchmarks and models. It is able to come close to and even sometimes outperform the two presented DA models, highlighting the importance of augmentations during source-domain training.

Model	Augmentation	W→N			K64 → K32			K64 → K16		
		Car	Ped	Cyc	Car	Ped	Cyc	Car	Ped	Cyc
SECOND	No Aug	17.84	4.44	0.22	73.50	41.11	39.10	50.71	16.63	17.58
	GT-Sampling	17.86	5.62	**3.68**	73.50	38.96	51.04	55.01	9.69	22.87
	Mixed GT-Sampling	17.28	2.36	0.03	**83.20**	42.80	41.75	48.72	17.43	17.98
	Shape Augmentation (SA)	17.23	4.56	0.20	72.01	45.80	43.54	53.07	24.45	21.93
	Line Downsampling (LD)	**20.92**	**7.64**	0.60	75.37	**47.35**	**53.88**	**65.30**	**41.16**	**37.89**
SECOND	ST3D [59]	20.19	5.11	3.35	61.94	-	-	52.17	-	-
	Beam-Distillation [54]	22.86	-	-	74.33	-	-	65.13	-	-
	Oracle	30.30	16.79	0.0	76.61	41.43	49.74	68.34	42.96	40.70
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**
PVRCNN	ST3D [59]	22.99	-	-	-	-	-	-	-	-
	Beam-Distillation [54]	25.63	-	-	-	-	-	-	-	-
	Oracle	37.85	24.56	1.67	81.45	51.75	61.55	72.71	53.05	49.8
VOTR-TSD	No Aug	21.32	7.0	3.48	76.29	49.78	49.76	55.61	21.71	24.38
	GT-Sampling	20.15	8.09	**6.34**	78.5	41.81	**62.13**	59.37	19.04	31.25
	Mixed GT-Sampling	21.54	8.22	2.41	59.49	38.64	40.68	35.86	20.27	16.82
	Shape Augmentation (SA)	20.26	7.15	3.43	75.5	51.81	50.44	57.36	29.21	27.84
	Line Downsampling (LD)	**24.82**	**9.41**	5.22	**80.3**	**54.81**	50.26	**69.39**	**49.32**	**50.88**
	Oracle	38.13	22.51	2.43	82.47	50.69	66.08	73.75	53.45	50.69

Table 12. 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 GT-Sampling deteriorates pedestrian detection.

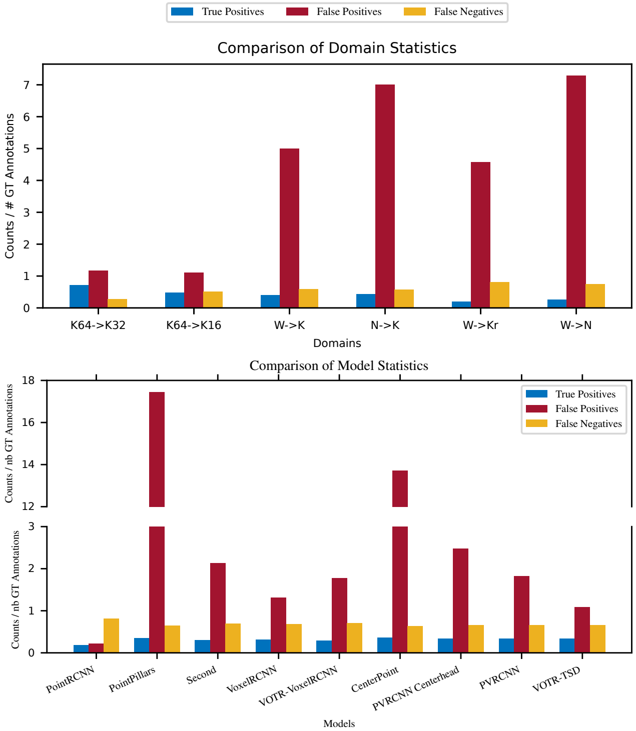

Figure 8. Number of TPs, FPs, and FNs, normalized by the corresponding number of groundtruth annotations.

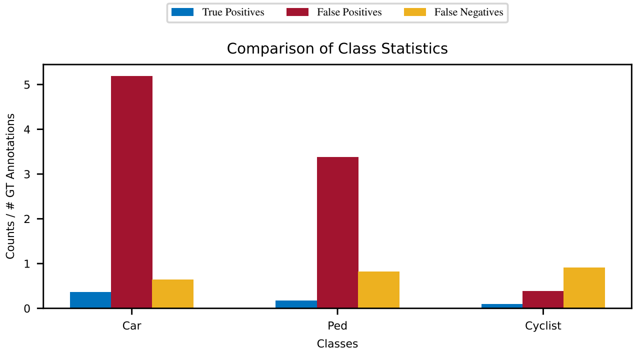

Figure 9. Number of TPs, FPs, and FNs per class across all models, normalized by the corresponding number of groundtruth annotations.

12. TP/FP/FN Analysis

In Fig. 8, we conduct further analysis, reporting the number of True Positives (TP), False Positives (FP) and False Negatives (FN) for each 3D object model on the six studied benchmarks. The analysis shows: (1) Going from high-to-low resolution results in more FN. The number of FN in K64→K16 is higher than the number of FN in K64→K32. The FN ratio is also very high on W→N and W→Kr, showing how challenging it is to detect objects when the target domain is sparser than the source domain. (2) Across different geograohical locations, the number of FPs is high. This number can be largely mitigated by tuning the anchor size on the target data. (3) Some models are very prone to FP like PointPillars and CenterPoint, which generate up to 17 FPs for every groundtruth label. (4) Point-based model PointRCNN fails to detect many objects, resulting in the largest number of FN among all studied detectors. (5) The number of FNs among detector shows little variations, showing there is still work to be done to improve the detection of objects in sparser target domains. (6) VOTR-TSD

Features	Architecture	Method	K64→K32 (R)			N→K (G+R)		
			Car	Ped	Cyc	Car	Ped	Cyc
Point	MLP	PointRCNN	76.8	50.86	59.42	12.14	26.27	0
Voxel	Conv	PointPillars	72.43	40.47	27.02	0	0	0
	Conv	Second	74.93	44.65	43.17	8.52	14.9	0.002
	Conv	VoxelRCNN	80.07	56.55	59.66	8.21	20.25	0
	ViT	VOTR-VoxelRCNN	80.07	52.01	48.44	21.17	27.63	0.68
Hybrid	Conv	CenterPoint	72.74	45.42	47.19	9.912	20.69	0.06
	Conv	PVRCNN Centerhead	71.99	45.43	35.82	26.13	25.26	0.016
	Conv	PVRCNN	78.12	54.48	52.25	14.42	16.6	0
	ViT	VOTR-TSD	77.21	48.87	51.68	27.1	26.42	4.12

Table 13. Multiple runs experiments. We train each model five times on each benchmark and report the average per class. Variations across different runs are found to be small.

shows the smallest number of FPs among the hybrid and voxel-based methods (PointRCNN has a smaller FP ratio, but it is likely because it generates fewwer bounding boxes than the rest of the models, as reflected in the high FN ratio). (7) Adding point features has a different effect on each backbone: when added to VOTR, point features decrease the number of FPs (compare VOTR-VoxelRCNN to VOTR-TSD). When added to 3D CNNs, the number of FPs slightly increase while FNs slightly decrease.

In Fig. 9, we report the number of TP, FP, and FN for the three classes across all models and datasets. Clearly, smaller and rarer classes in the source datasets (cyclists, pedestrians) suffer mostly from FN. On the other hand, a very large number of FP cars. While previously mentioned factors (point sparsity, object size) are mainly the cause of this discrepancy between classes, the labeling technique may also vary across datasets. For instance, the Vehicle class in Waymo includes cars, trucks, and even motorcycles, which is different from all other datasets like KITTI and NuScenes. This could lead to many FP detections.

13. Statistical Analysis

In Tab. 13, we report the results of multiple runs (5 per model) on two benchmarks (K64→K32 and N→N). Results show mall variations across experiments.

14. Recall Analysis

In Tab. 14, We measure the recall of PVRCNN at various overlaps (0.3, 0.5, and 0.7) in different settings and benchmarks. First, on W→K, the recall at IoU= 0.3, 0.5 is almost the same with or without anchor optimization. However, we notice a large drop in the recall at IoU= 0.7, indicating the existence of localization errors in this domain gap. The tuned anchor addresses this problem and boosts the recall to 0.61.

IoU	eval on W→K	
	w/o Anchor Tuning	w/ Anchor Tuning
0.3	0.95	0.95
0.5	0.88	0.91
0.7	0.34	0.61
IoU	eval on Kirk	
	Trained on Waymo	Trained on Kirk
0.3	0.78	0.74
0.5	0.72	0.70
0.7	0.51	0.53
IoU	eval on K64 w/o Anchor Optimization	
	Trained on W	Trained on N
0.3	0.95	0.44
0.5	0.88	0.39
0.7	0.34	0.17
IoU	eval on K64 w/ Anchor Optimization	
	Trained on W	Trained on N
0.3	0.95	0.54
0.5	0.91	0.51
0.7	0.61	0.38

Table 14. Analyis of Recall at multiple IoU for PVRCNN model on different benchmarks

In the second setting, we evaluate on Kirk a PVRCNN model trained on W and another one on Kirk. While the recall of the model trained on W is slightly better, the difference between the two is small. Both models suffer from many FN (confirmed by Fig. 8), but the model trained on W has fewer FP than the model trained on Kirk, which explains the difference in their AP but their similar recall.

In the third and fourth settings, we measure the recall of PVRCNN on KITTI after training on W and N. The model trained on N has significantly worse recall values at all overlaps, showcasing the challenge of training on different resolutions and field-of-view. The anchor optimization boosts the scores, as it addresses the location domain gap. However, the scores of N are still low due to the presence of the resolution domain gap.