28th *Computer Vision Winter Workshop*
Graz, Austria, February 12–14, 2025

DOI: 10.3217/978-3-99161-022-9-002

An Investigation of Beam Density on LiDAR Object Detection Performance

Christoph Griesbacher[1] Christian Fruhwirth-Reisinger[1,2]

[1]Institute of Visual Computing, TU Graz
[2]Christian Doppler Laboratory for Embedded Machine Learning
{griesbacher, reisinger}@tugraz.at

Abstract

Accurate 3D object detection is a critical component of autonomous driving, enabling vehicles to perceive their surroundings with precision and make informed decisions. LiDAR sensors, widely used for their ability to provide detailed 3D measurements, are key to achieving this capability. However, variations between training and inference data can cause significant performance drops when object detection models are employed in different sensor settings. One critical factor is beam density, as inference on sparse, cost-effective LiDAR sensors is often preferred in real-world applications. Despite previous work addressing the beam-density-induced domain gap, substantial knowledge gaps remain, particularly concerning dense 128-beam sensors in cross-domain scenarios.

To gain better understanding of the impact of beam density on domain gaps, we conduct a comprehensive investigation that includes an evaluation of different object detection architectures. Our architecture evaluation reveals that combining voxel- and point-based approaches yields superior cross-domain performance by leveraging the strengths of both representations. Building on these findings, we analyze beam-density-induced domain gaps and argue that these domain gaps must be evaluated in conjunction with other domain shifts. Contrary to conventional beliefs, our experiments reveal that detectors benefit from training on denser data and exhibit robustness to beam density variations during inference.

1. Introduction

Autonomous driving has been receiving increasing attention in recent years, as it has the potential to increase road safety, traffic efficiency, and reduce emissions. To enable the decision-making capabilities of Advanced Driver Assistance Systems (ADAS) or Automated Driving (AD) technologies, understanding the vehicle's immediate environment is crucial. Light Detection And Ranging (LiDAR) technology stands out as a particularly effective solution for this task through its ability to directly measure three-dimensional distances with high accuracy [27].

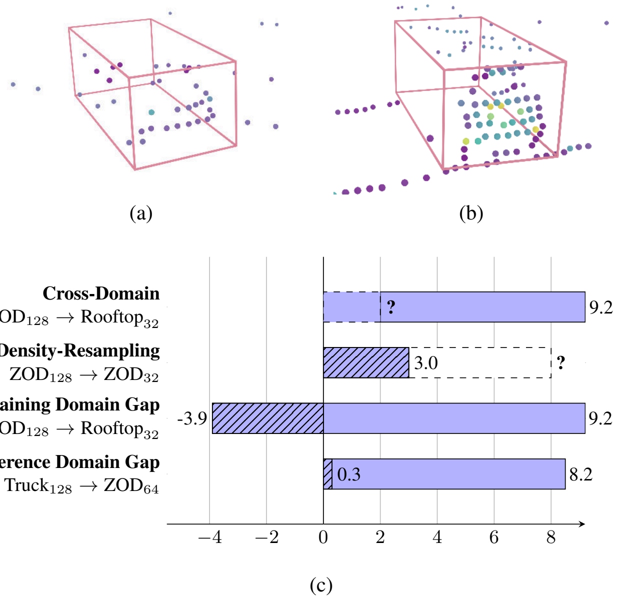

Figure 1. (a) Low-density and (b) high-density scan of vehicles at a similar distance. (c) Overall and beam-density-induced domain gap (in % for IOU=0.4) measured by different methods. The Cross-Domain and Density-Resampling methods fail to assess either the beam-density-induced or overall domain gap, while the Training and Inference Domain Gaps provide a complete picture.

LiDAR-based 3D object detection models have demonstrated impressive performance on established benchmarks [2, 5, 12, 25, 37]. However, their performance often drops significantly when applied across different datasets due to inherent differences between the source domain and the target domain. Typical examples are varying sensor configurations between source and target domain or adverse weather conditions covered by the target but not the source data. When these differences are substantial, the detection model struggles to generalize to the new domain, introducing a performance gap known as the *domain gap*. This challenge is particularly critical in real-world applications, where a domain gap is almost inevitable due to the variability between the training dataset and the diverse conditions encountered in deployment.

The common way to mitigate domain gaps is the application of domain adaptation methods. Such methods are oftentimes tailored towards a specific domain difference, such as different LiDAR resolutions [15, 20, 40] or varying object size distributions [23, 39, 44]. Thus, to successfully apply domain adaptation methods, the most significant domain shifts have to be identified first. A structured domain shift taxonomy is useful in this context, as it helps to categorize and systematically understand the specific shifts between domains, enabling the selection or design of a targeted adaptation technique.

Despite taxonomies of related work [6, 8, 24, 47], no study includes all the domain shifts between the datasets under investigation (described in Sec. 3.1). Thus, we introduce a domain shift taxonomy in Tab. 2. Keeping the goal of domain adaptation in mind, we distinguish between domain shifts that can effectively be addressed by domain adaptation methods and those that persist despite the application of domain adaptation. We call the former *non-persistent* and the later *persistent* domain shifts.

Motivated by the observation that the domain gap varies significantly when employing different detectors, we conduct an object detector architecture evaluation. While prior studies do not give particular attention to a thoughtful selection of an object detection model [31], we aim to identify detectors that are inherently robust against domain changes. By minimizing the initial domain gap, the reliance on domain adaptation is minimized, ensuring that the domain adaptation efforts focus on the most challenging aspects of domain gap. We find that (1) voxel-based detectors robustly detect objects, but have difficulties at precisely localizing them and (2) point-based detectors excel at localizing objects in cross-domain settings. Our experiments suggest that optimal cross-domain detection performance is achieved by combining voxel- and point-based approaches in a two-staged detector.

A particularly important domain shift stems from the number of LiDAR beams (see Fig. 1). High-density LiDAR sensors produce detailed point clouds with a high number of points, easing the accurate estimation of object sizes and positions. Low-density LiDAR sensors, which are often more affordable and more commonly used in large-scale deployments, capture fewer points, leading to sparser point clouds and less reliable detection results. This difference in beam density creates a domain shift when models trained on high-density LiDAR data are applied to low-density data and vice versa. To analyze the domain gap caused by varying beam densities, related studies utilize one of two approaches. The first approach [15] involves multiple datasets employing LiDAR sensors with varying beam densities which are subsequently compared. The second approach [8, 31, 40] is based on downsampling a dense dataset to create sparser twin-dataset with varying beam density which are subsequently compared.

This paper highlights the shortcomings of the existing methods. First, comparing the domain gap between two datasets does not guarantee that the observed domain gap actually stems from varying beam density or is caused by other domain shifts occurring between the investigated datasets. The second approach leads to ambiguous results because it analyzes the effect of beam density in isolation of other domain shifts. In real-world applications, the beam-density-induced domain gap is always accompanied by other effects influencing the domain gap. We show that the beam-density-caused domain gap has to be assessed *in conjunction* with other domain shifts to accurately evaluate its impact in real-world applications. Our experiments suggest that (1) in contrast to the results of related studies [8, 10, 31], it is more beneficial to train object detectors on dense data, independent of the density of the target data and (2) concerning the inference domain gap, detectors are robust against a change of up to 64 beams (see Fig. 1c). Our contributions can be summarized as follows:

- We introduce a domain shift taxonomy based on macro-, sensor-, and object-level domain shifts and distinguish between persistent and non-persistent domain shifts.
- We conduct a detector architecture evaluation where we compare different detectors by their inherent domain adaptation abilities.
- We investigate the domain gap induced by varying beam densities including 128-beam sensors on real-world datasets with consideration of other domain shifts.

2. Related Work

Object Detection: In LiDAR-based 3D object detection, architectural choices heavily influence detection performance. Voxel-based methods [43, 54], discretize LiDAR points into 3D grids, allowing for efficient feature extraction through sparse convolutions. Pillar-based methods [9, 18, 19, 33] convert the point cloud into a 2D BEV-image, sacrificing height information for computational efficiency. Operating directly on the raw points, point-based approaches [28, 30, 52] retain spatial details without quantization. Recent transformer-based models [38, 53, 55], provide an alternative to CNN-based models [3, 49, 50], capturing interactions across larger spatial regions. Detection heads in object detection are either anchor-based [21], relying on predefined anchor sizes, or anchor-free [46], which directly predict object centers to generate bounding boxes. Two-staged detectors [34, 36, 36] split the detection into a proposal and refinement stage, often improving accuracy over single-stage detectors but at a higher computational cost.

Concurrent to our work, Eskandar *et al.* [8] empirically test the impact of fundamental architecture choices. However, they chose different detectors to represent each architectural choice. While Eskandar *et al.* choose PointRCNN [34], VoTr [26] and PV-RCNN [35], we select the faster or better performing object detectors IA-SSD [52], DSVT [38] and PV-RCNN++ [36] for the point-based, Transformer-based and two-staged architectures.

Domain Gap Analysis: Recent works have extensively studied how specific domain shifts contribute to the overall domain gap. Wang *et al.* [39] analyzed the impact of geographical variations, concluding that differences in object size distribution can significantly affect detection performance. Another well-studied factor is weather [7, 13]: while LiDAR sensors are less susceptible to adverse weather than cameras, conditions such as snow [17], rain [42], or fog [16] still impair object detection. Concerning sensor-level domain shifts, Hu *et al.* [14] and Fang *et al.* [10] investigate the impact of different LiDAR mounting positions. There are also some recent works investigating the effect of varying beam densities [10, 31]. Richter *et al.* [31] perform a real-world study comparing a 32-beam and 64-beam LiDAR sensors utilizing a specially designed dataset, isolating the beam-density-induced domain gap. However, they do not analyze beam density in conjunction with other domain shifts such as geographic location or object size. Fang *et al.* [10] perform a systematic study regarding beam density on a simulated dataset. However, they did not test the transferability of their findings to real-world datasets.

Domain Adaptation: Domain adaptation methods aim to improve object detection performance across different datasets, addressing the challenges introduced by domain shifts. Broadly, domain adaptation approaches fall into one of three categories: domain alignment, feature alignment, and self-training.

The domain alignment methods SN [39], OT [39] and SAILOR [23] excel at handling object size discrepancies by rescaling ground truth bounding boxes during training or inference. For beam density shifts, methods like DTS [15], PDDA [20] and LiDAR-CS [10] employ resampling methods to align point cloud densities. ReSimAD [48] aligns more complex LiDAR sensor characteristics by reconstructing target scenes and rendering source-like point clouds. Feature alignment methods [22, 40, 41, 51], another approach, perform domain adaptation by alignment in feature space instead of aligning the point clouds directly. In self-training [4, 11, 29, 32, 44, 45], iterative refinement of pseudo-labels is used to gradually adapt the detector to the target domain.

While these domain adaptation methods effectively reduce the occurring domain gaps, they pay little attention to the underlying object detector. We show that a thoughtful selection of the object detector architecture can already close a portion of the domain gap, which reduces the reliance on domain adaptation methods and shifts the focus to more complex domain shifts which cannot be mitigated through architecture alone.

3. Preliminary Analysis

Our preliminary analysis lays the groundwork for this study by addressing three aspects. First, we introduce the non-public datasets involved in this study and detail their unique properties. Second, we establish a domain shift taxonomy, allowing us to systematically assess domain differences. Third, we conduct a detector architecture evaluation to identify models that are inherently robust to domain shifts, providing a foundation for effective domain adaptation.

	Truck	**Rooftop**	**ZOD**
Locations	Germany	Germany	15 European Countries
Ann. frames	40k	7.5k	100k
Sequences	2036	251	43468
Top LiDAR	OS2 (128-beam)	VLP 32c (32-beam)	VLS 128 (128-beam)
Mounting height	3.41m	1.78m	2.01m
Side LiDARs	64-beam	16-beam	16-beam
Front LiDAR	32-beam	-	-
Avg. pts per frame	178.4k	71.5k	254k
Points per beam	2048	1800	3270
Horizontal res.	0.18°	0.2°	0.11°
License	private	private	CC BY-SA

Table 1. Dataset overview.

3.1. Dataset Introduction

In this paper we leverage three datasets (see Tab. 1 and Fig. 2) for training and evaluation. The Rooftop and Truck datasets are private while the remaining Zenseact Open Dataset (ZOD) [1] is open-source. All datasets are specifically designed for autonomous driving applications and feature frame-wise LiDAR data. Concerning the dataset size, the Rooftop dataset is the smallest, with about 7.5k annotated frames, while the Truck and Zenseact Open datasets are substantially larger with 40k and 100k annotated frames. While the Rooftop and Truck datasets were both recorded in Germany, the ZOD contains data from 15 different European countries. Another substantial difference concerns the organization of frames. The Rooftop and Truck datasets are structured in sequences of 20 or 30 frames per sequence, while the ZOD consists of single frames, where, on average, only two frames belong to the same sequence. Regarding the sensor setup, the main LiDAR also differs between each dataset. The Zenseact and Truck datasets employ a dense 128-beam LiDAR, each from a different manufacturer, while the Rooftop dataset employs a sparse 32-beam main LiDAR. We conduct a detailed analysis of the differences between the datasets in the subsequent Sec. 3.2.

Some inherent differences between the datasets can be eliminated by dataset alignment. The size difference can be aligned rather easily by randomly subsampling of the larger datasets to match the size of the smallest dataset. We identified three major dataset alignment measures to address the frame content: coordinate, range and label-space alignment.

Coordinate Alignment: Datasets often differ in coordinate systems, leading to potential mismatches between LiDAR points and object labels. To address this, we align all data points and labels with the commonly used sensor coordinate system with a forward-pointing x-axis and

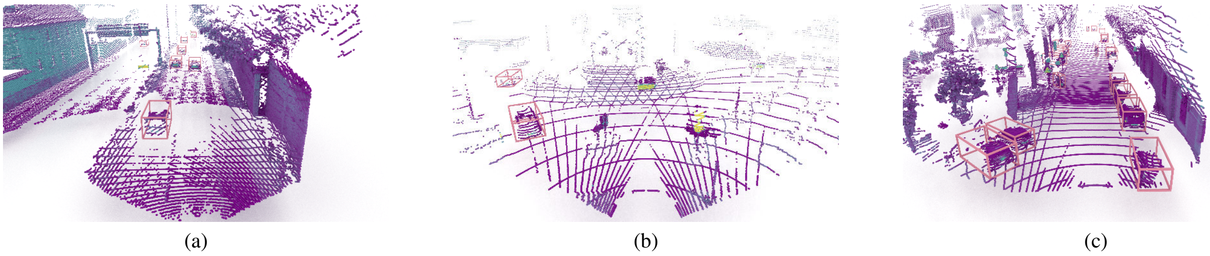

Figure 2. Comparison of the (a) Truck, (b) Rooftop and (c) Zenseact datasets. Differences in LiDAR beam density are clearly visible. Ground-truth objects are marked by red bounding boxes.

	Persistent	Non-persistent
Macro-level	Collection Area Type Geographical Location Frame Selection	Object Size Statistics Weather Conditions
Sensor-level	Sensor Setup Intensity Value Rate of Rotation Alignment Error	Beam Density Horizontal Resolution Field of View
Object-level	Labeling Quality	Label Space Definition Labeling Zone Object Definition

Table 2. Domain shift taxonomy. We differentiate between persistent and non-persistent to highlight domain shifts that can effectively be addressed by either readily available domain adaptation methods or dataset alignment measures.

an upward-pointing z-axis. Since sensors are mounted at different heights, we standardize the origin by aligning it with the ground plane.

Range Alignment: Standardizing the Field of View (FOV) across datasets allows the detector to learn consistent object regions. We define a forward detection range of 123.2 meters to support high-speed safety applications and limit the horizontal FOV to 120° to match the ZOD's labeled region. We mark objects truncated by the cropped FOV as "ignore" during training and evaluation, which prevents the generation of a loss from these objects.

Label-Space Alignment: Inherent label-space differences of the ground-truth annotations between datasets necessitate a mapping to standardize object classes. We categorize objects into four primary classes: *Vehicle*, *Truck*, *Single-track*, and *Pedestrian*. Single-track vehicles are composed of bicycle and motorcycles. Larger vehicles such as vans, trucks and trailers fall under the *Truck* class. To handle varying labeling conventions concerning single-tracked vehicles and their riders, we merge their bounding boxes encompassing both as a single object. A detailed mapping of the label-spaces between the three datasets can be found in the Supplementary.

3.2. Domain Shift Taxonomy

We propose a domain shift taxonomy which allows for a detailed and systematic investigation of possibly occurring domain shifts between aligned datasets. We distinguish between three main categories. Sensor-level domain shifts are directly caused by the mode of collection, while Object-level domain shifts concern the object definition and labeling. The remaining macro-level domain shifts are mainly caused by differences in dataset content. Keeping the final goal of domain adaptation in mind, we additionally differentiate between domain shifts that can effectively be reduced by domain adaptation methods, the *non-persistent domain shifts*, and those that persist despite domain adaptation methods, which we refer to as *persistent domain shifts*.

Concerning the *persistent domain shifts*, we notice a few macro-level differences. While the ZOD features a geographically diverse set of recording locations, the Rooftop and Truck datasets were exclusively recorded in Germany. Also, the types of areas differ between datasets: the ZOD features substantially more *City* frames compared to the remaining two datasets. Finally, we find differences that likely originate from the frame selection process for each dataset. We notice that there is a significantly lower number of overall objects in the Rooftop dataset compared to the Truck and Zenseact datasets. Especially the *Pedestrian* and *Cyclist* classes are significantly underrepresented such that the missing diversity of classes would dominate the domain gap. Thus, we resort to mainly perform dataset-wise comparisons between the *Vehicle* classes.

There are also significant differences on a sensor-level. The Truck dataset has a unique sensor setup as the sensors are mounted considerably higher compared to the other two datasets. The high mounting position has the consequence of a large blind spot right in front of the ego vehicle. The installation of an additional forward-facing LiDAR addresses this issue, resulting in a four-sensor setup.

In terms of object-level differences, we find disparities between the datasets caused by deficient labeling. More specifically, we notice missing ground truth labels for the Rooftop dataset, especially for distant objects that are hit by less LiDAR points. The implications are a noisy supervision signal for training and a distorted evaluation result as predominantly hard-to-detect objects are missing. The ZOD suffers from a similar problem, but hereby, the missing labels are caused by the labeling procedure. ZOD's labeling is based on the camera images. Slight height differences between the camera and LiDAR sensors cause objects to be occluded for the camera while visible for the

Detector	Backbone Architecture		Detection Head	Stages
SECOND [43]	Voxel	CNN	Anchor	Single
PointPillar [18]	Pillar	CNN	Anchor	Single
IA-SSD [52]	Point	CNN	Point	Single
CenterPoint [46]	Voxel	CNN	Center	Single
PVRCNN++ [36]	Point-Voxel	CNN	Center/Point	Two
DSVT [38]	Pillar	Transformer	Center	Single

Table 3. List of 3D object detection methods and their architectural properties.

LiDAR, resulting in missing labels.

We also identify many *non-persistent domain shifts*. In contrast to the previous class of domain shifts, the non-persistent ones can effectively be reduced or even eliminated by domain adaptation methods. Most prominently, the datasets employ LiDAR sensors with a differing number of beams as well as varying beam patterns. Furthermore, the ZOD is more diverse in terms of captured weather conditions as it also features adverse weather conditions such as fog or snow. We also notice differences in terms of object sizes. As the ZOD contains frames recorded in multiple different countries, the intra-dataset object size variability is higher.

Throughout our analysis, we find numerous domain shifts between the datasets. Most of the identified shifts cannot be isolated, making it infeasible to estimate the impact of individual domain shifts on the overall domain gap by a simple comparison between datasets. We provide detailed statistics and domain shift examples in the Supplementary Material.

3.3. Detector Architecture Evaluation

We identify six key differences among commonly used object detection architectures and select one object detector representative of each difference. This approach allows us to assess the impact of each architectural choice. An overview of the selected object detectors is given in Tab. 3. In terms of data representation, we choose SECOND [43] to represent voxel-based architectures, PointPillars [18] for the pillar-based representation, and IA-SSD [52] to represent the class of point-based object detectors. Furthermore, we select CenterPoint [46] to assess the effect of center-based detection heads. To reason about the effectiveness of two-staged methods, we employ the point-voxel-based detector PV-RCNN++ [36]. This detector uses a SECOND-like first-stage to extract bounding box proposals and a point-feature-based second-stage to refine the proposals for the final bounding box estimation. Lastly, we test the impact of different feature extractor architectures. As Transformer-based architectures have recently established themselves in the field of 3D object detection [38], we test their performance in comparison to the well-established sparse-convolution-based architectures.

4. Approach

To evaluate the impact of the beam density on the cross-domain performance, we first select a detector architecture that demonstrates robustness across domains. In our initial experiments, we simply evaluate the trained detectors across domains and group the detection results according to the domain shifts of interest. Results are reported in both high- and low-IOU settings to differentiate between localization and detection errors. For our analysis, we primarily focus on detection errors, which are assessed using low-IOU experiments, as localization errors can usually be mitigated through domain adaptation methods targeting object sizes [23, 39]. This cross-domain comparison operates under the assumption that the datasets are sufficiently similar to enable meaningful conclusions. However, as elaborated in Sec. 3.2, this assumption rarely holds due to persistent domain gaps arising from differences in sensor setups, environmental conditions, and other factors.

In our second set of experiments, we analyze the impact of varying beam densities by isolating it from other domain shifts. Following the approach of [8, 40], we generate beam-wise downsampled versions of one dataset. A detector is trained on each version and subsequently evaluated on the other versions of the same dataset. This approach allows us to focus on the impact of beam density, independent of other domain-specific properties. However, in real-world applications, a varying beam density is usually just one of many domain shifts occurring at test time. In such cases, other kinds of domain shifts may completely dominate the domain gap, rendering the effect of beam density negligible. On the contrary, it could also be the case that due to the cross-domain application, other more reliable features are missing, resulting in an increased domain gap caused by varying beam density. Focusing on a single domain shift in isolation fails to capture these complex interactions. Thus, more sophisticated experiments with the goal of capturing the domain gap by a certain domain shift *in conjunction* with other domain shifts is necessary.

To address the limitations of isolated domain shift analysis, we propose an experimental setup designed to account for interactions between beam density and other domain shifts. As in prior experiments, we utilize sparsified versions of datasets to analyze the impact of beam density, but this time in conjunction with other datasets. Our setup divides the domain gap into two components: the *training domain gap*, caused by differences in beam density during training, and the *inference domain gap*, caused by variations during evaluation, as described by Richter *et al.* [31]. To measure the training domain gap, we downsample the training dataset to create multiple versions, each representing a specific beam density level. By matching or mismatching the beam density with the evaluation dataset, we isolate the effects of beam density variation during training. Similarly, the inference domain gap can be attained by varying the beam density of the evaluation datasets while keeping the training datasets unchanged. These controlled experiments allow us to isolate the specific effects of beam density in a cross-domain setting. We note that in real-world applications, it is typically infeasible to disentangle the training and inference domain gaps, underscoring the relevance of these controlled experiments.

5. Experiments

To demonstrate the effectiveness of our assessment approach, we conduct experiments on the two private datasets Rooftop and Truck and the public Zenseact Open Dataset [1]. We first present our results for the object detector architecture evaluation, based on which we then assess the cross-domain and density-resampling domain gaps. Finally, we compare our training and inference domain gaps to the previously determined domain gaps. Detailed experiment results and additional information about implementation and model training can be found in the Supplementary Material.

5.1. Evaluation Metrics

We use the Intersection-over-Union (IOU)-based metric *3D average precision* $\mathrm{AP}_{S\rightarrow T}$ to assess the detection performance of an object detection model trained on the source domain $\mathbb{D}^S$ when evaluated on the target domain $\mathbb{D}^T$. By lowering the IOU threshold, we can additionally disentangle the *detection error* from the *localization error*. Thereby, localization errors are caused by objects that are detected but not localized accurately enough to be considered true positives, whereas detection errors represent entirely missed or wrongly classified objects. Wang *et al.* [39] demonstrated that the 3D average precision significantly increases when the IOU threshold is reduced from the commonly used threshold of 0.7 (70%) to approximately 0.4 (40%). At this threshold, the domain gap primarily reflects detection errors, which are of greater practical significance than localization errors. Localization errors can often be mitigated using domain adaptation methods such as ROS [44], SN, or OT [39]. Therefore, in our evaluation, we primarily focus on a reduced IOU threshold of 0.4 to better understand detection errors.

Detector	mAP ↑ IOU=0.7	mAP ↑ IOU=0.4	DG in % ↓ IOU=0.7	DG in % ↓ IOU=0.4
SECOND	30.6	70.4	47.6	**16.0**
PointPillars	23.0	63.9	56.4	21.5
IA-SSD	34.7	66.1	**41.5**	18.3
CenterPoint	28.6	68.1	49.7	18.0
PV-RCNN++	**37.2**	**71.2**	42.8	16.2
DSVT	33.4	68.4	47.7	20.1

Table 4. Detector comparison results overview. We calculate the cross-domain performance by averaging over all cross-domain results. We report the average domain gap and the cross-domain performance using the AP metric at the IOU thresholds of 0.7 and 0.4 for the *Vehicle* class.

While the cross-domain performance is well suited for comparing different detectors, it does not adequately capture the generalization ability of a certain detector across domains, as it is influenced by the inherent difficulty of the target dataset. To address this limitation, we employ the *domain gap* metric [44], which relates cross-domain performance to the detector's maximum achievable performance on the target domain ($\mathrm{AP}_{T\rightarrow T}$). This relative metric provides a detached view of domain generalization ability and allows for meaningful comparisons of detectors evaluated on target datasets with varying difficulty levels. The domain gap DG, expressed as a percentage of the maximum achievable performance, is defined as

$$DG = \frac{AP_{T\rightarrow T} - AP_{S\rightarrow T}}{AP_{T\rightarrow T}} \cdot 100 \tag{1}$$

5.2. Detector Architecture Evaluation

With the goal of finding a detector that is robust against domain changes, we examine the impact of each architectural choice on the overall performance in the cross-domain setting (see Tab. 4). The first architectural comparison concerns the voxel and pillar discretization methods. We find that the voxel-based detector SECOND [43] outperforms the pillar-based PointPillars [18] by a substantial margin. While PointPillars is the worst-performing detector across all metrics, SECOND exhibits surprisingly good performance in the low-IOU settings. This indicates that SECOND is good at detecting objects but fails to precisely locate them in 3D space. This discrepancy stems from the quantization process. During voxelization, the exact geometric structure is lost, hampering the precise localization of objects. In terms of different data representations, we also test the point-based detector IA-SSD [52]. This detector shows a very strong performance in the high-IOU setting, indicating that it is also good at predicting the 3D location of objects. This can be attributed to the detector's direct access to the point data.

Subsequently, we test the effects of different detection heads. More specifically, we compare anchor-heads, as employed in SECOND or PointPillars, with center-heads, as introduced in CenterPoint [46]. Contrary to expectations, center-heads result in degraded performance compared to anchor-heads.

Next, we apply PV-RCNN++ [36] to test the impact of an additional second stage. This detector outperforms all others in terms of cross-domain performance while staying competitive in terms of domain gap. Similar to IA-SSD, the second stage of PV-RCNN++ benefits from direct access to raw point data, which likely enhances its performance. We can conclude that, for our experiments, the addition of a second stage significantly benefits the object detectors regarding generalization abilities.

Lastly, we examine the effect of Transformer-based backbones. While DSVT [38] achieves excellent in-domain results, its cross-domain performance and domain

	Source→Target	avg. DG in % IOU=0.7 ↓	avg. DG in % IOU=0.4 ↓
Beam Density	Dense→Dense	23.9	14.4
	Dense→Sparse	36.4	11.1
	Sparse→Dense	67.4	23.0

Table 5. Domain gap in percent for the object detector PV-RCNN++. The different cross-domain settings are grouped and averaged by beam density. We report the average domain gap calculated with the AP metric at the IOU thresholds of 0.7 and 0.4 for the *Vehicle* class.

gap metrics are only moderate. Our experiments suggest that Transformer-based backbones do not benefit object detectors in terms of domain generalization.

These findings highlight the critical role of detector architecture in achieving robust domain generalization. Comparing the best and worst-performing detectors, we observe a performance difference of 61.7% in the high-IOU setting and 11.4% in the low-IOU settings. We further find that purely voxel-based detectors excel at detecting objects and the addition of point information drastically improves the localization error. The object detection architecture evaluation conducted by Eskandar *et al.* [8] yields similar conclusion concerning the effect of point information. However, our experiments do not support their finding that Transformer-based backbones improve cross-domain generalization.

5.3. Cross-domain Results

Beginning our examination of the domain gap induced by beam density, we conduct a simple cross-domain evaluation. As shown in Tab. 5, the sparsely-trained detector (trained on Rooftop) applied on denser datasets (Sparse→Dense) exhibits approximately twice the domain gap compared to applying a densely-trained detector (trained on Truck or ZOD) on a sparse dataset (Dense→Sparse). This trend persists when isolating the detection error by evaluating with reduced IOU threshold. In terms of domain generalization, we could conclude from this initial analysis that for these particular datasets, it is beneficial to train a detector on the dense datasets as they generalize towards sparse and dense datasets. As the effect of beam density is just one of many factors contributing to this observed domain gap, further analysis is required to make stronger statements.

5.4. Density-resampling Results

We continue by isolating the beam-density-induced domain gap in our second set of experiments. To keep track of the resampled density, we call the original dense dataset ZOD$_{128}$ and the sparser variants ZOD$_{64}$ and ZOD$_{32}$, where the index represents the number of beams. As shown in Tab. 6, the density-resampling analysis contrasts with the findings from the cross-domain analysis. In the high-IOU setting, the sparse-to-dense cases (top-right of the results matrix) give better results than their dense-to-sparse counterparts (bottom-left of the results matrix). More broadly, we notice that the performance differences between all datasets are comparably small. This indicates that the detectors generalize very well towards the same dataset when solely varying the sampling. However, in real-world applications, variations in sampling is usually accompanied by other kinds of domain shift. In the following experiments we thus investigate the beam-density-induced domain shift in the presence of other kinds of domain shifts by measuring the training and inference domain gaps.

	Target		
	ZOD$_{32}$	**ZOD**$_{64}$	**ZOD**$_{128}$
Source	DG in % ↓ IOU=0.7	DG in % ↓ IOU=0.7	DG in % ↓ IOU=0.7
ZOD$_{32}$	-	1.8	5.1
ZOD$_{64}$	4.3	-	1.9
ZOD$_{128}$	9.4	0.9	-
	DG in % ↓ IOU=0.4	DG in % ↓ IOU=0.4	DG in % ↓ IOU=0.4
ZOD$_{32}$	-	0.6	2.3
ZOD$_{64}$	2.7	-	1.6
ZOD$_{128}$	3.0	-0.9	-

Table 6. Density-caused domain gap for the density-resampling setting. We report the domain gap calculated with the AP metric at the IOU thresholds of 0.7 (top) and 0.4 (bottom) for the *Vehicle* class.

5.5. Training and Inference Domain Gap Results

We first examine the training domain gap in Tab. 8. Compared to the density-resampling setting (recall Tab. 6), the overall domain gap level is significantly higher, as many more factors contribute besides the beam density. In the high-IOU experiments, trends are consistent with the density-resampling case: larger differences in beam density lead to larger domain gaps. However, when isolating the detection error in the low-IOU setting, a different trend emerges. Densely-trained detectors show overall better performance for the dense *and* sparse target datasets, indicating that they are able to detect more objects than their sparsely-trained counterparts in the cross-domain case. This setup also allows us to quantify the impact of beam density on the overall domain gap. When evaluating on the Rooftop dataset, training on a denser 128-beam dataset *reduces* the domain gap from 13.1% to 9.2%, reducing the domain shift by almost one-third (shown in Fig. 1c).

Next, we analyze the inference domain gap in Tab. 7. The high-IOU results (top-left of the table) show greater domain gap variability between the Rooftop and Truck datasets than within each dataset across beam densities. This observation supports our earlier assumption about persistent dataset-caused domain gaps in cross-domain evaluation settings (recall Sec. 4). Overall, the results in-

	Target					
	ZOD$_{32}$	**ZOD**$_{64}$	**ZOD**$_{128}$	**ZOD**$_{32}$	**ZOD**$_{64}$	**ZOD**$_{128}$
Source	DG in % ↓	DG in % ↓	DG in % ↓	DG in % ↓	DG in % ↓	DG in % ↓
	IOU=0.7	IOU=0.7	IOU=0.7	IOU=0.4	IOU=0.4	IOU=0.4
Rooftop$_{32}$	55.0	58.0	67.4	13.9	11.8	13.4
Truck$_{128}$	28.6	19.0	16.3	16.9	8.5	8.2
	AP ↑	AP ↑	AP ↑	AP ↑	AP ↑	AP ↑
	IOU=0.7	IOU=0.7	IOU=0.7	IOU=0.4	IOU=0.4	IOU=0.4
Rooftop$_{32}$	25.1	26.7	22.4	60.5	69.0	73.0
Truck$_{128}$	39.7	51.4	57.6	58.4	71.6	77.4

Table 7. Inference domain gap caused by varying beam densities in a cross-domain setting. We report the domain gap (top) and the cross-domain performance (bottom) using the AP metric at the IOU thresholds of 0.7 (left) and 0.4 (right) for the *Vehicle* class.

	Target			
	Rooftop$_{32}$	**Truck**$_{128}$	**Rooftop**$_{32}$	**Truck**$_{128}$
Source	DG in % ↓	DG in % ↓	DG in % ↓	DG in % ↓
	IOU=0.7	IOU=0.7	IOU=0.4	IOU=0.4
ZOD$_{32}$	38.4	41.5	13.1	28.0
ZOD$_{64}$	41.2	37.2	11.1	23.7
ZOD$_{128}$	45.8	32.5	9.2	20.5

Table 8. Training domain gap caused by varying beam densities in a cross-domain setting. We report the domain gap calculated with the AP metric at the IOU thresholds of 0.7 (left) and 0.4 (right) for the *Vehicle* class.

dicate that the detector is relatively robust to beam density, provided the number of beams does not change drastically. In the high-IOU experiments, the domain gap increases by only 3% or less when doubling or halving beam density.

The previous analyses were exclusively done through the lens of the *domain gap* metric. For the inference domain gap, it is also interesting to examine the performance values themselves. Especially for the low-IOU setting, we can see in Tab. 7 that the performance (measured in AP) increases steadily with an increasing number of beams, despite the domain gap staying similar. This indicates that the observed performance gain is caused by easing the detection problem in contrast to better generalizability of the detectors. As the density increases, more LiDAR rays hit objects which makes it easier for the object to be detected.

In summary, the results provide a comprehensive view of the domain gap caused by varying beam densities. When isolating the effect of varying beam densities (see Tab. 6), the domain gap appears minor, favoring sparsely-trained detectors for domain generalization. This aligns with findings from related studies [8, 10, 31]. However, when analyzing the training domain gap in conjunction with other domain shifts, we find that densely-trained detectors exhibit better domain generalization in terms of detecting objects (see Tab. 8). Regarding inference domain gaps (see Tab. 7), results show that detectors generalize well as long as beam density changes are modest. Nonetheless, denser sampling reduces detection difficulty, leading to better performance irrespective of the detector's generalizability.

6. Conclusion

This study presented an investigation of the impact of beam density on LiDAR object detection performance in cross-domain scenarios during which we also explored optimal object detector architectures to address domain variability effectively. Our object detector architecture evaluation revealed that combining voxel- and point-based approaches delivers superior cross-domain performance by leveraging the complementary strengths of these representations. While Transformer-based backbones demonstrated strong performance in in-domain tasks, their cross-domain benefits were limited under the conditions tested. Our findings emphasize the importance of selecting a robust detector architecture as a prior step to domain adaptation.

We further investigated the impact of beam density on LiDAR object detection performance in cross-domain scenarios, offering insights into both training and inference domain gaps. We found that detectors trained on dense datasets generalize better across domains, particularly for detecting objects, where detection error (rather than localization error) is the primary concern. During inference, detectors showed robustness against moderate beam density changes, with denser configurations improving performance by reducing the difficulty of the detection task rather than enhancing generalizability.

A key insight from this study is that domain gaps, including those caused by beam density, should not be analyzed in isolation. Instead, we advocate for a holistic approach to domain adaptation, beginning with the selection of a detector intrinsically robust to domain changes. This minimizes the initial domain gap and allows adaptation efforts to focus on more complex types of domain shifts.

7. Acknowledgements

The financial support by the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development and the Christian Doppler Research Association is gratefully acknowledged.

References

[1] Mina Alibeigi, William Ljungbergh, Adam Tonderski, Georg Hess, Adam Lilja, Carl Lindstrom, Daria Motorniuk, Junsheng Fu, Jenny Widahl, and Christoffer Petersson. Zenseact Open Dataset: A large-scale and diverse multimodal dataset for autonomous driving. In *Proc. CVPR*, 2023. 3, 6

[2] 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 *Proc. CVPR*, 2020. 1

[3] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. LargeKernel3D: Scaling up Kernels in 3D Sparse CNNs. In *Proc. CVPR*, 2023. 2

[4] Zhuoxiao Chen, Yadan Luo, Zheng Wang, Mahsa Baktashmotlagh, and Zi Huang. Revisiting Domain-Adaptive 3D Object Detection by Reliable, Diverse and Class-balanced Pseudo-Labeling. In *Proc. ICCV*, 2023. 3

[5] Carlos A. Diaz-Ruiz, Youya Xia, Yurong You, Jose Nino, Junan Chen, Josephine Monica, Xiangyu Chen, Katie Luo, Yan Wang, Marc Emond, Wei-Lun Chao, Bharath Hariharan, Kilian Q. Weinberger, and Mark Campbell. Ithaca365: Dataset and driving perception under repeated and challenging weather conditions. In *Proc. CVPR*, 2022. 1

[6] 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 autonomous driving. In *Proc. CVPR*, 2023. 2

[7] Mariella Dreissig, Dominik Scheuble, Florian Piewak, and Joschka Boedecker. Survey on LiDAR Perception in Adverse Weather Conditions. In *IEEE Intelligent Vehicles Symposium*, 2023. 3

[8] George Eskandar, Chongzhe Zhang, Abhishek Kaushik, Karim Guirguis, Mohamed Sayed, and Bin Yang. An Empirical Study of the Generalization Ability of Lidar 3D Object Detectors to Unseen Domains. In *Proc. CVPR*, 2024. 2, 5, 7, 8

[9] Lue Fan, Ziqi Pang, Tianyuan Zhang, Yu-Xiong Wang, Hang Zhao, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Embracing Single Stride 3D Object Detector with Sparse Transformer. In *Proc. CVPR*, 2022. 2

[10] Jin Fang, Dingfu Zhou, Jingjing Zhao, Chenming Wu, Chulin Tang, Cheng-Zhong Xu, and Liangjun Zhang. LiDAR-CS Dataset: LiDAR Point Cloud Dataset with Cross-Sensors for 3D Object Detection. In *Proc. ICRA*, 2024. 2, 3, 8

[11] Christian Fruhwirth-Reisinger, Michael Opitz, Horst Possegger, and Horst Bischof. FAST3D: Flow-Aware Self-Training for 3D Object Detectors. In *Proc. BMVC*, 2021. 3

[12] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proc. CVPR*, 2012. 1

[13] 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)*, 2023. 3

[14] Hanjiang Hu, Zuxin Liu, Sharad Chitlangia, Akhil Agnihotri, and Ding Zhao. Investigating the Impact of Multi-LiDAR Placement on Object Detection for Autonomous Driving. In *Proc. CVPR*, 2022. 3

[15] Qianjiang Hu, Daizong Liu, and Wei Hu. Density-Insensitive Unsupervised Domain Adaption on 3D Object Detection. In *Proc. CVPR*, 2023. 2, 3

[16] Jiyoon Kim, Bum-jin Park, and Jisoo Kim. Empirical Analysis of Autonomous Vehicle's LiDAR Detection Performance Degradation for Actual Road Driving in Rain and Fog. *Sensors*, 2023. 3

[17] Akhil Kurup and Jeremy Bos. Dsor: A scalable statistical filter for removing falling snow from lidar point clouds in severe winter weather, 2021. 3

[18] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast Encoders for Object Detection from Point Clouds. In *Proc. CVPR*, 2019. 2, 5, 6

[19] Jinyu Li, Chenxu Luo, and Xiaodong Yang. PillarNeXt: Rethinking Network Designs for 3D Object Detection in LiDAR Point Clouds. In *Proc. CVPR*, 2023. 2

[20] Shuangzhi Li, Lei Ma, and Xingyu Li. Domain generalization of 3d object detection by density-resampling. In *Proc. ECCV*, 2025. 2, 3

[21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single Shot MultiBox Detector. In *Proc. ECCV*, 2016. 2

[22] 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 *Proc. ICCV*, 2021. 3

[23] Dusan Malic, Christian Fruhwirth-Reisinger, Horst Possegger, and Horst Bischof. SAILOR: Scaling Anchors via Insights into Latent Object Representation. In *Proc. WACV*, 2023. 2, 3, 5

[24] Sivabalan Manivasagam, Ioan Andrei Bârsan, Jingkang Wang, Ze Yang, and Raquel Urtasun. Towards zero domain gap: A comprehensive study of realistic LiDAR simulation for autonomy testing. In *Proc. ICCV*, 2023. 2

[25] Jiageng Mao, Minzhe Niu, Chenhan Jiang, Hanxue Liang, Jingheng Chen, Xiaodan Liang, Yamin Li, Chaoqiang Ye, Wei Zhang, Zhenguo Li, Jie Yu, Hang Xu, and Chunjing Xu. One million scenes for autonomous driving: ONCE dataset. In *Proc. NeurIPS*, 2021. 1

[26] Jiageng Mao, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiashi Feng, Xiaodan Liang, Hang Xu, and Chunjing Xu. Voxel Transformer for 3D Object Detection. In *Proc. ICCV*, 2021. 2

[27] Jiageng Mao, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. 3D Object Detection for Autonomous Driving: A Review and New Outlooks. In *IJCV*, 2022. 1

[28] Xuran Pan, Zhuofan Xia, Shiji Song, Li Erran Li, and Gao Huang. 3D Object Detection with Pointformer. In *Proc. CVPR*, 2021. 2

[29] Xidong Peng, Xinge Zhu, and Yuexin Ma. CL3D: Unsupervised Domain Adaptation for Cross-LiDAR 3D Detection. In *Proc. AAAI*, 2023. 3

[30] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proc. CVPR*, 2017. 2

[31] Jasmine Richter, F. Faion, Di Feng, Paul Benedikt Becker, Piotr Sielecki, and Claudius Glaeser. Understanding the Domain Gap in LiDAR Object Detection Networks. In *Uni-DAS e.V. Workshop Fahrerassistenz und automatisiertes Fahren*, 2022. 2, 3, 5, 8

[32] Cristiano Saltori, Stephane Lathuiliere, Nicu Sebe, Elisa Ricci, and Fabio Galasso. SF-UDA ^{3d} : Source-Free Unsupervised Domain Adaptation for LiDAR-Based 3D Object Detection. In *IEEE 3DV*, 2020. 3

[33] Guangsheng Shi, Ruifeng Li, and Chao Ma. PillarNet: Real-Time and High-Performance Pillar-based 3D Object Detection. In *Proc. ECCV*, 2022. 2

[34] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. In *Proc. CVPR*, 2019. 2

[35] 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 *Proc. CVPR*, 2020. 2

[36] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN++: Point-Voxel Feature Set Abstraction With Local Vector Representation for 3D Object Detection. In *IJCV*, 2022. 2, 5, 6

[37] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Sheng Zhao, Shuyang Cheng, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proc. CVPR*, 2020. 1

[38] Haiyang Wang, Chen Shi, Shaoshuai Shi, Meng Lei, Sen Wang, Di He, Bernt Schiele, and Liwei Wang. DSVT: Dynamic Sparse Voxel Transformer with Rotated Sets. In *Proc. CVPR*, 2023. 2, 5, 6

[39] Yan Wang, Xiangyu Chen, Yurong You, Li Erran, Bharath Hariharan, Mark Campbell, Kilian Q. Weinberger, and Wei-Lun Chao. Train in Germany, Test in The USA: Making 3D Object Detectors Generalize. In *Proc. CVPR*, 2020. 2, 3, 5, 6

[40] 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 *Proc. ECCV*, 2022. 2, 3, 5

[41] Maciej K Wozniak, Mattias Hansson, Marko Thiel, and Patric Jensfelt. Uada3d: Unsupervised adversarial domain adaptation for 3d object detection with sparse lidar and large domain gaps. *IEEE RA-L*, 2024. 3

[42] 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 *Proc. ICCV*, 2021. 3

[43] Yan Yan, Yuxing Mao, and Bo Li. SECOND: Sparsely Embedded Convolutional Detection. *Sensors*, 18(10), 2018. 2, 5, 6

[44] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. ST3D: Self-training for Unsupervised Domain Adaptation on 3D Object Detection. In *Proc. CVPR*, 2021. 2, 3, 6

[45] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. ST3D++: Denoised Self-Training for Unsupervised Domain Adaptation on 3D Object Detection. In *IEEE TPAMI*, 2022. 3

[46] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Centerpoint: Center-based 3D Object Detection and Tracking. In *Proc. CVPR*, 2021. 2, 5, 6

[47] 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 *Proc. CVPRW*, 2023. 2

[48] Bo Zhang, Xinyu Cai, Jiakang Yuan, Donglin Yang, Jianfei Guo, Renqiu Xia, Botian Shi, Min Dou, Tao Chen, Si Liu, Junchi Yan, and Yu Qiao. ReSimAD: Zero-Shot 3D Domain Transfer for Autonomous Driving with Source Reconstruction and Target Simulation. In *Proc. ICLR*, 2024. 3

[49] Gang Zhang, Junnan Chen, Guohuan Gao, Jianmin Li, and Xiaolin Hu. HEDNet: A Hierarchical Encoder-Decoder Network for 3D Object Detection in Point Clouds. In *Proc. NeurIPS*, 2023. 2

[50] Gang Zhang, Junnan Chen, Guohuan Gao, Jianmin Li, Si Liu, and Xiaolin Hu. SAFDNet: A Simple and Effective Network for Fully Sparse 3D Object Detection. In *Proc. CVPR*, 2024. 2

[51] Weichen Zhang, Wen Li, and Dong Xu. SRDAN: Scale-aware and Range-aware Domain Adaptation Network for Cross-dataset 3D Object Detection. In *Proc. CVPR*, 2021. 3

[52] Yifan Zhang, Qingyong Hu, Guoquan Xu, Yanxin Ma, Jianwei Wan, and Yulan Guo. Not All Points Are Equal: Learning Highly Efficient Point-based Detectors for 3D LiDAR Point Clouds. In *Proc. CVPR*, 2022. 2, 5, 6

[53] Chao Zhou, Yanan Zhang, Jiaxin Chen, and Di Huang. OcTr: Octree-based Transformer for 3D Object Detection. In *Proc. CVPR*, 2023. 2

[54] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In *Proc. CVPR*, 2018. 2

[55] Zixiang Zhou, Dongqiangzi Ye, Weijia Chen, Yufei Xie, Yu Wang, Panqu Wang, and Hassan Foroosh. Lidarformer: A unified transformer-based multi-task network for lidar perception. In *Proc. ICRA*, 2024. 2

DOI: 10.3217/978-3-99161-022-9-002

An Investigation of Beam Density on LiDAR Object Detection Performance

Supplementary Material

1. Dataset Introduction

In the following, we provide the detailed label map (see Tab. 1) from Chap. 3.1 and elaborate some of the domain shifts mentioned in Chap. 3.2 in more detail. We begin by showcasing the effect of geographically diverse locations in Fig. 1. The observed country-level size bias combined with a dataset-specific size-bias results in different average object sizes between the datasets (see Fig. 2). Fig. 3 shows differences in recording locations. We can see that the ZOD contains significantly more *City* frames compared to the other two datasets. As a consequence, a detector trained on the ZOD is more likely to assign objects that typically associated with *City* frames, such as *Cyclists*, to ambiguous objects than detectors trained on the other datasets. An example of this phenomenon is depicted in Fig. 4, where the detector trained on the ZOD detects a *Cyclist*, while the other detectors correctly detect a *Truck*. The bias introduced by the frame selection procedure can be seen in Fig. 5. The Rooftop dataset contains, on average, less objects per frame than the other two datasets. This difference is especially severe for the classes *Pedestrian* and *Cyclist*. Finally, we give an example for imperfect labeling of the Rooftop dataset in Fig. 6. The camera image shows a black car, which is captured by 8 points in the LiDAR image. However, no bounding box is assigned in the LiDAR frame.

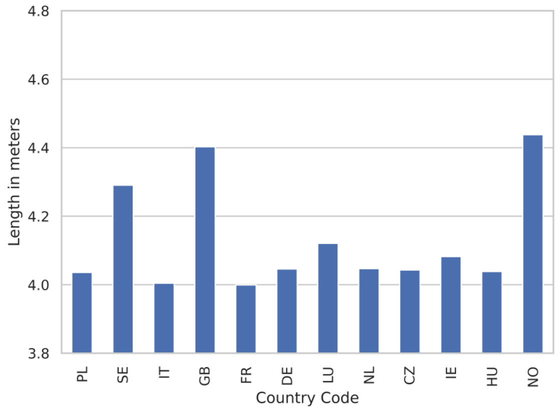

Figure 1. Average length of vehicles for different countries in the ZOD.

2. Experiments

2.1. Implementation Details

In the following, we summarize some implementation details which are shared across the object detection models.

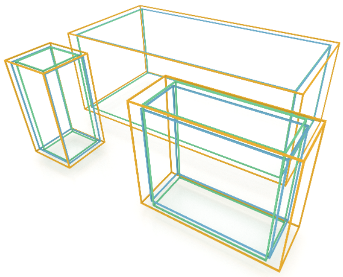

Figure 2. Comparison of average object sizes for the classes *Car*, *Pedestrian* and *Cyclist* for ZOD (green), Truck dataset (blue) the Rooftop dataset (orange). Object sizes of the Rooftop dataset are significantly larger on average.

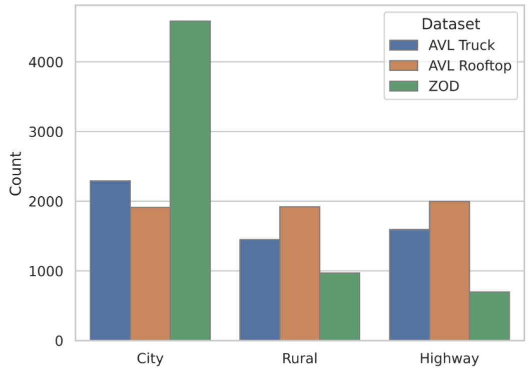

Figure 3. Recording area statistics.

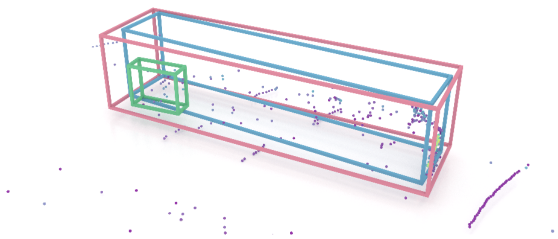

Figure 4. Example of a misclassification of an ambiguous object on a highway. The detector trained on the ZOD (green bounding box) is more likely to assign the class *Cyclist* to the ambiguous object compared to the detector trained on the Rooftop dataset (blue bounding box), which correctly identifies the object as *Truck* (red bounding box).

Codebase. All models were implemented in the codebase 3DTrans[1], which is an extension of the open-source

[1] https://github.com/PJLab-ADG/3DTrans

Detector Label-Space	Dataset Label-Space		
	Truck	Rooftop	Zenseact Open Dataset
Vehicle	Vehicle_Drivable_Car	Vehicle_Drivable_Car	Vehicle_Car
	Vehicle_Drivable_Van	Vehicle_Drivable_Van	Vehicle_Van
Truck	LargeVehicle_Bus	LargeVehicle_Bus	Vehicle_Bus
	LargeVehicle_TruckCab	LargeVehicle_TruckCab	
	Trailer	Trailer	Vehicle_Trailer
			Vehicle_Truck
	LargeVehicle_Truck	LargeVehicle_Truck	Vehicle_TramTrain
			Vehicle_HeavyEquip
Cyclist	Vehicle_Ridable_Motorcycle	Vehicle_Ridable_Motorcycle	VulnerableVehicle_Motorcycle
	Vehicle_Ridable_Bicycle	Vehicle_Ridable_Bicycle	VulnerableVehicle_Bicycle
Pedestrian	Human	Human	Pedestrian
DontCare	Dont_Care	PPObject	
	Other	PPObject_Stroller	
		PPObject_BikeTrailer	
		Vehicle_PMD	

Table 1. Label-space mapping between the detector label-space and the dataset label-spaces.

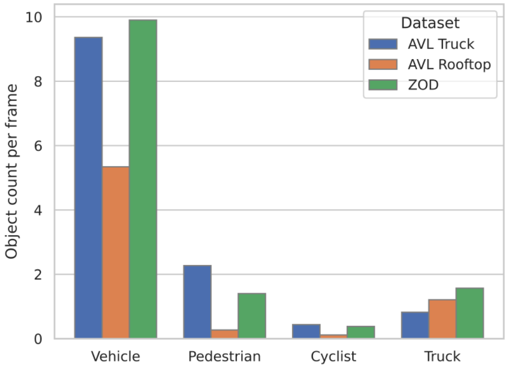

Figure 5. Class statistics.

3D object detecton codebase OpenPCDet [5]. Models developed in OpenPCDet can seamlessly be integrated into 3DTrans. All the models were already implemented in 3DTrans for the Waymo Open Dataset [4], with the exception of DSVT, which had to be adopted from the official OpenPCDet codebase. The ZOD dataloader has been implemented based the provided developement kit[2]. The dataloaders for the Rooftop and Truck datasets were implemented from scratch. Our implementations are based on PyTorch 2.1 and SpConv [1] version 2.3.6 for CUDA 12.0.

Hardware. We conducted the development and testing of the models on a workstation featuring a single RTX 4090 GPU. We trained the final models on a GPU with four RTX A6000 GPUs.

Schedule and Optimization. We train all object detectors on each dataset for 100 epochs. All the models employ the ADAM optimizer [2] and use a OneCycle learning-rate scheduler [3] with varying learning rate, momentum and weight-decay parameters depending on the model.

Data Representation. The voxel-based methods SECOND, CenterPoint, PV-RCNN++, and DSVT require a discretization of the point cloud into a voxel-representation before the object detection models can be applied. To this end, we adapt a voxel size of (0.1m, 0.1m, 0.15m) following the implementation of PV-RCNN++. For the pillar-based method PointPillars, we use a pillar-size of (0.32m, 0.32m, 6.0m).

2.2. Detector Architecture Search

In Tab. 2, we provide the raw data used to calculate the averaged results for the domain gap and performance, which we base our detector architecture selection on. These results are also used to conduct the initial cross-domain experiment in Sec. 5.3.

2.3. Domain Gap Results

In Tab. 3 and Tab. 4 we provide the performance values in Average Precision based on which the domain gaps in Sec. 5.4 and 5.5 are calculated.

[2] https://github.com/zenseact/zod

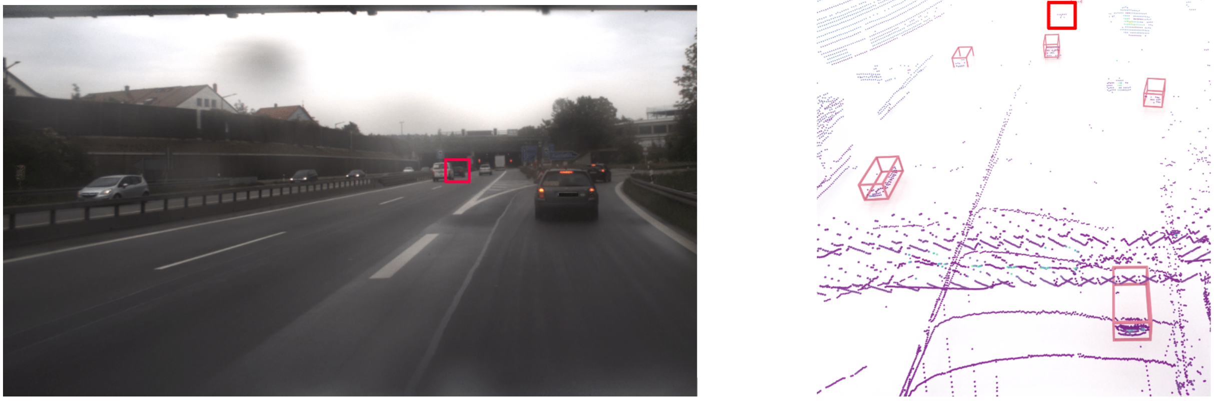

Figure 6. Example of missing ground truth label for the Rooftop dataset. The 3D bounding box of the red boxed car is missing even though it is clearly visible in the image and LiDAR data.

		Target		
		Truck	**Rooftop**	**Zenseact Open Dataset**
Source	Detector	AP ↑ IOU 0.7/0.4	AP ↑ IOU 0.7/0.4	AP ↑ IOU 0.7/0.4
Truck	SECOND	55.5/84.9	39.2/73.6	46.9/74.7
	PointPillars	49.7/82.4	32.7/67.3	33.5/67.7
	IA-SSD	58.6/82.4	41.3/69.5	53.9/72.1
	CenterPoint	54.5/82.4	36.9/75.7	43.6/74.2
	PV-RCNN++	65.5/86.4	45.5/74.0	57.6/77.4
	DSVT	60.4/86.3	41.0/72.1	53.2/77.3
Rooftop	SECOND	16.7/60.5	58.3/84.6	16.9/66.9
	PointPillars	13.6/56.3	51.8/82.3	13.6/65.1
	IA-SSD	20.6/59.9	55.1/82.2	29.9/71.7
	CenterPoint	13.9/54.4	58.3/84.0	11.7/58.9
	PV-RCNN++	21.0/58.3	61.4/84.3	22.4/73.0
	DSVT	20.9/55.2	64.6/86.4	19.9/62.6
Zenseact Open Dataset	SECOND	36.7/69.7	27.0/76.8	61.2/82.0
	PointPillars	25.3/59.5	19.1/67.6	56.7/79.8
	IA-SSD	33.7/54.0	28.7/69.3	63.8/78.6
	CenterPoint	38.7/69.8	26.9/75.5	57.7/82.7
	PV-RCNN++	44.2/68.6	32.7/75.8	68.7/84.3
	DSVT	38.1/66.2	27.6/77.0	66.4/84.3

Table 2. Detector comparison in terms of the cross-domain performance. We report the performance using the AP metric at an IOU threshold of 0.7/0.4 for the *Vehicle* class.

	Target		
	ZOD$_{32}$	**ZOD$_{64}$**	**ZOD$_{128}$**
Source	AP ↑ IOU=0.7	AP ↑ IOU=0.7	AP ↑ IOU=0.7
ZOD$_{32}$	55.7	62.3	65.3
ZOD$_{64}$	53.3	63.5	67.4
ZOD$_{128}$	50.4	62.9	68.7
	AP ↑ IOU=0.4	AP ↑ IOU=0.4	AP ↑ IOU=0.4
ZOD$_{32}$	70.3	77.8	82.3
ZOD$_{64}$	68.4	78.3	82.9
ZOD$_{128}$	68.2	79.0	84.3

Table 3. Density-caused domain gap for the density-resampling setting. We report the performance with the AP metric at the IOU thresholds of 0.7 (top) and 0.4 (bottom) for the *Vehicle* class.

	Target			
	Rooftop$_{32}$	**Truck$_{128}$**	**Rooftop$_{32}$**	**Truck$_{128}$**
Source	AP ↑ IOU=0.7	AP ↑ IOU=0.7	AP ↑ IOU=0.4	AP ↑ IOU=0.4
ZOD$_{32}$	37.9	38.3	73.3	62.2
ZOD$_{64}$	36.1	41.1	74.9	65.9
ZOD$_{128}$	33.3	44.2	76.6	68.7

Table 4. Training domain gap caused by varying beam densities in a cross-domain setting. We report cross-domain performance with the AP metric at the IOU thresholds of 0.7 (left) and 0.4 (right) for the *Vehicle* class.

References

[1] Spconv Contributors. Spconv: Spatially Sparse Convolution Library, 2022. 2

[2] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proc. ICLR*, 2015. 2

[3] Leslie N. Smith and Nicholay Topin. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates, 2018. 2

[4] Pei Sun, Henrik Kretzschmar, Shuyang Dotiwalla, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proc. CVPR*, 2020. 2

[5] OpenPCDet Development Team. OpenPCDet: An Open-source Toolbox for 3D Object Detection from Point Clouds, 2020. 2