Rethinking LiDAR Domain Generalization: Single Source as Multiple Density Domains

Jaeyeul Kim, Jungwan Woo, Jeonghoon Kim, and Sunghoon Im(✉)

DGIST, Daegu, South Korea
{jykim94, friendship1, jeonghoon, sunghoonim}@dgist.ac.kr

Abstract. In the realm of LiDAR-based perception, significant strides have been made, yet domain generalization remains a substantial challenge. The performance often deteriorates when models are applied to unfamiliar datasets with different LiDAR sensors or deployed in new environments, primarily due to variations in point cloud density distributions. To tackle this challenge, we propose a Density Discriminative Feature Embedding (DDFE) module, capitalizing on the observation that a single source LiDAR point cloud encompasses a spectrum of densities. The DDFE module is meticulously designed to extract density-specific features within a single source domain, facilitating the recognition of objects sharing similar density characteristics across different LiDAR sensors. In addition, we introduce a simple yet effective density augmentation technique aimed at expanding the spectrum of density in source data, thereby enhancing the capabilities of the DDFE. Our DDFE stands out as a versatile and lightweight domain generalization module. It can be seamlessly integrated into various 3D backbone networks, where it has demonstrated superior performance over current state-of-the-art domain generalization methods. Code is available at https://github.com/dgist-cvlab/MultiDensityDG.

Keywords: LiDAR Semantic Segmentation · Domain Generalization

1 Introduction

Light Detection and Ranging (LiDAR) provides detailed 3D data, making it indispensable for environmental perception in autonomous vehicles. Among the various LiDAR-based perception tasks, semantic segmentation plays a crucial role in understanding the driving scene by classifying each point into multiple classes. While LiDAR-based semantic segmentation [1, 13, 48] have been widely studied, their impressive performance is often constrained to scenarios where source and target datasets align perfectly. However, mismatches between these datasets can lead to significant performance declines. These are mainly attributed to two primary factors: environmental variations as exemplified by the

J. Kim and J. Woo—Both authors contributed equally to this work.

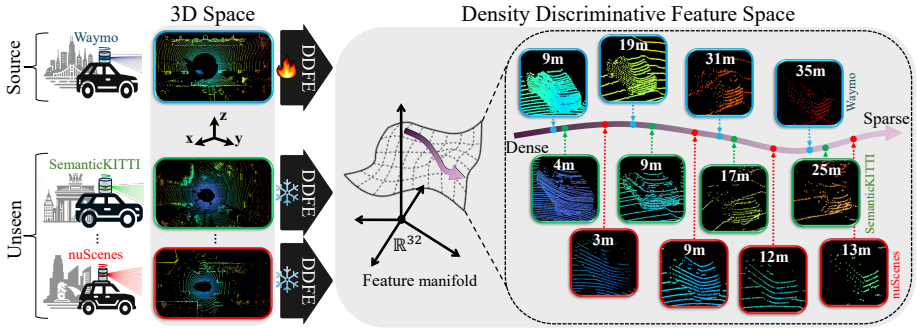


Fig. 1: Motivation of our DDFE: Leveraging Diverse Densities in a Source LiDAR for Domain Generalization - Despite the apparent differences in density distributions between Waymo (64-channel), SemanticKITTI (64-channel), and nuScenes (32-channel), they share regions of overlapping density distributions. For example, the observation that a vehicle at 35 meters in Waymo (top) has a similar density to one at 25 meters in SemanticKITTI (middle) and 12 meters in nuScenes (bottom) underscores this phenomenon. This understanding of varying local density within a source domain serves as a foundation of our domain generalization method. The proposed DDFE transforms features from 3D space to a unified density space without additional training on unseen data, enhancing domain generalization performance.

Waymo [35] in the USA, the SemanticKITTI [2] in Germany, and the nuScenes [4] in Singapore; and sensor-induced discrepancies, including differences in the number of beams and the field of view (FOV).

Numerous Unsupervised Domain Adaptation (UDA) studies [17, 30, 45] address these issues but require additional fine-tuning whenever the target domain changes. In contrast, models deployed with robust Domain Generalization (DG) techniques are increasingly sought after for their potential to enable real-time autonomous driving systems without needing constant fine-tuning. Nevertheless, this crucial field remains under-researched, and its potential has yet to be fully realized. Existing study [16] pinpoints the point cloud density distribution as a prime performance hindrance. While some solutions like point cloud sampling and completion-based methods attempt to remedy this, they fall short in achieving the desired performance [44] and require sequentially labeled data and knowledge of ego-motion [29].

In this paper, we introduce a novel perspective in domain generalization by addressing the challenges posed by density variations across different LiDAR sensors. Previous studies [14,16,38] have primarily focused on global density differences, typically considering datasets from 64-channel LiDARs like Waymo [35] denser than those from 32-channel LiDARs like nuScenes [4]. However, these approaches oversimplify the inherent complexity of LiDAR data, which exhibits a wide range of density spectra over distance, and interpret them as a single global density value. Contrary to the simplified view of global density comparisons, our research recognizes that LiDAR point clouds are composed of regions

with varying densities, as shown in Fig. 1. These variations are influenced by the distance of objects from the LiDAR sensor, leading to a more complex and nuanced understanding of density within LiDAR data.

To do so, we propose a Density Discriminative Feature Embedding (DDFE) module, designed to enhance the domain generalization capabilities of LiDAR-based segmentation networks by exploiting these density variations. The DDFE module incorporates a beam density estimation module that encodes the specific densities for each 3D voxel, enabling refined density discrimination across regions. The features are then modulated using attention mechanisms guided by beam density, further enhancing the model's ability to generalize across domains with varying density characteristics. Furthermore, we introduce a density soft clipping technique. This method constrains the density spectrum, ensuring it does not encompass density distributions from unseen domains that are absent in source datasets. To supplement this, we use density augmentation to widen the density spectrum of the source data, thereby enhancing its domain generalization capabilities. Extensive experiments validate the superiority of our method over the conventional domain adaptation and generalization methods across multiple 3D backbone networks.

In summary, our primary contributions include:

- We introduce a new perspective for domain generalization to overcome density variations caused by different LiDAR sensors, utilizing the diverse densities within point clouds observed in the source domain.
- We propose a density discriminative feature embedding module designed to identify areas with similar density distribution and amplify relevant features.
- We present a simple yet effective data augmentation strategy, aimed at broadening the density spectrum of source data.
- Extensive experiments demonstrate that our method outperforms state-of-the-art Domain Generalization and Domain Adaptation methods.

2 Related Work

2.1 LiDAR-based Semantic Segmentation

LiDAR point clouds pose unique challenges due to their irregular, unordered, and unstructured nature. Consequently, approaches to represent point cloud data have been broadly classified into three categories: Projection-based, point-based, and voxel-based methods. Projection-based methods transform a 3D point cloud into a 2D representation through either a spherical or bird's-eye view projection. By doing so, they can utilize lightweight models like 2D convolution [5, 8, 25, 47] or transformer [1,6] rather than 3D convolution. Nevertheless, such methods face the inherent limitation of 2D kernels not preserving the 3D geometric information of the real world. Point-based methods [9, 15, 36] directly extract features from the point cloud. These methods utilize all the 3D spatial information without distortion. However, the trade-off is a substantial demand on memory and computational resources. In voxel-based approaches [7, 11, 13, 48], point clouds

Table 1: LiDAR configuration for each dataset (f_{min}, f_{max}: the minimum and maximum LiDAR field of view, H_{b}, V_{b}: the number of horizontal and vertical beams).

	Waymo [35]	SemanticKITTI [2]	nuScenes [4]	Pandaset [42]	SemanticPOSS [24]
H_{b}	2560	2048	1080	1800	1800
V_{b}	64	64	32	64	40
$[f_{\text{min}}, f_{\text{max}}]$ (°)	$[-17.6, +2.4]$	$[-24.8, +2.0]$	$[-30.0, +10.0]$	$[-25.0, +15.0]$	$[-16.0, +7.0]$

are quantized into 3D grids, known as voxels, and features are extracted using 3D convolutional methods. Alongside sparse convolution [12], voxel-based methods achieve high performance by maintaining an actual geometric receptive field while ensuring an efficient computational load.

2.2 LiDAR-based Domain Adaptation

Variations in LiDAR sensor configurations, as shown in Tab. 1, or alterations in operating environments can lead to differences in the distribution of point cloud data, which, in turn, may adversely affect the perception capabilities of autonomous driving systems. To tackle this challenge without constructing new training datasets for every sensor type and location, research has increasingly focused on unsupervised domain adaptation techniques [17, 26, 30, 31, 39, 43, 45]. These methods strive to preserve performance in the target domain by utilizing labeled data from a source domain alongside unlabeled data from the target domain. C&L [44] leverages sequential data to train a voxel completion network and segments on the reconstructed canonical domain to overcome domain discrepancies. Rochan *et al.* [28] propose a range-view-based unsupervised domain adaptation method, aligning beam positions between training and target data. LiDAR-UDA [34] employs beam subsampling to mimic different LiDAR sensors, and utilizes cross-frame ensembling and a Learned Aggregation Model to acquire improved pseudo labels. Although these DA methods demonstrate effectiveness, they share a common limitation: the need for individual fine-tuning with target data upon each domain shift.

2.3 LiDAR-based Domain Generalization

Domain Generalization [16, 19, 21, 29, 32, 33, 41] aims to improve the performance in scenarios where the target domain data, unseen during training, is encountered. DGLSS [16] leverages sparsity invariance feature consistency and bridges the semantic correlation consistency between the source data and the sparse domain generated by beam sampling. LiDomAug [29] aggregates multiple frames to generate dense world models and augment data by sampling through randomized LiDAR configurations with additional knowledge of ego-motion and the sequentially labeled data. BEV-DG [21] employs a bird's-eye view for enhanced cross-modal learning and develops density-maintained vector modeling for efficient learning of domain-invariant features. LiDOG [32] utilizes semantic priors in a 2D bird's-eye view to extract domain-agnostic features. Sanchez *et*

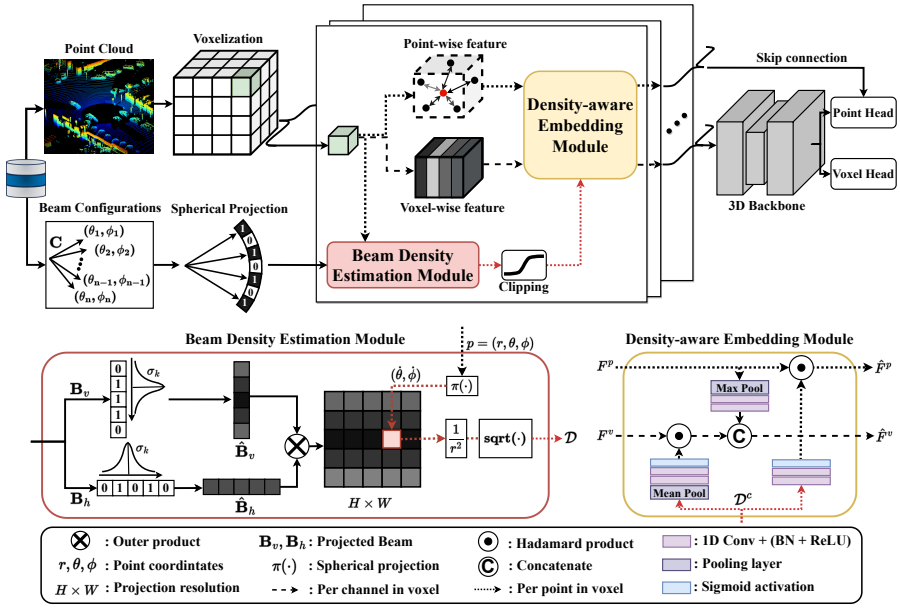


Fig. 2: Overview of the DDFE pipeline.

al. [33] introduce a label propagation method that integrates multi-frame aggregation and ego-motion, although it demands significant computational resources due to the adaptation of KPConv [36]. Distinctively, our approach advances a single-frame domain generalization approach that considers the inherent characteristics of LiDAR. Consequently, our method offers the distinct advantage of bypassing the need for ego-motion and sequentially labeled data during the training and inference phases.

3 Method

In this section, we detail our domain generalization method for LiDAR semantic segmentation, which capitalizes on density variations observable within a single source LiDAR. The proposed Density Discriminative Feature Embedding (DDFE) module is primarily composed of four components: Point-voxel feature encoding as detailed in Sec. 3.1, Beam density estimation module in Sec. 3.2, Density soft clipping in Sec. 3.3, and Density-aware embedding module in Sec. 3.4. Additionally, Sec. 3.5 outlines our density augmentation technique. The overall framework of the proposed DDFE module is shown in Fig. 2.

3.1 Point-voxel feature encoding

We introduce a technique to extract generalized representations across domains from point clouds. The intensity distribution in point clouds is influenced by

the specific LiDAR sensor used, which can adversely affect the performance of perception models during sensor transitions due to the variance in intensity information. Thus, we omit the use of LiDAR intensity values to enhance domain generalization performance. Furthermore, each LiDAR sensor has a unique distribution of measurement errors, particularly within the intra-voxel region. To minimize variance induced by localized sensing noise, we incorporate a direct voxel-wise feature encoding into the PointNet-based feature extraction [27]. This direct encoding, which bypasses localized information within the voxels, effectively dismisses grid size variations, such as those of 20cm. Given a point cloud $\mathbf{P} = \{p_i \in \mathbb{R}^3 \mid i = 1, ..., N\}$ containing N 3D points, we initially partition it into M 3D voxels. For each voxel v_j, it is defined as a set of points and is represented by $\mathbf{V} = \{v_j \in \mathbb{R}^{L\times 3} \mid j = 1, ..., M\}$, where L indicates the number of points contained in each voxel. The coordinates of these points are subsequently transformed into the form of $(\cos(\theta_j), \sin(\theta_j), \phi_j, r_j)$, where (θ_j, ϕ_j, r_j) correspond to spherical coordinates. These transformed coordinates are then encoded into voxel-wise features $F^v \in \mathbb{R}^{M\times 16}$ through an MLP. We also encode point-wise features $F^p \in \mathbb{R}^{N\times 16}$, which are essential to produce point-wise outputs. For each point k in voxel v_j centered at (x_j^c, y_j^c, z_j^c), we compute the offset as $(x_{jk} - x_j^c, y_{jk} - y_j^c, z_{jk} - z_j^c)$, with $k = 1, ..., L_j$. Here, L_j represents the number of points belonging to v_j. These offsets are then processed by a point head to generate point-wise features.

3.2 Beam density estimation module

In this section, we detail the method to compute the density expectation for each ray emitted from a LiDAR sensor. Since all rays from a LiDAR originate from a singular source and are emitted at a fixed angle, the density associated with each beam can be deduced using a spherical projection informed by the beam configuration. The LiDAR sensor configuration, accessible from low-level sources such as an SDK, determines the set of horizontal and vertical inclinations $\mathbf{C}_h$ and $\mathbf{C}_v$ for each beam in this manner:

$$\mathbf{C}_h = \left\{ \frac{2\pi i}{H_\mathrm{b}} \right\}_{i\in\{1,\dots,H_\mathrm{b}\}}, \ \mathbf{C}_v = \left\{ \frac{(f_\mathrm{max} - f_\mathrm{min})j}{V_\mathrm{b}} + f_\mathrm{min} \right\}_{j\in\{1,\dots,V_\mathrm{b}\}}, \tag{1}$$

where f_min and f_max denote the minimum and maximum LiDAR field of view, while H_b and V_b represent the number of horizontal and vertical beams, detailed in Tab. 1. To project LiDAR beam inclinations onto spherical projected image coordinates, we use the projection function $\pi : (\theta, \phi) \to (\dot{\theta}, \dot{\phi})$ defined as:

$$\dot{\theta} = \left\lfloor \frac{\theta}{2\pi} W \right\rfloor, \ \dot{\phi} = \left\lfloor \frac{\phi - f_\mathrm{min}^\mathrm{proj}}{f_\mathrm{max}^\mathrm{proj} - f_\mathrm{min}^\mathrm{proj}} H \right\rfloor, \tag{2}$$

where H and W are the height and width resolutions of the projected image, respectively set to $H = 512$, $W = 5120$. We define the projected image's field of view as $\left[f_\mathrm{min}^\mathrm{proj}, f_\mathrm{max}^\mathrm{proj}\right] = [-30.0, 15.0]$ to accommodate a range of LiDAR sensors.

Given this, beam configurations can be transformed into 1-D binary vectors $\mathbf{B}_v \in \mathbb{R}^H$ and $\mathbf{B}_h \in \mathbb{R}^W$ as follows:

$$\mathbf{B}_h(\dot{\theta}) = \mathbb{1}_{\mathbf{C}_h}(\theta),\ \mathbf{B}_v(\dot{\phi}) = \mathbb{1}_{\mathbf{C}_v}(\phi),\ \text{where } \mathbb{1}_{\mathbf{C}}(x) = \begin{cases} 1, & \text{if } x \in \mathbf{C} \\ 0, & \text{otherwise} \end{cases}, \tag{3}$$

where the indicator function $\mathbb{1}(\cdot)$ yields 1 when the azimuth or elevation (θ, ϕ) of a pixel $(\dot{\theta}, \dot{\phi})$ corresponds to the beam inclinations specified in $\mathbf{C}$ in Eq. (1). Here, $\mathbf{C}$ can be either $\mathbf{C}_h$ or $\mathbf{C}_v$ that represent the locations of projected vertical and horizontal beams, respectively. To compute the beam density, we convolve the binary vectors $\mathbf{B}_h$ and $\mathbf{B}_v$, with four distinct 1-D Gaussian kernels, each characterized by standard deviations $\sigma_k = \{10, 30, 50, 70\}$ as follows:

$$\hat{\mathbf{B}}_h = \mathbf{B}_h * \mathbf{G}_{\sigma_k},\ \hat{\mathbf{B}}_v = \mathbf{B}_v * \mathbf{G}_{\sigma_k}. \tag{4}$$

Finally, we define beam density $\mathcal{D}_i$ of point p_i as follows:

$$\mathcal{D}_i = \left[\sqrt{\hat{\mathbf{B}}_h^{(k)}(\dot{\theta}_i) \cdot \hat{\mathbf{B}}_v^{(k)}(\dot{\phi}_i)/r_i^2} \right]_{k=1}^{4}, \tag{5}$$

where r_i is the radial distance of a LiDAR point p_i in the spherical coordinates, and $[\cdot]_{k=1}^{4}$ denotes the concatenation operation. Taking into account that the density of the beam diminishes in proportion to the square of the distance due to its radial emission, we incorporate the inverse of r^2 in our computation.

3.3 Density soft clipping

When exposed to densities that deviate from those in the source dataset, the model is susceptible to performance degradation in an unseen domain. To alleviate this issue, we introduce a density soft clipping method that confines the density spectrum using the $\tanh(\cdot)$ function as follows:

$$\begin{gathered} \mathcal{D}_i^c = \tanh\left(\frac{\mathcal{D}_i - m}{l}\right) l + m, \\ m = \frac{\mathcal{P}_{90}(\mathcal{D}) + \mathcal{P}_{10}(\mathcal{D})}{2},\ l = \frac{\mathcal{P}_{90}(\mathcal{D}) - \mathcal{P}_{10}(\mathcal{D})}{2}, \end{gathered} \tag{6}$$

where $\mathcal{D}_i^c$ is clipped density of point p_i, and $\mathcal{P}_{90}$ and $\mathcal{P}_{10}$ are functions that extract 90^{th} and 10^{th} channel-wise percentile values from the density embedding of the training domain, respectively. Employing these percentile values allows us to discount the outlier density values. To calculate the percentile on the fly under memory constraints, we adopt the Reservoir Sampling [37] technique with a sample size $N = 1000$. Please refer to the supplementary material for a more detailed algorithm description.

3.4 Density-aware embedding module

For each point p_i, a point-wise density embedding feature $\mathcal{D}_i^c$ is obtained via the beam density estimation module incorporating density soft clipping. This feature serves as an input of both a point-wise attention function f_p and a voxel-wise attention function f_v. These mechanisms are pivotal for crafting domain-invariant density-discriminative features. In the point-wise attention framework, $\mathcal{D}_i^c$ is synchronized dimensionally with the point-wise feature F_i^p, facilitating the creation of a density discriminative feature $\hat{F}_i^p$ as follows:

$$\hat{F}_i^p = f_p(\mathcal{D}_i^c) \odot F_i^p, \tag{7}$$

where f_p consists of two 1D convolution layers followed by a sigmoid function. For voxel-wise attention, it begins by averaging the density embedding features for all points encapsulated within a voxel. This aggregated measure then undergoes a similar treatment as follows:

$$\hat{F}_j^v = \text{Concat}(f_v(\mathcal{D}_j^c) \odot F_j^v, g(F_j^p)),\ \mathcal{D}_j^c = \frac{1}{|L_j|} \sum_{p_i \in v_j} \mathcal{D}_i^c, \tag{8}$$

where $\mathcal{D}_j^c$ is a clipped density of voxel v_j and $\hat{F}_j^v$ is a voxel-wise density discriminative feature and f_v consists of two 1D convolution layers followed by a sigmoid function. An aggregation function g for the voxel-wise feature F_j^v is adjusted by voxel-wise attention and concatenated with the max pooled point-wise feature within the v_j. Finally, the extracted features are passed through a single 1D convolutional layer, resulting in 32-channel features. These density-aware features are subsequently fed into a 3D backbone for LiDAR semantic segmentation.

3.5 Density Augmentation

The DDFE module aims to align densities across varied sensor domains by considering each dataset as a collection of multiple density domains. This method, while effective in many scenarios, may face challenges when there's little to no overlap in the density ranges between the source and unseen datasets. To address this potential issue, we introduce a density augmentation method aimed at broadening the density spectrum covered by the training data. Several existing 3D point cloud augmentation strategies, such as [16, 22, 38] outlined in recent studies, attempt to address density variations either directly or by implication. We adapt the Mix3D [22] for our purposes, which we refer to as enhanced-Mix3D, incorporating random translations along the direction of ego-vehicle movement, along with additional rotational transformations to simulate a wider range of density variations. We also employ beam sampling technique [16, 38], selectively eliminating specific LiDAR beams, to amplify the density in the lower direction. During the training phase, we apply both enhanced-Mix3D and beam sampling augmentations with a set probability of 0.5, aiming to effectively simulate diverse density conditions. Please refer to the supplementary material for more detailed implementation.

4 Experiments

In this section, we demonstrate the domain generalization performance of our method through extensive experiments. The implementation details and experimental settings of the proposed method are detailed in Sec. 4.1. The configuration of datasets used for evaluation is detailed in Sec. 4.2. A comparative analysis with recent domain generalization and domain adaptation methods is presented in Sec. 4.3. The effectiveness of individual components within the proposed DDFE and computational cost are analyzed in Section Sec. 4.4. Detailed per-class performance metrics are available in the supplementary material.

4.1 Implementation Details

We employ the point head inspired by Cylinder3D [48] to produce point-wise outputs as shown in Fig. 2. During the training phase, we incorporate the Lovasz-Softmax loss $\mathcal{L}^{lovasz}$ [3] along with the Weighted Cross-Entropy loss $\mathcal{L}^{wce}$. The total loss is composed of an equal-weighted combination of point-wise and voxel-wise losses as follows:

$$\mathcal{L}_{total} = \mathcal{L}^{lovasz}_{point} + \mathcal{L}^{wce}_{point} + \mathcal{L}^{lovasz}_{voxel} + \mathcal{L}^{wce}_{voxel}. \tag{9}$$

We use the Adam optimizer with an initial learning rate of 1e-3. This rate is decreased by a factor of 0.99 with every epoch. The training is conducted over 30 epochs with a batch size of two on a single NVIDIA RTX 3090, and the epoch yielding the highest source validation mIoU is chosen. For voxelization, we adopt a cubic size of [20cm, 20cm, 20cm]. In cases where multiple point labels are found within a voxel, the voxel label is determined based on the predominant label, aligning with the approach in Cylinder3D [48].

4.2 Datasets

In our experimental setup, we employ three datasets: Waymo [35], nuScenes [4], and SemanticKITTI [2]. Notably, the Waymo and SemanticKITTI datasets use 64-channel LiDAR, while the nuScenes employs a 32-channel LiDAR. For detailed information, please refer to Tab. 1. In line with methods from previous studies [16, 29], we split the nuScenes sequences into 700 for training and 150 for validation. The Waymo dataset is partitioned into 798 sequences for training and 202 for validation. Regarding the SemanticKITTI dataset, we use sequences 00 to 10 for training, setting aside sequence 08 exclusively for validation. In integrating class variations across datasets, we follow the class mapping configurations from previous studies for fair comparisons.

4.3 Comparison to State-of-the-Art DA/DG Methods

The field of domain generalization of LiDAR-based semantic segmentation is still in its formative stages, with a lack of universally accepted experimental standards. This situation has led to a diversity of experimental designs across different studies. To facilitate fair comparisons, we align our experimental framework

Table 2: Comparison with domain generalization methods based on MinkowskiNet architecture using the mIoU. The best and the second best results are highlighted in **bold** and underline, respectively.

Method	DA	Source	W	K	N	Source	K	W	N	Source	N	W	K
Base		W	75.37	49.40	47.83	K	57.31	35.24	37.42	N	65.78	38.65	36.24
IBN-Net [23]			75.47	51.13	44.72		57.74	36.99	38.74		65.31	36.53	36.93
MLDG [20]			72.47	48.94	48.64		56.26	35.39	36.77		61.32	36.33	32.70
COSMIX (W) [30]	✓		-	-	-		49.35	39.46	38.94		-	-	-
COSMIX (K) [30]	✓		66.68	44.71	49.96		-	-	-		-	-	-
COSMIX (N) [30]	✓		65.68	40.99	47.98		49.98	38.05	43.25		-	-	-
DGLSS [16]			75.28	51.23	49.61		59.62	40.67	44.83		65.32	40.93	38.98
Ours			**76.15**	**57.07**	**56.75**		**62.50**	**42.73**	**49.43**		**68.16**	**45.98**	**46.52**

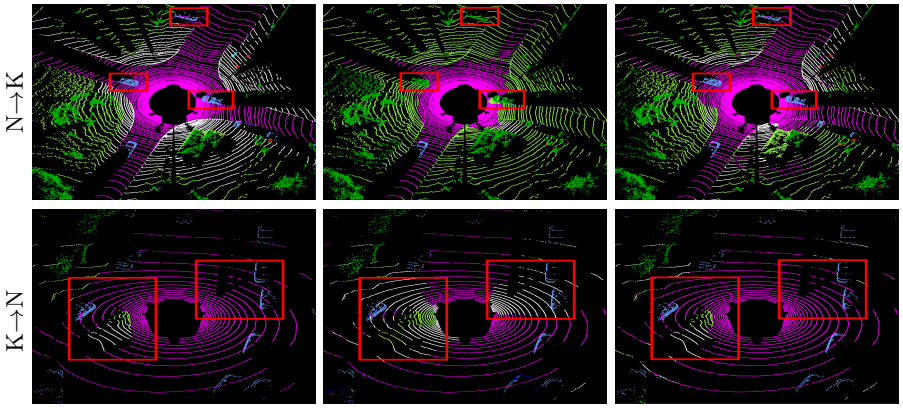
Table 3: Comparison with domain adaptation and data augmentation methods.

Backbone	Methods	K→N	N→K	Backbone	Methods	K→N	N→K
MinkNet42 [7]	Base	37.8	36.1	C&L [44]	Base	27.9	23.5
	CutMix [46]	37.1	37.6		SWD [18]	27.7	24.5
	Copy-Paste [10]	38.5	41.1		3DGCA [39]	27.4	23.9
	Mix3D [22]	43.1	44.7		C&L [44]	31.6	33.7
	PolarMix [40]	45.8	39.1		LiDomAug [29]	39.2	37.9
	LiDomAug [29]	45.9	48.3		LiDAR-UDA [34]	41.8	34.0
	Ours (v=5cm)	48.6	**51.3**		Ours (v=5cm)	42.5	**41.0**
	Ours (v=20cm)	**50.1**	46.3		Ours (v=20cm)	**47.1**	40.3

with those utilized in seminal works such as LiDomAug [29] and DGLSS [16]. This approach allows us to benchmark our method against a range of Domain Adaptation (DA) and Domain Generalization (DG) techniques, showcasing the effectiveness of our method in the context of evolving domain generalization challenges.

Experiments in the DGLSS Setting We compare our method with a domain adaptation method [30] and three domain generalization methods [16,20,23], following the experimental framework defined in DGLSS [16] using Waymo (W), SemanticKITTI (K), and nuScenes (N). We adopt MinkowskiNet [7] as our backbone network, aligning with the configuration utilized in DGLSS. The results in Tab. 2 indicate that the proposed method consistently outperforms DGLSS across all datasets. Impressively, our method not only demonstrates superior domain generalization performance but also dominates in the source-to-source settings (W→W, K→K, N→N). Unlike DGLSS, which tailors its augmentation strategies for each specific dataset, our method employs a uniform hyperparameter setting across all datasets, achieving enhanced domain generalization results. Our method achieves an average increase of +12.9% over DGLSS for unseen datasets using Waymo as the source data (W→K, W→N). With SemanticKITTI as the source, there's an average enhancement of +7.8% for other unseen domains (K→W, K→N), and employing nuScenes as the source data yields an average increase of +15.8% on other unseen datasets (N→W, N→K).

These results demonstrate the robustness of the proposed method in addressing domain discrepancies.



(a) Ground-truth (b) MinkNet42 (c) MinkNet42+Ours

Fig. 3: Qualitative comparison with MinkNet42 backbone. Top: Model trained on nuScenes, tested on SemanticKITTI (N→K). Bottom: Trained on SemanticKITTI, tested on nuScenes (K→N).

Experiments in the LiDomAug Setting We benchmark the proposed method against various augmentation methods [10, 22, 40, 46], domain adaptation methods [18, 34, 39, 44], and a domain generation method [29]. We adhere to the experimental setup established by LiDomAug [29], employing MinkNet42 [7] and C&L [44] as backbone networks, and utilizing SemanticKITTI and nuScenes datasets for evaluation. Unlike the DGLSS [16] setting which uses a voxel size of 20 cm, the LiDomAug setting uses a voxel size of 5 cm. We benchmark the proposed method against various augmentation methods [10, 22, 40, 46], domain adaptation methods [18, 34, 39, 44], and a domain generation method [29]. We adhere to the experimental setup established by LiDomAug [29], employing MinkNet42 [7] and C&L [44] as backbone networks, and utilizing SemanticKITTI and nuScenes datasets for evaluation. The comparative analysis in Tab. 3 shows that our method surpasses all considered augmentation methods in both (K→N) and (N→K) configurations. Particularly with MinkNet42 as the backbone, our method achieves a 5.9% and 5.8% increase in performance over LiDomAug in the (K→N) and (N→K) scenarios, respectively. This enhancement is significant, considering LiDomAug's reliance on ego-motion and multi-frame data integration for domain generalization. When utilizing C&L [44] as the backbone, the proposed method significantly improves mIoU by +34.5% and +8.4% for the scenario of training on SemanticKITTI and evaluating on nuScenes (K→N), compared to C&L and LiDomAug, respectively. Moreover, in the transition from the sparse 32-channel dataset (nuScenes) to the denser 64-channel dataset (SemanticKITTI) (N→K), our method shows a +21.7% increase in mIoU over C&L and +8.2% improvement over LiDomAug.

We further conduct an analysis with a voxel size of 20 cm, unlike the comparative methods that all use a voxel size of 5 cm. Increasing the voxel size from 5 cm to 20 cm results in reductions of 30.3% in training time and 62.5% in infer-

Table 4: Ablation study on individual components within our method using mIoU. Experiments evaluate the model's performance with and without the following: (a) Point-voxel encoding, (b) Density-aware embedding module, (c) Density clipping, and (d) Density-Augmentation. All experiments were conducted using a voxel size of 20 cm.

(a)	(b)	(c)	(d)	K→N	N→K
				40.7	31.4
✓				43.0 (+5.7%)	35.0 (+11.5%)
✓	✓			45.7 (+12.3%)	40.5 (+29.0%)
✓	✓	✓		46.2 (+13.5%)	41.8 (+33.1%)
✓	✓	✓	✓	**50.1** (+23.1%)	**46.3** (+47.5%)

ence time, while maintaining comparable performance—with a slight decrease of 3.5% on the MinkNet42 backbone and an increase of 4.7% on the C&L backbone. Based on these results, we recommend a voxel size of 20 cm (v=20cm) as the default setting for our proposed method. The effectiveness is further illustrated in the qualitative results depicted in Fig. 3, highlighting the substantial improvements our method brings to the domain generalization performance of the 3D backbone network. Please refer to the supplementary material for detailed results, including per-class IoU.

4.4 Ablation Studies

To validate the impact of individual components within our method, we conduct ablation studies. These studies concentrate on the evaluation of point-voxel encoding, the density-aware embedding module, density soft clipping, and density augmentation, providing insights into the significance of each element in enhancing the overall methodology.

Analysis of the individual components of our method In our comprehensive ablation study in Tab. 4, we thoroughly investigate the individual components of our method. We use the experimental setup proposed by LiDomAug [29] and employ MinkowskiNet [7] as the backbone. This study dissects the impact of our DDFE, which includes (a) point-voxel encoding, (b) density-aware embedding module, and (c) density clipping. Additionally, it examines (d) the impact of our density augmentation method.

The integration of all these elements results in a significant performance uplift +13.5% in the (K→N) scenario and +33.1% in the (N→K) scenario over the baseline model. Notably, the density-aware embedding module stands out for its substantial effect, closely followed by the point-voxel encoding, highlighting its critical role. Furthermore, the study validates the utility of density soft clipping in enhancing model flexibility towards novel density configurations. Meanwhile, the gains observed with density clipping highlight the inherent challenge in adapting to unfamiliar density landscapes, emphasizing the nuanced contributions of each component towards achieving robust domain generalization. Lastly,

Table 5: Comparison of data augmentation methods in the proposed DDFE. The best and the second best results are highlighted in **bold** and underline, respectively. 'D. V.' refers to Density Variation situation, and 'B. S.' to Beam Sampling augmentation.

Method	D. V.	K→N	N→K
DDFE		45.9	41.8
+ PolarMix [40]		48.2 (+5.0%)	42.3 (+1.2%)
+ B. S. [38]	✓	49.1 (+7.0%)	42.2 (+0.9%)
+ Mix3D [22]	✓	47.4 (+3.2%)	44.8 (+7.1%)
+ Mix3D + B. S. [38]	✓	49.1 (+7.0%)	44.2 (+5.7%)
+ E-Mix3D	✓	48.5 (+5.6%)	46.4 (+11.0%)
+ E-Mix3D + B. S. [38]	✓	**49.4** (+7.6%)	**46.5** (+11.2%)

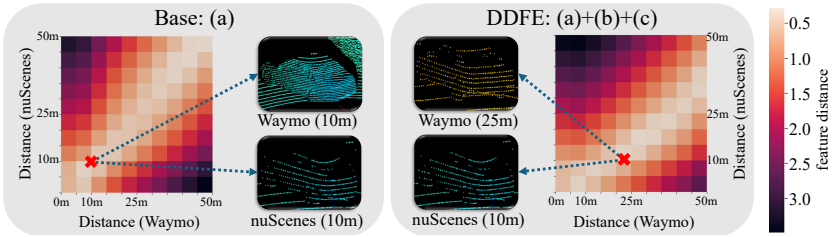


Fig. 4: Visualization of feature similarity matrices between the Waymo (64-channel) and the nuScenes (32-channel) datasets, with a focus on the distance of objects from the LiDAR. We utilize models trained with the nuScenes dataset for visualization. (a) A point-voxel feature encoding method. (b) A density-aware embedding module. (c) A density soft clipping. Our DDFE combines all modules (a), (b), and (c).

incorporating density augmentation into DDFE leads to further enhancements, with an additional 8.4% performance increase in the (K→N) scenario and 10.8% in the (N→K) scenario. This demonstrates the effectiveness of the proposed density augmentation scheme in significantly boosting the performance of DDFE.

Augmentation Tab. 5 shows the performance comparison when various augmentation methods are applied to the proposed DDFE. The experimental setup follows DGLSS [16], and the baseline involves applying the proposed DDFE to MinkowskiNet [7]. The proposed enhanced-Mix3D shows a 2.3% performance improvement over the original Mix3D [22] when trained on SemanticKITTI and tested on nuScenes (K→N), and a 3.6% improvement when trained on nuScenes and tested on SemanticKITTI (N→K). In conjunction with the beam sampling method, it achieves a 7.6% performance improvement in (K→N) and 11.2% in (N→K) compared to the baseline.

Analysis of Density Discriminative Feature Embedding (DDFE) We delve into the effectiveness of the DDFE module by examining the feature similarity between the source dataset, nuScenes, and the unseen dataset, Waymo,

as depicted in Fig. 4. Our focus is on the input voxel features $\hat{F}^v$ of the 3D backbone model generated by the DDFE. We initiate our analysis by setting a baseline that utilizes a point-voxel feature encoding method for comparison. Following this, we proceed to assess our enhanced method, which integrates a density-aware embedding module along with density soft clipping, to understand its impact on bridging the domain gap. Similarity metrics are derived from the L2 distance between averaged features across specified distance intervals (*e.g.*, 0-5m, ..., 45-50m) for each dataset. According to Fig. 4, the baseline consistently shows higher feature similarity across distances, influenced by its direct embedding of 3D point coordinates. Conversely, our integrated model with density-aware embedding and soft clipping significantly aligns feature similarity across consistent density distributions. It demonstrates the efficacy of the proposed modules in adapting to differences between the source and unseen datasets.

Computational Cost The inference time of our method for processing a single frame with the nuScenes is 44ms on an NVIDIA RTX 3090. The inclusion of our DDFE module into the MinkNet42 architecture accounts for an extra 8ms of this computation time, while the base MinkNet42 architecture alone requires 36ms. This integration modestly elevates the model's complexity, introducing approximately 23.8k parameters (+0.06%). Furthermore, the application of density augmentation during training adds an extra 60ms per scan on the nuScenes.

5 Conclusion

In this paper, we introduce a new perspective for domain generalization of LiDAR semantic segmentation by exploiting the concept of density diversity within a source domain. Based on this perspective, we propose the Density Discriminative Feature Embedding (DDFE) module. DDFE is designed to incorporate expected densities derived from LiDAR beams into a density-aware feature space, significantly enhancing the model's capability to distinguish between different densities. This novel approach improves the adaptability and accuracy of LiDAR semantic segmentation, enabling them to perform effectively across diverse domains, including those previously unseen. In addition to the core methodology, we also present a simple and effective data augmentation technique that extends the density spectrum of the source data. Extensive experiments on the SemanticKITTI, Waymo, and nuScenes datasets demonstrate the effectiveness of the proposed method in elevating the domain generalization performance of LiDAR semantic segmentation. A limitation of this study is its exclusive focus on semantic segmentation tasks. Thus, in future work, we plan to extend our domain generalization method to encompass other LiDAR-based perception tasks, such as 3D object detection. This extension can involve utilizing object-centric density discriminative features alongside point-wise density embedding.

Acknowledgements

This work was supported by the Digital Innovation Hub project supervised by the Daegu Digital Innovation Promotion Agency(DIP) grant funded by the Korea government(MSIT and Daegu Metropolitan City) in 2023(DBSD1-02, Vision picking system for the logistics industry based on artificial intelligence object recognition) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00210908).

References

1. Ando, A., Gidaris, S., Bursuc, A., Puy, G., Boulch, A., Marlet, R.: Rangevit: Towards vision transformers for 3d semantic segmentation in autonomous driving. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5240–5250 (2023)
2. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: Semantickitti: A dataset for semantic scene understanding of lidar sequences. In: Proceedings of IEEE International Conference on Computer Vision (ICCV). pp. 9297–9307 (2019)
3. Berman, M., Triki, A.R., Blaschko, M.B.: The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4413–4421 (2018)
4. Caesar, H., et al.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11621–11631 (2020)
5. Chen, T.H., Chang, T.S.: Rangeseg: Range-aware real time segmentation of 3d lidar point clouds. IEEE Transactions on Intelligent Vehicles **7**(1), 93–101 (2022). https://doi.org/10.1109/TIV.2021.3085827
6. Cheng, H.X., Han, X.F., Xiao, G.Q.: Transrvnet: Lidar semantic segmentation with transformer. IEEE Transactions on Intelligent Transportation Systems **24**(6), 5895–5907 (2023). https://doi.org/10.1109/TITS.2023.3248117
7. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3075–3084 (2019)
8. Cortinhal, T., Tzelepis, G., Erdal Aksoy, E.: Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In: Bebis, G., Yin, Z., Kim, E., Bender, J., Subr, K., Kwon, B.C., Zhao, J., Kalkofen, D., Baciu, G. (eds.) Advances in Visual Computing. pp. 207–222. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-64559-5_16
9. Fan, S., Dong, Q., Zhu, F., Lv, Y., Ye, P., Wang, F.Y.: Scf-net: Learning spatial contextual features for large-scale point cloud segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14504–14513 (2021)
10. Ghiasi, G., Cui, Y., Srinivas, A.J., Qian, R., Lin, T.Y., Cubuk, E.D., Le, Q.V., Zoph, B.: Simple copy-paste is a strong data augmentation method for instance segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2918–2928 (2021)

11. Graham, B., Engelcke, M., Van Der Maaten, L.: 3d semantic segmentation with submanifold sparse convolutional networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9224–9232 (2018)
12. Graham, B., Van der Maaten, L.: Submanifold sparse convolutional networks. arXiv preprint arXiv:1706.01307 (2017)
13. Hou, Y., Zhu, X., Ma, Y., Loy, C.C., Li, Y.: Point-to-voxel knowledge distillation for lidar semantic segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8479–8488 (2022)
14. Hu, Q., Liu, D., Hu, W.: Density-insensitive unsupervised domain adaption on 3d object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 17556–17566 (2023)
15. Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A.: Randla-net: Efficient semantic segmentation of large-scale point clouds. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11108–11117 (2020)
16. Kim, H., Kang, Y., Oh, C., Yoon, K.J.: Single domain generalization for lidar semantic segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 17587–17598 (2023)
17. Kong, L., Quader, N., Liong, V.E.: Conda: Unsupervised domain adaptation for lidar segmentation via regularized domain concatenation. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA). pp. 9338–9345. IEEE (2023)
18. Lee, C.Y., Batra, T., Baig, M.H., Ulbricht, D.: Sliced wasserstein discrepancy for unsupervised domain adaptation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10285–10295 (2019)
19. Lehner, A., Gasperini, S., Marcos-Ramiro, A., Schmidt, M., Mahani, M.A.N., Navab, N., Busam, B., Tombari, F.: 3d-vfield: Adversarial augmentation of point clouds for domain generalization in 3d object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 17295–17304 (2022)
20. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.: Learning to generalize: Meta-learning for domain generalization. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). vol. 32 (2018)
21. Li, M., Zhang, Y., Ma, X., Qu, Y., Fu, Y.: Bev-dg: Cross-modal learning under bird's-eye view for domain generalization of 3d semantic segmentation. In: Proceedings of IEEE International Conference on Computer Vision (ICCV). pp. 11632–11642 (2023)
22. Nekrasov, A., Schult, J., Litany, O., Leibe, B., Engelmann, F.: Mix3d: Out-of-context data augmentation for 3d scenes. In: Proceedings of International Conference on 3D Vision (3DV). pp. 116–125 (2021)
23. Pan, X., Luo, P., Shi, J., Tang, X.: Two at once: Enhancing learning and generalization capacities via ibn-net. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 464–479 (2018)
24. Pan, Y., Gao, B., Mei, J., Geng, S., Li, C., Zhao, H.: Semanticposs: A point cloud dataset with large quantity of dynamic instances. In: 2020 IEEE Intelligent Vehicles Symposium (IV). pp. 687–693 (2020). https://doi.org/10.1109/IV47402.2020.9304596
25. Peng, K., Fei, J., Yang, K., Roitberg, A., Zhang, J., Bieder, F., Heidenreich, P., Stiller, C., Stiefelhagen, R.: Mass: Multi-attentional semantic segmentation of lidar

data for dense top-view understanding. IEEE Transactions on Intelligent Transportation Systems **23**(9), 15824–15840 (2022). https://doi.org/10.1109/TITS.2022.3145588
26. Peng, X., Zhu, X., Ma, Y.: Cl3d: Unsupervised domain adaptation for cross-lidar 3d detection. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). vol. 37, pp. 2047–2055 (2023)
27. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 652–660 (2017)
28. Rochan, M., Aich, S., Corral-Soto, E.R., Nabatchian, A., Liu, B.: Unsupervised domain adaptation in lidar semantic segmentation with self-supervision and gated adapters. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA). pp. 2649–2655 (2022)
29. Ryu, K., Hwang, S., Park, J.: Instant domain augmentation for lidar semantic segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9350–9360 (2023)
30. Saltori, C., Galasso, F., Fiameni, G., Sebe, N., Ricci, E., Poiesi, F.: Cosmix: Compositional semantic mix for domain adaptation in 3d lidar segmentation. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 586–602. Springer (2022)
31. Saltori, C., Lathuiliére, S., Sebe, N., Ricci, E., Galasso, F.: Sf-uda 3d: Source-free unsupervised domain adaptation for lidar-based 3d object detection. In: Proceedings of International Conference on 3D Vision (3DV). pp. 771–780 (2020)
32. Saltori, C., Osep, A., Ricci, E., Leal-Taixé, L.: Walking your lidog: A journey through multiple domains for lidar semantic segmentation. In: Proceedings of IEEE International Conference on Computer Vision (ICCV). pp. 196–206 (2023)
33. Sanchez, J., Deschaud, J.E., Goulette, F.: Domain generalization of 3d semantic segmentation in autonomous driving. In: Proceedings of IEEE International Conference on Computer Vision (ICCV). pp. 18077–18087 (2023)
34. Shaban, A., Lee, J., Jung, S., Meng, X., Boots, B.: Lidar-uda: Self-ensembling through time for unsupervised lidar domain adaptation. In: Proceedings of IEEE International Conference on Computer Vision (ICCV). pp. 19784–19794 (2023)
35. Sun, P., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2446–2454 (2020)
36. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: Flexible and deformable convolution for point clouds. In: Proceedings of IEEE International Conference on Computer Vision (ICCV). pp. 6411–6420 (2019)
37. Vitter, J.S.: Random sampling with a reservoir. ACM Trans. Math. Softw. **11**(1), 37–57 (mar 1985). https://doi.org/10.1145/3147.3165
38. Wei, Y., Wei, Z., Rao, Y., Li, J., Zhou, J., Lu, J.: Lidar distillation: Bridging the beam-induced domain gap for 3d object detection. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 179–195. Springer (2022)
39. Wu, B., Zhou, X., Zhao, S., Yue, X., Keutzer, K.: Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA). pp. 4376–4382 (2019)
40. Xiao, A., Huang, J., Guan, D., Cui, K., Lu, S., Shao, L.: Polarmix: A general data augmentation technique for lidar point clouds. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 11035–11048 (2022)

41. Xiao, A., Huang, J., Xuan, W., Ren, R., Liu, K., Guan, D., El Saddik, A., Lu, S., Xing, E.P.: 3d semantic segmentation in the wild: Learning generalized models for adverse-condition point clouds. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9382–9392 (2023)
42. Xiao, P., Shao, Z., Hao, S., Zhang, Z., Chai, X., Jiao, J., Li, Z., Wu, J., Sun, K., Jiang, K., Wang, Y., Yang, D.: Pandaset: Advanced sensor suite dataset for autonomous driving. In: ITSC. pp. 3095–3101 (2021). https://doi.org/10.1109/ITSC48978.2021.9565009
43. Yang, J., Shi, S., Wang, Z., Li, H., Qi, X.: St3d: Self-training for unsupervised domain adaptation on 3d object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10368–10378 (2021)
44. Yi, L., Gong, B., Funkhouser, T.: Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15363–15373 (2021)
45. Yuan, Z., Wen, C., Cheng, M., Su, Y., Liu, W., Yu, S., Wang, C.: Category-level adversaries for outdoor lidar point clouds cross-domain semantic segmentation. IEEE Transactions on Intelligent Transportation Systems **24**(2), 1982–1993 (2023). https://doi.org/10.1109/TITS.2022.3219853
46. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of IEEE International Conference on Computer Vision (ICCV). pp. 6023–6032 (2019)
47. Zhang, Y., Zhou, Z., David, P., Yue, X., Xi, Z., Gong, B., Foroosh, H.: Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9601–9610 (2020)
48. Zhu, X., Zhou, H., Wang, T., Hong, F., Ma, Y., Li, W., Li, H., Lin, D.: Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9939–9948 (2021)