2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Density-aware Domain Generalization for LiDAR Semantic Segmentation

Jaeyeul Kim[1,*], Jungwan Woo[1,*], Ukcheol Shin[2], Jean Oh[2], and Sunghoon Im[1]

***Abstract*— 3D LiDAR-based perception has made remarkable advancements, leading to the widespread adoption of LiDAR in autonomous driving systems. Despite these technological strides, variations in LiDAR sensors and environmental conditions can significantly deteriorate the performance of perception models, primarily due to changes in the density of point clouds. Recent studies in domain generalization have aimed to mitigate this challenge; however, they often rely on the availability of sequential data and ego-motion, which limits their applicability. To address these limitations, we propose two novel methods that enable network operation in a density-aware fashion without any constraints, thereby ensuring consistent performance despite fluctuations in point cloud density. First, we design the network to be density-aware by utilizing the kernel occupancy information from the 3D sparse convolution as geometric features. Subsequently, we further enhance density awareness by incorporating voxel-wise density prediction as an auxiliary task in a self-supervised manner. Our method demonstrates superior performance over current state-of-the-art approaches, achieving this without the need for specific data prerequisites. Our approach is compatible with a variety of 3D backbone architectures, enhancing domain generalization performance by 18.4% while adding a minimal computational overhead of only 7ms.**

I. Introduction

3D LiDAR plays a critical role in autonomous driving systems due to its precision in measuring three-dimensional distances and its extensive coverage area. The application of LiDAR-based models has shown promising outcomes in a variety of autonomous driving perception tasks such as segmentation, detection, and tracking. Nevertheless, as shown in Fig. 1, these advancements face challenges as perception models significantly underperform when deployed in environments different from their training data or when there is a switch in the LiDAR equipment. Variations in region (e.g., USA [1], Singapore [2], Germany [3]) lead to differences in object size and class distribution, and differences in sensor technology (e.g., HDL-64E [3], HDL-32E [2]) result in disparities in point cloud density (Fig. 2).

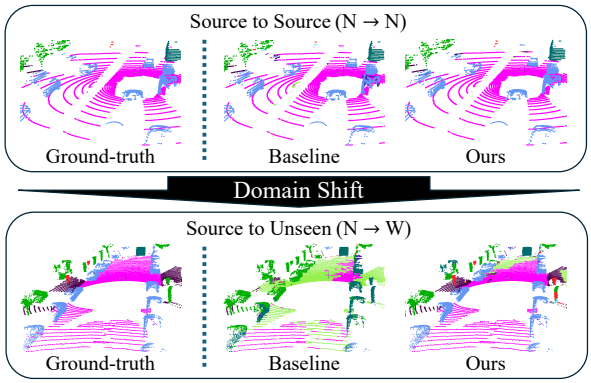


Fig. 1. Semantic segmentation results on the 32-ch nuScenes dataset (top) and the 64-ch Waymo dataset (bottom), after training on the nuScenes dataset. The vanilla baseline (middle) encounters a significant performance drop due to density mismatch as the domain changes, while applying our method (right) leads to remarkable improvements in unseen domains.

These environmental and sensor differences ultimately have a negative impact on model performance.

Various Domain Adaptation (DA) studies [4], [5], [6], [7] have made strides in addressing these domain-induced performance degradations. However, they often require detailed knowledge of the target domain and a dedicated fine-tuning stage. As a result, the focus has shifted towards Domain Generalization (DG) as a more flexible approach. Existing DG methods [4], [8] focus on identifying and countering the challenges posed by variations in point cloud density, a principal cause of performance degradation across domains. To address density variations caused by sensor disparities, numerous studies [4], [9], [10] utilize sequential data and ego-motion in the training or inference phases. This reliance on specific prerequisites restricts the practicality of these approaches, particularly when the available training or test datasets do not meet these conditions, such as in scenarios lacking ego-motion data [4], [10] or consisting of non-sequentially labeled data [4], [9].

In this paper, we present a new domain generalization framework designed to enhance model density awareness, thus improving their robustness and performance across varied domains without relying on specific preconditions. First, we propose a new density-aware sparse convolution technique that overcomes the limitations of the conventional sparse convolution method by directly integrating local density information. By utilizing each convolution kernel's occupancy

This work was supported by Korea Research Institute for defense Technology planning and advancement through Defense Innovation Vanguard Enterprise Project, funded by Defense Acquisition Program Administration(R230206), and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00210908). *(*Jaeyeul Kim and Jungwan Woo contributed equally to this work.)* (Corresponding author: Sunghoon Im.)

[1]Jaeyeul Kim, Jungwan Woo, and Sunghoon Im are with the Department of Electrical Engineering and Computer Science, DGIST, Daegu, 42988, Republic of Korea (email: {jykim94, friendship1, sunghoonim}@dgist.ac.kr)

[2]U. Shin and J. Oh are with Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, 15217, United States {ushin, hyaejino}@andrew.cmu.edu

pattern as additional geometric features, we allow the model to directly address local density variations. This strategy significantly improves the model's ability to generalize across a wide range of datasets and conditions. Second, we introduce a self-supervised density prediction auxiliary task to enhance the model's capability to perceive and adjust to variations in point cloud density. This sub-task compels the encoder to directly learn both density and semantic information, enhancing its responsiveness to fluctuations in density.

The proposed method stands out by not requiring any specific conditions related to the training or testing datasets, offering broad applicability across various backbone architectures. Our comprehensive experimental evaluations affirm the effectiveness of our method, showcasing its superior performance against both condition-independent and condition-dependent DA/DG approaches. This advancement signifies a substantial breakthrough in domain generalization, offering improved adaptability and performance for autonomous driving systems across a variety of environments.

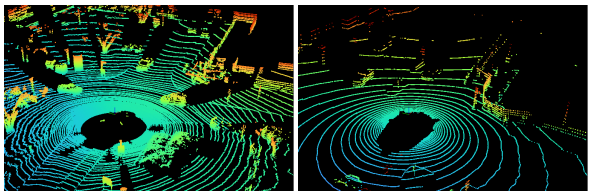
The contributions of our work are summarized as follows:

- We propose a density-aware sparse convolution module that innovatively utilizes the occupancy pattern of 3D sparse convolution kernels as additional geometric features, thereby enhancing the network's ability to perceive local density variations.
- We introduce a self-supervised auxiliary task focused on density prediction, further augmenting the model's ability to recognize variations in point cloud density.
- Our approach demonstrates superior performance over the current state-of-the-art domain generalization methods without relying on any prerequisites.

II. Related Work

A. LiDAR Data Processing

LiDAR point clouds, characterized by their unstructured, irregular, and unordered nature, pose significant challenges for direct convolution operations, leading to the development of three primary approaches for thier processing: Point-based [11], [12], [13], Projection-based [14], [15], [16], [17], [18], [19], and Voxel-based [20], [21], [22], [23] methods. Point-based techniques directly process LiDAR point clouds in their raw form, preserving the intricate details and spatial relationships. However, they are computationally intensive and demand high resources. Projection-based methods project 3D point clouds onto 2D planes, enabling efficient processing. Despite their efficiency, these methods can distort the three-dimensional receptive field due to the limitations of 2D convolution in accurately representing 3D spaces. Voxel-based methods utilize 3D convolution to maintain the integrity of the three-dimensional receptive field, with sparse convolution techniques [24], [25] mitigating the computational load for feasible real-time operations. To handle irregular density distributions in point clouds, PointConv [26] utilizes point cloud density to re-weight the convolution kernel to compensate for uneven sampling. Li et al. [27] propose a density-aware convolution module that re-weights convolution kernels based on point density.



SemanticKITTI nuScenes

Fig. 2. SemanticKITTI (HDL-64E) has a narrower FOV than nuScenes (HDL-32E) and also has four times the number of points, resulting in a significantly higher density.

B. LiDAR Domain Adaptation

Semantic segmentation models achieve satisfactory performance within the same domain but face performance degradation due to the domain gap when applied to new environments or sensor configurations. To mitigate this without extensive re-labeling or training, various Domain Adaptation (DA) strategies [28], [29], [30], [31] have been explored: Yi *et al.* [4] address the variance in point cloud density caused by differences in LiDAR sensors by transforming point clouds to a canonical domain using consecutive frames. Rochan *et al.* [5] introduce an unsupervised domain adaptation approach based on range views, which aligns beam positions across training and target datasets to maintain spatial consistency in the sensor data, facilitating model adaptation. LiDAR-UDA [7] utilizes LiDAR beam subsampling to simulate various LiDAR sensors, employing cross-frame ensembling and a Learned Aggregation Model (LAM) to generate more accurate pseudo labels. While these DA techniques provide satisfactory performance, they are limited by the need for target data statistics and require time-consuming fine-tuning.

C. LiDAR Domain Generalization

Domain Generalization (DG) approaches aim to overcome the limitations of DA by preparing models to perform well across unseen domains without requiring access to target data or undergoing fine-tuning. DGLSS [8] utilizes Sparsity Invariant Feature Consistency and Semantic Correlation Consistency to prevent performance degradation from density and scene distribution variations, respectively. LiDomAug [9] uses ego-motion and sequential data for data completion and generating new data forms through random LiDAR sensor configurations. BEV-DG [32] proposes Density-maintained Vector Modeling for domain-invariant feature extraction using bird's-eye view representations. LiDOG [33] enhances generalization performance by joint training in 3D and 2D birds-eye-view, bypassing the 3D density discrepancy with lower dimensions. Sanchez *et al.* [10] propose a strategy for improving generalization through label propagation and multi-frame aggregation with known ego motion. However, its use of the computationally heavy KPConv [11] limits real-time applications. Our work contributes to this field by presenting a novel DG method that enhances model adaptability to density variations in point clouds without relying on specific conditions like ego-motion or sequential data.

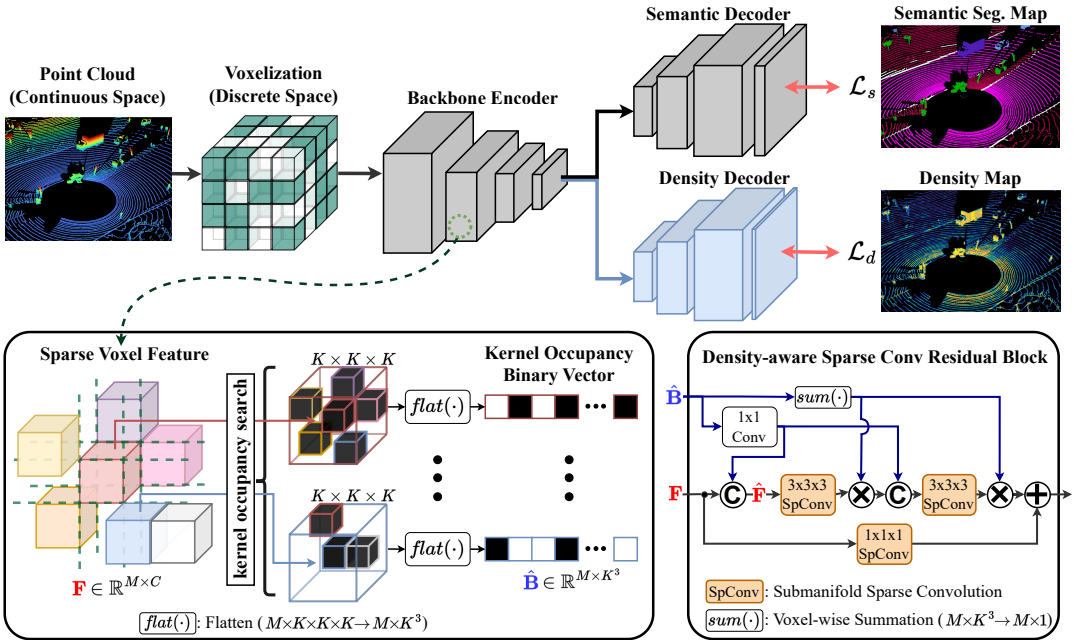


Fig. 3. Overall framework and density-aware sparse convolution method. The kernel occupancy binary vector is computed by checking for the presence of voxels in the kernel area, $K \times K \times K$ region centered on each voxel.

III. Proposed Method

In this section, we present a density-aware domain generalization method illustrated in Fig. 3. First, we detail novel density-aware sparse convolution, focusing on adapting sparse convolution to leverage density information effectively in Section III-A. We then describe a self-supervised density prediction task designed to enhance model generalization in Section III-B. Lastly, we outline the loss functions used in training our framework in Section III-C.

A. Density-aware sparse convolution

1) Preliminary: sparse tensor and sparse convolution: LiDAR sensors produce sparse point cloud data by emitting rays and generating points upon contact, leaving most space empty. Traditional dense convolution on this data demands substantial memory and computational resources. To mitigate this, we adopt sparse voxel-based approaches, transforming sparse point cloud data into a sparse tensor representation, thereby enabling efficient processing through sparse convolution techniques [24], [25]. The construction of a sparse tensor $\mathbf{T}$, comprising M voxels, is defined as follows:

$$\mathbf{T} = \{(\mathbf{p}_i, \mathbf{f}_i) \mid \mathbf{p}_i \in \mathbf{P}, \mathbf{f}_i \in \mathbf{F}\}, \tag{1}$$

where $\mathbf{p}_i$ and $\mathbf{f}_i$ represent the position and feature vector of each voxel, respectively. The sets $\mathbf{P}$ and $\mathbf{F}$ consist of 3D coordinates and C-dimensional feature vectors for each voxel:

$$\begin{aligned} \mathbf{P} &= \{\mathbf{p}_i \in \mathbb{R}^3 \mid i = 1, ..., M\}, \\ \mathbf{F} &= \{\mathbf{f}_i \in \mathbb{R}^C \mid i = 1, ..., M\}. \end{aligned} \tag{2}$$

Sparse convolution operations are then applied to $\mathbf{T}$ for efficient data processing, focusing solely on non-empty voxels. To ensure computational efficiency, we employ submanifold sparse convolution [25] that maintains the output indices identical to the input indices.

2) Local occupancy embedding: Sparse convolution performs weight multiplication and summation only for the occupied parts of the kernel, but it does not directly employ the occupancy status within the kernel as a feature. This means that the local density information is not directly accessible during the convolution process. To bridge this gap, we introduce a method that embeds kernel occupancy information within the sparse convolution operation to make it density-aware. Occupancy information for voxels in a sparse tensor $\mathbf{T} = (\mathbf{P}, \mathbf{F})$ is encoded by conducting a kernel occupancy search within the kernel area $\mathbf{D}_K$, defined as:

$$\mathbf{D}_K = \left\{ (x, y, z) \mid x, y, z \in \left[-\frac{K-1}{2}, \frac{K+1}{2} \right] \cap \mathbb{Z} \right\}. \tag{3}$$

This approach allows us to compute the kernel occupancy binary vector $\mathbf{B}_\mathbf{T} \in \mathbb{R}^{M \times K \times K \times K}$, representing the presence or absence of voxels within the kernel's vicinity, as follows:

$$\mathbf{B}_\mathbf{T}[\mathbf{p}_i, \mathbf{d}_j] = \begin{cases} 1 & \text{if } (\mathbf{p}_i - \mathbf{d}_j) \in \mathbf{P} \\ 0 & \text{otherwise} \end{cases}, \text{ where } \mathbf{d}_j \in \mathbf{D}_K. \tag{4}$$

Finally, the occupancy vector $\mathbf{B}_\mathbf{T}$ is flattened to $\hat{\mathbf{B}}_\mathbf{T} \in \mathbb{R}^{M \times K^3}$ for further processing. Through embedding local occupancy information, our method enhances the density-awareness of sparse convolution, potentially improving the processing of sparse point cloud data.

3) Sparse convolution residual block: The kernel occupancy binary vector encodes critical information about sparsity and the geometric shape from the occupancy data. To effectively leverage this information alongside traditional sparse convolution operation, we integrate the occupancy

information via a 1×1 convolution, referred to as $\text{Conv}_{1\times1}$, generating kernel occupancy features. These features are then concatenated with the original input features $\mathbf{F}$ to enrich them, as follows:

$$\hat{\mathbf{F}} = [\mathbf{F};\ \text{Conv}_{1\times1}(\hat{\mathbf{B}}_\mathbf{T})]. \tag{5}$$

This process enhances the input features by adding spatially relevant occupancy information, providing a more comprehensive input for the subsequent convolution steps.

With the enhanced sparse tensor $\hat{\mathbf{T}} = (\mathbf{P}, \hat{\mathbf{F}})$ prepared, sparse convolution is applied to yield the output sparse tensor $\mathbf{T}^{out} = (\mathbf{P}^{out}, \mathbf{F}^{out})$. To mitigate the imbalance in sparse kernel weights for the feature $\mathbf{f}_i^{out} \in \mathbf{F}^{out}$, we normalize these features by the count of contributing input tensors, thereby adjusting for the variance in kernel occupancy:

$$\hat{\mathbf{f}}_i^{out} = \mathbf{f}_i^{out} / \sum_{\mathbf{d}_j \in \mathbf{D}_K} (\mathbf{B}_\mathbf{T}[\mathbf{p}_i, \mathbf{d}_j]). \tag{6}$$

In the design of networks utilizing sparse convolution, it is common practice to compute sparse convolution layers multiple times within the same spatial area, with changes to the spatial size between layers being infrequent. Based on this observation, we propose a strategy to compute the kernel occupancy binary vector and its embedding just once per module block, instead of after each sparse convolution operation. This methodology, as depicted in the bottom right of Fig. 3, streamlines the integration of occupancy information into the convolution process, significantly enhancing the efficiency of our density-aware sparse convolution framework.

B. Self-supervised density estimation

To train a model that is both density-aware and capable of capturing a generalized representation, we introduce an auxiliary task focused on predicting the local density at a voxel level. Our framework is architecturally designed with a shared encoder Φ_{enc}, which processes input features $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_M] \in \mathbb{R}^{M\times3}$ into a latent representation $\mathbf{Z} \in \mathbb{R}^{m\times\text{feat}}$. Here, M represents the voxel count prior to encoding, and m denotes the voxel count post-encoding. This latent representation is then utilized by two distinct decoders: one for semantic segmentation $\Phi^{\mathcal{S}}_{\text{dec}}$ and the other for density estimation $\Phi^{\mathcal{D}}_{\text{dec}}$. The operations of these components are formalized as follows:

$$\mathbf{Z} = \Phi_{\text{enc}}(\mathbf{V}),\ \mathcal{S}^{\text{pred}} = \Phi^{\mathcal{S}}_{\text{dec}}(\mathbf{Z}),\ \mathcal{D}^{\text{pred}} = \Phi^{\mathcal{D}}_{\text{dec}}(\mathbf{Z}), \tag{7}$$

where $\mathcal{S}^{\text{pred}} \in \mathbb{R}^{M\times\text{class}}$ and $\mathcal{D}^{\text{pred}} \in \mathbb{R}^{M\times1}$ represent semantic segmentation and density prediction, respectively. This dual-decoder approach facilitates an exhaustive understanding and representation of scenes, presenting a robust framework for self-supervised learning in density estimation and semantic segmentation tasks.

Self-supervised label generation. In addressing the challenge of generating voxel-wise density labels, we introduce a self-supervised method utilizing k-Nearest Neighbor (k-NN) search, which leverages the distance from the nearest k neighbor points for each point, as shown in Fig. 4. First, neighboring voxels are searched in voxel space with $k = 10$.

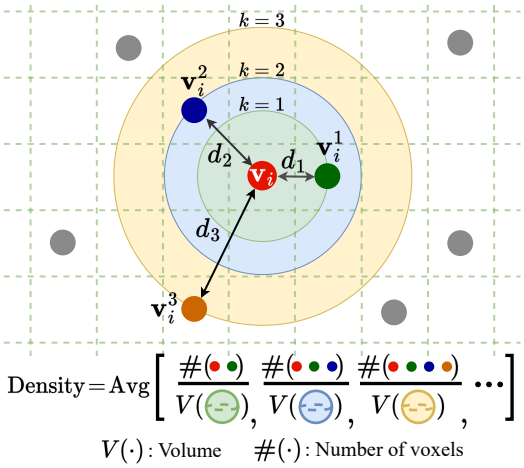


Fig. 4. Density calculation based on the k-Nearest Neighbors (k-NN) algorithm. Density is defined as the number of points within a sphere.

As our framework operates on a voxel-based 3D network, k-NN is applied to the voxelized input $\mathbf{V} \in \mathbb{R}^{M\times3}$ in the following manner:

$$\{\mathbf{v}_i^1, \mathbf{v}_i^2, \ldots, \mathbf{v}_i^k\} = \text{k-NN}(\mathbf{V}, k),\ \forall i \in \{1, \ldots, M\}. \tag{8}$$

By employing the Euclidean distance to neighboring voxels from each voxel, we compute the inter-voxel density. The concept of density, typically defined as mass per volume, is analogously applied to measuring the concentration of voxels within a given volume. Specifically, density ρ is proportional to the number of occupied voxels n within the volume of a sphere with radius r, expressed as $\rho \propto n/r^3$. This approach allows for the estimation of density based on voxel count within a specified volume.

To mitigate noise in the density labels, we compute the density for each number of adjacent voxels ranging from 1 to k, subsequently averaging these calculations. This smoothing technique ensures a more reliable density estimation. Furthermore, to normalize the variance in scale, a logarithmic transformation is applied to the density values. Consequently, the density label $\mathcal{D}_i^{\text{label}}$ for each voxel i is determined through the following refined process, which encapsulates the steps from neighbor searching to logarithmic adjustment for scale variance as follows:

$$\mathcal{D}_i^{\text{label}} = \log\left(1 + \frac{1}{k}\sum_{j=1}^{k} \frac{j+1}{\left\|\mathbf{v}_i - \mathbf{v}_i^j\right\|_2^3}\right),\ \forall i \in \{1, \ldots, M\}. \tag{9}$$

C. Loss function

For the prediction of inter-voxel density, we employ the Smooth-L1 loss. For semantic segmentation, our model utilizes a dual-loss approach, incorporating both Cross-Entropy loss and Lovasz-Softmax loss [36] as follows:

$$\begin{aligned} \mathcal{L}_{Density} &= \text{Smooth}_{L_1}(\mathcal{D}^{\text{pred}} - \mathcal{D}^{\text{label}}), \\ \mathcal{L}_{Semantic} &= \text{CE}(\mathcal{S}^{\text{pred}}, \mathcal{S}^{\text{label}}) + \text{Lovasz}(\mathcal{S}^{\text{pred}}, \mathcal{S}^{\text{label}}). \end{aligned} \tag{10}$$

TABLE I

COMPARISON OF DOMAIN GENERALIZATION PERFORMANCE USING WAYMO, SEMANTICKITTI, AND NUSCENES DATASETS. † DENOTES THE DOMAIN ADAPTATION SCHEME. BOLD FONT INDICATES THE HIGHEST PERFORMANCE, WHILE UNDERLINING DENOTES THE SECOND HIGHEST PERFORMANCE.

Method	W→W	W→K	W→N	K→K	K→W	K→N	N→N	N→W	N→K
Base	<u>75.37</u>	49.40	47.83	57.31	35.24	37.42	<u>65.78</u>	38.65	36.24
IBN-Net [34]	**75.47**	51.13	44.72	57.74	36.99	38.74	65.31	36.53	36.93
MLDG [35]	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	<u>49.96</u>	-	-	-	-	-	-
COSMIX (N) [30]†	65.68	40.99	47.98	49.98	38.05	43.25	-	-	-
DGLSS [8]	75.28	<u>51.23</u>	49.61	**59.62**	<u>40.67</u>	<u>44.83</u>	65.32	<u>40.93</u>	<u>38.98</u>
Ours	74.61	**53.87**	**52.98**	<u>58.23</u>	**42.78**	**46.92**	**67.64**	**45.29**	**40.09**

The aggregated loss function utilized during training, which combines the contributions from both the semantic segmentation and density prediction tasks, is defined as:

$$\mathcal{L}_{Total} = \lambda_1 \cdot \mathcal{L}_{Semantic} + \lambda_2 \cdot \mathcal{L}_{Density}, \tag{11}$$

where λ_1 and λ_2 are weighting coefficients that balance the relative importance of the semantic segmentation and density prediction losses, respectively. Through empirical evaluation, we have determined the optimal values for these coefficients to be $\lambda_1 = 1$ and $\lambda_2 = 10$, ensuring a balanced contribution to the total loss and thereby optimizing our model's performance across both tasks.

IV. EXPERIMENTS

This section outlines the comprehensive experimentation conducted to validate the effectiveness of our method. We detail the model implementation and experimental setups in Section IV-A. We compare our method against existing condition-free domain generalization techniques in Section IV-B. We evaluate the proposed method alongside condition-required domain adaptation and generalization methods in Section IV-C. Subsequently, Section IV-D is dedicated to dissecting the influence of each component within our framework. Throughout these experiments, we adhere to the class map settings established in prior studies to ensure our evaluations are consistent with recognized benchmarks. Additionally, Section IV-E investigates the computational demands of our approach.

A. Implementation details

Our evaluation encompasses datasets such as the Waymo (64-ch) [1], SemanticKITTI (64-ch) [3], and nuScenes (32-ch) [2] datasets. Since each dataset has its own class categorization, we adopt the label unification protocol from previous research [4], [5], [8] to consolidate them. We utilize the mean Intersection over Union (mIoU) for our primary quantitative evaluation. Voxelization is performed with a voxel size of 20cm, and our training scheme includes a learning rate scheduler that decreases the rate by 0.99 every epoch, beginning at 1e-3. With a batch size of 8 and the Adam optimizer for training, we further enhance the model with various augmentation techniques such as rotation, scale adjustment, translation, and beam sampling, as per the strategies in [8].

B. Comparison to condition-free DG method

Our method stands out by functioning without the specific conditions often required by many Domain Adaptation (DA) and Domain Generalization (DG) techniques. To demonstrate its efficacy, we compare it with DGLSS [8], a leading condition-free DG method. DGLSS employs the Waymo, SemanticKITTI, and nuScenes datasets for training, focusing on a common set of 11 classes across these platforms. We adopt MinkowskiNet [21] as the backbone network, just as with DGLSS, for a fair comparison.

The comparative results in Table I, spanning a total of 9 scenarios, involve using each of the three datasets—Waymo (W), SemanticKITTI (K), nuScenes (N)—as the source data, with the other two serving as unseen data. In the source-to-source scenarios (W→W, K→K, N→N), the proposed method exhibits comparable performance to DGLSS. However, in the six scenarios testing on unseen datasets (W→K, W→N, K→W, K→N, N→W, N→K), the proposed method achieves an average performance improvement of 18.38% over the base model and significantly outperforms the state-of-the-art domain generalization method, DGLSS by a margin of 5.88%.

Further analysis, including class-wise performance on the SemanticKITTI dataset after training with it as the source, is presented in Table II. While our method shows a slight decline of 2.33% in source-to-source scenarios (K→K) compared to DGLSS, it registers substantial gains on unseen datasets—nuScenes and Waymo—with improvements of 4.66% and 5.19%, respectively. More impressively, our method surpasses DGLSS across all key classes crucial for autonomous driving environments, such as 'Car', 'Pedestrian', 'Drivable surface', and 'Sidewalk', across the three datasets. These findings underscore our method's resilience to domain shifts and its superior generalization capabilities.

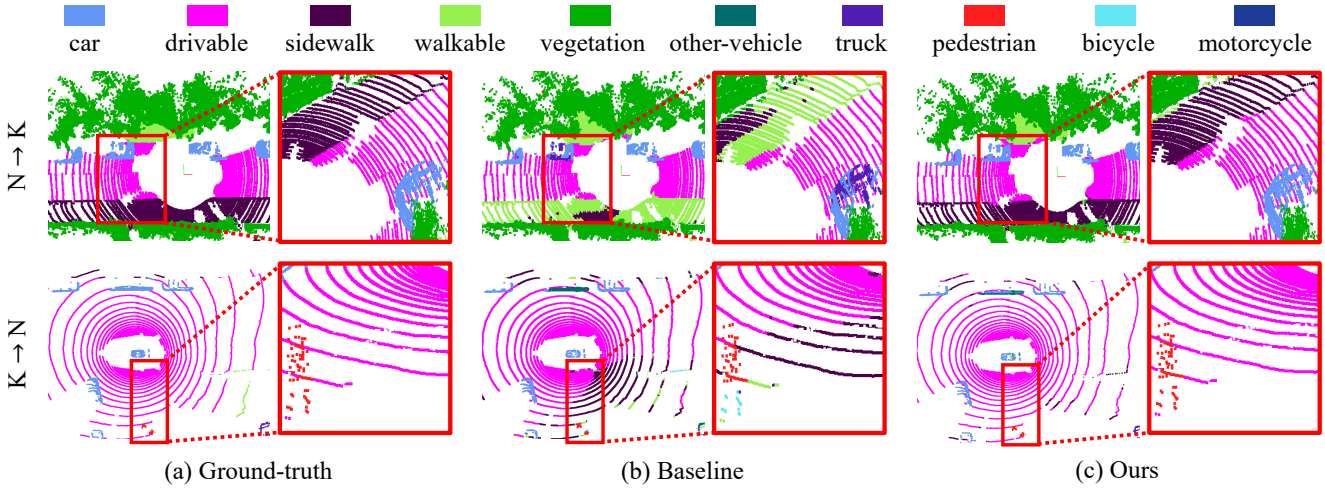
C. Comparison to condition-required DG/DA methods

Our method excels even without the need for specific conditions such as ego-motion or sequential data, showcasing its effectiveness against DA/DG methods that rely on these prerequisites. The comparison results in Table III, using the

TABLE II

QUANTITATIVE COMPARISONS BY CLASS WHEN USING SEMANTICKITTI AS THE SOURCE. BOLD FONT INDICATES THE BEST PERFORMANCE.

Scenario	Method	Car	Bicycle	Motor-cycle	Truck	Other vehicle	Pedes-trian	Drivable surface	Side-walk	Walk-able	Vege-tation	mIoU
K→K	Base	91.23	10.04	**35.69**	52.89	37.95	40.99	83.86	62.78	66.34	**91.33**	57.31
	DGLSS [8]	92.76	11.99	27.09	**72.50**	**45.95**	36.39	84.76	65.64	67.98	91.28	**59.62**
	Ours	**92.99**	**14.20**	33.68	33.84	37.02	**52.98**	**87.32**	**67.91**	**70.45**	91.90	58.23
K→N	Base	68.91	**2.51**	12.18	11.30	20.35	29.47	80.17	31.91	40.19	77.18	37.42
	DGLSS [8]	76.36	1.51	**35.18**	26.47	25.49	37.09	82.03	38.12	44.20	81.79	44.83
	Ours	**80.47**	1.27	19.45	**33.74**	**27.16**	**49.87**	**85.85**	**42.37**	**46.54**	**82.51**	**46.92**
K→W	Base	72.12	2.52	4.52	7.77	13.36	40.86	64.92	30.12	34.84	81.40	35.24
	DGLSS [8]	82.26	**4.85**	**9.72**	16.80	**17.67**	52.55	68.20	35.91	33.33	85.41	40.67
	Ours	**86.65**	4.06	8.22	**21.58**	16.39	**58.15**	**72.74**	**35.97**	**36.71**	**87.37**	**42.78**



(a) Ground-truth (b) Baseline (c) Ours

Fig. 5. Qualitative comparison when using the C&L semantic network as backbone. The top row represents results trained on nuScenes and tested on SemanticKITTI (N→K), while the bottom row represents results trained on SemanticKITTI and tested on nuScenes (K→N).

TABLE III

QUANTITATIVE COMPARISONS WHEN USING THE COMPLETE & LABEL BACKBONE. BOLD FONT INDICATES THE HIGHEST PERFORMANCE, WHILE UNDERLINING DENOTES THE SECOND HIGHEST PERFORMANCE.

Backbone	Methods	K→N	N→K
C&L [4]	Base	27.9	23.5
	SWD [37]	27.7	24.5
	3DGCA [28]	27.4	23.9
	C&L [4]	31.6	33.7
	LiDAR-UDA [7]	<u>41.8</u>	34.0
	LiDomAug [9]	39.2	<u>37.9</u>
	Ours	**43.1**	**39.0**

semantic network from Complete & Label [4] as the backbone, highlight our method's performance against various condition-required DA/DG approaches. Following the class types and experimental setups of LiDAR-UDA [7] and LiDomAug [9], we conduct tests using the SemanticKITTI and nuScenes datasets.

The outcomes demonstrate that our method surpasses the state-of-the-art domain adaptation technique LiDAR-UDA [7], achieving a performance boost of 3.11% on the nuScenes dataset when trained with SemanticKITTI (K→N) and a significant 14.71% improvement in the reverse scenario (N→K). This is noteworthy, especially considering LiDAR-UDA's reliance on multi-frame data for DA, whereas our approach, a condition-free DG method, attains superior results. Furthermore, our method also shows a 9.95% performance increase against LiDomAug [9] in the (K→N) scenario and a 2.90% enhancement in the opposite direction, underlining its robustness and adaptability across different domain settings. Qualitative results are illustrated in Fig. 5.

Table IV shows the comparison of the proposed method with the range view based domain adaptation methods. Our experiments utilize MinkowskiNet [21] as the 3D backbone network, highlighting the inherent advantage of the voxel-based 3D network in achieving superior generalization performance over 2D rangeview-based alternatives. With MinkowskiNet as the backbone, the improvements are 6.90% and 28.62% for the (K→N) and (N→K) scenarios, respectively. These findings underscore the efficacy of our approach, demonstrating its potential for application in diverse settings without the dependency on specific conditions for training or testing phases.

D. Ablation studies

We conduct ablation studies to further validate the efficacy of each component of our method. The ablation experiments in

TABLE IV

QUANTITATIVE COMPARISON WITH RANGE-VIEW BASED DA METHODS. BOLD FONT INDICATES THE HIGHEST PERFORMANCE, WHILE UNDERLINING DENOTES THE SECOND HIGHEST PERFORMANCE.

Backbone	Method	K→N	N→K
SalsaNext [14]	Base	20.1	12.6
	CORAL [38]	33.3	23.2
	MEnt [39]	33.1	17.1
	AEnt [39]	30.4	18.3
	(M+A)Ent [39]	32.0	22.8
	SWD [37]	30.1	18.1
	Rochan *et al.* [5]	34.5	23.5
MinkNet [21]	Base	<u>40.6</u>	<u>31.8</u>
	Ours	**43.4**	**40.9**

TABLE V

ABLATION STUDY ON THE PROPOSED DENSITY-AWARE SPARSE CONVOLUTION (DASC) METHOD AND THE DENSITY PREDICTION AUXILIARY TASK.

Backbone	DASC	Density pred	K→N	N→K
C&L [4]			35.9	31.6
	✓		<u>41.3</u>	<u>37.4</u>
		✓	39.5	34.4
	✓	✓	**43.1**	**39.0**

Table V, using C&L [4] as the backbone network, specifically assess the contributions of Density-Aware Sparse Convolution (DASC) and the auxiliary density prediction task. For a fair comparison, the results of the baseline are reproduced using the beam augmentation [8], along with various schemes, including rotation and scale augmentation.

DASC alone accounts for a substantial performance uplift of 15.04% in the SemanticKITTI to nuScenes (K→N) scenario and an even more significant 18.35% in the reverse nuScenes to SemanticKITTI (N→K) setting, compared to the baseline. Applying the auxiliary density prediction task results in a performance increase of 10.03% for (K→N) and 8.86% for (N→K). The synergy of DASC with the auxiliary density prediction task culminates in even more pronounced improvements, pushing performance gains to 20.06% for (K→N) and 23.42% for (N→K) over the baseline.

Notably, even though our method is density-aware, it shows additional performance gains when using Mix3D.

Moreover, we explore the integration of our method with established augmentation techniques such as Mix3D [40] and PolarMix [41] as presented in Table VI. Notably, even though our method is density-aware, it shows additional performance improvements when using Mix3D. Mix3D enhances performance by expanding the range of densities encountered during training, allowing our model to handle unseen density regions more effectively.

TABLE VI

QUANTITATIVE EVALUATION OF AUGMENTATION APPLICATION.

Backbone	Methods	K→N	N→K
C&L [4]	Ours	43.1	39.0
	Ours+Mix3D [40]	**45.7**	41.2
	Ours+PolarMix [41]	44.8	**41.5**

E. Computational costs

Autonomous vehicles require real-time processing capabilities, making factors like usage pivotal, especially given the challenge of incorporating high-performance desktop GPUs into such compact systems. The proposed Density-Aware Sparse Convolution (DASC) comprises simple 1×1 convolutions, ensuring the added computational overhead is kept to a minimum. On an NVIDIA RTX A6000, the MinkowskiNet base model exhibits a computational speed of 33 ms per frame, and adding DASC increases the computational load by 7 ms, totaling 40 ms of operation time in the SemanticKITTI. The density task operates only during the training phase; thus, it does not affect the computation speed or memory during the test phase. In an RTX A6000 setup, the MinkowskiNet base model takes 11 minutes per epoch, whereas incorporating the density task extends this duration by merely 4 minutes per epoch. Given the substantial enhancements, our method brings to domain generalization performance, such a minor increase in computational demands is considered inconsequential.

V. CONCLUSION

In this paper, we propose a novel approach aimed at enhancing the robustness of 3D networks against domain shifts, particularly addressing the challenge of varying point cloud densities. First, we propose a density-aware sparse convolution module that leverages additional geometric features derived from the kernel occupancy information in submanifold sparse convolution. In addition, we introduce a voxel-wise density prediction as an auxiliary task to enhance density-aware feature extraction. These strategies collaboratively endow any 3D backbone network with density awareness, thereby significantly boosting domain generalization capabilities. Our extensive experimental evaluations demonstrate that our method not only excels in condition-free domain generalization scenarios but also outperforms existing domain adaptation and domain generalization methods that depend on specific prerequisites. In the future, we plan to research on multi-source domain generalization that can be effectively used even in the presence of multiple source data.

REFERENCES

[1] P. Sun *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2446–2454.

[2] H. Caesar *et al.*, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 621–11 631.

[3] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 9297–9307.

[4] L. Yi, B. Gong, and T. Funkhouser, "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)*, 2021, pp. 15 363–15 373.

[5] M. Rochan, S. Aich, E. R. Corral-Soto, A. Nabatchian, and B. Liu, "Unsupervised domain adaptation in lidar semantic segmentation with self-supervision and gated adapters," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2649–2655.

[6] L. Kong, N. Quader, and V. E. Liong, "Conda: Unsupervised domain adaptation for lidar segmentation via regularized domain concatenation," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9338–9345.

[7] A. Shaban, J. Lee, S. Jung, X. Meng, and B. Boots, "Lidar-uda: Self-ensembling through time for unsupervised lidar domain adaptation," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2023, pp. 19 784–19 794.

[8] H. Kim, Y. Kang, C. Oh, and K.-J. Yoon, "Single domain generalization for lidar semantic segmentation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 17 587–17 598.

[9] K. Ryu, S. Hwang, and J. Park, "Instant domain augmentation for lidar semantic segmentation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 9350–9360.

[10] J. Sanchez, J.-E. Deschaud, and F. Goulette, "Domain generalization of 3d semantic segmentation in autonomous driving," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2023, pp. 18 077–18 087.

[11] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 6411–6420.

[12] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Randla-net: Efficient semantic segmentation of large-scale point clouds," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 108–11 117.

[13] S. Fan, Q. Dong, F. Zhu, Y. Lv, P. Ye, and F.-Y. Wang, "Scf-net: Learning spatial contextual features for large-scale point cloud segmentation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 14 504–14 513.

[14] T. Cortinhal, G. Tzelepis, and E. Erdal Aksoy, "Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds," in *Advances in Visual Computing: 15th International Symposium, ISVC 2020, San Diego, CA, USA, October 5–7, 2020, Proceedings, Part II 15*. Springer, 2020, pp. 207–222.

[15] Y. Zhang, Z. Zhou, P. David, X. Yue, Z. Xi, B. Gong, and H. Foroosh, "Polarnet: An improved grid representation for online lidar point clouds semantic segmentation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9601–9610.

[16] K. Peng, J. Fei, K. Yang, A. Roitberg, J. Zhang, F. Bieder, P. Heidenreich, C. Stiller, and R. Stiefelhagen, "Mass: Multi-attentional semantic segmentation of lidar data for dense top-view understanding," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15 824–15 840, 2022.

[17] T.-H. Chen and T. S. Chang, "Rangeseg: Range-aware real time segmentation of 3d lidar point clouds," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 1, pp. 93–101, 2022.

[18] A. Ando, S. Gidaris, A. Bursuc, G. Puy, A. Boulch, and R. Marlet, "Rangevit: Towards vision transformers for 3d semantic segmentation in autonomous driving," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 5240–5250.

[19] H.-X. Cheng, X.-F. Han, and G.-Q. Xiao, "Transrvnet: Lidar semantic segmentation with transformer," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 6, pp. 5895–5907, 2023.

[20] B. Graham, M. Engelcke, and L. Van Der Maaten, "3d semantic segmentation with submanifold sparse convolutional networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 9224–9232.

[21] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3075–3084.

[22] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin, "Cylindrical and asymmetrical 3d convolution networks for lidar segmentation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 9939–9948.

[23] Y. Hou, X. Zhu, Y. Ma, C. C. Loy, and Y. Li, "Point-to-voxel knowledge distillation for lidar semantic segmentation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 8479–8488.

[24] B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Pensky, "Sparse convolutional neural networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 806–814.

[25] B. Graham and L. Van der Maaten, "Submanifold sparse convolutional networks," *arXiv preprint arXiv:1706.01307*, 2017.

[26] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2019, pp. 9621–9630.

[27] X. Li, L. Wang, M. Wang, C. Wen, and Y. Fang, "Dance-net: Density-aware convolution networks with context encoding for airborne lidar point cloud classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 166, pp. 128–139, 2020.

[28] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer, "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)*, 2019, pp. 4376–4382.

[29] C. Saltori, S. Lathuiliére, N. Sebe, E. Ricci, and F. Galasso, "Sf-uda 3d: Source-free unsupervised domain adaptation for lidar-based 3d object detection," in *International Conference on 3D Vision (3DV)*, 2020, pp. 771–780.

[30] C. Saltori, F. Galasso, G. Fiameni, N. Sebe, E. Ricci, and F. Poiesi, "Cosmix: Compositional semantic mix for domain adaptation in 3d lidar segmentation," in *Proceedings of European Conference on Computer Vision (ECCV)*. Springer, 2022, pp. 586–602.

[31] Z. Yuan, C. Wen, M. Cheng, Y. Su, W. Liu, S. Yu, and C. Wang, "Category-level adversaries for outdoor lidar point clouds cross-domain semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 1982–1993, 2022.

[32] M. Li, Y. Zhang, X. Ma, Y. Qu, and Y. Fu, "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)*, 2023, pp. 11 632–11 642.

[33] C. Saltori, A. Osep, E. Ricci, and L. Leal-Taixé, "Walking your lidog: A journey through multiple domains for lidar semantic segmentation," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2023, pp. 196–206.

[34] X. Pan, P. Luo, J. Shi, and X. Tang, "Two at once: Enhancing learning and generalization capacities via ibn-net," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2018, pp. 464–479.

[35] D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[36] M. Berman, A. R. Triki, and M. B. Blaschko, "The lovz-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)*, 2018, pp. 4413–4421.

[37] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht, "Sliced wasserstein discrepancy for unsupervised domain adaptation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10 285–10 295.

[38] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*. Springer, 2016, pp. 443–450.

[39] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2517–2526.

[40] A. Nekrasov, J. Schult, O. Litany, B. Leibe, and F. Engelmann, "Mix3d: Out-of-context data augmentation for 3d scenes," in *International Conference on 3D Vision (3DV)*, 2021, pp. 116–125.

[41] A. Xiao, J. Huang, D. Guan, K. Cui, S. Lu, and L. Shao, "Polarmix: A general data augmentation technique for lidar point clouds," in *Neural Information Processing Systems (NeurIPS)*, 2022, pp. 11 035–11 048.