This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

BEV-DG: Cross-Modal Learning under Bird's-Eye View for Domain Generalization of 3D Semantic Segmentation

Miaoyu Li[1], Yachao Zhang[2 ‡], Xu Ma[3], Yanyun Qu[1 ‡], Yun Fu[3]
[1]School of Informatics, Xiamen University
[2] Tsinghua Shenzhen International Graduate School, Tsinghua University
[3] Department of ECE, Northeastern University
limiaoyu@stu.xmu.edu.cn, yachaozhang@sz.tsinghua.edu.cn, yyqu@xmu.edu.cn

Abstract

Cross-modal Unsupervised Domain Adaptation aims to exploit the complementarity of 2D-3D data to overcome the lack of annotation in an unknown domain. However, the training of these methods relies on access to target samples, meaning the trained model only works in a specific target domain. In light of this, we propose cross-modal learning under bird's-eye view for Domain Generalization (DG) of 3D semantic segmentation, called BEV-DG. DG is more challenging because the model cannot access the target domain during training, meaning it needs to rely on cross-modal learning to alleviate the domain gap. Since 3D semantic segmentation requires the classification of each point, existing cross-modal learning is directly conducted point-to-point, which is sensitive to the misalignment in projections between pixels and points. To this end, our approach aims to optimize domain-irrelevant representation modeling with the aid of cross-modal learning under bird's-eye view. We propose BEV-based Area-to-area Fusion (BAF) to conduct cross-modal learning under bird's-eye view, which has a higher fault tolerance for point-level misalignment. Furthermore, to model domain-irrelevant representations, we propose BEV-driven Domain Contrastive Learning (BDCL) with the help of cross-modal learning under bird's-eye view. We design three domain generalization settings based on three 3D datasets, and BEV-DG significantly outperforms state-of-the-art competitors with tremendous margins in all settings.

1. Introduction

Semantic segmentation of LiDAR point clouds is fundamental for numerous vision applications, such as robotics, autonomous driving and virtual reality. Given a LiDAR frame, the goal is to separate each point in the point cloud into a cluster with corresponding semantic labels.

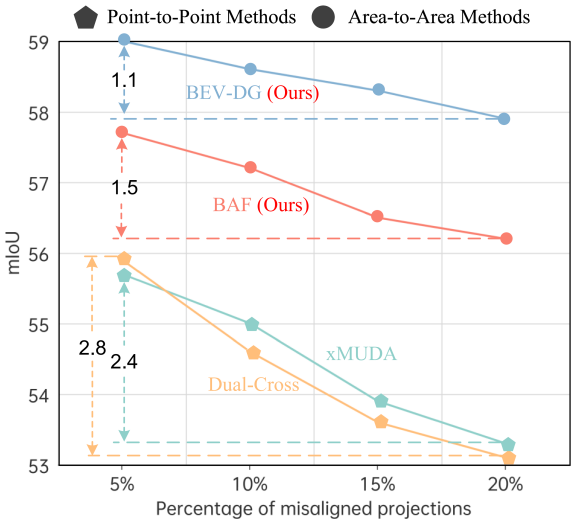

Figure 1. DG results of methods under different levels of point-to-pixel misalignment. The models are trained on A2D2 and SemanticKITTI datasets, and tested on nuScenes dataset. Area-to-area methods significantly outperform point-to-point methods under each level of misalignment. Moreover, point-to-point methods degrade more dramatically with the increasing misalignment.

Recently, some 3D semantic segmentation approaches continuously refresh the performance leader-boards on several benchmark datasets [2, 3, 4, 7, 9]. Nevertheless, the training and testing data for these approaches originate from identical datasets (domains). As each dataset has a different configuration of LiDAR sensors, these methods can significantly degrade under domain shift. Specifically, due to the number of laser beams varying from LiDAR to LiDAR, the obtained point cloud is also quite diverse in terms of density (resolution), which results in a tremendous domain gap. To improve the generalization of the model,

‡Corresponding Author

some Unsupervised Domain Adaptation (UDA) methods [14, 21, 25, 34, 36] are proposed for point cloud semantic segmentation in a single-modal or cross-modal manner. However, the training of these UDA methods relies on the target domain data, which makes them only generalize well to a specific target domain.

To this end, we are focused on investigating Domain Generalization (DG) for 3D semantic segmentation. Compared to UDA, DG is more challenging as it can not access the target domain during training, and the model should generalize well to an unseen target domain. Currently, many cross-modal UDA methods [14, 21, 25, 36] are proposed for 3D semantic segmentation. To mitigate the negative effect of domain shift, they utilize cross-modal learning to prompt information interaction between two modalities (image and point cloud). The mechanism behind this idea is that if one modality is sensitive to one type of shift while the other is robust, and the robust modality can guide the sensitive modality. In light of this, we solve the DG problem using cross-modal learning on multi-modal data.

However, current cross-modal learning achieves cross-modal matching through a point-to-point manner, wherein the 2D pixels and 3D points are matched using pre-existing projections. Due to the inaccuracy of the extrinsic calibration between LiDAR and camera [6], there is a more or less point-level misalignment in the projections. As a result, existing cross-modal UDA methods degrade significantly when extending them to DG task. Unlike UDA, which allows fine-tuning on the target domain, the target domain is unavailable in the DG setting. Thus these point-to-point cross-modal UDA methods are more sensitive to inaccurate cross-modal matching caused by point-level misalignment, as seen in Fig. 1. Moreover, to model domain-irrelevant representations, some cross-modal UDA methods [25, 36] introduce adversarial learning, which is highly responsive to hyperparameters and challenging to train.

To tackle these concerns, we propose cross-modal learning under BEV for domain generalization of 3D semantic segmentation, which is inspired by 3D object detection methods [16, 35, 39] that use the additional bird's-eye view of one modality (point cloud) to better the target posture and boundary. For different modalities (image and point cloud), with the help of an auxiliary bird's-eye view, we alleviate the cross-modal matching error caused by point-to-pixel misalignment and optimize the domain-irrelevant representation modeling. Specifically, we first propose BEV-based Area-to-area Fusion (BAF). Instead of conducting cross-modal learning point-to-point, we divide the point cloud and its corresponding image into areas with the help of a unified BEV space. And then, based on point-to-pixel projections, we match areas from two modalities to conduct area-to-area fusion. The cross-modal matching between areas has a higher fault tolerance for point-level misalignment. Because two projected point and pixel are more likely to be located in the same area than sharing the same accurate location. In this way, we significantly mitigate the influence of point-level misalignment and achieve accurate cross-modal learning in an area-to-area manner.

Furthermore, BEV-driven Domain Contrastive Learning (BDCL) is proposed to optimize domain-irrelevant representation modeling. First, with the aid of cross-modal learning under bird's-eye view, we generate the BEV feature map in a voxelized manner. This process is greatly affected by point cloud density, which makes the BEV feature map highly domain-relevant. Thus, using the BEV feature map to drive contrastive learning can provide stronger supervision for learning domain-irrelevant features. However, domain attribute, *i.e.*, LiDAR configuration, is reflected in the global density of the point cloud. Therefore, we propose Density-maintained Vector Modeling (DVM) to transform the BEV feature map into a global vector that maintains density perception. Then, we build contrastive learning that constrains consistency between BEV vectors before and after changing domain attributes. Moreover, as the BEV vectors contain domain-retentive multi-modal information, BDCL can push both 2D and 3D networks to learn domain-irrelevant features jointly.

We can summarize our contributions as follows:

• We propose BEV-DG for domain generalization of 3D semantic segmentation. With the aid of cross-modal learning under bird's-eye view, we optimize domain-irrelevant representation modeling in a constraint manner.

• To relive the cross-modal learning from the suffering of misalignment in point-to-pixel projections, we propose BEV-based area-to-area fusion. The accurate area-to-area cross-modal learning under bird's-eye view can more efficiently promote the information interaction between modalities to confront the domain shift.

• We propose BEV-driven domain contrastive learning, where the Density-maintained Vector Modeling (DVM) is introduced to generate the global vector that sufficiently embodies domain attributes. Furthermore, with the help of Density Transfer (DT), we build contrastive learning based on these vectors, pushing 2D and 3D networks to learn domain-irrelevant features jointly.

• We design three generalization settings based on three 3D datasets and provide the results of some competitors by extending cross-modal UDA methods to the DG setting. Comprehensive experimental results illustrate that BEV-DG consistently outperforms both the baseline and state-of-the-art approaches across all evaluated generalization scenarios.

2. Related Work

Point Cloud Semantic Segmentation. In recent years, significant advancements have been achieved in deep

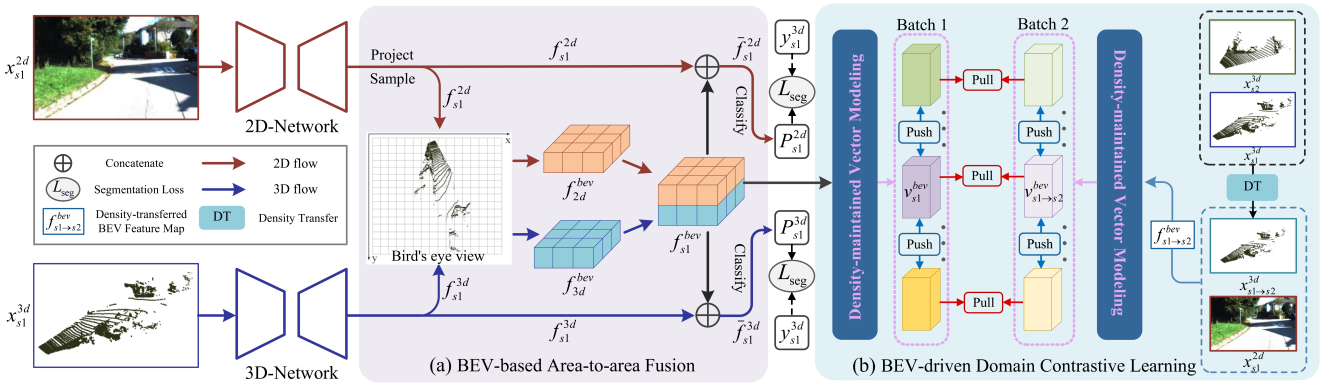

Figure 2. The framework of BEV-DG. When it is trained on the first source domain, the input is paired samples $\{\mathbf{x_{s1}^{2d}}, \mathbf{x_{s1}^{3d}}\}$ with 3D label y_{s1}^{3d}. Through module (a), the BEV feature map f_{s1}^{bev} is obtained based on 2D and 3D features $\{\mathbf{f_{s1}^{2d}}, \mathbf{f_{s1}^{3d}}\}$. Using point-to-pixel projections, we sample from the dense 2D feature map to generate 2D features of the length N, *i.e.*, the number of 3D points. Then, f_{s1}^{bev} is fused with f_{s1}^{2d} and f_{s1}^{3d}, respectively, to generate predictions. Moreover, f_{s1}^{bev} is fed into the module (b) to generate density-maintained BEV vector v_{s1}^{bev} to drive contrastive learning for domain-irrelevant representation modeling.

learning-based point cloud semantic segmentation[10, 12, 15, 27, 40]. However, these methods rely on fine-grained annotations, which are costly to obtain. To this end, weakly supervised semantic segmentation has attracted increasing attention [31, 32, 33, 37, 38] as it reduces the reliance on labels. These fully or weakly supervised methods achieve impressive performance on current datasets. However, these methods utilize training and testing data derived from a shared dataset. Due to different data collection devices, their data distribution is usually different. Therefore, whether the model trained on one dataset can generalize well to others is a critical question to consider.

Domain Adaptation for 3D Point Clouds. An increasing interest task, Unsupervised Domain Adaptation (UDA), has emerged in the past few years for 3D vision. 3D-CoCo [35] utilizes highly transferable BEV features to expand the concept of contrastive instance alignment to encompass point cloud detection, aiming to push the model to acquire domain-irrelevant features. For 3D segmentation, Complete and Label [34] designs a completion network for recovering the underlying surfaces of point clouds. The reconstructed 3D surfaces act as a standardized domain, enabling the transfer of semantic labels across various LiDAR sensors. However, these UDA methods only utilize 3D modality (point cloud), neglecting the value of 2D images.

As 3D datasets often consist of 3D point clouds and corresponding 2D images, utilizing multi-modality to solve domain shift problems for point clouds is convenient. Some cross-modal UDA methods [14, 21, 25, 36] have recently been proposed for 3D semantic segmentation. Beneath the surface of such techniques lies the same essence of cross-modal learning, *i.e.*, the fusion of multi-modal information. With the help of projections between points and pixels, these methods achieve cross-modal learning through the constraint of the consistency between point and pixel predictions. This point-to-point manner is significantly affected by misalignment in projections. To achieve this goal, in this paper, our focus is directed towards achieving more accurate cross-modal learning that is less influenced by point-level misalignment.

Domain Generalization. Domain Generalization (DG) centers around the ability of the model to generalize to unseen domains. Compared to UDA, the model can not access data originating from the target domain during the training phase. The common methods include learning domain-irrelevant features based on multiple source domains [13, 17, 18, 20, 24, 26] and enlarging the available data space with augmentation [5, 28, 30, 41, 42]. Recently, certain methods have also utilized regularization within the context of meta-learning [19] and Invariant Risk Minimization (IRM) [1] framework for DG. However, these methods are focused on 2D vision tasks. There has been relatively limited research focused on exploring domain generalization for 3D point clouds. 3D-VField [17] attempts to improve the generalization of 3D object detectors to out-of-domain data by using proposed adversarial augmentation to deform point clouds during training. Compared to these methods, our BEV-DG aims to utilize multi-modal data to investigate DG for 3D semantic segmentation.

3. Method

3.1. Overview of BEV-DG

Problem Definition. In DG for 3D semantic segmentation, the problem is framed with the expectation of having paired 2D images and 3D point clouds. The DG task aims to exploit the knowledge from two source domains, $S1$ and $S2$, to achieve generalization to the target domain T. Each domain contains images and point clouds $\{x^{2d}, x^{3d}\}$. For the sake of simplicity, we exclusively utilize the front camera

image and the corresponding LiDAR points that are projected onto it. The model is trained on $S1$ and $S2$, where only the 3D label $y^{3d}_{s1/2}$ exists, and tested on T. In the DG task, the model is not exposed to the target domain during its training phase, *i.e.*, T is unseen. Due to images and point clouds being heterogeneous, x^{2d} and x^{3d} are fed into 2D network h^{2d} and 3D network h^{3d} to output segmentation results P^{2d} and P^{3d}, respectively. Cross-modal DG aims to utilize the complementarity of x^{2d} and x^{3d} to improve both P^{2d} and P^{3d}. Furthermore, the projections between 2D pixels and 3D points are supplied by preexisting data.

Method Overview. Our approach consists of BEV-based Area-to-area Fusion (BAF) and BEV-driven Domain Contrastive Learning (BDCL), aiming to optimize domain-irrelevant feature modeling with the aid of cross-modal learning under bird's-eye view. The overall framework is depicted in Fig. 2. Using the training on the initial source domain as an illustration, the training for the second domain is identical. Given the paired 2D and 3D samples (*i.e.*, x^{2d}_{s1} and x^{3d}_{s1}), we first input them into BAF to generate BEV feature map (*i.e.*, f^{bev}_{s1}) and output segmentation results (*i.e.*, p^{2d}_{s1} and p^{3d}_{s1}). Specifically, we process 2D and 3D features (*i.e.*, f^{2d}_{s1} and f^{3d}_{s1}) in a unified BEV space to generate 2D and 3D BEV feature maps (*i.e.*, f^{bev}_{2d} and f^{bev}_{3d}). Next, we concatenate them to produce f^{bev}_{s1} and fuse it with f^{2d}_{s1} and f^{3d}_{s1} respectively, obtaining $\bar{f}^{2d}_{s1}$ and $\bar{f}^{3d}_{s1}$ to further generate the predictions. In addition, f^{bev}_{s1} is fed into BDCL to help the networks learn domain-irrelevant features. Through Density-maintained Vector Modeling (DVM), we use generated BEV vector (*i.e.*, v^{bev}_{s1}) to form positive and negative pairs with other BEV vectors. For the positive pair, we generate a density-transferred BEV vector (*i.e.*, $v^{bev}_{s1\to s2}$) with the help of Density Transfer (DT), which transforms the density of x^{3d}_{s1} into the second domain point cloud x^{3d}_{s2}.

3.2. BEV-based Area-to-area Fusion

Previous methods conduct cross-modal learning in a point-to-point manner based on the projections between 3D points and 2D pixels. However, due to the inaccuracy of extrinsic calibration between LiDAR and camera [6], more or less point-level misalignment exists, hindering the effectiveness of such methods. Given this perspective, we propose BAF to conduct cross-modal learning under bird's-eye view in an area-to-area manner. In this way, we can effectively mitigate the influence of point-level misalignment and achieve more accurate cross-modal learning.

BEV Transformation. Camera captures data in perspective view and LiDAR in 3D view. This view discrepancy makes it difficult to appropriately divide the areas of the image and point cloud and set the matching relationship between them. To this end, we introduce a unified BEV space to transform the image and point cloud into the same view. For point cloud x^{3d}_{s1}, we first quantize it along the x-axis and y-axis to generate pillar voxels evenly, as shown in module (a) of Fig. 2. These voxels can be regarded as different areas of point cloud under the bird's-eye view. As a result, the points are assigned to these areas according to their coordinates. The feature of a voxel is obtained by max-pooling the features of points inside it. For example, the feature in the i,j-th grid cell is:

$$\begin{aligned} f^{3d}_{i,j} =& MAX(\{h^{3d}(p^{3d}) \mid (i-1)w < p^{3d}_x < iw, \\ & (j-1)w < p^{3d}_y < jw\}), \end{aligned} \tag{1}$$

where $f^{3d}_{i,j} \in \mathcal{R}^{1\times C_{3d}}$. C_{3d} is the number of channels of 3D features. MAX denotes the max pooling operation. The size of a grid cell is $w \times w$. p^{3d}_x/p^{3d}_y is the x/y coordinate of 3D point p^{3d}, *i.e.*, its locations in the BEV space. Finally, the BEV feature map of x^{3d}_{s1} can be formulated as follows:

$$f^{bev}_{3d} = \left\{ f^{3d}_{i,j} \mid i \in \{1,2\ldots,W\}, j \in \{1,2\ldots,L\} \right\}, \tag{2}$$

where $f^{bev}_{3d} \in \mathcal{R}^{W\times L\times C_{3d}}$. W and L denote the number of grid cells along the x-axis and y-axis, respectively.

How to transform the image into bird's-eye view is a challenging problem. To tackle it, existing methods [22, 23] usually utilize depth estimation or transformer, which are very complex and costly. In contrast, we simply use the point-to-pixel projections provided by data prior to conduct view transformation for images. Specifically, for a pixel p^{2d} in image x^{2d}_{s1}, which projects to point p^{3d}, its accurate locations in the BEV space may be different from p^{3d}_x and p^{3d}_y due to misalignment. However, to transform the image into bird's-eye view, we only need to determine the proximate locations of pixels, *i.e.*, the voxels in which pixels are located. A pillar voxel covers much more space than a point, and even if p^{2d} mismatches p^{3d}, they are still likely located in the same voxel. So we determine the voxels where pixels are located based on the corresponding 3D points, effectively mitigating the influence of misalignment. Finally, we can obtain 2D BEV feature map f^{bev}_{2d} as follows:

$$\begin{aligned} f^{2d}_{i,j} =& MAX(\{h^{2d}(p^{2d}) \mid (i-1)w < p^{3d}_x < iw, \\ & (j-1)w < p^{3d}_y < jw \ \}), \end{aligned} \tag{3}$$

$$f^{bev}_{2d} = \left\{ f^{2d}_{i,j} \mid i \in \{1,2\ldots,W\}, j \in \{1,2\ldots,L\} \right\}. \tag{4}$$

Area-to-area Fusion. After BEV transformation, we divide the image and point cloud into areas using the same criteria and obtain 2D and 3D features of these areas, *i.e.*, f^{bev}_{2d} and f^{bev}_{3d}. Compared to point-to-point cross-modal learning, our method does not need to match pixels and points based on projections that may be misaligned. Instead, we just need to match their areas. Compared with sharing the same accurate location in BEV space, two projected point and pixel are more likely to be located in the same voxel (area), which

means matching between areas based on point-to-pixel projections has a higher fault tolerance for point-level misalignment. So we directly concatenate f_{2d}^{bev} and f_{3d}^{bev}, followed by a linear layer with ReLU, to achieve area-to-area fusion:

$$f_{s1}^{bev} = ReLU(FC_1(f_{2d}^{bev} \oplus f_{3d}^{bev})), \tag{5}$$

where f_{s1}^{bev} is the fusion BEV feature map of x_{s1}^{2d} and x_{s1}^{3d}. Next, we further fuse this area-level information with initial point-level features for final semantic segmentation:

$$\bar{f}_{p^{3d}} = ReLU(FC_2(h^{3d}(p^{3d}) \oplus f_{i,j}^{bev})), \tag{6}$$

where $f_{i,j}^{bev}$ is the feature of voxel where point p^{3d} is located, *i.e.*, the i,j-th feature in f_{s1}^{bev}. $\bar{f}_{s1}^{3d}$ consists of all N fused point features $\bar{f}_{p^{3d}}$. The process to obtain $\bar{f}_{s1}^{2d}$ is identical. This fusion provides bird's-eye-view multi-modal contextual information for each point (pixel) in a point-to-area manner. As only matching between the point (pixel) with the area where it is located, this manner is also less susceptible to point-level misalignment. In summary, each stage of BAF effectively mitigates the impact of misalignment, achieving more accurate cross-modal learning.

3.3. BEV-driven Domain Contrastive Learning

Previous methods usually utilize adversarial learning to model domain-irrelevant representations, which are susceptible to hyperparameters and challenging to train. Considering this, we introduce BDCL, which conducts contrastive learning between different domains and samples with the help of cross-modal learning under bird's-eye view. Specifically, we promote the consistency between samples before and after changing the domain attributes, providing additional supervision for learning domain-irrelevant features. For contrastive learning, the stronger the domain relevance of the sample features, the stronger the supervision will be. Thus we choose BEV feature map f_{s1}^{bev} generated by cross-modal learning under bird's-eye view to drive the contrastive learning. It is produced in a voxelized manner, which makes it highly related to the point cloud density. Concretely,due to the fixed size of a voxel, the quantity of points it encompasses is notably contingent on the density. Therefore, compared to the initial point-level features f_{s1}^{2d}/f_{s1}^{3d}, f_{s1}^{bev} has stronger domain relevance. Moreover, as f_{s1}^{bev} contains domain-retentive multi-modal information, BDCL can push both 2D and 3D networks to learn domain-irrelevant features jointly.

Our proposed BDCL consists of two components: (1) Density-maintained Vector Modeling (DVM); (2) building contrastive learning to help 2D and 3D networks jointly learn domain-irrelevant features.

Density-maintained Vector Modeling (DVM). As the LiDAR configuration determines the global density of the point cloud, the domain attribute of a sample should be characterized by its global feature. However, The BEV feature map consists of area-level features. Thus, we must transform it into a global vector that can embody domain attributes well. For a point cloud, the distribution of points inside it varies greatly. Concretely, the density of the part near the LiDAR is greater than the part away from the LiDAR. Thus, directly modeling the global vector from f_{s1}^{bev} by equally treating points in different areas can not maintain the perception of density. In light of this, we propose DVM to transform $f_{s1}^{bev} \in \mathcal{R}^{W \times L \times C}$ into BEV vector $v_{s1}^{bev} \in \mathcal{R}^{1 \times C}$ without undermining the density perception of feature. Specifically, in the BEV space, we find the density of the point cloud is well reflected in the distribution of areas, as shown in Fig. 3. This distribution pattern helps us model density-maintained vector from BEV feature map f_{s1}^{bev}. More analysis of DVM can be seen in the supplementary material. We can generate the BEV vector as follows:

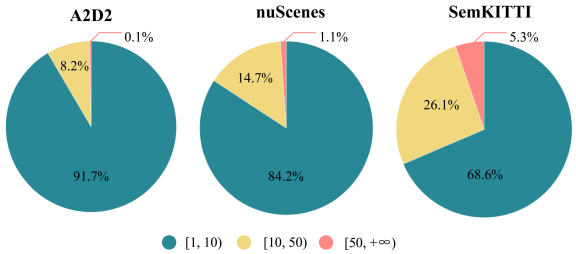

Figure 3. The distribution of areas in BEV space over dataset. We divide point clouds from three datasets into areas in the BEV space. Following that, we classify the areas into three groups according to the number of points inside them. Different distribution patterns can be seen clearly when we show the percentage of each type of area: points are spread more densely in nuScenes (32 beam) and SemanticKITTI (64 beam) than in A2D2 (16 beam).

$$\begin{aligned} v_{s1}^{bev} = & \frac{N_{[1,10)}}{N_{all}} MAX\left(f_{[1,10)}^{bev}\right) + \frac{N_{[10,50)}}{N_{all}} MAX\left(f_{[10,50)}^{bev}\right) \\ & + \frac{N_{[50,+\infty)}}{N_{all}} MAX\left(f_{[50,+\infty)}^{bev}\right), \end{aligned} \tag{7}$$

where $N_{[1,10)}/N_{[10,50)}/N_{[50,+\infty)}$ is the number of areas with $[1,10)/[10,50)/[50,+\infty)$ points inside. N_{all} is the number of all areas. $f_{[1,10)}^{bev}/f_{[10,50)}^{bev}/f_{[50,+\infty)}^{bev}$ is the feature set of areas with $[1,10)/[10,50)/[50,+\infty)$ points inside.

Architecture of BDCL. To form the negative and positive pairs of BEV vectors, we first utilize Density Transfer (DT) in Dual-Cross [21] to generate approximate BEV vectors of the other source domain. Concretely, we transform the densities of point clouds in the current batch into densities of point clouds from the other source domain, as depicted in module (b) of Fig. 2. In this process, the semantic content of point clouds and their corresponding images remain unchanged. Using these synthetic point clouds and their corresponding images, we can generate a new batch of density-transferred BEV vectors, which share the same

semantic content but perceive the density of the other domain. On the one hand, we push the BEV vectors in the same batch (domain) away from each other in the representation space. Since negative sample pairs come from a single domain and share identical domain attribute, domain-irrelevant representations are learned to contrast them. On the other hand, we pull the BEV vectors that share the same semantic content but from different batch (domain) close in the representation space, promoting the networks to learn domain-irrelevant features jointly. The contrastive loss of the first source domain can be formulated as follows:

$$\mathcal{L}_{ct}^{s1} = -\frac{1}{B}\sum_{i=1}^{B} log \frac{exp(v_i^{s1} \cdot v_i^{s1\to s2}/\tau)}{\sum_{j=1}^{B} exp(v_i^{s1} \cdot v_j^{s1}/\tau) + \sum_{k=1}^{B} exp(v_i^{s1\to s2} \cdot v_k^{s1\to s2}/\tau)}, \tag{8}$$

where B is the batch size, v_i^{s1} is the i-th BEV vector in the first domain batch, and $v_i^{s1\to s2}$ is the corresponding density-transferred BEV vector of v_i^{s1}. τ is the temperature hyper-parameter that controls the concentration level. The contrastive loss of the second source domain is the same.

3.4. Overall Objective Function

The segmentation loss of the first source domain can be formulated as follows:

$$\mathcal{L}_{seg}^{s1}(x_{s1}, y_{s1}^{3d}) = -\frac{1}{NC}\sum_{n=1}^{N}\sum_{c=1}^{C} y_{s1}^{(n,c)} \log p_{x_{s1}}^{(n,c)}, \tag{9}$$

where x_{s1} is either x_{s1}^{2d} or x_{s1}^{3d}, N denotes the number of points, while C stands for the number of categories. The segmentation loss of the second source domain is the same. So the final segmentation loss and contrastive loss can be written as:

$$\mathcal{L}_{seg} = \mathcal{L}_{seg}^{s1} + \mathcal{L}_{seg}^{s2}, \tag{10}$$

$$\mathcal{L}_{ct} = \mathcal{L}_{ct}^{s1} + \mathcal{L}_{ct}^{s2}. \tag{11}$$

Finally, we train the model on source domains using Eq. 12:

$$\mathcal{L}_{all} = \mathcal{L}_{seg} + \lambda_{ct}\mathcal{L}_{ct}, \tag{12}$$

where λ_{ct} is the trade-off to control the contrastive loss.

4. Experiments

4.1. Datasets and Generalization Settings

We conduct experiments using three autonomous driving datasets acquired by different LiDAR configurations. (1) A2D2 [9]: The point clouds are acquired by a Velodyne 16-beam LiDAR. The LiDAR frames are labeled point by point. The data is divided into ∼28K training frames and ∼2K validation frames. (2) nuScenes [4]: It contains ∼40K LiDAR frames annotated with 3D bounding boxes. Following previous methods [14, 21, 25, 36], we assign point-wise labels based on the 3D bounding box where points are located. Unlike the A2D2 dataset, it uses a 32-beam LiDAR sensor. We train our model on ∼28K frames and evaluate on ∼6K frames. (3) SemanticKITTI [3]: Different from A2D2 and nuScenes, it uses a Velodyne 64-beam LiDAR. We use sequences 00-07 and 09-10 to train model and evaluate it on sequence 08, resulting in ∼19K frames for training and ∼4K frames for evaluation. For each dataset, the RGB camera and LiDAR are synchronized and calibrated. The projections from 3D points to 2D pixels are given by data. We solely utilize the image of front camera and the corresponding projected LiDAR points for simplicity and consistency across datasets. Only 3D annotations are used for 3D semantic segmentation.

To evaluate the performance of BEV-DG, we design three generalization settings. (1) A,S→N: the network is trained on samples from A2D2 and SemanticKITTI, but tested on samples from nuScenes. (2) A,N→S: we train the network with A2D2 and nuScenes, but test it with SemanticKITTI. (3) N,S→A: the network is trained on nuScenes and SemanticKITTI, but tested on A2D2. We define 5 shared classes between the three datasets: car, truck, bike, person and background. All of them are commonplace and crucial for safety in self-driving scenarios.

4.2. Implementation Details

Backbone Network. To ensure an equitable comparison, we utilize the identical backbone network as the previous methods [14, 21, 25, 36]. Concretely, the 2D network is an adapted U-Net [29] with a pre-trained ResNet34 [11] encoder from ImageNet [8]. The 3D network employs SparseConvNet [10] with a U-Net [29] architecture, featuring six levels of downsampling. Using a voxel size of $5cm$ in SparseConvNet, we guarantee the presence of only one 3D point within each voxel. We conduct training and evaluation of our model using the PyTorch deep learning framework on a single NVIDIA TITAN RTX GPU.

Configuration of Parameters. We select a batch size of 8 and employ the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We implement a learning schedule based on iterations, starting with an initial learning rate of 0.001. The rate is divided by 10 at $80k$ and $90k$ iterations. BEV-DG undergoes training for $100k$ iterations in each generalization scenario. The w in Eq. 1 is set to $0.5m$. We set both τ in Eq. 8 and λ_{ct} in Eq. 12 to 0.01. The accuracy is assessed using the mean Intersection over Union (mIoU).

4.3. Experimental Results

We assess the effectiveness of our approach across three distinct domain generalization scenarios, *i.e.*, A,S→N, A,N→S and N,S→A, and compare our method with some representative state-of-the-art competitors. These approaches share the same 2D and 3D backbone networks as

Table 1. Quantitative results (mIoU, %) in various DG scenarios. The baseline architecture only contains 2D and 3D backbone networks with segmentation heads. The training of it is supervised by segmentation loss of source domains. In 'Oracle', the baseline model is trained and tested on the target domain. 'Avg' represents the ensemble outcome achieved by averaging the softmax-predicted probabilities from both 3D and 2D outputs. The highest value is indicated in red, while the second highest value is indicated in blue.

Method	A,S→N			A,N→S			N,S→A		
	2D	3D	Avg	2D	3D	Avg	2D	3D	Avg
Baseline	48.5	49.4	53.9	32.1	51.4	44.7	48.0	45.0	48.8
xMUDA[14]	49.8	49.1	55.9	32.8	52.0	44.9	50.9	44.9	50.8
xMUDA Fusion[14]	46.8	49.0	52.8	32.3	56.5	44.7	51.5	45.4	50.4
DsCML[25]	48.2	47.6	52.3	31.6	51.3	43.8	52.2	46.1	51.7
SSE-xMUDA[36]	44.9	48.6	53.9	36.1	52.7	47.3	55.3	44.8	52.0
Dual-Cross[21]	50.8	48.1	56.0	32.3	55.5	42.6	53.1	41.3	50.4
BEV-DG	58.0	59.3	59.0	47.9	54.7	60.2	55.0	55.1	56.7
Oracle	64.8	57.9	69.0	55.5	72.8	70.7	81.7	53.1	82.4

Table 2. Ablation experiment results (mIoU, %) of two modules.

	BAF	BDCL	A,S→N			A,N→S		
			2D	3D	Avg	2D	3D	Avg
#1	Baseline		48.5	49.4	53.9	32.1	51.4	44.7
#2	✓		56.7	57.9	57.8	45.5	50.8	52.9
#3		✓	50.2	49.6	57.1	34.2	51.9	49.4
#4	✓	✓	58.0	59.3	59.0	47.9	54.7	60.2

ours. BEV-DG outperforms others in all DG settings.

We provide some qualitative results in Fig. 4 and elaborate on the 3D semantic segmentation comparison outcomes in Tab. 1. It is noticeable that across all three generalization settings, BEV-DG consistently enhances results for both 2D and 3D in comparison to competitors. On A,S→N, BEV-DG outperforms the baseline by 9.5% (2D), 9.9% (3D) and 5.1% (Avg). In the first row of Fig. 4, we notice that the baseline misclassifies a person as a trunk, whereas BEV-DG avoids this error. Furthermore, when compared to the second-best values, BEV-DG outperforms them by 7.2% (2D), 9.9% (3D) and 3.0% (Avg). On A,N→S, BEV-DG outperforms the baseline by 15.8% (2D), 3.3% (3D) and 15.5% (Avg). In the third row, it is evident that the baseline misclassifies a car, while BEV-DG avoids this misclassification. In addition, BEV-DG outperforms the second-best values by 11.8% (2D) and 12.9% (Avg) but is slightly worse on 3D. On N,S→A, BEV-DG outperforms the baseline by 7.0% (2D), 10.1% (3D) and 7.9% (Avg). In the second row, it is evident that the baseline struggles to accurately classify the bike and person, whereas BEV-DG accurately identifies them. Compared to the second-best values, BEV-DG outperforms them by 9.0% (3D) and 4.7% (Avg) but is slightly worse on 2D. These results demonstrate that BEV-DG significantly improves the generalization ability of the model through BEV-based area-to-area fusion and BEV-driven domain contrastive learning.

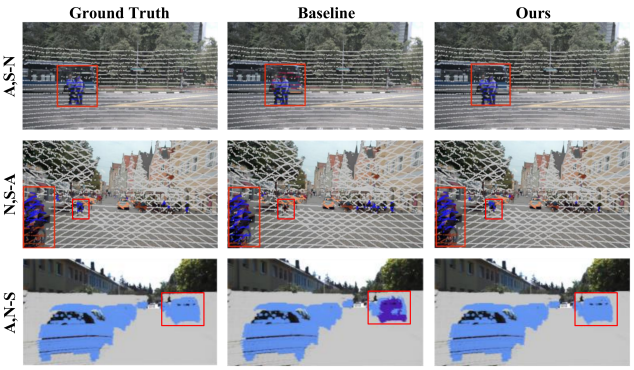

Figure 4. Qualitative results in three domain generalization settings: A,S→N, N,S→A and A,N→S. Additional qualitative results are available in the supplementary material.

4.4. Ablation Experiments

Impacts of BAF and BDCL. To show the impacts of BAF and BDCL, we systematically present the performance of each module by incrementally integrating them into the baseline. The corresponding outcomes are detailed in Tab. 2. In BAF, we conduct cross-modal learning area-to-area under BEV, aiming to alleviate the influence of misalignment. In BDCL, DVM generates density-maintained BEV vectors for contrastive learning, enabling the 2D and 3D networks to jointly learn domain-irrelevant features.

In Tab. 2, comparing #1 and #2 reveals that BAF leads to substantial enhancements in Avg performance, with improvements of 3.9% and 8.2% on A,S→N and A,N→S settings, respectively. More importantly, compared to the point-to-point methods [14, 21, 25, 36] in Tab. 1, BAF consistently outperforms all of them on both 2D and 3D modalities, indicating it effectively mitigates the influence of point-level misalignment. When applying BDCL to baseline, *i.e.*, removing the fusion between f_{s1}^{bev} and f_{s1}^{2d}/f_{s1}^{3d} in BEV-DG, we can observe that BDCL brings about en-

Table 3. Ablation experiment results (mIoU, %) of BEV transformation and fusion of BEV feature maps.

	Method	A,S→N			A,N→S		
		2D	3D	Avg	2D	3D	Avg
#1	Ours w/o Trans.	56.1	56.2	56.2	38.7	38.9	38.7
#2	Ours w/o Fusion	53.6	48.8	58.3	37.2	50.9	45.3
#3	Ours-full	58.0	59.3	59.0	47.9	54.7	60.2

Table 4. Ablation experiment results (mIoU, %) of DVM and DT.

	Method	A,S→N			A,N→S		
		2D	3D	Avg	2D	3D	Avg
#1	Ours w/o DVM	56.1	56.2	56.7	40.4	47.8	50.0
#2	Ours w/o DT	56.3	57.1	57.0	46.7	41.5	49.1
#3	Ours-full	58.0	59.3	59.0	47.9	54.7	60.2

hancements of 3.2% and 4.7% on A,S→N and A,N→S scenarios in Avg performance, respectively. Furthermore, by applying both BAF and BDCL to the baseline, we present the results in #4. BEV-DG brings 9.5% (2D), 9.9% (3D) and 5.1% (Avg) enhancements on A,S→N and 15.8% (2D), 3.3% (3D) and 15.5% (Avg) enhancements on A,N→S. These results evaluate the effects of our BAF and BDCL.

Analysis of BAF. There are two crucial steps in the BAF module. One is that we transform initial point-level features into BEV feature maps. The other is that we further fuse the 2D and 3D BEV feature maps for following prediction and contrastive learning. To further demonstrate the effects of BAF, we conduct additional experiments by removing the two steps from BEV-DG respectively, and results are shown in Tab. 3. We first experiment by removing the BEV transformation step. Specifically, in module (a), we directly concatenate initial 2D and 3D features using point-to-pixel projections for semantic segmentation. And then, we generate a global vector by performing max pooling on these concatenated point-level features to drive contrastive learning in module (b). The results are shown in #1 of Tab. 3. Comparing #3 and #1, we can find a sharp drop in the performance, indicating that the BEV feature is the crucial reason why our method works.

Next, we experiment by removing the fusion of 2D and 3D BEV feature maps. Specifically, in module (a), we directly fuse initial 2D features with the 2D BEV feature map for semantic segmentation. The 3D branch is the same. Moreover, in module (b), contrastive learning is conducted by using the 2D BEV vector and 3D BEV vector, respectively. The results are displayed in #2 of Tab. 3. Comparing between #2 and #3, it is evident that "ours-full" gains 4.4% (2D), 10.5% (3D) and 0.7% (Avg) improvements on A,S→N and 10.7% (2D), 3.8% (3D) and 14.9% (Avg) improvements on A,N→S, which demonstrates the effectiveness of area-to-area fusion.

Analysis of BDCL. In BDCL, we propose DVM to generate the density-maintained vector, which can sufficiently embody the domain attributes, to drive contrastive learning. Moreover, we utilize DT to generate the density-transferred vector to form positive pairs. Therefore, to provide further insight into the impact of BDCL, we perform supplementary experiments by individually excluding the two components from BEV-DG, and results are shown in Tab. 4. We first experiment by removing DVM. Specifically, we directly perform max pooling on BEV feature map $f_{s1}^{bev} \in \mathcal{R}^{W \times L \times C}$ to generate a global vector with a size of $1 \times C$ to drive contrastive learning. Compared with density-maintained vector v_{s1}^{bev}, it can not maintain the perception of point cloud density because it is generated by treating different areas of the point cloud equally. The results are shown in #1 of Tab. 4. Comparing between #1 and #3, we can find that "ours-full" gains 1.9% (2D), 3.1% (3D) and 2.3% (Avg) enhancements on A,S→N and 7.5% (2D), 6.9% (3D) and 10.2% (Avg) enhancements on A,N→S, which demonstrates the effectiveness of DVM. It also confirms that more domain-related features for contrastive learning can achieve better domain-irrelevant feature learning.

Next, we experiment by removing DT. Specifically, in module (b), we replace density-transferred vector $v_{s1\to s2}^{bev}$ with a copy of v_{s1}^{bev}. The results are shown in #2 of Tab. 4. Comparing between #2 and #3, we can find that "ours-full" achieves 2.0% and 11.1% improvements on A,S→N and A,N→S in Avg respectively, which indicates that DT can help the BDCL model domain-irrelevant features by introducing density discrepancy.

Analysis of Misalignment. To demonstrate the influence of point-to-pixel misalignment, we evaluate two representative point-to-point methods (xMUDA [14] and Dual-Cross [21]) on A,S→N. Furthermore, we compare them with our proposed BAF and BEV-DG, which conduct cross-modal learning area-to-area. We randomly select a fraction of the points in the point cloud and perturb their projections to pixels. The results (Avg) are shown in Fig. 1. We can observe that BAF and BEV-DG with 20% misalignment even perform better than xMUDA and Dual-Cross with 5% misalignment. Moreover, these point-to-point methods degrade more dramatically with increasing misalignment. These results suggest that approaches based on point-to-point cross-modal learning are more sensitive to point-level misalignment. In contrast, with the help of cross-modal learning under bird's-eye view, our BAF and BEV-DG effectively mitigate the influence of misalignment.

Hyperparameter Sensitivity Analysis. To investigate the impact of two important hyperparameters, *i.e.*, area size w and contrastive loss weight λ_{ct}, we conduct additional experiments on A,S→N by changing their values in BEV-DG. w is critical to BAF because it determines the size of an area, directly affecting the effectiveness of area-to-area cross-modal learning. We change the value of w between $0.05m$ and $1m$. The outcomes are depicted in Fig. 5. We

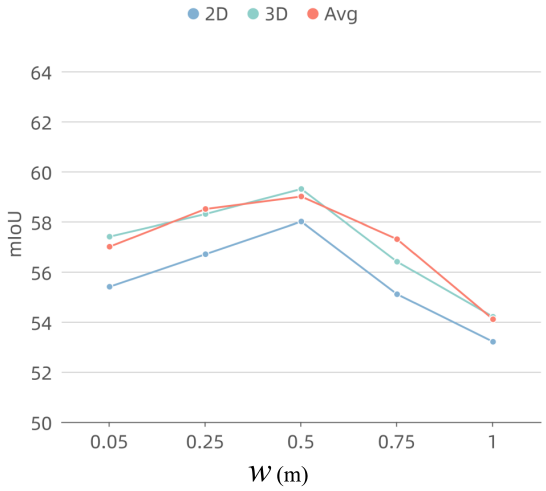

Figure 5. Results of BEV-DG with different w.

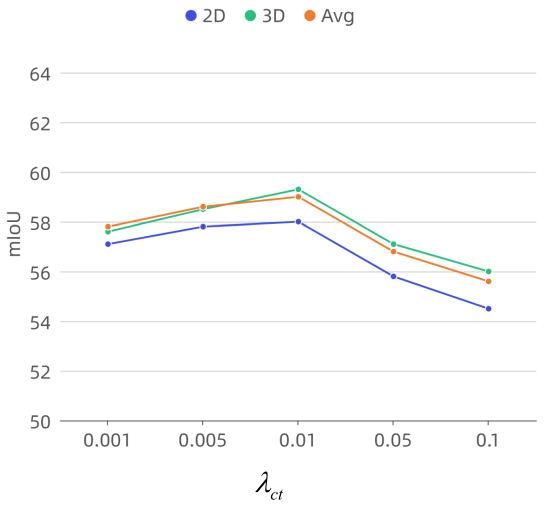

Figure 6. Results of BEV-DG with different λ_{ct}.

can observe that the performance of BEV-DG is best when w is set to $0.5m$. For λ_{ct}, it controls the importance of contrastive learning loss, which is crucial to BDCL. We change the value of λ_{ct} between 0.001 and 0.1. The results are presented in Fig. 6. We can find that the model works best when λ_{ct} is 0.01 and our method is not sensitive to λ_{ct}. Taking into account the aforementioned information, we assign the values of w and λ_{ct} as $0.5m$ and 0.01 respectively.

Area Distribution in the BEV Space. The area distribution in BEV space depends on how to divide areas with different points inside into different types. In BEV-DG, we divide areas into three types, *i.e.*, $[1, 10)/[10, 50)/[50, +\infty)$, because the distribution pattern generated by this criterion can obviously embody the difference in point cloud density of datasets. To verify the rationality of our criterion for classification, we show the area distribution generated by some other criteria in Fig. 7. We can observe that these distribution patterns can not obviously embody domain attributes compared to ours, which is shown in Fig. 3.

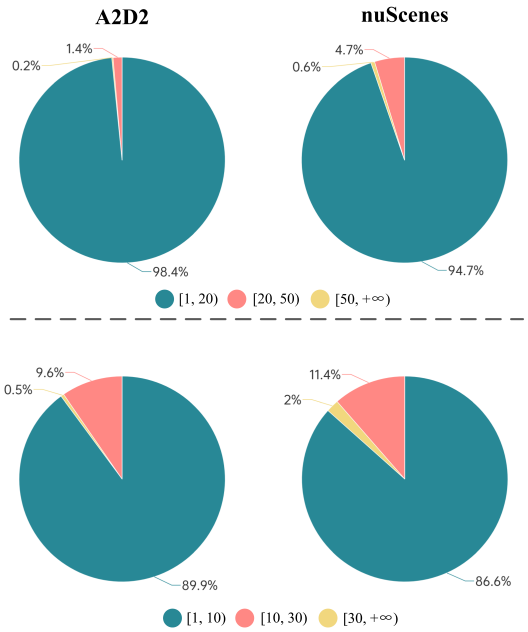

Figure 7. The distribution of areas in BEV space over datasets.

5. Conclusion

This paper proposes cross-modal learning under BEV for domain generalization of 3D semantic segmentation, aiming to optimize domain-irrelevant representation modeling with the aid of cross-modal learning under BEV. Specifically, we propose BEV-based area-to-area fusion to achieve cross-modal learning under BEV, which has a higher fault tolerance for point-level misalignment. Accurate cross-modal learning can more efficiently utilize the complementarity of multi-modality to confront the domain shift. Furthermore, we propose BEV-driven domain contrastive learning to optimize domain-irrelevant representation modeling. With the help of cross-modal learning under BEV and density-maintained vector modeling, we generate the BEV vector to drive contrastive learning, pushing the networks to learn domain-irrelevant features jointly. Extensive experimental results on three designed generalization settings highlight the superiority of our BEV-DG.

Acknowledgment

Miaoyu Li, Yachao Zhang , and Yanyun Qu were supported by the National Natural Science Foundation under Grant No. 6217224, the China Post-doctoral Science Foundation No. 2023M731957, the CCF-Lenovo Blue Ocean Research Fund.

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[2] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *CVPR*, pages 1534–1543, 2016.

[3] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *ICCV*, pages 9297–9307, 2019.

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020.

[5] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *CVPR*, pages 2229–2238, 2019.

[6] Shoubin Chen, Jingbin Liu, Xinlian Liang, Shuming Zhang, Juha Hyyppä, and Ruizhi Chen. A novel calibration method between a camera and a 3d lidar with infrared images. In *ICRA*, pages 4963–4969, 2020.

[7] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[9] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, et al. A2d2: Audi autonomous driving dataset. *arXiv preprint arXiv:2004.06320*, 2020.

[10] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, pages 9224–9232, 2018.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[12] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *CVPR*, pages 11108–11117, 2020.

[13] Shoubo Hu, Kun Zhang, Zhitang Chen, and Laiwan Chan. Domain generalization via multidomain discriminant analysis. In *UAI*, pages 292–302, 2020.

[14] Maximilian Jaritz, Tuan-Hung Vu, Raoul de Charette, Emilie Wirbel, and Patrick Pérez. xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation. In *CVPR*, pages 12605–12614, 2020.

[15] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *ICCV*, pages 4558–4567, 2018.

[16] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, pages 12697–12705, 2019.

[17] Alexander Lehner, Stefano Gasperini, Alvaro Marcos-Ramiro, Michael Schmidt, Mohammad-Ali Nikouei Mahani, Nassir Navab, Benjamin Busam, and Federico Tombari. 3d-vfield: Adversarial augmentation of point clouds for domain generalization in 3d object detection. In *CVPR*, pages 17295–17304, 2022.

[18] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, number 1, 2018.

[19] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. In *ICCV*, pages 1446–1455, 2019.

[20] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *CVPR*, pages 5400–5409, 2018.

[21] Miaoyu Li, Yachao Zhang, Yuan Xie, Zuodong Gao, Cuihua Li, Zhizhong Zhang, and Yanyun Qu. Cross-domain and cross-modal knowledge distillation in domain adaptation for 3d semantic segmentation. In *ACMMM*, pages 3829–3837, 2022.

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

[23] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *arXiv preprint arXiv:2205.13542*, 2022.

[24] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *ICCV*, pages 5715–5725, 2017.

[25] Duo Peng, Yinjie Lei, Wen Li, Pingping Zhang, and Yulan Guo. Sparse-to-dense feature matching: Intra and inter domain cross-modal learning in domain adaptation for 3d semantic segmentation. In *ICCV*, pages 7108–7117, 2021.

[26] Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient domain generalization via common-specific low-rank decomposition. In *ICML*, pages 7728–7738. PMLR, 2020.

[27] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, pages 5099–5108, 2017.

[28] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *CVPR*, pages 12556–12565, 2020.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015.

[30] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. *arXiv preprint arXiv:1804.10745*, 2018.

[31] Xian Shi, Xun Xu, Ke Chen, Lile Cai, Chuan Sheng Foo, and Kui Jia. Label-efficient point cloud semantic segmentation: An active learning approach. *arXiv preprint arXiv:2101.06931*, 2021.

[32] Jiacheng Wei, Guosheng Lin, Kim-Hui Yap, Tzu-Yi Hung, and Lihua Xie. Multi-path region mining for weakly supervised 3d semantic segmentation on point clouds. In *CVPR*, pages 4384–4393, 2020.

[33] Xun Xu and Gim Hee Lee. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In *CVPR*, pages 13706–13715, 2020.

[34] Li Yi, Boqing Gong, and Thomas Funkhouser. Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds. In *CVPR*, pages 15363–15373, 2021.

[35] Zeng Yihan, Chunwei Wang, Yunbo Wang, Hang Xu, Chaoqiang Ye, Zhen Yang, and Chao Ma. Learning transferable features for point cloud detection via 3d contrastive cotraining. *NeurIPS*, pages 21493–21504, 2021.

[36] Yachao Zhang, Miaoyu Li, Yuan Xie, Cuihua Li, Cong Wang, Zhizhong Zhang, and Yanyun Qu. Self-supervised exclusive learning for 3d segmentation with cross-modal unsupervised domain adaptation. In *ACMMM*, pages 3338–3346, 2022.

[37] Yachao Zhang, Zonghao Li, Yuan Xie, Yanyun Qu, Cuihua Li, and Tao Mei. Weakly supervised semantic segmentation for large-scale point cloud. In *AAAI*, pages 3421–3429, 2021.

[38] Yachao Zhang, Yanyun Qu, Yuan Xie, Zonghao Li, Shanshan Zheng, and Cuihua Li. Perturbed self-distillation: Weakly supervised large-scale point cloud semantic segmentation. In *ICCV*, pages 15520–15528, 2021.

[39] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *CVPR*, pages 9601–9610, 2020.

[40] Zhiyuan Zhang, Binh-Son Hua, and Sai-Kit Yeung. Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. In *ICCV*, pages 1607–1616, 2019.

[41] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *AAAI*, pages 13025–13032, 2020.

[42] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *ECCV*, pages 561–578, 2020.