Leveraging Smooth Deformation Augmentation for LiDAR Point Cloud Semantic Segmentation

Shoumeng Qiu, Jie Chen, Chenghang Lai, Hong Lu, Xiangyang Xue, Jian Pu*

***Abstract*—Existing data augmentation approaches on LiDAR point cloud are mostly developed on rigid transformation, such as rotation, flipping, or copy-based and mix-based methods, lacking the capability to generate diverse samples that depict smooth deformations in real-world scenarios. In response, we propose a novel and effective LiDAR point cloud augmentation approach with smooth deformations that can enrich the diversity of training data while keeping the topology of instances and scenes simultaneously. The whole augmentation pipeline can be separated into two different parts: scene augmentation and instance augmentation. To simplify the selection of deformation functions and ensure control over augmentation outcomes, we propose three effective strategies: residual mapping, space decoupling, and function periodization, respectively. We also propose an effective prior-based location sampling algorithm to paste instances on a more reasonable area in the scenes. Extensive experiments on both the SemanticKITTI and nuScenes challenging datasets demonstrate the effectiveness of our proposed approach across various baselines. The codes are publicly available at https://github.com/skyshoumeng/SmoothDA.**

***Index Terms*—LiDAR Augmentation, Smooth Deformation, Semantic Segmentation.**

I. Introduction

LiDAR point cloud semantic segmentation plays a crucial role in environment understanding for autonomous driving [3]–[5]. It is also very important for downstream tasks of autonomous driving, such as trajectory prediction [6] and motion planning [7]. Data augmentation is proven to be one of the most crucial and practical techniques in enhancing model performance without additional computation costs in the test phase [8]–[12]. This is especially true for tasks such as LiDAR point cloud semantic segmentation [3], [13]–[16], where creating a large dataset is extremely difficult and requires extensive labor work.

Data-driven deep models often require abundant data to sufficiently understand complex LiDAR point clouds in real-world scenarios. In contrast to images with lattice structures, point clouds are unordered sets of points without inherent structure [17]. Therefore, while data augmentation is relatively common for images [18]–[21], it has been relatively underexplored for LiDAR point clouds [1], [22], [23].

Most of the existing methods are based on rigid transformations, such as the commonly used rotation and flipping. Despite some great progress has been made in recent years, such as the copy-paste based approaches [24] or Mix-based approaches [1], [25]. PointMixup [26] proposed to produce new examples through an interpolation between two scans of point clouds. Copy-Paste [24] proposed to simply copy instances from other scenes and then paste them into the current scenes directly. PolarMix [1] proposed to enrich the diversity of the point cloud through two cross-scan augmentation strategies. They all lack consideration of cases where smooth deformations happen in real-world scenarios [17], [27], such as walking people or a winding road, which is also very important for the diversity of the datasets. Only a few methods are based on local deformations now, such as [17], [28], [29]. PointAugment [28] proposed an adversarial learning framework to optimize an augmented neural network and a task-specific network jointly. PointWOLF [17] proposed to generate the augmented results by applying locally weighted transformations centered at multiple anchor points in the object. PA-AUG [29] proposed to divide instances into partitions and then stochastically apply five different augmentation methods to each local region. However, the above approaches only apply to the point clouds of objects well. This is attributable to the LiDAR point clouds in the outdoor environments were distributed over a wide range [30]–[32], the method should be effective and the augmented results should be reasonable everywhere instead of a single object. In addition, compared with the simple rigid transformations, the augmentation with local deformation transformation is more sophisticated and uncontrollable [17], [27]. Overall, these factors result in augmentation approaches based on deformations for the LiDAR point cloud have not been fully investigated.

In this paper, we focus on LiDAR point cloud augmentation with the aim of alleviating the issue of data scarcity for 3D semantic segmentation. Specifically, we propose a novel and effective augmentation approach with smooth deformations for the LiDAR point clouds semantic segmentation task. To simplify the selection and design of the deformation functions, three strategies were proposed: residual mapping, space decoupling, and function periodization. First, drawing inspiration from the residual mapping in ResNet [33], instead of designing a function that maps the source point to the target point directly, we only need to design a function that maps the source point to the offset between the source and target point. Second, we decouple the raw transformation

Manuscript received xx, xx, 2023. This work is supported by Shanghai Municipal Science and Technology Major Project (No. 2018SHZDZX01), ZJLab, Shanghai Center for Brain Science and Brain-Inspired Technology (Corresponding author: Jian Pu.)

S. Qiu, J. Chen, C. Lai, H. Lu and X. Xue are with Shanghai Key Lab of Intelligent Information Processing and the School of Computer Science, Fudan University, Shanghai 200433, China (emails: {smqiu21, chenj19, chlai21}@m.fudan.edu.cn, {honglu, xyxue}@fudan.edu.cn)

J. Pu is with the Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University, Shanghai, 200433, China (e-mail: jianpu@fudan.edu.cn)

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2023.3348805

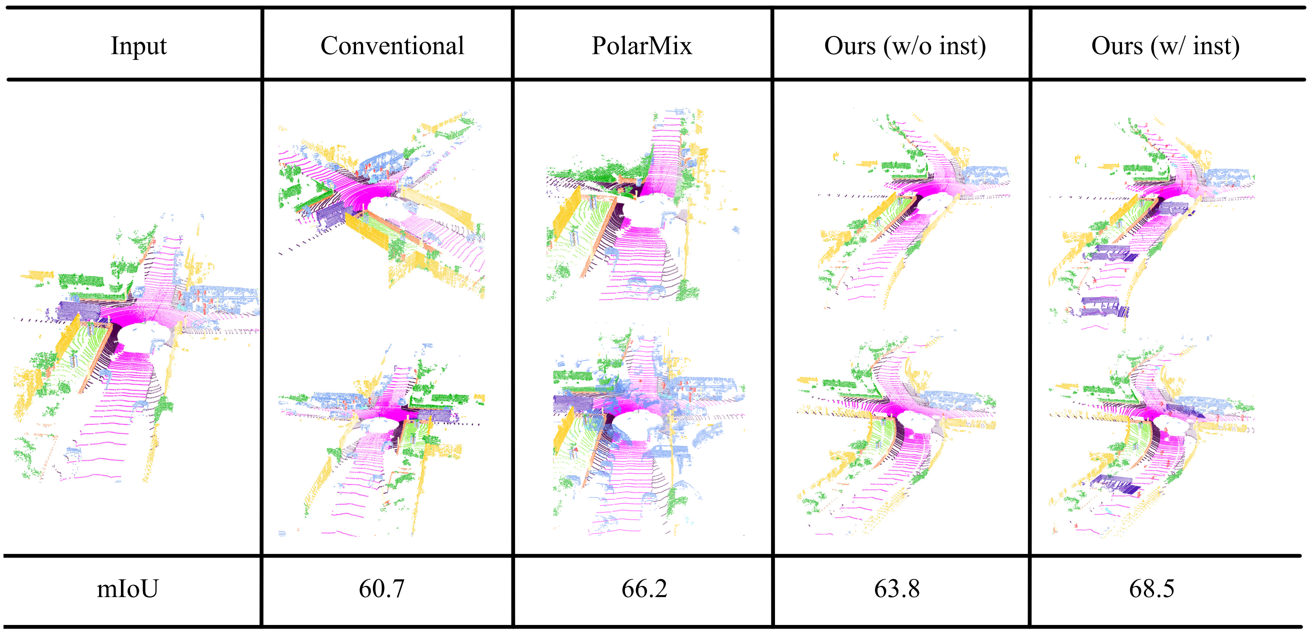

Fig. 1. The overall comparison between the conventional method, recent state-of-the-art (SOTA) method PolarMix [1], and our proposed augmentation approach with smooth deformation. The second row shows some augmented samples generated by the corresponding method. The third row is the final performance of the model trained with each augmentation approach on the SemanticKITTI validation set [2].

mapping $\{x, y, z\} \rightarrow \{x', y', z'\}$ into three different mapping pairs, making the design of the mapping functions easier and the augmentation process more controllable. Third, given the LiDAR point clouds were distributed over a wide range, we adopt a periodization to the function, allowing concentration on a localized area during augmentation function design. In addition, to address the unbalanced class in the datasets [2], [31], we further propose an effective prior-based location sampling algorithm, from which a more reasonable location can be obtained when we perform the copy-paste-based instance augmentation operation. Finally, we evaluate the effectiveness of our proposed approach on both the challenging SemanticKITTI and nuScenes datasets across different baselines and the experimental results show significant improvement in performance on both datasets. The overall comparison between the conventional method, the recent SOTA method Polarmix [1], and our proposed smooth deformation augmentation approach on the Semantic KITTI val set is shown in Figure 1.

Our contributions are summarized as follows:

1. We propose a novel LiDAR point cloud augmentation approach with smooth deformations for the semantic segmentation task, our approach can enrich the training data diversity and boost the performance of baselines effectively.

2. We propose three different strategies: residual mapping, space decoupling, and function periodization, to simplify the selection and design of the deformation augmentation functions, and also make the augmentation results more flexible and controllable.

3. We propose a simple and effective prior-based location sampling algorithm, which can place the augmented instances in a more feasible area.

4. We conduct extensive experiments and show substantial and consistent improvements in performance by adopting our proposed augmentation approach.

II. Related Work

In this section, we give a brief overview of the data augmentation approaches for point clouds and LiDAR semantic segmentation tasks. We further divide the data augmentation part into two different categories: Augmentation with Rigid Transformation and Augmentation with Deformation and divide the LiDAR semantic segmentation part into 2D-based methods, 3D-based methods, and Fusion-based Methods.

A. Data Augmentation

Augmentation with Rigid Transformation. Conventional augmentation methods including rotation, flipping, scaling, and perspective transformation are widely used in many recent works, such as [34]–[37], which are typically useful in many cases. Inspired by Mixup [25], PointMixup [26] proposed an interpolation method that produces new examples through an optimal assignment of the path function between two point clouds. Autoaugment [38] proposed to use of reinforcement learning technologies to find the best choices and orders of the augmentation actions to achieve the best performance. In Fast AutoAugment [38], a more efficient search strategy is proposed based on density matching, which does not require any back-propagation for network training for each policy evaluation. InstraBoost [39] proposed to generate a location probability map to explore the feasible locations where instances can be placed. By sampling feasible locations from the local contour similarity heatmap, a significant performance improvement can be achieved. Lidar-Aug [40] proposed a plug-and-play rendering-based LiDAR augmentation module to enrich the training data and boost the performance of the model. By leveraging the rendering technique to compose the augmented objects into the real background frames, the occlusion constraints are automatically enforced. RS-Aug [41] proposed a Realistic Simulator based data augmentation approach, and a heuristic search based object insertion scheme is also proposed to enhance rendering quality with collision and distance constraints. Both of the above methods require the use of simulators to enrich the diversity of objects. SageMix [42] proposed a saliency-guided Mixup augmentation for point clouds to make sure that the salient local structures are preserved. Polarmix [1] proposed to enrich the distribution of the point cloud and preserve the fidelity of the point cloud through two cross-scan augmentation strategies: scene-level and instance-level respectively. For the scene-level augmentation, points within the same azimuth angle range are exchanged, while for the instance-level augmentation, the points selected from another scan were rotated for multiple copies and then pasted into another scan. In [43], the authors proposed Point Augmentation (PA)-RCNN which aim for small object detection task through generating complementary features. In [44], the authors proposed to improve the robustness of LiDAR-based perception methods in adverse weather with different data augmentation techniques, which demonstrates that data augmentation can effectively enhance the model's generalization ability in different scenarios. LidarAugment [45] introduced a search-based approach for LiDAR point clouds augmentation. However, the augmentation policies in the search space still come from conventional augmentation methods. They all lack consideration of cases where smooth deformation happens in real-world scenarios.

Augmentation with Deformation. PointAugment [28] proposed an adversarial learning strategy to jointly optimize an augmented network and a task-specific network. The learnable point augmentation function was formulated with a shape-wise transformation and a point-wise displacement, and a specific loss function was carefully designed to enable the model to adjust the augmentation magnitude based on the learning state of the model for the main task dynamically, which allowed the generation augmented samples that are more suitable for any training stage of the task. In PointWOLF [17], it proposed to generate the augmented samples by locally weighted transformations centered at multiple anchor points, the method can produce diverse and realistic augmented samples with smoothly varying deformations formulated as a kernel regression. PatchAugment [27] proposed a new augmentation framework, in which different augmentation techniques were applied to different local neighborhoods. In PA-AUG [29], it also divides instances into partitions and stochastically applies five augmentation methods to each local region, then the rich information of labels can be better utilized in augmentation. 3D-VField [46] introduced a new data augmentation approach that generates reasonably deformed objects via vector fields learned in an adversarial fashion. The method is targeted for object detection task and therefore operates only on the instances level. Therefore, these methods are specifically designed for the objects, and not suitable for large-scale LiDAR point clouds.

B. LiDAR Semantic Segmentation

Point cloud semantic segmentation is one of the most fundamental tasks for autonomous driving [3], [15], [16], which aims to provide precise semantic information about the surrounding environment. As the 3D Light Detection and Ranging (LiDAR) sensor can capture more precise and farther-away distance measurements of the surrounding environment than conventional visual cameras, it has gradually become an indispensable device in many other scenarios. Currently, the methods for LiDAR semantic segmentation can be categorized into three categories: 2D-based, 3D-based, and fusion-based methods.

2D-based Methods. 2D-based Methods can also be referred aa projection-based methods, which can be further divided projection-based methods into two different categories: Range View projection (RV) and Bird's-Eye-View projection (BEV). For the range-based methods: RangeNet++ [47] proposed a deep-learning-supported approach to exploit the potential of range images and 2D convolutions, and a GPU-accelerated post-processing K-Nearest-Neighbor (KNN) approach is further proposed to recover consistent semantic information during inference for entire LiDAR scans. KPRNet [48] improved the convolutional neural network architecture for the feature extraction of the range image, and the commonly used post-processing techniques such as KNN were replaced with KPConv [49], which is a learnable pointwise component and allows for more accurate semantic class prediction. For BEV-based approaches, which are consistent with the currently popular representation in BEV space [50], [51]. PolarNet [52] proposed to use the polar Bird's-Eye-View representation to balance the spatial distribution of points in the coordinate system, and a ring convolution operation was also developed that was more suitable for the polar Bird's-Eye-View representation. Panoptic-PolarNet [34] was proposed based on PolarNet, which is a proposal-free LiDAR point cloud panoptic segmentation network and can cluster instances on top of the semantic segmentation efficiently.

3D-based Methods. 3D-based Methods can be further divided into point-based methods and voxel-based methods. For the point-based method, KPConv [49] proposed a new kind of 3D point convolution that operates on point clouds without any intermediate representation. RandLA-Net [53] proposed an efficient network architecture for directly inferring per-point semantics on large-scale 3D point clouds. It uses random point sampling instead of more complex point sampling approaches for efficiency. it also introduced a local feature aggregation module to preserve geometric details by progressively increasing the receptive field for each 3D point. For voxel-based methods, which generally achieve better performance than point-based methods. MinkNet [54] proposed a generalized 3D sparse convolution and an auto-differentiation library for sparse tensors was proposed. SPVCNN [55] proposed a lightweight 3D module that can boost the performance on the 3D scene understanding tasks effectively. In Cylinder3D [56], it introduced a novel Cylindrical and Asymmetrical 3D Convolution framework, which can effectively and robustly explore the 3D geometric pattern and tackle the difficulties

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2023.3348805

caused by sparsity and varying density of point clouds. (AF)2-S3Net [36] proposed a multibranch attentive feature fusion module in the encoder and an adaptive feature selection module with feature map re-weighting in the decoder. It fuses the voxel-based learning and point-based learning methods into a unified framework to process the potentially large 3D scene effectively.

Fusion-based Methods. As 2D-based (range and BEV) methods and 3D-based (point and voxel) methods have different advantages while suffering from their own shortcomings in the semantic segmentation task [57]. So it is intuitive to fuse information from different views together for better segmentation performance. AMVNet [58] proposed an assertion-based multiview(range, BEV) fusion network for LiDAR semantic segmentation, where the features of individual views were fused later with an assertion-guided sampling strategy. RPVNet [57] devised a deep fusion framework with multiple and mutual information interactions among three (range, point, and voxel) different views to make feature fusion more effective and flexible. GFNet [59] introduced a geometric flow network to better explore the geometric correspondence between two different views(range, BEV) in an align-before-fuse manner. CPGNet [60] proposed a cascade Point-Grid Fusion Network (CPGNet) effective feature extraction with minimal information loss and a consistency loss for better inference performance.

III. Method

In this section, we first provided the details of smooth deformation augmentation function selection and design in Section III-A, including three strategies: residual mapping, space decoupling, and function periodization. Then we describe the whole non-rigid augmentation pipeline for LiDAR point clouds in Section III-B, which also contains three different modules: the instance augmentation module, prior-based location sampling module, and scene augmentation module.

A. Deformation Function Design

The deformation augmentation function presents an expansive search space [61], [62], making the selection of appropriate augmentation functions a considerable challenge. To address this issue, we identify specific desired properties to narrow down the search space. Specifically, we categorized the desired properties from three different aspects, namely, continuity of function, scale consistency, and computational efficiency.

First, it is imperative that augmentation functions exhibit continuity or smoothness; without this attribute, instances risk disintegration after augmentation. Second, the size of the augmented instance should remain relatively consistent with its original dimensions, as the size is a critical attribute of an object. For instance, it would be incongruous to encounter a five-meter giant or a mere ten-centimeter individual. Finally, the function should preferably be computationally efficient in practice.

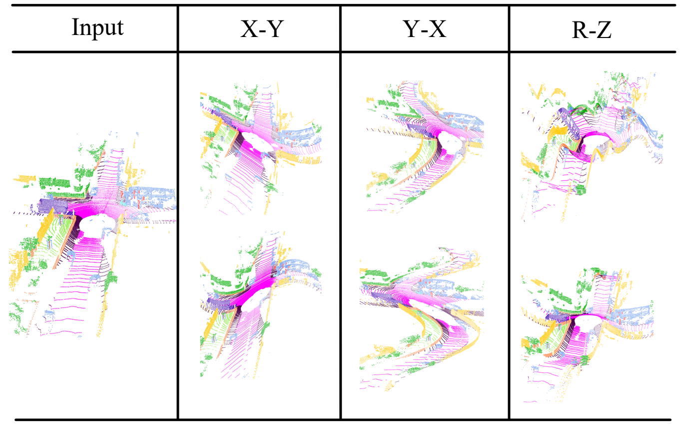

Fig. 2. The visualization of the space decoupling strategy. Here, we assume that the horizontal direction is the x-axis, and the vertical direction is the y-axis. For the second (X-Y) column, the y-coordinate is determined as a function of the x-coordinate. In the third column (Y-X), the x-coordinate is derived as a function of the y-coordinate. Lastly, in the column labeled (R-Z), the x-coordinate is defined as a function of the radius r.

1) Residual Mapping: For a deformation augmentation operation, establishing a direct mapping between the raw points and augmented points is challenging due to the vast number of points and their expansive range.

Inspired by the deep residual network [33], we propose to focus on generating the residual coordinates for each point rather than directly computing the final augmented coordinates. Specifically, for a given point cloud scan $P = \{p_0, p_1, \ldots, p_{N-1}\}$, the deformation augmentation function $\phi()$ doesn't aim to yield the target coordinates $P' = \{p_0^{'}, p_1^{'}, \ldots, p_{N-1}^{'}\}$,

we only need to compute the residual coordinates, represented by $P^{'} - P$, thus the augmentation process becomes:

$$P' = P + \alpha\phi(P), \tag{1}$$

where α represents a scale parameter. The residual coordinate generation has the following advantages: first, as mentioned in [33], the residual function is relatively simple and easier to learn, which also makes the design of the augmentation function easier. Second, the augmented samples were only affected by the residual branch, so the magnitude of augmentation can be easily controlled by α.

2) Space Decoupling: While the residual mapping strategy can make the design of the augmentation function much easier, the task remains complex due to the interdependence of spatial coordinates. Generally, the augmentation function takes the coordinates $\{x, y, z\}$ of each point as input, and the corresponding offset $\{x' - x, y' - y, z' - z\}$ is generated. The coupling between three coordinates makes controlling augmented results more difficult. However, in the real-world scenario, the $\{x, y, z\}$ three-dimensional space is not always coupled. For instance, the road may be undulating while straightforward, or a winding and twisting road but very smooth at the same time. Based on the above observations, we propose space decoupling to further simplify the augmentation function design. Specifically, for the raw coupled mapping:

$$\{x, y, z\} \rightarrow \{x', y', z'\}, \tag{2}$$

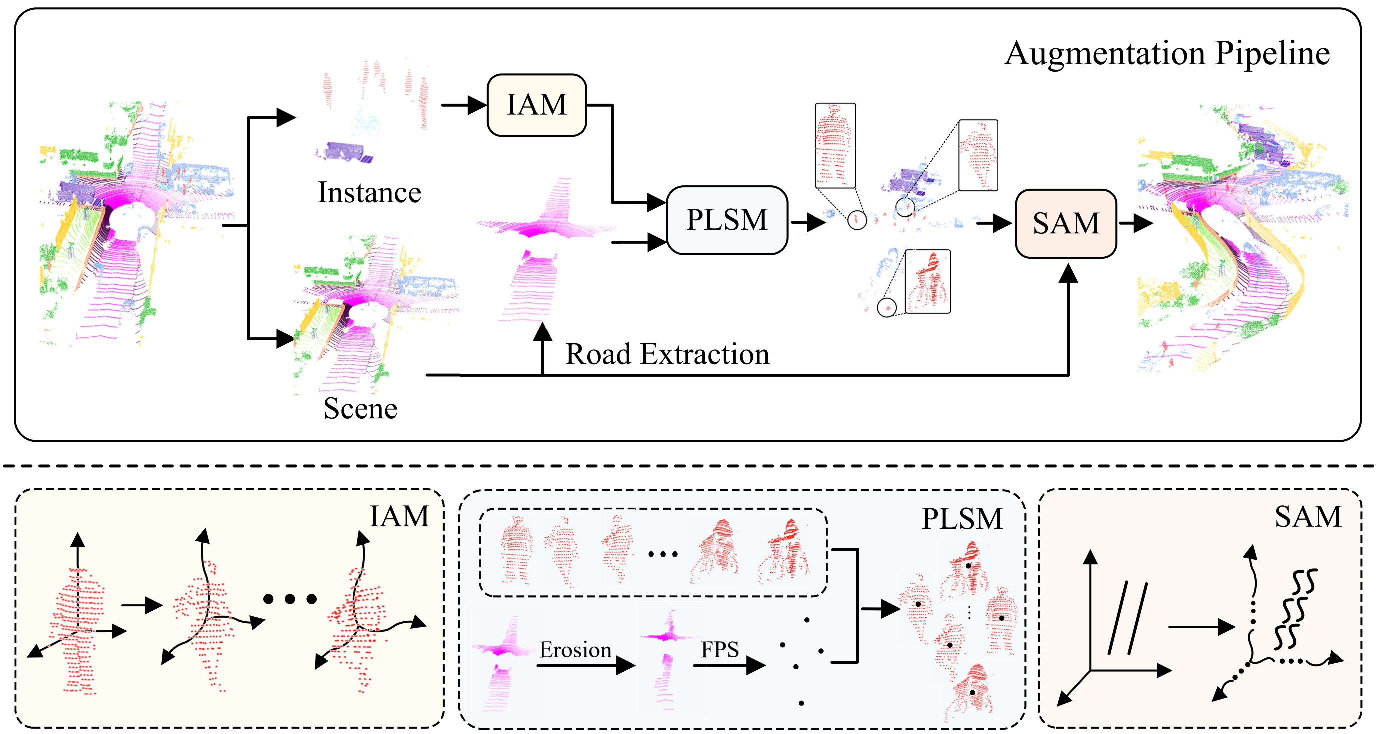

Fig. 3. The pipeline of our proposed augmentation approach. Given a scan of a point cloud, it is first divided into two different parts: 'scene' and 'things' [63]. Then Instance Aug Module (IAM) takes the instances as input, the augmented instances are input into the prior-based Location Sampling (PLSM) module and then they are placed in the sampled locations on the road area. Next, the augmented instances and the scene are combined to form a new scan with more instances. Finally, the new scan is input into the Scene Aug Module (SAM) to obtain the final augmented result. The augmented point clouds are taken as input for model training.

we decouple it into three different mapping pairs:

$$\{x \to y^{'} - y\}; \{y \to x^{'} - x\}; \{r \to z^{'} - z\}, \tag{3}$$

where $r = \sqrt{(x^2 + y^2)}$. The above equation shows that for each mapping pair, we only need to consider the impact of one dimension on another dimension, without taking into account the impact of all dimensions, which simplifies the design and selection of the augmentation function. The concept of space decoupling is illustrated in Figure 2. From the visualization, it can be seen that for each decoupled pair, the generated samples are both intuitive and plausible within real-world contexts.

3) Function Periodization: With residual mapping and space decoupling, the selection and design of the augmentation function have been largely simplified. However, as the coordinates of the input points were distributed over a wide range in the real-world scenario [2], [31], the functions for augmentations should be under control over a wide range of inputs. The commonly used functions, such as functions from the binomial family or gaussian family may not handle this situation well, as their outputs could exhibit significant variations across different regions of a point cloud.

To overcome the aforementioned challenges, we propose periodizing a simple augmentation function which tailored for a localized area to encompass the entire scan of points. Despite its simplicity, the periodization strategy greatly facilitates the selection and design of the augmentation. which also means that the selected function can be adapted to both the instances and scenes, making our method more concise and unified.

Finally, the overall deformation augmentation for the point cloud can be expressed as:

$$\begin{aligned} x' &= x + \phi(y, f_y * \xi(.), \beta_y * \xi(.)) * \alpha_y * \xi(.), \\ y' &= y + \phi(x, f_x * \xi(.), \beta_x * \xi(.)) * \alpha_x * \xi(.), \\ z' &= z + \phi(r, f_z * \xi(.), \beta_z * \xi(.)) * \alpha_z * \xi(.), \end{aligned} \tag{4}$$

where $\phi(.)$ is a periodic function, $\xi(.)$ generates random numbers between 0-1, r is the radius distance, $f_x * \xi(.)$, $f_y * \xi(.)$ and $f_z * \xi(.)$ controlling the frequency of deformation augmentation function, $\beta_x * \xi(.)$, $\beta_y * \xi(.)$ and $\beta_z * \xi(.)$ controlling the phase of the periodic function, α_x, α_y, and α_z is the amplitude of augmentation, with distinct values designated for instance and scene augmentation.

Next, we provide further descriptions and explanations about Equation 4. First, we observe that the augmentation is applied in a residual manner. Specifically, for each dimension x, y, and z, the change before and after augmentation can be simply represented as $x' = x + x_{\text{offset}}$, $y' = y + y_{\text{offset}}$, and $z' = z + z_{\text{offset}}$, respectively. Then, for the offset generation, taking x_{offset} as an example, from the first line of Equation 4, we can see that x_{offset} only depends on y among the three x, y, and z dimensions. This simplifies the generation of x_{offset} as we only need to consider the y dimension rather than interference from x and z. It should be noted that the simplification remains consistent with potential real-world scenarios like a winding, twisting, yet smooth road. Finally, since we chose $\phi(.)$ as a periodic function, we only need to ensure the offset output from $\phi(.)$ is reasonable within one period. This frees us

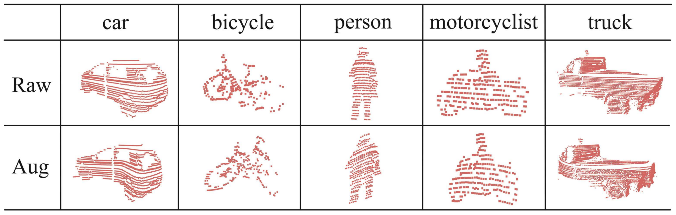

Fig. 4. Visualization of augmented instances.

from considering the large-scale distribution of LiDAR point cloud scenes when designing and selecting parameters for the augmentation function.

B. Augmentation Pipeline

The overall framework of our proposed approach is illustrated in Figure 3. It consists of three main modules: an Instance Augmentation Module (IAM), a Prior-based Location Sampling Module (PRLS), and a Scene Augmentation Module (SAM). First, the raw point cloud is separated into two different parts through the ground truth label: instance and scene [63]. For the instance branch, the separated instances are fed to the instance augmentation module for the deformation augmentation. Next, we use the prior-based location sampling module to generate multiple candidate locations. Then we paste each instance on the sampled location. Finally, the whole point cloud is fed to the scene augmentation module and the global deformation transformation is performed.

1) Instance Augmentation: As the instance may be distributed over a wide range in the point cloud, a decentralization operation is performed before the deformation augmentation operation is performed. Although the deformation augmentation operation can increase the diversity of the samples, the unbalanced count of classes in the datasets was not yet solved.

To address this, we propose to generate more samples with diversity by applying the augmentation operation on each instance multiple times with different augmentation parameters. Specifically, for each instance I, we generate more than one augmented sample $I_{aug} = \{\mathcal{F}(I, \theta_1), \mathcal{F}(I, \theta_2), \ldots, \mathcal{F}(I, \theta_k)\}$, where $\mathcal{F}(,)$ is the augmentation operation, θ_i is the different augmentation parameters, k is the number of augmented samples we want to generate. We visualize some augmented instances in Figure 4.

Then we need to paste the augmented instances into the scene. In copy-paste [24], the instances are copied from other scenes and pasted directly, the place in one scene may be inappropriate for another. In PolarMix [1], the instances were simply cut from another scan and then rotated before being pasted to the current scan. Both methodologies overlook a further exploration regarding the placement of instances. To tackle these challenges, we further propose a more reasonable and effective prior-based location sampling algorithm.

2) Prior-based Location Sampling: In [24], the authors utilize a plane equation to represent the road, ensuring that augmented instances remain grounded. While this provides a foundational approach, we go a step further for the prior-based location generation.

Algorithm 1 Semantic Segmentation Model Training Procedure with Our Proposed Augmentation Approach

Require:
Dataset $\mathcal{D}$: LiDAR points, semantic label, and panoptic label: P, Y_S, Y_P;
Deformation augmentation operation $\mathcal{F}(,)$, Parameter space Θ_1, Θ_2 for scene augmentation and instance augmentation respectively. Max augmented times for each instance n;
Initialized Segmentation Model M_{init};
Maximum training iteration MAX$_{iter}$.
Ensure: Trained Model $M_{trained}$.
while $iter <$ MAX$_{iter}$ **do**
  Sample batch of D: $B \sim \mathcal{D}$, $B = \{P_0, ..., P_{len(B)}\}$;
  Augmentation results: $R_{aug} = \emptyset$
  for P in B **do**
    Separate each scan P into scene P_S and instances P_I with ground-truth label Y_S;
    Divide P_I into separated instance: $Instance = \{I_1, I_2, ..., I_n\}$ with ground-truth label Y_P;
    // For instance augmentation //
    Initialize instance bank $IB = \emptyset$
    for inst in $Instance$ **do**
      Sampling augmentation times K in [0, n];
      for k in $range(K)$ **do**
        Sampling parameters θ_1 in Θ_1:
        $inst_{aug}$ = $\mathcal{F}(inst, \theta_1, centering = True)$
        IB.append($inst_{aug}$)
      end for
    end for
    // For prior-based FPS //
    Separate points of road P_r with label Y_S;
    Farthest Sampling $length(IB)$ points:
      $\{p_1, p_2, ..., p_{length(IB)}\}$ = Prior-based FPS(P_r)
    for i in $range(length(IB))$ **do**
      Paste instance IB[i] at p_i on scene P_S;
    end for
    // For scene augmentation //
    Sampling augmentation parameters θ_2 in Θ_2:
    $scene_{aug}$ = $\mathcal{F}(P_S, \theta_2, centering = False)$
    R_{aug}.append($scene_{aug}$)
  end for
  Input R_{aug} to model and obtain the predictions;
  Back-propagate and update parameters of the model;
  $iter$ = $iter$ + 1;
end while
Return the trained model $M_{trained}$.

Specifically, given a scan of point clouds P, first, an appropriate Region of interest (ROI) area is cropped, and only the points residing within this ROI undergo subsequent farthest point sampling operation. This methodology addresses the prevalent issue of outliers in point clouds, which is extremely unfriendly for the FPS algorithm, as the sampled points may have a large probability of being outliers. Second, we separate the points of the road by the ground-truth label and project the points to a predefined Bird's eye view(BEV) map following the

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2023.3348805

TABLE I

QUANTITATIVE COMPARISON OF MINKNET [54] AND SPVCNN [55] TRAINED WITH THE PROPOSED AUGMENTATION APPROACH AND OTHER METHODS. THE RESULTS ARE REPORTED IN TERMS OF THE MIOU ON THE SEMANTICKITTI VALIDATION SET. CGA INDICATES CONVENTIONAL GLOBAL AUGMENTATION WHICH INCLUDES RANDOM SCALING AND RANDOM ROTATION. IT CAN BE SEEN THAT OUR APPROACH CLEARLY ACHIEVES THE BEST SEMANTIC SEGMENTATION PERFORMANCE ACROSS DIFFERENT AUGMENTATION METHODS.

Methods	*car*	*bicycle*	*motorcycle*	*truck*	*other-vehicle*	*person*	*bicyclist*	*motorcyclist*	*road*	*parking*	*sidewalk*	*other-ground*	*building*	*fence*	*vegetation*	*trunk*	*terrain*	*pole*	*traffic-sign*	mIoU
MinkNet	95.9	3.7	44.9	53.2	42.1	53.7	68.9	0.0	92.8	43.0	80.0	1.8	90.5	60.0	87.4	64.5	73.3	62.1	43.7	55.9
+CGA	96.3	8.7	52.3	63.2	51.6	63.5	74.4	0.1	93.3	46.6	80.4	0.8	90.3	60.0	88.0	65.1	74.5	62.8	46.8	58.9(+3.9)
+CutMix	96.0	10.2	59.3	78.7	52.1	63.4	79.4	0.0	93.5	47.8	80.7	1.6	90.3	61.0	87.5	66.2	73.3	64.0	46.8	60.6(+5.7)
+CopyPaste	96.6	18.4	62.8	76.3	64.6	68.9	82.8	1.0	93.1	45.3	80.2	1.4	90.5	60.7	88.1	67.8	74.6	63.7	49.1	62.4(+6.5)
+Real3DAug	96.5	39.1	71.9	60.9	67.0	67.6	81.0	15.3	91.8	42.6	80.2	1.6	89.9	59.2	88.0	66.3	74.3	63.8	48.6	63.5(+7.6)
+Mix3D	96.3	29.6	61.8	68.5	55.4	72.7	77.7	1.0	94.3	52.9	81.7	0.9	89.1	55.5	88.3	69.3	74.6	65.2	50.3	62.4(+6.5)
+PolarMix	96.3	51.2	75.6	63.4	63.9	71.9	85.6	4.9	93.6	45.8	81.4	1.4	91.0	62.8	88.4	68.5	75.0	64.6	49.9	65.0 (+9.1)
+Ours	97.6	55.9	78.9	86.2	75.8	74.3	87.0	13.8	93.8	44.1	80.4	1.6	90.4	60.7	89.1	66.5	77.0	63.6	52.5	**67.9(+12.0)**
SPVCNN	94.9	9.1	55.8	66.5	33.7	61.8	75.9	0.2	93.1	45.3	79.6	0.4	91.4	62.7	87.5	66.2	72.9	62.8	42.7	58.0
+CGA	96.1	21.8	57.8	69.2	49.8	66.7	80.8	0.0	93.4	44.8	80.1	0.2	90.9	62.9	88.5	64.8	75.7	63.6	46.2	60.7(+2.7)
+CutMix	96.1	21.4	59.6	71.2	54.2	66.8	81.8	0.0	93.5	49.6	81.1	2.2	90.9	63.1	87.9	66.9	74.1	63.8	49.8	61.7(+3.7)
+CopyPaste	96.0	32.4	66.4	67.1	52.9	74.8	84.3	3.6	93.3	46.9	80.2	2.5	91.1	64.1	88.1	67.0	73.9	64.0	51.6	63.2(+5.2)
+Real3DAug	95.9	44.1	73.4	49.2	48.4	70.3	85.5	12.0	92.8	45.7	79.7	2.9	89.4	57.0	89.2	67.6	76.7	63.7	48.9	62.8(+4.8)
+Mix3D	96.0	32.4	66.4	67.1	52.9	74.8	84.3	3.6	93.3	46.9	80.2	2.5	91.1	64.1	88.1	67.0	73.9	64.0	51.6	63.7(+5.7)
+PolarMix	96.5	53.9	79.7	68.5	64.9	75.6	87.8	7.5	93.5	47.3	81.2	1.1	91.2	63.8	88.2	68.2	74.2	64.5	49.4	66.2 (+8.5)
+Ours	97.6	56.1	77.7	85.6	75.0	79.4	86.9	24.7	93.9	47.0	80.7	2.2	89.9	57.0	88.2	67.1	74.4	64.9	52.9	**68.5(+10.7)**

TABLE II

QUANTITATIVE COMPARISON OF MINKNET [54] AND SPVCNN [55] TRAINED WITH THE PROPOSED AUGMENTATION APPROACH AND OTHER METHODS. THE RESULTS ARE REPORTED IN TERMS OF THE MIOU ON THE SEMANTICKITTI TEST SET. IT CAN BE SEEN THAT OUR APPROACH CLEARLY ACHIEVES BETTER PERFORMANCE THAN POLARMIX.

Methods	*car*	*bicycle*	*motorcycle*	*truck*	*other-vehicle*	*person*	*bicyclist*	*motorcyclist*	*road*	*parking*	*sidewalk*	*other-ground*	*building*	*fence*	*vegetation*	*trunk*	*terrain*	*pole*	*traffic-sign*	mIoU
MinkNet+PM	96.2	56.5	57.1	47.9	52.3	64.7	70.3	16.8	89.2	64.5	72.8	30.0	91.4	67.0	85.8	70.8	70.6	59.7	64.9	64.7
MinkNet+Ours	97.0	59.5	59.1	49.8	56.2	66.5	69.1	46.3	89.3	64.7	72.8	28.4	92.0	67.7	85.0	70.4	69.4	60.9	64.6	**66.8(+2.1)**
SPVCNN+PM	96.2	57.4	61.1	48.3	50.6	69.2	72.8	24.3	89.0	63.3	72.7	30.3	91.0	66.1	85.3	70.7	69.8	60.6	65.5	65.5
SPVCNN+Ours	97.0	60.8	59.3	48.5	54.1	67.6	73.2	39.1	89.6	64.1	73.5	28.9	92.0	67.8	84.8	70.6	68.7	60.8	66.1	**66.6(+1.1)**

approach in Pointpillar [64]. Depending on whether the grid of maps has points or not, the grids were classified into two different states: valid and empty, which are represented by “1” and “0” respectively. Third, an erosion operation is performed on the BEV map. Without the erosion operation, the sampled locations will be mainly distributed at the boundaries of the road. We refer to this as boundary effects, which are obviously not appropriate in the real-world scenario. Finally, the Farthest Point Sampling (FPS) is adopted for generating more uniform distributed locations for instance to be pasted.

3) Scene Augmentation: The scene augmentation procedure is more simple and straightforward compared to instance augmentation. Specifically, we first paste the instances generated from IAM at the sampled location on the scene. Then the whole point clouds can be processed according to the equation 4 but with different parameter settings $\{\ f_x,\ f_y,\ f_z,\ \beta_x,\ \beta_y,\ \beta_z,\ \alpha_x,\ \alpha_y,\ \alpha_z\ \}$. We summarize the overall semantic segmentation training procedure with our proposed smooth deformation augmentation approach in Algorithm 1. Note that our method operates only on the model input data, allowing for seamless integration into the training process of current segmentation models.

IV. EXPERIMENTS

A. Dataset and Metrics

Dataset. We evaluate our proposed nonrigid augmentation approach over two LiDAR datasets of driving scenes that have been widely adopted for benchmarking in semantic segmentation. The first is SemanticKITTI [2], which is a large-scale dataset for semantic scene understanding using LiDAR sequences. It is based on the KITTI Vision Benchmark and has a dense semantic annotation for the entire KITTI Odometry Benchmark. It has a total of 43,551 scans sampled from 22 sequences and collected in different cities in Germany. In the dataset, over 21,000 are available for training (sequences 00 to 10), the rest (sequences 11 to 21) are used as the test set, and sequence 08 is often used as the validation set. It has 19 classes for training and evaluation, and the details of each class are listed in Table I. The second is nuScenes-lidarseg [31], which has 40,000 scans captured in a total of 1000 scenes of 20s duration. It is collected with a 32 beams LiDAR sensor and is sampled at 20Hz. The dataset was split into training and validation sets officially. After similar classes were merged and rare classes were removed, there remained 16 classes for the LiDAR semantic segmentation.

Evaluation Metric. To evaluate our proposed method, we follow the official guidance to leverage means intersection

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2023.3348805

TABLE III

QUANTITATIVE COMPARISON OF POLARNET [52] AND CYLINDER3D [56] TRAINED WITH THE PROPOSED AUGMENTATION APPROACH AND OTHER METHODS. THE RESULTS ARE REPORTED IN TERMS OF THE MIOU ON BOTH THE SEMANTICKITTI VALIDATION AND TEST SET. $*$ REPRESENTS THE RESULT REPRODUCED BY THE OFFICIALLY RELEASED CODE.

Methods	*car*	*bicycle*	*motorcycle*	*truck*	*other-vehicle*	*person*	*bicyclist*	*motorcyclist*	*road*	*parking*	*sidewalk*	*other-ground*	*building*	*fence*	*vegetation*	*trunk*	*terrain*	*pole*	*traffic-sign*	mIoU
PolarNet	91.5	30.7	38.8	46.4	24.0	54.1	62.2	0.0	92.4	47.1	78.0	1.8	89.1	45.5	85.4	59.6	72.3	58.1	42.2	53.6
+PolarMix (val)	92.5	37.5	47.9	74.6	34.5	61.6	76.2	0.0	92.8	43.7	78.6	2.1	90.4	54.2	84.1	53.6	66.8	59.1	41.7	57.5(+3.9)
+Ours	93.1	46.7	58.1	70.7	35.8	64.3	82.4	0.0	93.9	46.5	79.9	6.2	87.1	42.0	86.6	53.3	74.2	62.2	42.0	**58.9(+5.3)**
PolarNet	93.8	40.3	30.1	22.9	28.5	43.2	40.2	5.6	90.8	61.7	74.4	21.7	90.0	61.3	84.0	65.5	67.8	51.8	57.5	54.3
+PolarMix (test)	94.1	33.3	29.9	24.8	36.8	53.0	61.5	34.0	89.9	61.7	71.8	15.6	90.3	63.3	84.3	63.3	64.2	53.9	58.2	57.0(+2.7)
+Ours	93.8	52.3	33.8	28.9	32.3	54.1	62.6	34.0	90.4	63.0	74.0	20.3	90.6	63.4	83.9	65.9	67.7	54.7	59.6	**59.2(+4.9)**
Cylinder3D*	96.1	50.7	67.1	79.0	53.4	74.4	85.6	0.0	92.7	40.7	78.3	5.4	90.6	60.2	86.2	67.7	69.8	64.6	48.2	63.7
+PolarMix (val)	96.3	53.2	67.3	64.8	60.4	75.1	87.7	10.6	94.1	47.8	80.6	2.5	89.9	57.9	87.2	70.5	72.1	65.4	50.8	64.9(+1.2)
+Ours	95.8	52.7	78.3	79.8	56.1	76.3	84.1	14.1	94.5	43.0	81.3	0.2	90.1	56.5	86.7	68.9	71.0	65.0	52.2	**65.6(+1.9)**
Cylinder3D*	96.7	60.1	57.4	43.2	49.6	70.0	65.1	12.0	91.6	64.6	76.0	24.3	90.0	63.4	84.8	70.7	67.6	62.0	64.0	63.9
+PolarMix (test)	96.2	64.3	57.1	30.8	47.8	72.0	67.6	30.7	91.4	65.9	76.2	18.6	90.4	64.1	84.4	72.7	67.4	62.6	64.3	64.5(+0.6)
+Ours	96.4	66.0	59.1	31.2	49.8	73.8	69.4	36.5	91.8	67.0	77.0	20.6	90.7	65.0	84.9	73.4	68.2	63.6	65.2	**65.8(+1.9)**

TABLE IV

QUANTITATIVE COMPARISON OF MINKNET [54] AND SPVCNN [55] TRAINED WITH THE PROPOSED AUGMENTATION APPROACH AND OTHER METHODS ON THE NUSCENES VALIDATION SET. OUR APPROACH ACHIEVES SIGNIFICANT IMPROVEMENTS OVER THE TWO DIFFERENT BASELINES.

Methods	MinkNet	SPVCNN	FLOPs	Time (ms)
None	67.1	68.4	-	-
+CGA	70.2(+3.1)	69.1(+0.7)	1.0 M	1.27
+CutMix	70.4(+3.3)	71.7(+3.3)	0.1 M	0.02
+Copy-Paste	70.8(+3.7)	71.3(+2.9)	-	-
+Mix3D	70.1(+3.0)	70.5(+2.1)	-	-
+PolarMix	72.0(+4.9)	72.1(+3.7)	0.6 M	0.95
+Ours	**73.3(+6.2)**	**73.5(+5.1)**	2.6 M	2.13

over-union (mIoU) as the evaluation metric as defined in [2]. The evaluation metric be formulated as:

$$IoU_c = \frac{TP_c}{TP_c + FP_c + FN_c}, \tag{5}$$

where TP_c, FP_c, and FN_c represent true positive, false positive, and false negative of class c respectively. The final mIoU is the mean value of IoU over all classes in the dataset.

B. Implementation Details

In our experiments, we adopt the cosine function as the augmentation function $\phi(.)$. This function aptly aligns with the characteristics delineated in our analysis. For the $\{x \rightarrow y' - y\}$, $\{y \rightarrow x' - x\}$, and $\{r \rightarrow z' - z\}$ mapping, the wavelength $1/f_x$, $1/f_y$, and $1/f_z$ of the cosine function is randomly sampled from a uniform distribution within the interval [1/10π, 1/30π], the phase of the cosine function is randomly sampled from a uniform distribution within the interval [0, π], and amplitude of the function is randomly sampled from a uniform distribution within the interval [0, 1] and [0, 10] for the instance and scene, respectively. The max augmented times n for each instance are set to 4 in all our experiments. For the ROI crop operation in the prior-based location sampling module, we randomly select an ROI area with size 70m $\times$ 70m from the predefined range [-50m, 50m] in the training stage.

C. Experimental Results

We evaluate our proposed augmentation approach over SemanticKITTI [2] and nuScenes-lidarseg [31] datasets across MinkNet [54], SPVCNN [55], PolarNet [52] and Cylinder3D [56] baselines. We choose MinkNet and SPVCNN as they are also adopted in other augmentation methods, which allows us to make a more comprehensive and fair comparison. In addition, we further chose PolarNet and Cylinder3D to further prove the effectiveness and generalization of our proposed approach. To clarify the definition, we use CGA to represent conventional global augmentation which includes random scaling and random rotation.

For the MinkNet and SPVCNN baselines on the SemanticKITTI val set, the evaluation results are shown in Table I. It can be seen that a significant improvement can be achieved with our approach, which suppresses the baseline by 12.0% and 10.7% mIoU respectively, and suppresses the PolarMix method by 2.9% and 2.2% respectively. For the comparison results on the SemanticKITTI test set, as the annotations of test data are not available, predicted segmentation results are submitted to the online server for a fair evaluation to prevent overfitting to the test set. The result is shown in Table II. We can see that our approach achieves clear performance gain compared with the PolarMix [1]. We suppress the Polarmix by 2.1% and 1.1% on the MinkNet and SPVCNN baseline, respectively.

We also conduct experiments with the PolarNet [52] and Cylinder3D [56] baselines on both the SemanticKITTI val and test set to demonstrate the effectiveness and generalization of our approach, and the results is shown in Table III. Better performance is also achieved over the two different baselines. Specifically, we suppress the PolarNet and Cylinder3D baselines by 5.3% and 4.9% on the SemanticKITTI val set. Compared with PolarMix, we achieved an additional performance gain of 1.4% and 2.2%, respectively. On the SemanticKITTI test set, we suppress the baseline by 1.9% and 1.9%. Compared with PolarMix, we attained an additional performance gain of 0.7% and 1.3%, respectively.

To further demonstrate the generalization of our method, we further conduct experiments with the MinkNet and SPVCNN

TABLE V
QUANTITATIVE COMPARISON OF TWO MAIN CATEGORIES: INSTANCE AND SCENE. THE EXPERIMENTAL RESULTS ARE CONDUCTED ON BOTH THE SEMANTICKITTI VALIDATION AND TEST SET. FOR THE STATISTICS OF BASELINES AND POLARMIX METHODS, WE USE THE RESULTS REPORTED IN WORKS [1], [52].

Methods	val		test	
	Instance (mIoU)	Scene (mIoU)	Instance (mIoU)	Scene (mIoU)
MinkNet	45.3	63.6	-	-
+PolarMix	64.1	65.7	57.7	69.7
+Ours	71.2	65.4	62.9	69.6
SPVCNN	49.7	64.1	-	-
+PolarMix	66.8	65.7	60.0	69.5
+Ours	72.9	65.3	62.5	69.7
PolarNet	43.5	61.0	38.1	66.0
+PolarMix	53.1	60.6	45.9	65.1
+Ours	56.4	61.3	49.0	66.7
Cylinder3D	63.3	64.0	60.6	66.7
+PolarMix	67.7	62.5	61.9	66.7
+Ours	67.2	64.5	63.8	67.6

TABLE VI
QUANTITATIVE COMPARISON BETWEEN CDA, POINTWOLF [17] AND OUR SMOOTH DEFORMATION AUGMENTATION APPROACH FOR THE INSTANCE AUGMENTATION OPERATION IN THE IAM. RESULTS ARE REPORTED ON THE SEMANTICKITTI VALIDATION SET.

Method	CDA	POINTWOLF [17]	Ours
mIoU	68.0	68.4	68.5

baselines on the nuScenes-lidarseg dataset, the evaluation results are shown in Table IV. It can be seen that our approach achieves obvious improvements over the two different baseline models. We suppress the baseline model by 6.2% and 5.1% mIoU, respectively. Compared with the PolarMix, we achieve a 1.3% performance gain on the MinkNet baseline and suppress the PolarMix by 1.4% on the SPVCNN baseline.

In addition, Table IV compares computational costs and practical inference times across augmentation methods. For FLOPs, we assume 100,000 points per scan, ignoring instance operations due to their small contribution. We conducted inference experiments on an Intel(R) Core(TM) i5-12500H @ 2.50GHz CPU, running each method 100 times and averaging. Our method consumes more computational resources and CPU time than others to generate augmented samples. However, in training, we experimentally found negligible differences in speed between the augmentation methods.

Taking the experimental results analysis one step further, we categorize the mIoU performance into two main categories [2]: the Instance mIoU and the Scene mIoU. Here, we aim to compare the effectiveness of different methods on scene and instance levels, as both our approach and the PolarMix contain two main components: scene-level augmentation and instance-level augmentation. Results of the analysis are shown in Table V. It can be seen that we achieve significant improvements on the instance level as we have more specific designs for instance augmentation (IAM and PLSM) compared with scene augmentation (only scene deformation augmentation is adopted), but we still achieve competitive results for the Scene mIoU. Specifically, for the MinkNet and SPVCNN baseline, compared with PolarMix, the performance improvements on the Instance mIoU remains significant, with 7.1% and 6.1% improvements on the val set, and 5.2% and 2.5% improvements on the test set respectively. For the mIoU of scene, our advantages over Polarmix become less obvious or even slightly worse as our augmentation operation for the scene is relatively simple, but we still achieve obvious performance improvements over the baseline model, 1.8% and 1.2% improvements compared with the MinkNet and SPVCNN baseline. For the PolarNet baseline, we achieve a consistent and significant performance improvement on both the val and test set, suppress the PolarMix method by 3.3% instance mIoU and 0.7% scene mIoU on the val set, and improve the performance by 3.1% instance mIoU and 1.6% scene mIoU on the test set. For the Cylinder3D baseline method, we are slightly worse than the PolarMix by 0.5% instance mIoU on the val set, but suppress PolarMix by 2% scene mIoU. On the test set, we suppress the PolarMix method by 1.9% instance mIoU and 0.9% scene mIoU, respectively. The above experimental results demonstrate that our method holds advantages over the PolarMix in both instance and scene aspects, and also prove the effectiveness of the IAM, PLSM, and SAM modules we designed.

TABLE VII
ABLATION STUDY OF EACH COMPONENT IN OUR PROPOSED APPROACH ON THE FINAL PERFORMANCE OF THE SPVCNN MODEL ON THE SEMANTICKITTI VALIDATION SET.

SPVCNN	SAM	IAM	Copy paste	PLSM	mIoU
√					58.0
√	√				63.8
√	√		√		65.8
√	√			√	67.6
√	√	√	√		67.3
√	√	√		√	68.5

As for object-level augmentation used in the Instance Augmentation Module (IAM), we conduct experiments and make comparisons with other popular augmentation methods for instance including Conventional Data Augmentation (CDA) and POINTWOLF [17], where CDA including rotation, flipping, scaling, and point-wise jittering as in [65]. For a fair comparison, we simply replace the method used in our paper for instance augmentation with the above methods. The experimental results are shown in Table VI. It can be seen that our method has a slight advantage compared to methods tailored for object augmentation.

D. Analysis

Here, we give a more in-depth analysis of why our approach can deliver such promising performance improvement. The overall improvement in performance comes from two aspects: the performance improvement of instance classes and the performance improvement of scene classes. Correspondingly, our method can be divided into two parts: augmentation of the scene and augmentation of the instance.

Firstly, for the scene part, our method considers smooth deformations — a non-rigid point cloud transformation that has not been fully utilized in other augmentation techniques. However, such deformation transformations commonly exist in reality, like curved roads. Unlike other augmentation methods employing only rigid transformations, our smooth deformation augmentation preserves the continuity and topological structure of point clouds while simultaneously altering the 3D representations during training. We believe this critically enhances the model's generalization ability to diverse scenarios by encouraging better learning and utilization of structural information for semantic prediction. Furthermore, experimental results in Table V clearly demonstrate significant improvements in scene segmentation performance with our technique.

Regarding the instance part, two primary factors limit instance segmentation performance: data imbalance and lack of diversity. Compared to scene classes spanning hundreds of thousands of points, instance point clouds typically contain only hundreds or thousands of points. Many existing augmentation methods solely apply global transformations without considering instance-level augmentations, resulting in poor instance segmentation performance. To address these factors, we propose tailored solutions. Firstly, to mitigate data imbalance, we develop an improved instance copy-paste algorithm with prior location selection. Unlike original copy-paste randomly pasting instances scene-wide, we incorporate constraints on viable pasting locations to mimic realistic scenarios better. For instance, pedestrians and vehicles generally occupy roads; thus, we limit pasting to road areas. However, roads provide limited space, so random pasting risks instance conflicts. To resolve this, we apply farthest point sampling (FPS) of road point clouds and pasting instances across the distributed FPS locations, nearly eliminating inter-instance conflicts. Secondly, to improve instance diversity, we apply non-rigid smooth deformations, which can demonstrably enhance model generalization [17]. As Table VI shows, our smooth deformation instance augmentations further improve model performance.

E. Ablation Study

To verify the effectiveness of each component, we conduct ablation studies of the SPVCNN model on the SemanticKITTI validation set. The experimental results are shown in Table VII. We can see that only the scene augmentation can obtain a 5.8% mIoU improvement, and with the copy-paste instance augmentation, the performance can be improved by 2% mIoU. Together with our instance deformation augmentation, the performance can be further improved by 1.5% mIoU. When the copy-pasted instance augmentation operation is replaced with our prior-based location sampling module, the performance can obtain another 1.2% mIoU improvement. All the above results demonstrate the effectiveness of each component of our proposed approach.

F. Sensitivity Analysis

We further conduct a sensitivity analysis for the frequency parameters f_x, f_y, f_z, and amplitude parameters $\alpha_x, \alpha_y, \alpha_z$ in equation 4, as these are the very crucial parameter for our proposed approach. For simplicity, in the experiments, we performed scaling by multiples based on the default experimental parameter settings as mentioned in Section IV-B. The experiment results are shown in Figure 6.

In the experiments, we have established four distinct sets of coefficients for the sensitivity analysis, $0.25\times$, $0.5\times$, $1\times$, and $2\times$, respectively. Notably, the performance for both the frequency parameters f_* and amplitude parameters a_* slightly declined on both the left and right of $1\times$, which is also the default parameter setting chosen for our experiments. The best performance is always obtained at $1\times$ location.

G. Visualization

To demonstrate the model's capability in responding to scene changes after training with our approach, we visualize changes in feature representations for point clouds before and after augmentation for both instance and scene classes, and also make comparisons with the baseline model. Notably, for the road class, we employ the erosion operation mentioned in Section III-B2, removing boundary areas since features are easily affected by surroundings in these areas. The results are shown in Figure 5. It can be seen that after training with our augmentation approach, the extracted features exhibit better consistency despite structural changes in objects and scenes. We believe that the consistency of feature representation is crucial for achieving robustness segmentation results in real-world scenarios.

In addition, to provide a more intuitive understanding of our augmentation approach, we visualize different scenes before and after augmentation in Figure 7. We show five augmentation results for each scene generated from different augmentation parameters. The first three columns mainly show the augmentation in the $x-y$ plane, while the last two columns show z-direction augmentation. The results demonstrate that our approach can generate a wide variety of samples, which will be highly beneficial for model learning.

Finally, Figure 8 visualizes segmentation results for qualitative comparison. In the first two rows, our augmentation approach provides correct segmentations while PolarMix incorrectly labels instances. Specifically, in the first row, PolarMix incorrectly classifies the *other-vehicle* as a *car*. In the second row, it mislabels the *fence* as a *building*. Notably, PolarMix inconsistently labels classes bearing similarity, requiring precise discrimination between vehicles or man-made structures. Our augmentations enable a deeper understanding of subtle inter-class differences, improving generalization. Both methods incorrectly predict parts of *other-ground* as *vegetation* or *terrain* in the third row — a challenging case even for humans lacking scene context. Further augmentation advances targeting contextual reasoning could provide correct predictions. Overall, qualitative results demonstrate that our augmentations produce consistently accurate segmentations compared with PolarMix, overcoming limitations posed by small inter-class discrepancies. However, complex scenes warrant further augmentation to deeply understand relationships and ambiguity.

V. Conclusion

In this work, we proposed a novel and effective augmentation approach with smooth deformation for the LiDAR point

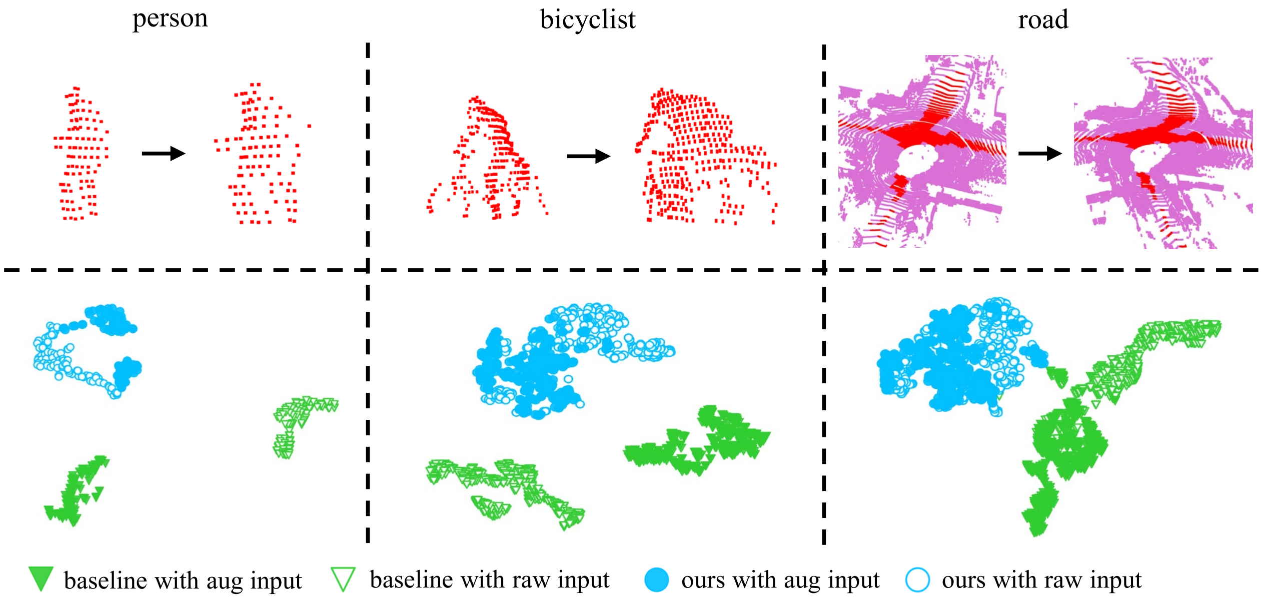

Fig. 5. Visualization of the spatial distribution of features for both instances and scenes before and after deformation augmentation. The first row shows the point cloud before and after augmentation. The second row shows t-SNE visualization of corresponding point cloud features. Green and blue represent results from the baseline and proposed model, respectively. Solid shapes indicate models take point clouds after augmentation as input, while hollow shapes indicate models take point clouds before augmentation as input.

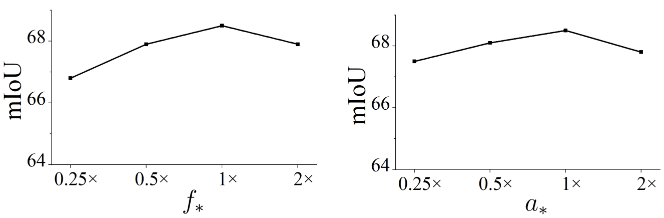

Fig. 6. Sensitivity analysis on the augmentation frequency parameters f_* (left) and amplitude parameters a_* (right) in equation 4. For convenience, the default parameter settings in Section IV-B are used as the baseline (1×). Experiments are then conducted by scaling the default settings by factors of 0.25, 0.5 and 2, respectively.

cloud semantic segmentation task. The overall augmentation pipeline has two main components: scene augmentation and instance augmentation. To simplify the selection and design of the smooth deformation augmentation functions and make the augmentation results more flexible and controllable, three different effective strategies were proposed: residual mapping, spacing decoupling, and function periodization, respectively. We also propose an effective prior-based location sampling algorithm that aims to paste the augmented instance on a more feasible area in the scene. As a result, our approach can enrich the diversity of training data and boost the performance of various baselines consistently and significantly. Finally, we conduct extensive experiments on both the SemanticKITTI and nuScenes challenging datasets. The experimental results demonstrate that our method shows an obvious advantage compared to other methods.

REFERENCES

[1] A. Xiao, J. Huang, D. Guan, K. Cui, S. Lu, and L. Shao, "Polarmix: A general data augmentation technique for lidar point clouds," *NeurIPS*, vol. 35, pp. 11 035–11 048, 2022.

[2] 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 *ICCV*, 2019, pp. 9297–9307.

[3] J. Mei and H. Zhao, "Incorporating human domain knowledge in 3-d lidar-based semantic segmentation," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 2, pp. 178–187, 2019.

[4] S. Qiu, F. Jiang, H. Zhang, X. Xue, and J. Pu, "Multi-to-single knowledge distillation for point cloud semantic segmentation," in *ICRA*, 2023, pp. 9303–9309.

[5] Z. Zhao, W. Zhang, J. Gu, J. Yang, and K. Huang, "Lidar mapping optimization based on lightweight semantic segmentation," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 3, pp. 353–362, 2019.

[6] Z. Wang, J. Guo, H. Zhang, R. Wan, J. Zhang, and J. Pu, "Bridging the gap: Improving domain generalization in trajectory prediction," *IEEE Transactions on Intelligent Vehicles*, pp. 1–13, 2023.

[7] S. Teng, X. Hu, P. Deng, B. Li, Y. Li, Y. Ai, D. Yang, L. Li, Z. Xuanyuan, F. Zhu *et al.*, "Motion planning for autonomous driving: The state of the art and future perspectives," *IEEE Transactions on Intelligent Vehicles*, 2023.

[8] J. Yoo, N. Ahn, and K.-A. Sohn, "Rethinking data augmentation for image super-resolution: A comprehensive analysis and a new strategy," in *CVPR*, 2020, pp. 8375–8384.

[9] C. Luo, Y. Zhu, L. Jin, and Y. Wang, "Learn to augment: Joint data augmentation and network optimization for text recognition," in *CVPR*, 2020, pp. 13 746–13 755.

[10] A. Dabouei, S. Soleymani, F. Taherkhani, and N. M. Nasrabadi, "Supermix: Supervising the mixing data augmentation," in *CVPR*, 2021, pp. 13 794–13 803.

[11] S. Li, K. Gong, C. H. Liu, Y. Wang, F. Qiao, and X. Cheng, "Metasaug: Meta semantic augmentation for long-tailed visual recognition," in *CVPR*, 2021, pp. 5212–5221.

[12] C. Wang, C. Ma, M. Zhu, and X. Yang, "Pointaugmenting: Cross-modal augmentation for 3d object detection," in *CVPR*, 2021, pp. 11 794–11 803.

[13] 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, 2021.

[14] G. Xian, C. Ji, L. Zhou, G. Chen, J. Zhang, B. Li, X. Xue, and J. Pu, "Location-guided lidar-based panoptic segmentation for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1473–1483, 2022.

[15] Y. Qian, X. Wang, Z. Chen, C. Wang, and M. Yang, "Hy-seg: A hybrid

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2023.3348805

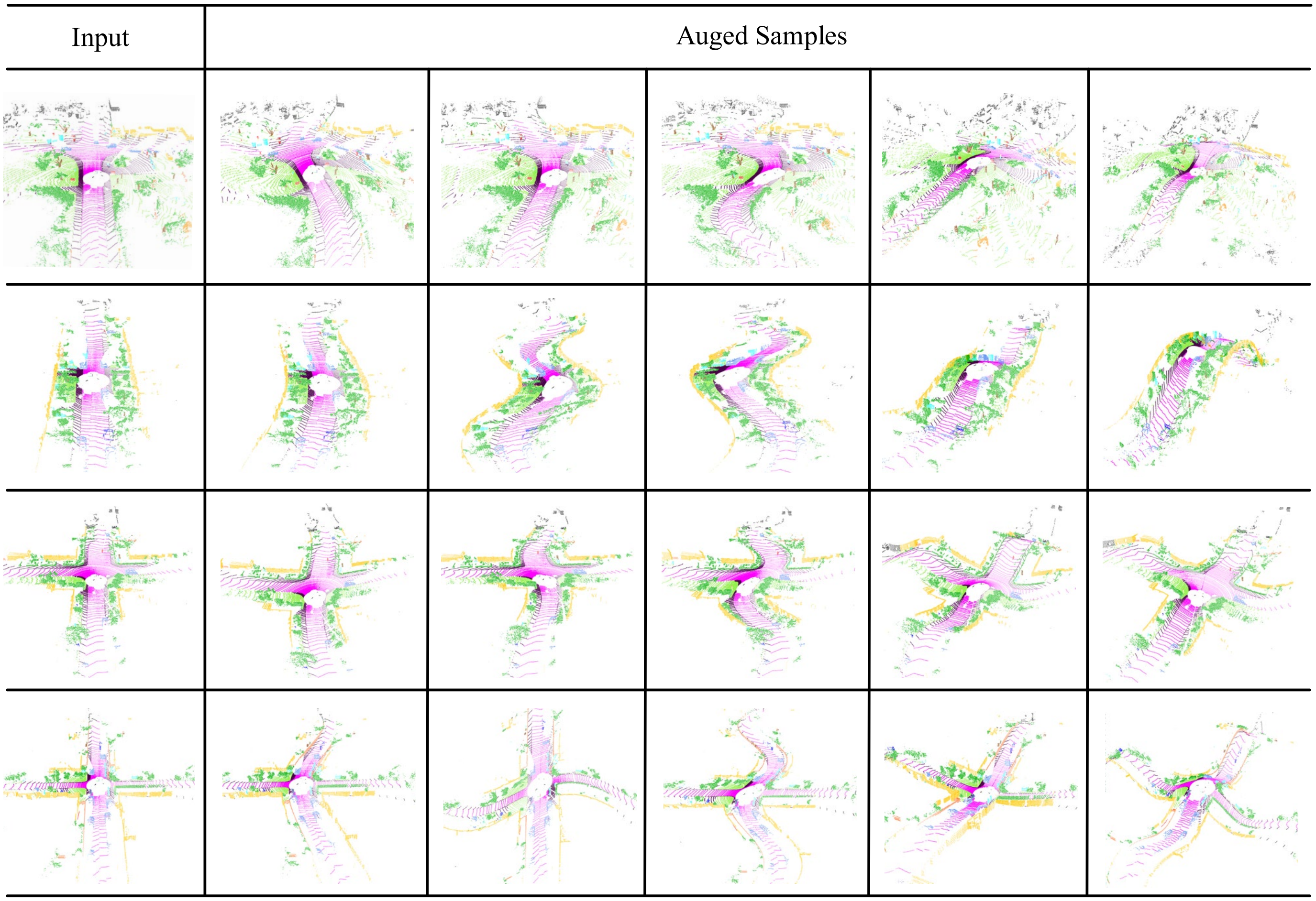

Fig. 7. Visualization of some augmented scenes. For each scene, we show five different augmentation results generated from different parameters. The first three columns mainly show the augmentation on the x-y plane, while the latter two columns mainly show the augmentation along the z direction. It can be seen that our method can generate a large number of samples with extensive diversity.

method for ground segmentation using point clouds," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1597–1606, 2022.

[16] P. Ni, X. Li, W. Xu, D. Kong, Y. Hu, and K. Wei, "Robust 3d semantic segmentation based on multi-phase multi-modal fusion for intelligent vehicles," *IEEE Transactions on Intelligent Vehicles*, 2023.

[17] S. Kim, S. Lee, D. Hwang, J. Lee, S. J. Hwang, and H. J. Kim, "Point cloud augmentation with weighted local transformations," in *ICCV*, 2021, pp. 548–557.

[18] E. Hoffer, T. Ben-Nun, I. Hubara, N. Giladi, T. Hoefler, and D. Soudry, "Augment your batch: Improving generalization through instance repetition," in *CVPR*, 2020, pp. 8129–8138.

[19] C. Gong, D. Wang, M. Li, V. Chandra, and Q. Liu, "Keepaugment: A simple information-preserving data augmentation approach," in *CVPR*, 2021, pp. 1055–1064.

[20] X. Zhang, N. Tseng, A. Syed, R. Bhasin, and N. Jaipuria, "Simbar: Single image-based scene relighting for effective data augmentation for automated driving vision tasks," in *CVPR*, 2022, pp. 3718–3728.

[21] S. Li, M. Xie, K. Gong, C. H. Liu, Y. Wang, and W. Li, "Transferable semantic augmentation for domain adaptation," in *CVPR*, 2021, pp. 11 516–11 525.

[22] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *ICCV*, 2019, pp. 6023–6032.

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

[24] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *CVPR*, 2021, pp. 2918–2928.

[25] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *ICLR*, 2018.

[26] Y. Chen, V. T. Hu, E. Gavves, T. Mensink, P. Mettes, P. Yang, and C. G. Snoek, "Pointmixup: Augmentation for point clouds," in *ECCV*, 2020, pp. 330–345.

[27] S. V. Sheshappanavar, V. V. Singh, and C. Kambhamettu, "Patchaugment: Local neighborhood augmentation in point cloud classification," in *ICCV*, 2021, pp. 2118–2127.

[28] R. Li, X. Li, P.-A. Heng, and C.-W. Fu, "Pointaugment: an auto-augmentation framework for point cloud classification," in *CVPR*, 2020, pp. 6378–6387.

[29] J. Choi, Y. Song, and N. Kwak, "Part-aware data augmentation for 3d object detection in point cloud," in *IROS*, 2021, pp. 3391–3397.

[30] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012, pp. 3354–3361.

[31] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *CVPR*, 2020, pp. 11 621–11 631.

[32] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *CVPR*, 2020, pp. 2446–2454.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[34] Z. Zhou, Y. Zhang, and H. Foroosh, "Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation," in *CVPR*, 2021, pp. 13 194–13 203.

[35] M. Hahner, D. Dai, A. Liniger, and L. Van Gool, "Quantifying data

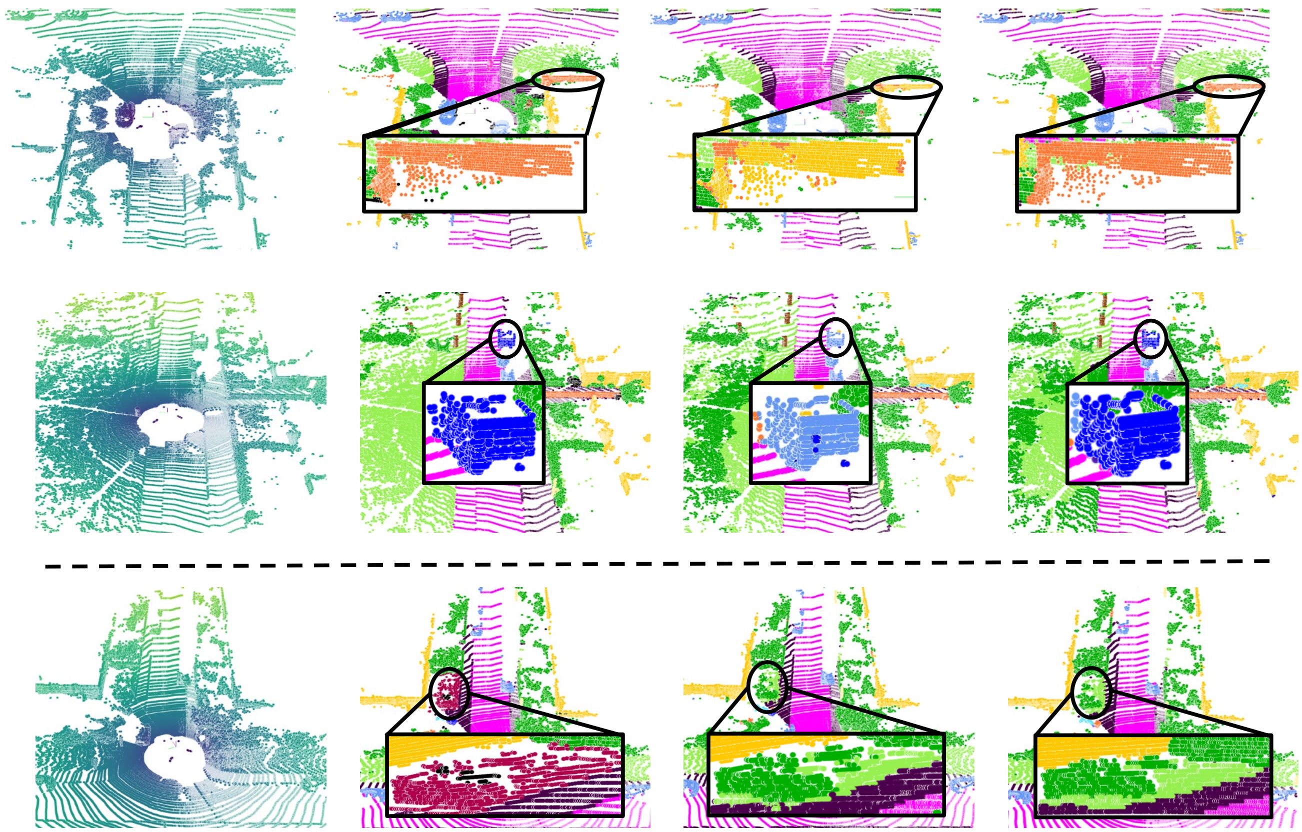

Fig. 8. Visualization of the segmentation results of different methods. The first two rows show the performance of the model trained with our approach outperforming PolarMix, and in the third row, we present a failure case for both PolarMix and our method.

augmentation for lidar based 3d object detection," *arXiv preprint arXiv:2004.01643*, 2020.

[36] R. Cheng, R. Razani, E. Taghavi, E. Li, and B. Liu, "2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network," in *CVPR*, 2021, pp. 12 547–12 556.

[37] M. Ye, R. Wan, S. Xu, T. Cao, and Q. Chen, "Drinet++: Efficient voxel-as-point point cloud segmentation," *arXiv preprint arXiv:2111.08318*, 2021.

[38] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *CVPR*, 2019, pp. 113–123.

[39] H.-S. Fang, J. Sun, R. Wang, M. Gou, Y.-L. Li, and C. Lu, "Instaboost: Boosting instance segmentation via probability map guided copy-pasting," in *ICCV*, 2019, pp. 682–691.

[40] J. Fang, X. Zuo, D. Zhou, S. Jin, S. Wang, and L. Zhang, "Lidar-aug: A general rendering-based augmentation framework for 3d object detection," in *CVPR*, 2021, pp. 4710–4720.

[41] P. An, J. Liang, J. Ma, Y. Chen, L. Wang, Y. Yang, and Q. Liu, "Rs-aug: Improve 3d object detection on lidar with realistic simulator based data augmentation," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[42] S. Lee, M. Jeon, I. Kim, Y. Xiong, and H. J. Kim, "Sagemix: Saliency-guided mixup for point clouds," *arXiv preprint arXiv:2210.06944*, 2022.

[43] W. Lu, D. Zhao, C. Premebida, L. Zhang, W. Zhao, and D. Tian, "Improving 3d vulnerable road user detection with point augmentation," *IEEE Transactions on Intelligent Vehicles*, 2023.

[44] S. Teufel, J. Gamerdinger, G. Volk, C. Gerum, and O. Bringmann, "Enhancing robustness of lidar-based perception in adverse weather using point cloud augmentations," in *2023 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2023, pp. 1–6.

[45] Z. Leng, G. Li, C. Liu, E. D. Cubuk, P. Sun, T. He, D. Anguelov, and M. Tan, "Lidar augment: Searching for scalable 3d lidar data augmentations," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7039–7045.

[46] A. Lehner, S. Gasperini, A. Marcos-Ramiro, M. Schmidt, M.-A. N. Mahani, N. Navab, B. Busam, and F. Tombari, "3d-vfield: Adversarial augmentation of point clouds for domain generalization in 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 295–17 304.

[47] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "Rangenet++: Fast and accurate lidar semantic segmentation," in *IROS*, 2019, pp. 4213–4220.

[48] D. Kochanov, F. K. Nejadasl, and O. Booij, "Kprnet: Improving projection-based lidar semantic segmentation," *arXiv preprint arXiv:2007.12668*, 2020.

[49] 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 *ICCV*, 2019, pp. 6411–6420.

[50] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *ECCV*, 2022, pp. 1–18.

[51] C. Yang, Y. Chen, H. Tian, C. Tao, X. Zhu, Z. Zhang, G. Huang, H. Li, Y. Qiao, L. Lu *et al.*, "Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision," in *CVPR*, 2023, pp. 17 830–17 839.

[52] 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 *CVPR*, 2020, pp. 9601–9610.

[53] 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 *CVPR*, 2020, pp. 11 108–11 117.

[54] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *CVPR*, 2019, pp. 3075–3084.

[55] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, and S. Han, "Searching

efficient 3d architectures with sparse point-voxel convolution," in *ECCV*, 2020, pp. 685–702.

[56] 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 *CVPR*, 2021, pp. 9939–9948.

[57] J. Xu, R. Zhang, J. Dou, Y. Zhu, J. Sun, and S. Pu, "Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation," in *CVPR*, 2021, pp. 16 024–16 033.

[58] V. E. Liong, T. N. T. Nguyen, S. Widjaja, D. Sharma, and Z. J. Chong, "Amvnet: Assertion-based multi-view fusion network for lidar semantic segmentation," *arXiv preprint arXiv:2012.04934*, 2020.

[59] H. Qiu, B. Yu, and D. Tao, "Gfnet: Geometric flow network for 3d point cloud semantic segmentation," *arXiv preprint arXiv:2207.02605*, 2022.

[60] X. Li, G. Zhang, H. Pan, and Z. Wang, "Cpgnet: Cascade point-grid fusion network for real-time lidar semantic segmentation," in *ICRA*. IEEE, 2022, pp. 11 117–11 123.

[61] E. Zeidler, *Nonlinear functional analysis and its applications: III: variational methods and optimization*. Springer Science & Business Media, 2013.

[62] N. S. Papageorgiou and P. Winkert, *Applied Nonlinear Functional Analysis: An Introduction*. Walter de Gruyter GmbH & Co KG, 2018.

[63] M. Aygun, A. Osep, M. Weber, M. Maximov, C. Stachniss, J. Behley, and L. Leal-Taixé, "4d panoptic lidar segmentation," in *CVPR*, 2021, pp. 5527–5537.

[64] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *CVPR*, 2019, pp. 12 697–12 705.

[65] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *NeurIPS*, vol. 30, 2017.

Shoumeng Qiu received the Master's degree from Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, and University of Chinese Academy of Sciences, China, in 2021. He is currently a Ph.D. student at Fudan University. His current research interests include 3D point cloud understanding, semantic segmentation, machine learning and artificial intelligence.

Jie Chen received the B.S. degree from Xiamen University, Xiamen, China, in 2017. He is currently a Ph.D. student with the School of Computer Science, Fudan University, Shanghai, China. His research interests include machine/deep learning, graph neural networks and causality.

Chenghang Lai received the M.S. degree in computer science from the University of Central China Normal UniversityWuHan, China, in 2021. He is currently working toward the PhD degree with the School of Computer Science, Fudan University. His research interests include deep learning, machine learning, and multimodal knowledge extraction and reasoning.

Hong Lu received the BEng and MEng degrees in computer science and technology from Xidian University, Xi'an, China, in 1993 and 1998, respectively, and the PhD degree from Nanyang Technological University, Singapore, in 2005. From 1993 to 2000, she was a lecturer and researcher in the School of Computer Science and Technology, Xidian University. From 2000 to 2003, she was a research student in the School of Electrical and Electronic Engineering, Nanyang Technological University. Since 2004, she has been with the School of Computer Science, Fudan University, Shanghai, China, where she is currently an associate professor. Her current research interests include image and video processing, computer vision, machine learning, and pattern recognition.

Xiangyang Xue received the BS, MS, and PhD degrees in communication engineering from Xidian University, Xi'an, China, in 1989, 1992, and 1995, respectively. He is currently a professor of computer science with Fudan University, Shanghai, China. His research interests include multimedia information processing and machine learning.

Jian Pu received a Ph.D. degree from Fudan University, Shanghai, China, in 2014. Currently, he is a Young Principal Investigator at the Institute of Science and Technology for Brain-Inspired Intelligence (ISTBI), Fudan University. He was an associate professor at the School of Computer Science and Software Engineering, East China Normal University from 2016 to 2019, and a postdoctoral researcher of the Institute of Neuroscience, Chinese Academy of Sciences in China from 2014 to 2016. His current research interest is to develop machine learning and computer vision methods for autonomous driving.