Class-Balanced PolarMix for Data Augmentation of 3D LIDAR Point Clouds Semantic Segmentation

Bo Liu[1,2], *Xiao Qi*[3*]

[1] *School of Computer Science and Artificial Intelligence, Chaohu University, China*
[2] *School of Computer Science and Engineering, Macau University of Science and Technology, Macau*
[3] *Department of Information Technology and Cybersecurity, Shanghai Police College, China*
liubo@chu.edu.cn, qixiao0513@sina.com

Abstract

3D LIDAR point clouds are extensively utilized in various domains, and data augmentation techniques for these point clouds can enhance network model convergence during training while also reducing the requisite data volume. Notably, PolarMix represents a seminal contribution to data enhancement in the realm of 3D LIDAR point Clouds Semantic Segmentation. It markedly augments the number of instances per class through swapping and rotate-paste mechanisms. Rotate-paste involves rotating and pasting selected class instances around the Z-axis multiple times. However, when capturing real-world scenarios using LiDAR point clouds, a pronounced class imbalance is observed, wherein certain classes dominate in sample numbers, while others are sparsely represented. Regrettably, PolarMix overlooks this class imbalance, leading to unequal treatment of all classes. To rectify this, we introduce the Class-Balanced PolarMix (CB-PolarMix), which operates in a cascading manner to diversify the training distribution and further optimize data augmentation outcomes. The cornerstone of CB-PolarMix lies in its adaptive reinforcement of foreground classes based on their distribution patterns. More specifically, our approach tweaks the pasting process for each class contingent upon its historical prediction accuracy. Experimental results from the SemanticPOSS and SemanticKitti datasets, utilizing the MinkowskiNet and SPVCNN models respectively, underscore the efficacy of the proposed CB-PolarMix.

Keywords: 3D LIDAR point cloud, Data augmentation, PolarMix, Class-Balanced, Semantic segmentation

1 Introduction

3D LIDAR point cloud refers to a set of three-dimensional data points captured by lidar sensors, which are used to represent objects and their surroundings in a virtual environment. 3D LIDAR point clouds are generated through a process called lidar detection, which emits radio waves and captures the reflections of these waves off objects in the environment. The reflected signals are then processed to determine the distance, speed, and direction of the objects, creating a point cloud of three-dimensional data. This data can be collected using various types of lidar sensors which can capture high-resolution point clouds with accuracy and precision. The organizational structure of 3D LIDAR point clouds typically consists of a large number of individual data points, each with its own set of coordinates in a three-dimensional space. Additionally, various algorithms and techniques can be applied to the point cloud data to extract meaningful information, such as object classification, segmentation, and tracking.

Effective data augmentation techniques play a crucial role in enhancing the performance of semantic segmentation tasks for 3D point clouds. These methods have the potential to increase the accuracy, robustness, and overall performance of the segmentation model. The method commonly used now is still global augmentation, which cannot operate across samples and cannot focus on augmenting local areas. The recently emerged Polarmix [1] approach stands out as a significant milestone in the field of data enhancement for 3D LIDAR point Clouds Semantic Segmentation. This technique boosts the count of each instance by employing swap and rotate-paste operations, thereby achieving notable outcomes. The rotate-paste procedure involves rotating and pasting selected class instances multiple times around the Z-axis. However, the Polarmix method faces challenges due to the significant class imbalance evident in semantic segmentation tasks. Consequently, the accuracy of semantic segmentation varies considerably among different classes. For instance, certain classes may exhibit an accuracy exceeding 90%, while others struggle with accuracy levels below 10%. One limitation of the Polarix method is its uniform application of rotate-paste for class instances, which does not account for class imbalance.

To address this limitation and account for class imbalance, we introduce the CB-PolarMix method. This approach is designed as a cascade cycle method, aiming to mitigate the issues associated with class imbalance in semantic segmentation tasks. By considering the disproportionate distribution of instances among different classes, the CB-PolarMix method offers a more equitable and effective way to enhance the accuracy of semantic segmentation across all classes. In essence, the CB-

*Corresponding Author: Xiao Qi; E-mail: qixiao0513@sina.com
DOI: https://doi.org/10.70003/160792642025012601006

PolarMix method represents an advancement in data augmentation [2-3] techniques for 3D point clouds, particularly in addressing class imbalance challenges. Through its cascade cycle approach, it offers a promising avenue to improve the overall performance of semantic segmentation tasks. Figure 1 shows our general improvement idea, which is to introduce a class balance strategy into the traditional PolarMix method.

The main contributions of this research are three-fold:

(1) Unbalanced class distribution issue in recent PolarMix: Regarding the task of semantic segmentation for 3D point clouds, we have observed the presence of an unbalanced class distribution issue in the recent PolarMix approach. Consequently, we have initiated a study to address and progressively resolve this problem.

(2) CB-PolarMix Method: We propose a novel strategy called CB-PolarMix. Upon examining the semantic segmentation results from the original PolarMix, we observed a significant fluctuation in precision across different classes. Based on these findings and leveraging the original PolarMix, we propose to paste a higher number of instances for classes exhibiting lower segmentation precision. This approach is intended to enhance the model's learning capacity by providing an increased volume of learning samples.

(3) Performance improvement: Based on the MinkowskiNet [4] and SPVCNN [5] models, we performed experiments on both the semanticKitti [6] and semanticPOSS [7] datasets. The experimental results show that the CB-PolarMix method we proposed performs better than the original PolarMix method in all the experiments conducted.

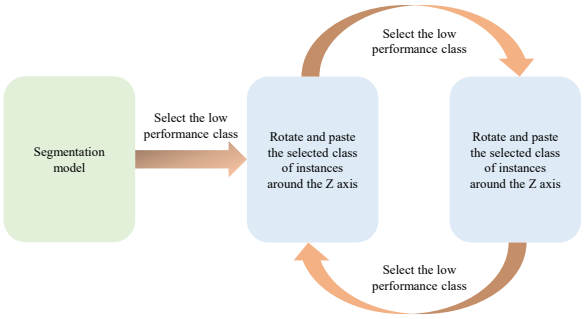


Figure 1. Schematic diagram of CB-PolarMix based on cyclic cascade way
(The process begins with applying segmentation model (such as a model with the original PolarMix data augmentation) followed by a class balancing strategy. Classes with low semantic segmentation performance are selected for further processing. This cyclical process of rotating and pasting underperforming classes continues until optimal results are achieved, thus improving the overall semantic segmentation performance.)

2 Related Work

3D LIDAR point cloud datasets. The field of 3D point cloud technology has been rapidly advancing, with numerous 3D LIDAR point cloud datasets emerging. Particularly noteworthy is the swift development of datasets specifically designed for semantic segmentation tasks.

In 2019, Jens Behley et al. made a significant contribution with the SemanticKITTI dataset. This large-scale point cloud dataset was designed with the goal of facilitating semantic segmentation and scene understanding tasks. It was derived from the KITTI [8] Vision Benchmark Suite and provides granular semantic annotations for each point in the point clouds. With 28 diverse semantic classes such as car, building, person, vehicle, bicycle, and more, the dataset provides a comprehensive resource for researchers. The SemanticKITTI dataset, gathered from real-world traffic scenes in Germany, employs a Velodyne HDL-64E S3 LIDAR mounted on a vehicle to collect data.

Following this, in 2020, Yancheng Pan et al. introduced the SemanticPOSS dataset. This dataset was collected at Peking University in China and offers instance-level annotations. It comprises 6 road sequences, labelled from 00 to 05, and includes a total of 2988 diverse LIDAR scans with point-wise labeling across 14 classes. The campus scenes captured in the SemanticPOSS dataset showcase a wide variety of features, including pedestrians, riders, cars, and more, providing a rich resource for research.

In the same year, Holger Caesar et al. presented the nuScenes [9] dataset, further expanding the available resources. This comprehensive and large-scale dataset, designed for autonomous driving research and development, encapsulates data from various weather conditions, lighting conditions, and different times of the day, captured across multiple cities. The nuScenes lidarseg dataset, focusing specifically on semantic segmentation, assigns semantic labels to individual points in the 3D point cloud with predefined semantic classes such as car, pedestrian, cyclist, road, sidewalk, and more. The dataset provides more than 1,000 scenes, each containing multiple sweeps of a LiDAR sensor mounted on a moving vehicle, synchronized with other sensor modalities including cameras and lidars for multimodal analysis and fusion.

In 2022, Aoran Xiao et al. introduced the SemanticLiDAR [10] dataset, a robust synthetic LiDAR dataset. This dataset, rich in detail and scope, features point-wise annotated point clouds which are accurately shaped geometrically and comprehensively categorized into semantic classes. The SemanticLiDAR dataset, collected from a variety of virtual environments with diverse scenes and layouts, boasts over 19 billion points distributed across 32 semantic classes. The richness and diversity of this dataset make it a crucial resource in the field of LiDAR technology.

Further enriching the field in 2023, Aoran Xiao et al. proposed the SemanticSTF [11] dataset. This adverse-weather point cloud dataset features dense point-level annotations, and is designed to facilitate the study of 3D Semantic Segmentation under various adverse weather conditions. The SemanticSTF dataset extends the STF [12] Detection Benchmark by providing point-wise annotations of 21 semantic classes under four typical adverse weather conditions frequently encountered in autonomous driving:

dense fog, light fog, snow, and rain.

Semantic segmentation network model for 3D LIDAR point cloud. 3D semantic segmentation, a process that assigns point-wise semantic labels to point clouds, has been gaining significant attention. This is largely due to the rapid advancements in artificial neural networks in recent years. As a result, we have witnessed a surge in the emergence of abundant and diverse models specifically designed for 3D semantic segmentation.

PointNet [13] is a pioneering deep learning architecture designed specifically for processing point clouds, which are sets of points in a 3D coordinate system. These point clouds are commonly used in the field of computer vision and are particularly relevant in robotics and 3D object recognition. The groundbreaking aspect of PointNet is its ability to directly consume raw point cloud data, while preserving the distinctive properties of points in the 3D space, such as invariance to ordering and transformations. The architecture of PointNet consists of a series of layers of multi-layer perceptrons (MLPs) and max-pooling layers. The MLPs are used to extract local features from each point, and the max-pooling layers are used to capture the most prominent feature across the point cloud, effectively providing a form of global information.

PointNet++ [14] is an extension of the original PointNet architecture. It was developed to address some of the limitations of PointNet, specifically its inability to capture local structures induced by the metric space points live in, and to model fine-grained patterns.PointNet++ introduces hierarchical neural networks that apply PointNet recursively on nested partitions of the input point set. In simpler terms, it breaks down the original point cloud into smaller subsets (clusters of points), applies the PointNet architecture to each subset, and then aggregates the results. This approach allows PointNet++ to capture both local and global features more effectively, leading to improved performance in tasks like object classification, part segmentation, and semantic segmentation.

SplatNet [15] is a versatile and powerful tool for point cloud processing, offering a unique approach to encoding spatial information and handling varying point cloud densities. What sets SplatNet apart from other point cloud processing methods is its unique approach to encoding spatial information. Rather than directly consuming the raw point cloud data, SplatNet first projects the 3D points onto a learned high-dimensional lattice, splatting each point's features onto the lattice vertices. This process, known as "splatting", allows for more efficient and effective encoding of the spatial relationships between points. Once the data is splatted onto the lattice, SplatNet applies a series of convolutional and pooling layers to extract features from the data. This is similar to the process used by convolutional neural networks (CNNs) for image processing, but adapted for the high-dimensional lattice structure. Finally, the features are "de-splatted" back onto the original points, ensuring that the output of the network is a function of the input point cloud. One of the key benefits of SplatNet is its ability to handle point clouds of varying density. Because the splatting process effectively normalizes the density of the point cloud, SplatNet can process sparse and dense point clouds equally well.

The PointSeg [16] model is specifically designed to handle this segmentation task. It uses a PointNet-based architecture to learn features from the raw point cloud data, and then applies a series of convolutional and fully connected layers to perform the segmentation. One of the key features of PointSeg is its ability to handle large-scale point clouds. It does this by dividing the input point cloud into smaller, manageable blocks, processing each block separately, and then aggregating the results. This makes PointSeg particularly suitable for processing the large, complex point clouds typically encountered in real-world scenarios.Another notable feature of PointSeg is its use of an auxiliary loss function to help guide the learning process. This auxiliary loss function is based on the distances between the points and their corresponding segment centers, which encourages the model to learn meaningful segmentations of the point cloud.

The architecture of SqueezeSeg [17] is inspired by the SqueezeNet model for image classification, which is known for its small model size and fast processing speed. SqueezeSeg leverages these benefits while adapting the model for point cloud data. To do this, SqueezeSeg first projects the 3D point cloud into a 2D spherical surface, similar to an image. Each point in the point cloud is transformed into a pixel in this 2D image, with the pixel's intensity representing the point's features. This transformation allows SqueezeSeg to use standard 2D convolutions, which are more efficient than 3D convolutions. Once the data is transformed, SqueezeSeg applies a series of convolutional and pooling layers to extract features from the data and perform the segmentation. It also uses a technique known as "squeeze-and-excitation" to recalibrate the channel-wise feature responses, improving the model's ability to focus on the most relevant features. A key advantage of SqueezeSeg is its efficiency. The model is small and fast, making it suitable for real-time applications, such as autonomous driving, where quick processing of point cloud data is crucial.

Considering the typically low quality of point cloud data acquired from different LiDAR systems in real-world scenarios, which often includes unwanted noise and irrelevant data points, a robust multi-task learning network [18] is introduced to preprocess LiDAR data. Moreover, point cloud classification has become a crucial field of research in various emerging applications such as robotics and autonomous driving. To address this, a novel hierarchical local-global framework [19] is proposed specifically designed for processing 3D point clouds.

Data augmentation for 3D point clouds. In recent years, there has been a surge in the development of data augmentation techniques for 3D point clouds. An abundance of data augmentation strategies for 3D point clouds have emerged in recent years, each bringing unique methodologies to enrich the training data. For instance, PointMixUp [20] method extends the concept of "mixup" from image data to point cloud data. The "mixup" technique was originally proposed for image data, and it involves creating new training samples by taking convex

combinations of pairs of images and their labels. The PointMixup technique adapts this concept for point cloud data.Specifically, PointMixup generates new point clouds by interpolating between pairs of point clouds. Each point in the new point cloud is a weighted combination of points from two original point clouds, with the weights randomly sampled from a beta distribution. In addition to the point coordinates, PointMixup also interpolates the associated features and labels of the points. This ensures that the new point cloud has meaningful features and labels, consistent with the interpolated points. By generating new training samples in this way, PointMixup helps to increase the diversity of the training data, which can help to improve the performance of the model. It also provides a form of regularization, which can help to prevent overfitting.

PointCutMix [21] is an adaptation of the CutMix data augmentation technique, originally developed for image data, specifically for point cloud data. The original CutMix technique works by cutting and pasting regions between two images and their labels to generate new images and labels. PointCutMix extends this concept to point clouds. In the PointCutMix technique, a random portion of a point cloud is replaced with a portion from another point cloud. The replaced portion (or "cut") can be a random sample of the points, a spherical region, or any other defined shape. The points in the cut region are then replaced with points from another point cloud (the "mix"). The labels for the new point cloud are also mixed accordingly. If a point is from the first point cloud, it retains its original label, and if it is from the second point cloud, it takes the label from that cloud. By generating new training samples in this way, PointCutMix helps to increase the diversity of the training data and improve the robustness of the model. It can also provide a form of regularization, helping to prevent overfitting.

GT-Aug [22-23] introduces a different approach by cutting instances and integrating them into other LiDAR scans. This technique is particularly designed for object detection tasks, requiring 3D bounding boxes for object cutting.

The Mix3D [24] technique works by creating new 3D samples by mixing two existing samples. Unlike traditional blending methods, Mix3D not only mixes the features (such as the coordinates and other attributes of the points) but also the labels associated with each point or 3D object. Specifically, Mix3D generates a new 3D data sample by taking a convex combination of two existing samples. Each point (or voxel, in the case of volumetric data) in the new sample is a weighted average of the corresponding points in the two original samples. The weights are usually randomly chosen from a distribution, such as the beta distribution. The labels of the new sample are also generated as a weighted combination of the labels of the original samples. This ensures that the new samples have meaningful labels, consistent with the mixed data. By generating new training samples in this way, Mix3D helps to increase the diversity of the training data, which can lead to improved model performance. Furthermore, the mixing process introduces a form of regularization, which can help to prevent overfitting.

In essence, these various techniques underscore the continued growth and innovation in data augmentation methods for 3D point clouds, each contributing uniquely to the enrichment of training data.

3 The Proposed Method: CB-PolarMix

Our proposed CB-PolarMix method is built on the basis of the original Polarmix and is a supplement or improvement to Polarmix. The key improvement strategy is to perform class balancing for several classes that have lower precision after semantic segmentation using the original Polarmix, that is, to paste instances of these classes more times.

3.1 Preliminaries

A 3D LIDAR point cloud is a refined data structure that encapsulates discrete points in a three-dimensional space. This structure comprises a multitude of points, each itemized with its coordinates in this 3D space. These points are typically sourced from techniques such as laser scanning, cameras, or other sensory devices. Every point in the point cloud possesses distinct attributes, which often include color, intensity, normals, and more. These attributes grant additional insights about the objects encapsulated in the point cloud. Figure 2 presents a visualization of a point cloud, from this vantage point, we can observe a traffic road scene with cars and the silhouettes of buildings, punctuated by trees. When we apply color-coding based on semantic segmentation labels to different classes, the results of semantic segmentation become intuitively visible, then we can clearly discern the green grassland, the brown tree trunks, and the blue cars.

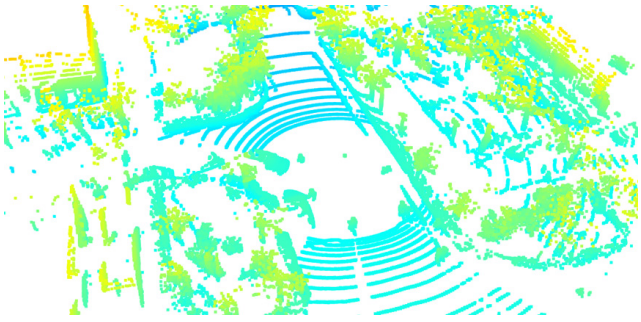


Figure 2. The visualization of a 3D point cloud

PolarMix is a data augmentation method specifically designed for 3D LIDAR point cloud data, employing the concept of mixing. It enhances point cloud distributions and preserves their fidelity through two cross-scan augmentation strategies that involve cutting, editing, and mixing point clouds along the scanning direction. The first step, known as scene-level swapping, involves the exchange of point cloud sectors between two LiDAR scans. These scans are divided along the azimuth axis, allowing for the swapping of corresponding sectors. This

process generates new variations of the point cloud data while preserving their spatial distribution. The second step, referred to as instance-level rotation and paste, involves selecting specific point instances from one LiDAR scan, rotating them at various angles (resulting in multiple copies), and subsequently pasting these rotated instances into other scans. This approach effectively creates new point cloud instances with minimal distortions to their original spatial structure. By combining these two strategies, PolarMix enhances the quality and diversity of 3D LIDAR point cloud data, leading to improved performance in semantic segmentation tasks. The resulting models are more robust and accurate, making PolarMix an essential tool for advancing the field of 3D point clouds semantic segmentation.

Algorithm 1. Class-balanced PolarMix method

Input: Point cloud data

Output: Point cloud data after data augmentation

Initialize instance_classes0, instance_classes1, Omega0 and Omega1. Here, instance_classes0 and Omega0 refer to the object instances and rotation angles required for operations in the original polarmix, while instance_classes1 and Omega1 represent the object instances and rotation angles that need further operations in the class-balanced polarmix that we propose.

```
1:  Procedure PolarMix(pts1, labels1, pts2, labels2, alpha, beta,
    instance_classes0, instance_classes1, Omega0, Omega1):
2:      /*Initialize output points and labels as pts1, labels1*/
3:      pts_out, labels_out=pts1, labels1
4:      /*Swapping*/
5:      If random number is less than 0.5 then
6:          Swap pts_out and labels_out with parts of pts2 and
    lebels2 between angles alpha and beta
7:      /*Ratate-Pasting*/
8:      If random number is less than 1.0 then
9:          Rotate-copy pts2 and labels2 with instance_classes0
    and Omega0 to get pts_copy and labels_copy
10:     Append pts_copy and labels_copy to pts_out and
    labels_out
11:
12:     /*Keep on Ratate-Pasting based on the output of the
    original polarmix*/
13:     Rotate-copy pts2 and labels2 with instance_classes1
    and Omega1 to get pts_copy1 and labels_copy1
14:     Append pts_copy1 and labels_copy1 to pts_out and
    labels_out
15:
16:     /*Return the enhanced point cloud data*/
17:     return pts_out, labels_out
18:
19: End Procedure
```

In the PolarMix approach, rotation and pasting operations for selected object instances are applied uniformly; all chosen instances experience an identical number of rotations and pastes. Theoretically, repeated rotation and pasting can amplify the instance count within a single lidar scan frame, thereby enriching the model's learning dataset. Nonetheless, we assert that this consistent application of rotation and pasting does not optimally enhance the accuracy of semantic segmentation.

Upon observing the original PolarMix's semantic segmentation results, we noticed a significant variance in precision across different classes. For instance, using MinkowskiNet on the SemanticKITTI dataset, we found that the precision of semantic segmentation for classes such as car, bicyclist, road, and building could reach 80% or even over 90%. However, for classes like motorcyclist and other-ground, the precision was merely around 10%. This stark disparity in semantic segmentation precision across different classes prompted our idea to adjust the original PolarMix. Consequently, we propose to paste more instances for classes with lower segmentation precision, thereby providing more learning samples for the model. This approach can improve the semantic segmentation precision of these classes, thereby enhancing the overall model's precision for semantic segmentation on a particular dataset.

3.2 The Algorithm of CB-PolarMix

The CB-PolarMix algorithm is built on the foundation of the original PolarMix algorithm, and it needs to be cascaded with the original PolarMix to fully realize the true potential of the CB-PolarMix algorithm. The basic idea of the CB-PolarMix algorithm is to select several classes with low semantic segmentation performance from the output results of the original PolarMix algorithm and further rotate and paste them. The purpose of doing this is to increase the number of instances of these underperforming classes, thereby providing the model with more learning samples. As for the specific algorithm of CB-PolarMix, we provide a detailed explanation with the pseudocode below. In this pseudocode, the code from line 1 to line 10 represents the original PolarMix operations, which roughly means to perform swapping or rotation and pasting operations on any point on the point cloud. On this basis, we further carry out CB-PolarMix. lines 13 and 14 indicate that the class instances in the instance_classes1 array are rotated and pasted according to the angles in the Omega1 array. And the classes in the instance_classes1 array are selected from the output results of the original PolarMix algorithm, specifically selecting several classes that have low semantic segmentation performance.

4 Experiments

4.1 Datasets Preprocessing

We conducted experiments utilizing both the SemanticKITTI and SemanticPOSS datasets. For SemanticKITTI, we followed the prevalent approach of using 19 semantic classes from the dataset for evaluation, in alignment with procedures adopted by other researchers. SemanticKITTI comprises 22 road sequences, labeled from sequence 00 to sequence 21. In accordance with common practices, sequence 08 was designated as the validation set,

while sequences 00, 01, 02, 03, 04, 05, 06, 07, 09, and 10 formed the training set.

The SemanticPOSS dataset, amassed at Peking University, encompasses 6 road sequences, labeled as 00 to 05, and includes a total of 2988 diverse LIDAR scans in the same data format as SemanticKITTI. Consistent with standard practices, sequence 03 served as the validation set, while sequences 00, 01, 02, 04, and 05 composed the training set. This dataset involves 17 classes in total. In our experiments, we remapped these 17 classes to 14, as detailed in Table 1. The "unlabeled" class was excluded from our experiments, and we concentrated on the evaluation of the remaining 13 classes.

Table 1. Mapping relationship for remapping the SemanticPOSS dataset from 17 classes to 14 classes

Class numbers (old)	Class numbers (new)	Labels
0	0	1 person
4	0	2+ person
5	1	Rider
6	2	Car
7	3	Trunk
8	4	Plants
9	5	Traffic sign 1 (standing sign)
10	5	Traffic sign 2 (hanging sign)
11	5	Traffic sign (high/big hanging sign)
12	6	Pole
13	7	Trashcan
14	8	Building
15	9	Cone/Stone
16	10	Fence
17	11	Bike
21	12	Ground
22	23	Unlabeled

In relation to the SemanticPOSS dataset, initially, we opted for the original PolarMix approach, selecting instances from classes 0, 1, 2, 5, 6, 7, 9 and 11 for rotation and pasting, with each class receiving equal pasting instances. We then implemented class balancing, selecting classes 3, 5, 6 and 7 for additional rotations and pasting operations.

Regarding the SemanticKITTI dataset, in accordance with the original PolarMix paper, we initially selected instances from classes 0, 1, 2, 3, 4, 5, 6 and 7 for rotation and pasting, with an equal number of pastes applied across all classes. Subsequently, we implemented class balancing, selecting classes 7 and 11 for further rotations and pasting operations.

4.2 Visualization Settings

In order to offer a vivid visualization of our experimental outcomes, we have allocated unique colors to each class within the SemanticPOSS and SemanticKitti datasets. These color designations are aligned with the official guidelines stipulated by the respective datasets. Such color codings allow for an intuitive visual comparison between the predicted point cloud and the actual ground truth during the 3D point cloud data visualization process.

The color of each point serves as a straightforward indicator for assessing the accuracy of semantic segmentation predictions. For instance, consider Figure 3, which is a screenshot from a 3D point cloud scan within the SemanticKitti dataset. The color annotations enable us to distinctly discern entities in the scene, such as green plants, blue cars, and the brown tree trunk, among others. This use of color, therefore, not only enhances the visual appeal but also facilitates an easier understanding of the segmentation results.

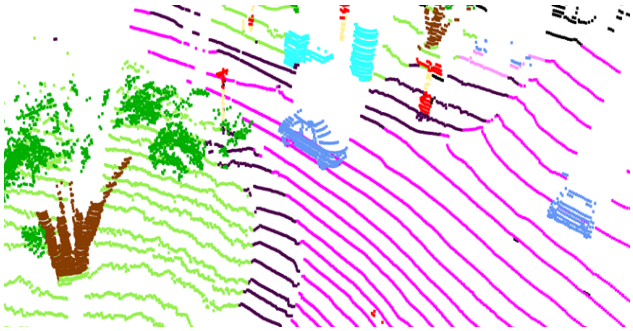


Figure 3. Illustration of color labels

4.3 Hyperparameter Setting and Training

Regarding the settings for rotation angles and pasting times, in the initial rotation and pasting, we maintained the settings from the original PolarMix paper, which were as follows:

$$\omega_0 = \left[np.random.random() \times \frac{2\pi}{3}, \; (np.random.random() + 1) \times \frac{2\pi}{3} \right] \quad (1)$$

For subsequent class balancing, our settings for rotation and pasting were as follows:

$$\omega_1 = \left[\frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4} \right] \quad (2)$$

Table 2 shows the experimental environment and parameter configurations. The training procedure was executed on an NVIDIA Tesla V100-16G GPU, utilizing CUDA version 10.2, Python version 3.8, and PyTorch version 1.6.0 for the model's training. Stochastic Gradient Descent (SGD) was employed as the optimizer during the training process, operating at a learning rate of 2.4e-1. To mitigate potential overfitting, we incorporated a weight decay of 1.0e-4. Additionally, faster convergence was promoted through the use of a momentum value set at 0.9. Nesterov momentum was also harnessed to expedite the optimization process. We implemented a cosine warmup scheduler to dynamically adjust the learning rate. This

scheduler gradually amplifies the learning rate in the initial stages and adheres to a cosine annealing schedule thereafter, optimizing the training process effectively.

Table 2. Experimental environment and parameter configurations

Item	Version
Python	3.8
PyTorch	1.6.0
Optimizer	Stochastic Gradient Descent (SGD)
Learning rate	0.24
Momentum	0.9
Weight decay	0.0001

4.4 Segmentation Model

MinkowskiNet is a type of neural network that is specifically designed for processing sparse, high-dimensional data, such as point clouds, which are often encountered in fields like robotics, autonomous vehicles, and 3D vision. It's named after the Minkowski space, a mathematical framework used for describing space-time in physics, but in the context of neural networks, it refers to the Minkowski Engine — a key component that enables efficient processing of sparse tensor inputs. MinkowskiNet uses a generalized form of convolution suitable for high-dimensional sparse data. Unlike traditional convolutional neural networks (CNNs) that assume data is dense (like pixels in an image), MinkowskiNet is designed to work with data where most of the space is empty, which is typical for point clouds. By focusing on non-empty data points, MinkowskiNet avoids unnecessary computations on empty regions of space, which greatly improves computational efficiency and memory usage.

SPVCNN, or Sparse Voxel Convolutional Neural Networks, is also a type of 3D convolutional neural network (CNN) designed to process point cloud data. What sets SPVCNN apart is its efficient handling of sparse 3D data. Traditional 3D CNNs often struggle with the computational and memory demands of handling 3D data, as a large portion of this data can be empty or irrelevant space. SPVCNN tackles this issue by using voxelization to divide the 3D space into a set of 3D pixels, or voxels. It then only processes the voxels that contain relevant data, greatly reducing the computational resources required.

5 Experiment Results and Discussion

5.1 Evaluation Metrics

Intersection over Union (IoU) is a common metric used for the evaluation of semantic segmentation tasks, including 3D point cloud data. The formula for IoU for a single class is as follows:

$$IoU = \frac{(Area\ of\ Overlap)}{(Area\ of\ Union)} \tag{3}$$

The "Area of Overlap" is the intersection of the predicted segmentation and the ground truth (i.e., the correctly predicted points), while the "Area of Union" is the union of the predicted segmentation and the ground truth (i.e., all points that were predicted to be a part of this class plus all points that should be a part of this class).

In essence, the IoU score reflects the overlap between the predicted segmentation and the ground truth. An IoU of 1 indicates a perfect match (complete overlap), while an IoU of 0 indicates no overlap.

Mean Intersection over Union (mIoU) is an extension of IoU, which averages the IoU scores across all classes. The formula for mIoU is as follows:

$$mIoU = \frac{Sum(IoU\ of\ each\ class)}{number\ of\ classes} \tag{4}$$

mIoU takes into account the performance of the model across all classes, providing a more holistic view of the model's performance. This makes it particularly useful for datasets with multiple classes, as it ensures that the model performs well not just on a single class, but across all classes.

5.2 Comparison with State-of-the-art

Table 3. Comparison of results based on SemanticPOSS dataset and MinkowskiNet model

MinkowskiNet model+SemanticPOSS dataset		
Method / Class	+PolarMix	+CN-PolarMix (Ours)
Person	61.6	60.9
Rider	65.6	66.0
Car	77.3	72.3
Truck	33.1	40.7
Trunk	78.5	80.9
Traffic sign	47.5	55.5
Pole	41.2	37.3
Trashcan	39.0	44.6
Building	79.6	83.3
Cone/Stone	42.2	41.1
Fence	63.3	63.2
Bike	54.8	56.2
Ground	80.6	81.2
mIoU	58.8	60.2

Table 3 presents the findings of a comparative study between PolarMix and our proposed CB-PolarMix, utilizing SemanticPOSS as the dataset and MinkowskiNet as the model. The experiment initially involved rotation and pasting operations on classes 0, 1, 2, 5, 6, 7, 9, and 11. The results obtained from training with the MinkowskiNet model are shown in the +PolarMix column. Upon examination, it is evident that the semantic segmentation accuracy for classes 3, 5, 6, and 7 is notably low. The IoUs for these classes stand at 33.1%, 47.5%, 41.2%, and 39.0% respectively, which negatively impacts the overall mIoU. To enhance the mIoU, classes 3, 5, 6, and 7 were subjected to additional rotation and pasting based on the algorithmic principles of CB-PolarMix, using the previously mentioned

parameter settings of Omega1. The subsequent results, obtained after training with the MinkowskiNet model, are displayed in the +CB-PolarMix column. It can be observed that the mIoU has improved from 58.8% to 60.2%.

Table 4 showcases another comparative experiment between PolarMix and our proposed CB-PolarMix, employing SemanticPOSS as the dataset and SPVCNN as the model. Similar to the first experiment, initial rotation and pasting operations were conducted on classes 0, 1, 2, 5, 6, 7, 9, and 11. The outcomes derived from training with the SPVCNN model are presented in the +PolarMix column. A closer look reveals that the semantic segmentation accuracy for classes 3, 5, 6, and 7 remains suboptimal. Following the algorithmic tenets of CB-PolarMix, classes 3, 5, 6, and 7 underwent additional rotation and pasting based on the earlier defined parameter settings of Omega1. The test results, achieved after training with the SPVCNN model, are depicted in the +CB-PolarMix column. Notably, the mIoU has risen from 57.7% to 58.3%. In light of these experiments, given the IoU of the 9th class is only 39.8%, it may be beneficial to incorporate the 9th class into CB-PolarMix to augment the number of rotations and pastings.

Table 5 presents the experiment results from the SemantiKitti dataset. The CB-PolarMix results were obtained through our experiments, while other methods' results are cited from reference [1]. The table is divided into two sections: the top half shows results from the MinkowskiNet model and the bottom half displays results from the SPVCNN model. Initially, rotation and pasting operations were performed on classes 0 to 7. Results after training with the MinkowskiNet model are shown in the +PolarMix row. It was observed that the semantic segmentation accuracy for classes 7 and 11 was low, with IoUs of only 4.9% and 1.4% respectively, which lowered the overall mIoU. To improve this, classes 7 and 11 were rotated and pasted again according to the Omega1 parameter settings, based on the algorithmic concept of CB-PolarMix. After training with the MinkowskiNet model, test results are displayed in the +CB-PolarMix row. It can be seen that the mIoU increased from 65.0% to 67.7%. Similarly, for the SPVCNN model, initial rotation and pasting operations were performed on classes 0 to 7. Test results after training with the SPVCNN model are shown in the +PolarMix row. Once again, it was observed that the semantic segmentation accuracy for classes 7 and 11 was low. Therefore, classes 7 and 11 were rotated and pasted again according to the Omega1 parameter settings, based on the algorithmic concept of CB-PolarMix. After training with the SPVCNN model, test results are displayed in the +CB-PolarMix row. It was observed that the mIoU increased from 66.2% to 67.6%.

Table 4. Comparison of experimental results based on SemanticPOSS dataset and SPVCNN model

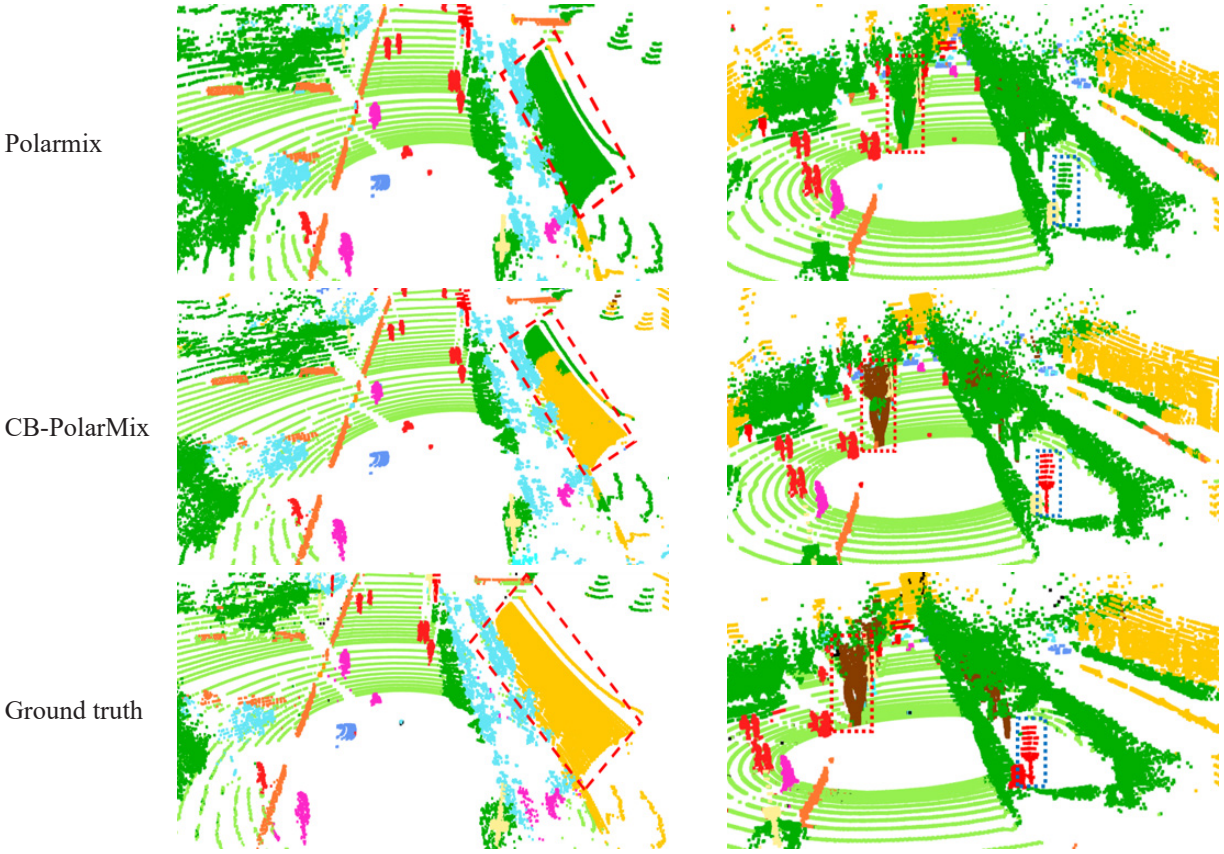
SPVCNN model+SemanticPOSS dataset		
Method / Class	+PolarMix	+CN-PolarMix (Ours)
Person	59.0	59.6
Rider	63.9	64.1
Car	66.2	67.3
Truck	31.1	34.4
Trunk	76.5	77.1
Traffic sign	54.2	51.2
Pole	45.8	43.6
Trashcan	43.3	44.5
Building	76.8	76.3
Cone/Stone	39.8	44.0
Fence	61.2	60.9
Bike	52.5	54.5
Ground	79.9	80.4
mIoU	57.7	58.3

Table 5. Comparison of experimental results based on SemanticKitti dataset

Methods	Car	Bicycle	Motorcycle	Truck	Other-vehicle	Person	Bicyclist	Motorcyclist	Road	Parking	Side walk	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
+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
+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
+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
+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
+CB-PolarMix (Ours)	96.8	53.7	76.5	79.8	67.8	74.1	88.4	13.6	93.7	48.9	81.7	11.2	91.1	63.5	88.2	66.8	74.1	65.4	51.2	67.7
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
+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
+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
+Mix3D	96.5	35.9	65.0	66.6	60.2	75.3	83.3	0.0	93.8	49.0	81.1	1.4	90.6	60.0	89.2	70.2	76.4	64.8	50.5	63.7
+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
+CB-PolarMix (Ours)	97.1	54.5	79.7	78.1	74.9	72.2	87.4	4.7	93.7	49.1	81.8	12.9	91.2	63.5	87.6	68.3	72.8	64.6	50.9	67.6

Figure 4 displays the experimental results obtained from running the MinkowskiNet model on the SemanticPOSS dataset. Two sets of visualized data were selected: the first from sequence 000405 of the third sequence in the SemanticPOSS dataset, and the second from sequence 000205 of the third sequence in the SemanticPOSS dataset. Each set of visualized data forms a column for convenient comparison and observation, with each row corresponding to the visualized results of PolarMix, CB-PolarMix, and Ground truth. In the first column, it is evident that compared to Ground truth, PolarMix misclassified the building on the right (outlined in red dashed lines) as green vegetation, while CB-PolarMix correctly identified most of the building. In the second column, compared to Ground truth, PolarMix misclassified the traffic sign (outlined in blue dashed lines) as green vegetation, and also misclassified the tree trunk (outlined in blue-red dashed lines) as green vegetation. These visualization results visually demonstrate the effectiveness of CB-PolarMix.

Figure 4. Visualization of comparative experiment results (The content within the dashed line box demonstrates the various performances of semantic segmentation.)



5.3 Analysis and Discussion

In fact, we conducted additional experiments in the process of obtaining the above results, with the aim of achieving better outcomes. Table 6 and Table 7 display the results with different times of paste operations. The right column in Table 6 and Table 7 represent instances underwent Rotate-paste operations following

$$\varphi_1 = \left[\frac{\pi}{3}, \frac{2\pi}{3}, \pi, \frac{4\pi}{3}, \frac{5\pi}{3}, 2\pi\right] \tag{5}$$

It's clear that the times of Rotate-paste operations is six. The left column in Table 6 and Table 7 represent instances underwent Rotate-paste operations following

$$\varphi_2 = \left[\frac{2\pi}{3}, \frac{4\pi}{3}, 2\pi\right] \tag{6}$$

Here, the times of Rotate-paste operations is obviously three. We can clearly see that the results on the left are better, indicating that the outcomes are better when the number of Rotate-paste operations is three. The reason for this outcome is that as the number of Rotate-paste operations increases, many instances become mixed together, which is not conducive to effective differentiation and learning. The experiments revealed that a lower times of Rotate-paste operations resulted in superior learning outcomes due to less instance blending and enhanced distinguishability.

Table 6. Comparison results with different Rotate-paste operation times based on MinkowskiNet model and SemanticPOSS dataset

SemanticPOSS		
Paste times / Class	3	6
Person	60.9	60.9
Rider	66.0	66.1
Car	72.3	75.8
Truck	40.7	42.2
Trunk	80.9	78.8
Traffic sign	55.5	51.5
Pole	37.3	38.0
Trashcan	44.6	46.8
Building	83.3	78.9
Cone/Stone	41.1	36.5
Fence	63.2	62.2
Bike	56.2	57.1
Ground	81.2	80.9
mIoU	60.2	59.7

Table 7. Comparison results with different rotate-paste operation times based on MinkowskiNet model and SemanticKitti dataset

SemanticKitti		
Paste times / Class	3	6
Car	96.8	96.7
Bicycle	53.7	53.4
Motorcycle	76.5	78.4
Truck	79.8	80.7
Other-vehicle	67.8	67.9
Person	74.1	72.4
Bicyclist	88.4	86.9
Motorcyclist	13.6	13.4
Road	93.7	93.5
Parking	48.9	47.5
Sidewalk	81.7	80.7
Other-ground	11.2	6.8
Building	91.1	91.1
Fence	63.5	64.9
Vegetation	88.2	89.3
Trunk	66.8	68.1
Terrain	74.1	76.7
Pole	65.4	64.8
Traffic-sign	51.2	50.3
mIoU	67.7	67.6

6 Conclusion

In this paper, we introduce an advanced data augmentation technique designed for the semantic segmentation of 3D point clouds. Our approach builds upon the existing PolarMix method by incorporating a two-stage sequential process to improve segmentation accuracy. During the first stage, we implement the original PolarMix technique and then integrate a class balancing strategy. In the second stage, classes that exhibited lower segmentation accuracy in the first stage are subjected to rotation and replication. This action effectively increases the sample size for these classes, leading to enhanced overall segmentation accuracy. We refer to this refined approach, which incorporates a class balancing strategy based on a sequential methodology, as CB-PolarMix. We tested our method on two datasets, SemanticPOSS and SemanticKitti, using two distinct models: MinkowskiNet and SPVCNN. The experimental outcomes underscore the effectiveness of our method in improving the accuracy of semantic segmentation, particularly for classes with initially lower accuracy, thereby elevating the average segmentation accuracy across all classes.

The primary contributions of this study are twofold: firstly, the introduction of an efficient data augmentation technique, CB-PolarMix, tailored for enhancing the accuracy of semantic segmentation in 3D point clouds; secondly, the demonstration of its efficacy through experiments conducted on two datasets using different models, offering a valuable strategy for improving semantic segmentation tasks involving 3D point clouds.

Acknowledgements

This work is funded in part by the Key Project of Nature Science Research for Universities of Anhui Province of China (No. 2022AH051720), and in part by Anhui Province Higher Education Talent Fund Project (No. YQYB2023033), and in part by Excellent Research and Innovation Team of Anhui Higher Education Institutions (No. 2024AH010022).

References

[1] A. Xiao, J. Huang, D. Guan, K. Cui, S. Lu, L. Shao, Polarmix: A general data augmentation technique for lidar point clouds, *Advances in Neural Information Processing Systems*, New Orleans, USA, 2022, pp. 11035-11048.

[2] P. Singh, J. Chopra, A. Singh, N. Nijhawan, Kritika, Deep Learning Innovations for Enhanced Drusen Detection in Retinal Images, *International Journal of Performability Engineering*, Vol. 19, No. 12, pp. 779-787, December, 2023.

[3] V. Arya, T. Kumar, Boosting X-ray scans feature for enriched diagnosis of pediatric pneumonia using deep learning models, *International Journal of Performability Engineering*, Vol. 19, No. 3, pp. 175-183, March, 2023.

[4] C. Choy, J. Y. Gwak, S. Savarese, 4d spatio-temporal convnets: Minkowski convolutional neural networks, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Long Beach, CA, USA, 2019, pp. 3075-3084.

[5] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, S. Han, Searching efficient 3d architectures with sparse point-voxel convolution, *European conference on computer vision*, Glasgow, United Kingdom, 2020, pp. 685-702.

[6] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, J. Gall, Semantickitti: A dataset for semantic

scene understanding of lidar sequences, *Proceedings of the IEEE/CVF international conference on computer vision*, Seoul, South Korea, 2019, pp. 9297-9307.

[7] Y. Pan, B. Gao, J. Mei, S. Geng, C. Li, H. Zhao, Semanticposs: A point cloud dataset with large quantity of dynamic instances, *2020 IEEE Intelligent Vehicles Symposium (IV)*, Las Vegas, Nevada, USA, 2020, pp. 687-693.

[8] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: The kitti dataset, *The International Journal of Robotics Research*, Vol. 32, No. 11, pp. 1231-1237, September, 2013.

[9] H. Caesar, V. Bankiti, A.-H. Lang, Sourabh Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, O. Beijbom, nuscenes: A multimodal dataset for autonomous driving, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Seattle, WA, USA, 2020, pp. 11621-11631.

[10] A. Xiao, J. Huang, D. Guan, F. Zhan, S. Lu, Transfer learning from synthetic to real lidar point cloud for semantic segmentation, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vancouver, Canada, 2022, pp. 2795-2803.

[11] A. Xiao, J. Huang, W. Xuan, R. Ren, K. Liu, D. Guan, A. E. Saddik, S. Lu, E. Xing, 3d semantic segmentation in the wild: Learning generalized models for adverse-condition point clouds, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Vancouver, Canada, 2023, pp. 9382-9392.

[12] M. Bijelic, T. Gruber, F. Mannan, F. Kraus, W. Ritter, K. Dietmayer, F. Heide, Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020, pp. 11682-11692.

[13] C.-R. Qi, H. Su, K. Mo, L.-J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, Hawaii, USA, 2017, pp. 652-660.

[14] C.-R. Qi, L. Yi, H. Su, L.-J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, *Advances in neural information processing systems*, Long Beach, CA, USA, 2017, pp. 5105-5114.

[15] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M. Yang, J. Kautz, Splatnet: Sparse lattice networks for point cloud processing, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Salt Lake City, UT, USA, 2018, pp. 2530-2539.

[16] Y. Wang, T. Shi, P. Yun, L. Tai, M. Liu, Pointseg: Real-time semantic segmentation based on 3D LIDAR point cloud, *arXiv preprint arXiv:1807.06288*, September, 2018. https://arxiv.org/abs/1807.06288

[17] B. Wu, A. Wan, X. Yue, K. Keutzer, Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3D LIDAR point cloud, *2018 IEEE international conference on robotics and automation (ICRA)*, Brisbane, Australia, 2018, pp. 1887-1893.

[18] L. Zhao, Y. Hu, X. Yang, Z. Dou, L. Kang, Robust multi-task learning network for complex LiDAR point cloud data preprocessing, *Expert Systems with Applications*, Vol. 237, No. Part B, Article No. 121552, March, 2024.

[19] W. Zhou, Y. Zhao, Y. Xiao, X. Min, J. Yi, TNPC: Transformer-based network for point cloud classification, *Expert Systems with Applications*, Vol. 239, Article No. 122438, April, 2024.

[20] Y. Chen, V. Hu, E. Gavves, T. Mensink, P. Mettes, P. Yang, CGM Snoek, Pointmixup: Augmentation for point clouds, *Computer Vision–ECCV 2020: 16th European Conference*, Glasgow, UK, 2020, pp. 330-345.

[21] D. Lee, J. Lee, J. Lee, H. Lee, M. Lee, S. Woo, S Lee, Regularization strategy for point cloud via rigidly mixed sample, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Nashville, TN, USA, 2021, pp. 15900-15909.

[22] Y. Yan, Y. Mao, B. Li, Second: Sparsely embedded convolutional detection, *Sensors*, Vol. 18, No. 10, Article No. 3337, October, 2018.

[23] J. Fang, X. Zuo, D. Zhou, S. Jin, S. Wang, L. Zhang, Lidar-aug: A general rendering-based augmentation framework for 3d object detection, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Nashville, TN, USA, 2021, pp. 4710-4720.

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

Biographies

Bo Liu received the M.S. degree in computer science from Shanghai Ocean University in 2013 and currently pursuing a Ph.D. in Artificial Intelligence at the Macau University of Science and Technology. He currently holds the position of Associate Professor at Chaohu University, his academic interests lie in employing deep learning approaches to improve image detection outcomes.

Xiao Qi received a master degree in software engineering from the University of Science and Technology of China. She is currently a lecturer with the Department of Information Technology and Cybersecurity, Shanghai Police College. Her research interests include network security, data security, federated learning.