
PointCutMix: Regularization Strategy for Point Cloud Classification

Jinlai Zhang¹ * Lyujie Chen² * Bo Ouyang² Binbin Liu² Jihong Zhu^{1 2 3} Yujin Chen¹ Yanmei Meng¹
Danfeng Wu³

Abstract

As 3D point cloud analysis has received increasing attention, the insufficient scale of point cloud datasets and the weak generalization ability of networks become prominent. In this paper, we propose a simple and effective augmentation method for the point cloud data, named PointCutMix, to alleviate those problems. It finds the optimal assignment between two point clouds and generates new training data by replacing the points in one sample with their optimal assigned pairs. Two replacement strategies are proposed to adapt to the accuracy or robustness requirement for different tasks, one of which is to randomly select all replacing points while the other one is to select k nearest neighbors of a single random point. Both strategies consistently and significantly improve the performance of various models on point cloud classification problems. By introducing the saliency maps to guide the selection of replacing points, the performance further improves. Moreover, PointCutMix is validated to enhance the model robustness against the point attack. It is worth noting that when using as a defense method, our method outperforms the state-of-the-art defense algorithms. The code is available at: <https://github.com/cuge1995/PointCutMix>.

1. Introduction

With the rapid development of autonomous driving and robotics industries, making machines understand the real three-dimensional world has become a guarantee for safe and efficient task execution (Guo et al., 2020). As a commonly used format for 3D data representation that can

be directly obtained by Light Detection And Ranging (LiDAR) sensors, the point cloud has been widely applied in many computer vision fields (Lang et al., 2019; Chen et al., 2019; Rao et al., 2020), such as 3D object detection (Shi et al., 2020; Bhattacharyya & Czarnecki, 2020), point cloud segmentation (Liu et al., 2020b), and point cloud classification (Qi et al., 2017b; Liu et al., 2019c; Wang et al., 2019). Following the pioneering work of PointNet (Qi et al., 2017a), a series of deep-learning-based methods brought the performance of these tasks to a higher level. However, due to the complexity and costs of fine-grained 3d point cloud annotations (Xu & Lee, 2020), the scale of existing point cloud datasets is much smaller than that of the image datasets (Chen et al., 2020), resulting in the overfitting and poor generalization of these methods (Jing & Tian, 2020). Although researchers have explored several data augmentation techniques for point cloud analysis, such as rotation, scaling, and jittering (Yan et al., 2020; Liu et al., 2020b), these kinds of augmentations ignore the shape complexity of the samples (Li et al., 2020), thus lead to insufficient training.

Over the past few years, mixed sample data augmentation (MSDA) for images has attached increasing interest which aims to create new training data by mixing the original training samples according to some rules (Harris et al., 2020; Guo et al., 2019). There are two mainstream methods in MSDA. The first one is MixUp (Zhang et al., 2018), which interpolates between training samples by performing weighting on the whole image and its label. Another method is CutMix (Yun et al., 2019). It inserts a rectangle region from one image into another and then performs weighting on the image and its label by the ratio of the region size. In comparison, CutMix achieves better results across various models and datasets in image classification, weakly supervised object localization, and transfer learning to object detection.

In this paper, inspired by the success of MSDA in the image domain, we propose an MSDA strategy to the point cloud data, named PointCutMix. To adapt to its unordered feature, we first calculate the optimal assignment of two point clouds refer to MSN (Liu et al., 2020a). Then, two PointCutMix methods that replace the points in one sample with their optimal assigned pairs in another sample are formulated, *i.e.*,

*Equal contribution ¹College of Mechanical Engineering, Guangxi University, Nanning, China ²Department of Computer Science and Technology, Tsinghua University, Beijing, China ³Department of Precision Instrument, Tsinghua University, Beijing, China. Correspondence to: Jihong Zhu <jhzhu@mail.tsinghua.edu.cn>.

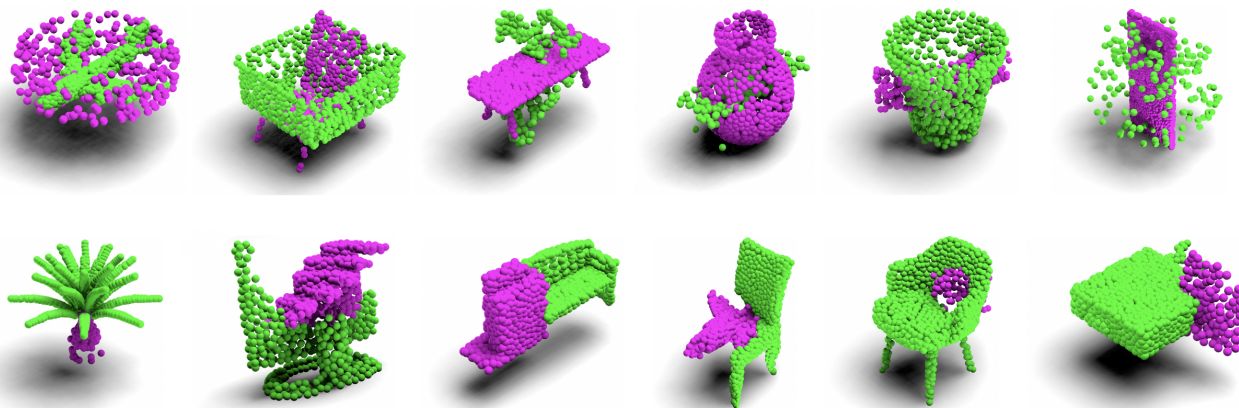


Figure 1. Some mixed samples produced by PointCutMix-R (top row) and PointCutMix-K (bottom row). The data generated by PointCutMix-R looks like two objects cross each other while the samples from PointCutMix-K are the obvious combination of two object parts.

PointCutMix-R and PointCutMix-K. The former randomly selects all replacing points while the latter selects k nearest neighbors of one random chosen point. Experimental results demonstrate that both methods achieve consistent and significant improvements in object-level classification task on ModelNet40 (Wu et al., 2015) and ModelNet10 (Wu et al., 2015) datasets. We also at the first time exploit PointCutMix for point-wise classification, *i.e.*, point cloud segmentation task. It is observed that PointCutMix can evidently improve the recognition accuracy for the uncommon categories. Inspired by the successful use of attention maps in CutMix (Walawalkar et al., 2020), we further introduce the saliency maps to guide the selection of replacing points which achieves better results. Additionally, we validate that PointCutMix can enhance the robustness of different models under the point cloud attack. When using as a defense method, under the point dropping attack (Zheng et al., 2019), our PointCutMix-ModelNet40 pre-trained models surpass the state-of-the-art defense algorithm IF-Defense (Wu et al., 2020) by a large margin without using any transformation on the adversarial point clouds. We also perform defense to other point cloud attacks and achieve promising results. Extensive experiments verify the effectiveness of our method. We believe this simple regularization strategy could be applied to various tasks and help future research in the 3D computer vision community.

2. Related Work

Deep learning on point cloud. PointNet (Qi et al., 2017a) is the first work that processes the point cloud using deep neural networks where the shared pointwise multi-layer perceptions (MLPs) followed by the max-pooling operation are used for point cloud learning. After that, the recent works focus on efficiently capturing local features (Qi et al.,

2017b; Yan et al., 2020; Zhao et al., 2019; Yang et al., 2019b) and investigating convolutional kernels for 3D point clouds. Liu et al. (Liu et al., 2019c) proposed RS-CNN which implemented the convolution using an MLP in the local subset of points. DensePoint (Liu et al., 2019b) defined a single-layer perceptron with a nonlinear activator as convolution. In KPConv (Thomas et al., 2019), by using a set of learnable kernel points, the rigid and deformable Kernel point convolution operators were proposed. Some other researchers have explored graph-based networks, where each point in a point cloud is considered as a vertex of a graph. DGCNN (Wang et al., 2019) constructed a graph in the feature space and MLP is used for each edge. To simplify the process of points agglomeration, the Dynamic Points Agglomeration Module based on graph convolution was proposed by Liu et al. (Liu et al., 2019a). In RGCNN (Te et al., 2018), a graph was constructed by connecting all points with each other in the point cloud. To utilize the local structural information, LocalSpecGCN (Wang et al., 2018) used the spectral convolution network to a local graph.

Mixed sample data augmentation. Mixed Sample Data Augmentation (MSDA) is a strategy that produces new training data by mixing samples according to some rules (Harris et al., 2020). Training with the mixed data, the model would learn multiple features in a balanced way (Taghanaki et al., 2020) and achieve better performance. Therefore, MSDA has become the mainstream data augmentation approach and dominated modern image classification for years (Harris et al., 2020; Guo et al., 2019; Zhang et al., 2018; Yun et al., 2019; Verma et al., 2019). Among them, MixUp (Zhang et al., 2018) and Cutmix (Yun et al., 2019) are two classical methods that have been widely used in various computer vision research (He et al., 2019) and competition (Dolhan-sky et al., 2020). MixUp (Zhang et al., 2018) interpolates

the training samples by performing weighting on the whole image and its label. CutMix (Yun et al., 2019) inserts a rectangle region from one image into another one and then performing weighting on the image and its label by the ratio of the region size. The experiment results show that CutMix has better performance improvement across different datasets and networks. Our work can be viewed as an extension of CutMix (Yun et al., 2019) for the point cloud.

Data augmentation on point cloud. Although random rotation, jittering, and scaling are commonly used in point cloud learning (Qi et al., 2017a;b), the data augmentation for point clouds has obviously not been studied systematically compared to the image domain. Recently, PointAugment (Li et al., 2020) and PointMixup (Chen et al., 2020) were proposed for point cloud data augmentation. PointAugment is the first auto-augmentation framework for the point cloud which optimizes the augmentor and classifier networks jointly. However, the additional augmentor network and the complicated adversarial training process makes it less practical. PointMixup extends Mixup (Zhang et al., 2018) to point cloud by interpolation between point cloud samples. However, for point cloud networks like PointNet++ (Qi et al., 2017b) and RS-CNN (Liu et al., 2019c) that local features are important for point cloud learning, this approach is easy to fall into locally ambiguous and unnatural. In this paper, we proposed PointCutMix to naturally combine two point clouds.

3. PointCutMix

3.1. Problem setting

The goal of a standard point cloud classification task is to learn a function $f : x \rightarrow [0, 1]^C$ that maps a point cloud to a one-hot class label for a total of C classes. Here $x \in \mathbf{R}^{N \times d}$ represents a set of 3D points $\{P_i | i = 1, \dots, N\}$ which either sampled from a shape or pre-segmented from a scene point cloud. N is the point number and each point P_i is a vector with d channels. In this paper, we simplicite use the 3d coordinate features. So $d = 3$ and $P_i \in \mathbf{R}^3$. The optimal parameters θ of function f can be learned by minimizing the loss as

$$\theta^* = \arg \min_{\theta} \sum_{x \in \mathcal{D}} \mathcal{L}_{\mathcal{D}}(f(x), y) \quad (1)$$

where $f(x)$ is the network output, y is the ground truth with respect to x , \mathcal{D} is the training set, and \mathcal{L} represents the training loss function.

3.2. Optimal assignment of point clouds.

To perform MSDA, it requires a one-to-one correspondence between the minimal unit of two samples. For image, this unit is pixel while for point cloud data, that is a single point.

In the image domain, the pixels are arranged in a grid form. By merely resizing or cropping two images to the same size, it is natural to make them correspond according to their coordinate. However, the point clouds are permutation-invariant and orderless. It is essential to define the one-to-one correspondence between points based on rules other than position.

Following the method in PointMixup (Chen et al., 2020) and MSN (Liu et al., 2020a), we define the optimal assignment ϕ^* between two point clouds x_1, x_2 as the optimal assignment of Earth Mover’s Distance (EMD) (Rubner et al., 2000) function. The EMD calculates the minimum total displacement required for matching each point in x_1 to the corresponding point in x_2 . We define the assignment function in the EMD as:

$$\phi^* = \arg \min_{\phi \in \Phi} \sum_i \|x_{1,i} - x_{2,\phi(i)}\|_2 \quad (2)$$

where $\Phi = \{\{1, \dots, N\} \mapsto \{1, \dots, N\}\}$ give one-to-one correspondences between the two point clouds. After given the optimal assignment ϕ^* (Chen et al., 2020), the EMD is then defined as:

$$\text{EMD} = \frac{1}{N} \sum_i \|x_{1,i} - x_{2,\phi^*(i)}\|_2 \quad (3)$$

where $\phi^*(i)$ denotes the index of optimal assignment point of $x_{1,i}$ in x_2 .

3.3. Algorithm

The key idea of PointCutMix is to create a new training point cloud (\tilde{x}, \tilde{y}) given two distinct training point clouds (x_1, y_1) and (x_2, y_2) . Here, x is the training point cloud and y is the corresponding label. After obtaining the optimal assignment ϕ^* between two samples, we define $x_{2,\tilde{i}} = x_{2,\phi^*(i)}$ and the combining operation as

$$\begin{aligned} \tilde{x} &= B \cdot x_1 + (I_N - B) \cdot \tilde{x}_2 \\ \tilde{y} &= \lambda y_1 + (1 - \lambda) y_2 \end{aligned} \quad (4)$$

where $B = \text{diag}\{b_1, b_2, \dots, b_N\}$ and $b_i \in \{0, 1\}$ indicates which sample the point belongs to. When $b_i = 1$, the i^{th} point is chosen from x_1 , otherwise it will be replaced by the optimal assigned point in x_2 . I_N is an identity matrix. $\lambda \in [0, 1]$ is the PointCutMix ratio, sampled from the beta distribution $Beta(\beta, \beta)$, which means $n = \lfloor \lambda \times N \rfloor$ points will be kept and the rest points will be replaced.

To perform cutting and pasting in the point cloud, we propose two replacement methods to construct the diagonal matrix B . The first method, abbreviated as PointCutMix-R, is to randomly sample n points from x_1 as a subset x_1^s . Those points are marked 1 in B , indicating that they will

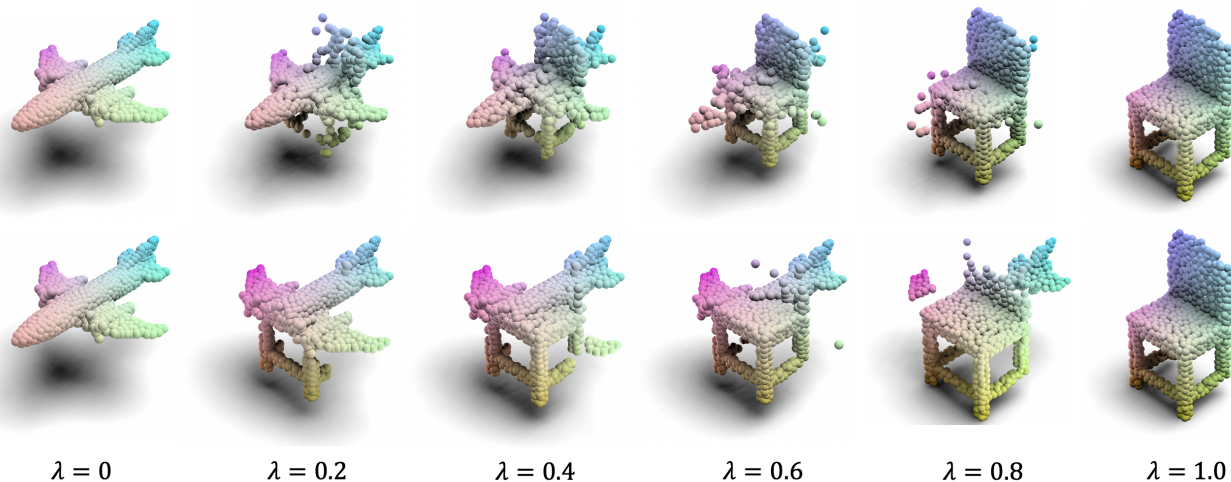


Figure 2. The visualization of the mixed samples between a plane and a chair under different replacement ratio λ . The samples in the first and second row are generated by PointCutMix-R and PointCutMix-K respectively.

not be replaced. The rest points are marked 0. In addition, to retain the local characteristics of the point cloud, we come up with the second method, noted as PointCutMix-K, which randomly sample one central point p from x_1 , and then finding its $n - 1$ nearest neighbors. We combine p and its nearest neighbors to form x_1^s and marked those points as 1 in B . Similarly, the rest points are marked 0 and be replaced. In Figure 1, we show the visualization of some mixed samples of PointCutMix-R and PointCutMix-K. It can be seen that the samples produced by PointCutMix-R look like two objects cross together while the mixed data from PointCutMix-K are the obvious combination of two object parts.

We also introduce a hyperparameter $\rho \in [0, 1]$ to indicate the probability of each point cloud to be augmented during the training. When $\rho = 0$, PointCutMix will not be used which is equivalent to the baseline model. On the contrary, $\rho = 1$ means all point clouds will be augmented. Therefore, the training loss can be denoted as

$$\sum_{x \in \mathcal{D}} (1 - \mathbb{1}_\rho) \mathcal{L}_{\mathcal{D}}(f(x), y) + \mathbb{1}_\rho \mathcal{L}_{\mathcal{D}}(f(\tilde{x}), \tilde{y}) \quad (5)$$

where $\mathbb{1}_\rho = 1$ with a probability ρ , otherwise it equals to 0.

3.4. Analysis

The difference of replacement methods. In Figure 2, we list the visualization of mixed samples between a plane and a chair produced by PointCutMix-R (top row) and PointCutMix-K (bottom row) under different replacement ratios λ . The ratios from left to right are 0, 0.2, 0.4, 0.6, 0.8,

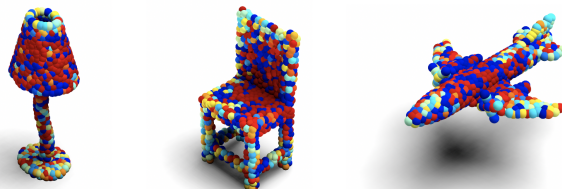


Figure 3. Saliency maps of different point clouds. Points with higher values are colored as red and the color of irrelevant points is closer to blue.

and 1.0. We can see that the samples in the top row are a little messy and like two objects fuse. Especially when λ close to 0 or 1, such as $\lambda = 0.2$ or $\lambda = 0.8$, only one of the objects can be easily recognized. Those hard to distinguished points actually perform like a kind of noise. We infer that this characteristic will impair the performance for learning classification task, but can improve the robustness of the model. This assumption has been verified in the experiment in Section 4. On the contrary, the mixed samples from PointCutMix-K are relatively regular, like a natural combination of two object parts. Since at least a part of each object can be identified, it provides more features for learning the classification task.

Is attention works for PointCutMix? Inspired by the successful use of attention maps to guide the cutting and pasting region of an object in CutMix (Walawalkar et al., 2020), we speculate that the selection of central point p in PointCutMix-K can also be guided rather than random. So

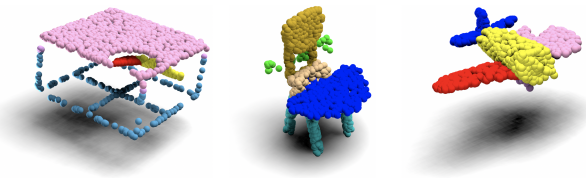


Figure 4. Mixed samples of point cloud segmentation problem using PointCutMix-K.

we try to obtain the contribution of each point to the classification result with saliency map (Zheng et al., 2019). Then the point with greater contribution has a higher probability to be selected as the central point. Through the visualization of the saliency maps shown in Figure 3, we find that the points with high contributions (in red color) are not scattered uniformly. For example, they gather more in the lampshades, the seat of the chair, and the fuselage of the airplane. In the next section, we will examine whether this strategy improves the accuracy of the model.

Extending to point cloud segmentation. So far, we have elaborated on how to apply our method to the object-level classification. Natural intuition is to extend to the point-wise classification problem, *i.e.*, point cloud segmentation. However, we find that the existing augmentation methods are limited by their fusion strategies, thus fail to complete this task. For example, when applying PointMixup to fuse a new point cloud, the semantic information of each point has lost, making it hard to get the semantic labels for new data. On the contrary, since our method is simply cutting and pasting points, the semantic information can be persisted. So in this paper, we at the first time perform augmentation to point cloud segmentation task. Specifically, we mix two point clouds using the same method mentioned before. The point-wise labels are replaced along with the points. For datasets that also contain object-level labels, the object-level annotation is fused referred to Section 3.3. In Figure 4, we show some mixed samples for the point cloud segmentation problem.

4. Experiments

In this section, we conduct extensive experiments to verify the effectiveness of PointCutMix. At first, we find the optimal hyperparameters through several comparative experiments. Then we assess our method from two aspects, one of which is to evaluate how much it improves the accuracy of object-level point cloud classification and point-wise segmentation while the other one is to evaluate the generalization ability and robustness of the model trained with augmented data provided by PointCutMix.

4.1. Setup

Datasets. We evaluate PointCutMix on two object-level point cloud classification datasets and a point-wise segmentation dataset, *i.e.*, ModelNet40 (Wu et al., 2015), ModelNet10 (Wu et al., 2015), and ShapeNet Parts (Yi et al., 2016). ModelNet40 contains 12311 samples in 40 categories. Among them, 9843 samples are used for training and 2468 for testing. ModelNet10 is a subset of ModelNet40. It contains a total of 4899 samples in 10 categories, of which 3991 samples are used for training and the rest are used for testing. ShapeNet Parts consists of 16,880 3D samples in 16 categories and 50 part labels, of which 14,006 for training and 2,874 for testing.

Networks. Since PointCutMix is a general data augmentation method, it is agnostic to the network architecture that is employed. Therefore, we select four popular networks in 3D computer vision area (Li et al., 2020; Zhao et al., 2020) for evaluation, *i.e.*, PointNet (Qi et al., 2017a), PointNet++ (Qi et al., 2017b), RS-CNN (Liu et al., 2019c), and DGCNN (Wang et al., 2019). As mentioned in Section 2, PointNet only uses global information while other three models take the local information into account.

Implementation details. Our work is implemented using PyTorch (Paszke et al., 2017) on NVIDIA GeForce GTX 2080Ti GPU. All networks take 1024 points as input and are trained for 300 epochs with a batch size of 16. For PointNet, PointNet++, and RS-CNN, we use the Adam (Kingma & Ba, 2014) optimizer with an initial learning rate of 0.001 and a decay rate of 0.5 every 20 epochs, which is the same configuration as the original released paper and code. We train DGCNN with SGD optimizer with an initial learning rate of 0.1. The minimum learning rate is 0.001 and the momentum of SGD is 0.9. The cosine annealing strategy is used to decay the learning rate.

4.2. Comparative experiments

We perform the comparative experiments in ModelNet40 dataset using the experimental settings described in implementation details.

Influence of ρ . We first compare the performance of PointCutMix-K on four representative models under different values of ρ to figure out whether our method is useful and how much of the data need to be augmented during the training. The results are illustrated in Figure 5. For pointNet++, RS-CNN, and DGCNN, we observe that even with only 25% of the samples are augmented ($\rho = 0.25$), the accuracy is greatly improved, which proves that our method is very essential and effective. Under different values of ρ , there is no much difference in accuracy for three models. However, PointNet performs in a completely different way. It is improved when ρ is small, but the accuracy is signif-

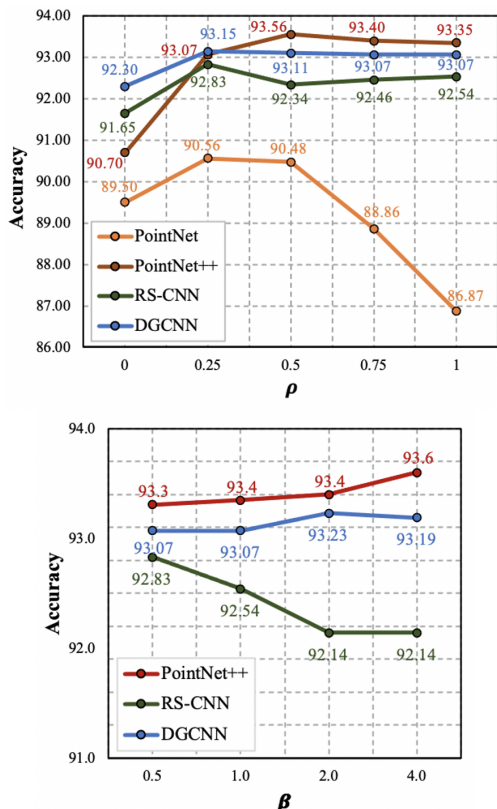


Figure 5. Performance of various models with PointCutMix-K under different value of ρ and β . In the upper plot, $\beta = 1$. In the lower plot, $\rho = 1$.

icantly dropped when ρ reaches 1. We speculate that this is because the coordinate feature of a single point has no actual information. The object-level classification must rely on the learning of the relationship between points. However, PointNet lacks the ability to learn local features since it performs MLP for all points in the object together, which makes it difficult to distinguish the replaced region.

In the following experiments, although each model reaches its optimal performance with different values of ρ , we choose $\rho = 1.0$ for the object-level point cloud classification task in order to make a fair comparison to other augmentation methods. Here we do not report PointNet since our method is not suitable for it. While for the point cloud segmentation task, we select $\rho = 0.5$ for better performance.

Influence of β . Next, we investigate the influence of β , *i.e.*, whether there is a difference in choosing the different number of replaced points at augmentation. From the results in Figure 5, it can be seen that the difference of accuracy under various values of β for PointNet++ and DGCNN is very small, but RSCNN prefers a small value of β . To use

Table 1. ModelNet40 classification results. PointMixup-U and PointMixup-A represent the results on unaligned and pre-aligned ModelNet40 with input mixup.

Method	PointNet++	RS-CNN	DGCNN
baseline	90.7	91.7	92.3
PointMixup-U	91.7	-	-
PointMixup-A	92.7	-	92.9
PointAugment	92.9	92.7	93.4
PointCutMix-R	92.8	91.9	92.8
PointCutMix-K	93.4	92.5	93.1
PointCutMix-S	93.4	92.7	93.2

Table 2. ModelNet10 classification results.

Method	Pointnet++	RS-CNN	DGCNN
baseline	93.3	94.2	94.8
PointAugment	95.8	96.0	96.7
PointCutMix-R	96.3	95.7	95.2
PointCutMix-K	95.7	95.6	95.7

the same hyperparameter for all models and simplify the selecting process of α , we select the $Beta(1, 1)$, *i.e.*, the uniform distribution in the subsequent experiments.

4.3. Point cloud classification

After determining the hyperparameters, we conduct point cloud classification experiments on ModelNet40 and ModelNet10 to evaluate various data augmentation methods, including conventional data augmentation (baseline) (Qi et al., 2017b), PointMixup (Chen et al., 2020), PointAugment (Li et al., 2020), PointCutMix-R, and PointCutMix-K. In addition, to verify the influence of attention maps mentioned in Section 3.4, we introduce the saliency map to guide the selection of central point p . This strategy is named PointCutMix-S. The results of baseline models refer to PointAugment. The models trained with PointCutMix methods are implemented with the settings in our implementation details. The saliency maps are produced by corresponding pre-trained baseline models during the training. The results of PointMixup and PointAugment refer to their original papers.

From Table 1 and Table 2, we observe that our methods consistently outperform PointMixup and have comparative results to PointAugment. This is a very impressive result because PointCutMix is much simpler than the existing methods. PointMixup needs to pre-align the point clouds of the training and test sets in the horizontal facing direction. But our method does not rely on any pre-process for the input point clouds. PointAugment uses an additional network

Table 3. Comparison on the ShaperNet part segmentation dataset. pIoU means part-average Intersection-over-Union. We perform the experiment using the settings described in Section 4.4.

Method	pIoU	air-plane	bag	cap	car	chair	ear-phone	guitar	knife	lamp	laptop	motor-bike	mug	pistol	rocket	skate-board	table
PointNet++	85.0	82.2	81.7	81.5	77.7	90.1	76.7	90.9	87.3	83.8	95.2	69.9	94.2	82.6	56.2	76.6	82.8
+PointCutMix	85.5	82.6	85.9	83.7	78.3	90.7	72.5	90.9	87.7	84.3	95.3	70.7	95.1	82.4	62.3	74.9	83.4

for data augmentation. It requires much more memory cost, which is not practical in real applications. In comparison, PointCutMix uses little computing resources and time but still achieves better performance.

PointCutMix-R occasionally has better results than PointCutMix-K on ModelNet10. However, in most cases across two datasets, PointCutMix-K performs better. The results also show that the saliency maps have limited help for the performance. Considering the addition calculation time and memory consumption used for generating the saliency maps during training, we hold that PointCutMix-K is a more versatile and efficient strategy.

4.4. Point cloud segmentation

To explore the extensibility of our method, we at the first time apply augmentation to the point cloud segmentation task. Here we train the baseline model and PointCutMix-S for 251 epochs with a batch size of 16. We use Adam (Kingma & Ba, 2014) optimizer with an initial learning rate of 0.001 and a decay rate of 0.5 every 20 epochs. In Table 3, we report the part-average Intersection-over-Union results. It shows that PointCutMix makes an improvement of 0.5% over the PointNet++ baseline. Although the improvement is not as significant as that for the object-level classification task, there is a special finding that the accuracy gains mainly come from the uncommon categories. Specifically, the ShapeNet Parts dataset (Yi et al., 2016) has an uneven distribution of training data, where the table has 5271 samples but the bag, cap, and rocket have only 76, 55, and 66 samples respectively. Training with our PointCutMix augmentation method, over 6.1 pIoU improvement is made for the rocket part-segmentation.

We infer the reason is that through the fusion of training samples in PointCutMix, the frequency of occurrence of uncommon categories is greatly increased. This also enlightens us that by carefully adjusting the ratio of selecting different categories of samples for augmentation, the unbalanced distribution problem might be effectively alleviated, which is worth exploring in the future.

4.5. Robustness test

After verifying the accuracy improvement of PointCutMix on the point cloud classification, we then use the adversarial

Table 4. Classification accuracy of ModelNet40 under point dropping attack (Zheng et al., 2019), the dropping points is 200.

Model	Baseline	PointCutMix-R	PointCutMix-K
PointNet++	68.96	86.18	87.97
RS-CNN	56.97	82.50	83.10
DGCNN	55.06	81.16	85.86

attack to investigate whether this regularization strategy can enhance the robustness of the model. As we know, deep neural networks are vulnerable to adversarial examples, which have been extensively studied in 2D images (Dong et al., 2018; Akhtar & Mian, 2018). Recently, point perturbation attack (Xiang et al., 2019), kNN attack (Tsai et al., 2020), and point dropping attack (Zheng et al., 2019) are proposed for 3D point cloud. In this paper, our method is trained after the normalization of point clouds. Since the point perturbation attack and the kNN attack don't perform normalization of point clouds during the attack and the generated point clouds may not center within a unit sphere, we only consider the point dropping attack in our robustness test.

We report the recognition accuracy after the point dropping attack on the test set of ModelNet40 in Table 4, where the results of baseline models refer to IF-Defense (Wu et al., 2020). It is observed that the baseline model dramatically degrade. But all models trained with PointCutMix-R and PointCutMix-K still have more than 80% accuracy. It verifies that our method can significantly improve the robustness of the model.

4.6. Point cloud defense

Motivated by the impressive performance under point drop attack, we consider applying our method to the point cloud defense. We surprisingly find that using the pre-trained models trained with PointCutMix augmentation as defense methods outperforms the state-of-the-art defense algorithm IF-Defense (Wu et al., 2020) by a large margin. Specifically, we first generate adversarial point clouds by point dropping attack on the pre-trained baseline model provided by (Wu et al., 2020). We then compare the classifiers trained using PointCutMix augmentation method on these generated adversarial point clouds to several recent developed defense methods, *i.e.*, SRS (Yang et al., 2019a), SOR (Zhou et al.,

Table 5. Classification accuracy of various defense methods on ModelNet40 under point dropping attack (Zheng et al., 2019), kNN attack (Tsai et al., 2020) and point perturbation attack (Xiang et al., 2019). Drop 200 and Drop 100 denote the dropping points is 200 and 100 respectively. * denotes that results are reported in IF-Defense (Wu et al., 2020). We report the best result of three IF-Defense methods. The best and second-place results for each row are emphasized as blue and bold.

Attack	Model	No Defense*	SRS*	SOR*	DUP-Net*	IF-Defense*	PointCutMix-R	PointCutMix-K
Drop 200	PointNet++	68.96	39.63	69.17	72.00	79.09	87.32	89.02
	DGCNN	55.06	23.82	59.36	36.02	73.30	87.36	88.82
Drop 100	PointNet++	80.19	64.51	74.16	76.38	84.56	89.51	91.17
	DGCNN	75.16	49.23	64.68	44.45	83.43	89.59	91.05
kNN	PointNet++	0.00	49.96	61.35	74.88	85.62	83.35	70.71
	DGCNN	20.02	41.25	55.92	35.45	82.33	80.27	68.76
point perturbation	PointNet++	0.00	73.14	77.67	80.63	86.99	86.71	84.93
	DGCNN	0.00	50.20	76.50	42.67	85.53	83.14	76.69

2019), DUP-Net (Zhou et al., 2019) and IF-Defense (Wu et al., 2020). From the results listed in Table 5, we observe that PointCutMix-R and PointCutMix-K consistently surpass all defense methods under two point dropping attacks. The improvement of recognition accuracy can reach up to 15% in a certain case, which fully proves the scalability and effectiveness of our method. It is worth noting that unlike previous defense methods that need to alter the adversarial point clouds which might cause information loss, our method just uses very limited computing power to classify the adversarial point clouds, which is a more natural defense approach.

Moreover, to verify the generalization of PointCutMix in defense, we also test with the kNN attack (Tsai et al., 2020) and the point perturbation attack (Xiang et al., 2019). We first perform normalization on the generated adversarial point clouds to limit all points into a unit sphere and then test it with deep point cloud classification networks trained using PointCutMix-K and PointCutMix-R. The kNN attack smoothes the attack by using a k-Nearest Neighbor loss, which is hard to defense by the simple method such as statistical outlier removal (Zhou et al., 2019). On the contrary, for the point perturbation attack (Xiang et al., 2019) where the attacked point clouds are messy, it can be easily defended by simple random sampling and statistical outlier removal (Zhou et al., 2019). As shown in Table 5, although these two attacks are very unfavorable for our models that are trained with normalized point clouds, PointCutMix-R still achieves second place and has a very close performance to the state-of-the-art defense method in all cases. We can also find that PointCutMix-R constantly surpasses PointCutMix-K in the defense of two attacks, proving the assumption in Section 3.4 that models trained with PointCutMix-R achieve better robustness.

From the above analysis, we can conclude that PointCutMix has strong generalization ability across various point cloud attack algorithms and the defense approach is very simple and computing cost-effective.

5. Conclusion

In this paper, we propose PointCutMix, a regularization strategy for point cloud classification. We conduct extensive experiments to verify the effectiveness of our method. For the object-level point cloud classification problem, the results show that PointCutMix evidently improves the performance of networks that learned with local features. While for the point-wise segmentation task, PointCutMix alleviates the unbalanced distribution problem and enhances the performance of uncommon categories. We also validate that PointCutMix significantly enhances the robustness of the model. By applying our method as a defense method, it outperforms the SOTA defense algorithm. We hope this simple regularization strategy could be applied to more tasks and help future researches.

In the future, we plan to extend our work to 3D object detection (Shi et al., 2020). However, due to the point cloud is different from images, there are still some challenges. For example, in 3D object detection, the point cloud of KITTI (Geiger et al., 2012) and ModelNet are very different, thus it is hard to directly use the pre-trained model of the classification network in the 3D detection task. Moreover, we also plan to apply our PointCutMix-R and PointCutMix-K to defense methods to recently proposed attacks AdvPC (Hamdi et al., 2020) and LG-GAN (Zhou et al., 2020).

References

- Akhtar, N. and Mian, A. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access*, 6:14410–14430, 2018.
- Bhattacharyya, P. and Czarnecki, K. Deformable pv-rcnn: Improving 3d object detection with learned deformations. *arXiv preprint arXiv:2008.08766*, 2020.
- Chen, Y., Liu, S., Shen, X., and Jia, J. Fast point r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9775–9784, 2019.
- Chen, Y., Hu, V. T., Gavves, E., Mensink, T., Mettes, P., Yang, P., and Snoek, C. G. Pointmixup: Augmentation for point clouds. In *European Conference on Computer Vision*, pp. 330–345. Springer, 2020.
- Dolhansky, B., Bitton, J., Pflaum, B., Lu, J., Howes, R., Wang, M., and Ferrer, C. C. The deepfake detection challenge dataset. *arXiv preprint arXiv:2006.07397*, 2020.
- Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., and Li, J. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9185–9193, 2018.
- Geiger, A., Lenz, P., and Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361. IEEE, 2012.
- Guo, H., Mao, Y., and Zhang, R. Mixup as locally linear out-of-manifold regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3714–3722, 2019.
- Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., and Bennamoun, M. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020.
- Hamdi, A., Rojas, S., Thabet, A., and Ghanem, B. Advpc: Transferable adversarial perturbations on 3d point clouds. In *European Conference on Computer Vision*, pp. 241–257. Springer, 2020.
- Harris, E., Marcu, A., Painter, M., Niranjana, M., and Hare, A. P.-B. J. Fmix: Enhancing mixed sample data augmentation. *arXiv preprint arXiv:2002.12047*, 2(3):4, 2020.
- He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., and Li, M. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 558–567, 2019.
- Jing, L. and Tian, Y. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., and Beijbom, O. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12697–12705, 2019.
- Li, R., Li, X., Heng, P.-A., and Fu, C.-W. Pointaugument: an auto-augmentation framework for point cloud classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6378–6387, 2020.
- Liu, J., Ni, B., Li, C., Yang, J., and Tian, Q. Dynamic points agglomeration for hierarchical point sets learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7546–7555, 2019a.
- Liu, M., Sheng, L., Yang, S., Shao, J., and Hu, S.-M. Morphing and sampling network for dense point cloud completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 11596–11603, 2020a.
- Liu, Y., Fan, B., Meng, G., Lu, J., Xiang, S., and Pan, C. Densepoint: Learning densely contextual representation for efficient point cloud processing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5239–5248, 2019b.
- Liu, Y., Fan, B., Xiang, S., and Pan, C. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8895–8904, 2019c.
- Liu, Z., Hu, H., Cao, Y., Zhang, Z., and Tong, X. A closer look at local aggregation operators in point cloud analysis. In *European Conference on Computer Vision*, pp. 326–342. Springer, 2020b.
- Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. 2017.
- Qi, C. R., Su, H., Mo, K., and Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017a.
- Qi, C. R., Yi, L., Su, H., and Guibas, L. J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017b.

- Rao, Y., Lu, J., and Zhou, J. Global-local bidirectional reasoning for unsupervised representation learning of 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5376–5385, 2020.
- Rubner, Y., Tomasi, C., and Guibas, L. J. The earth mover’s distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.
- Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., and Li, H. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10529–10538, 2020.
- Taghanaki, S. A., Hassani, K., Jayaraman, P. K., Khasahmadi, A. H., and Custis, T. Pointmask: Towards interpretable and bias-resilient point cloud processing. *arXiv preprint arXiv:2007.04525*, 2020.
- Te, G., Hu, W., Zheng, A., and Guo, Z. Rgcnn: Regularized graph cnn for point cloud segmentation. In *Proceedings of the 26th ACM international conference on Multimedia*, pp. 746–754, 2018.
- Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., and Guibas, L. J. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6411–6420, 2019.
- Tsai, T., Yang, K., Ho, T.-Y., and Jin, Y. Robust adversarial objects against deep learning models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 954–962, 2020.
- Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., and Bengio, Y. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pp. 6438–6447. PMLR, 2019.
- Walawalkar, D., Shen, Z., Liu, Z., and Savvides, M. Attentive cutmix: An enhanced data augmentation approach for deep learning based image classification. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3642–3646. IEEE, 2020.
- Wang, C., Samari, B., and Siddiqi, K. Local spectral graph convolution for point set feature learning. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 52–66, 2018.
- Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.
- Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.
- Wu, Z., Duan, Y., Wang, H., Fan, Q., and Guibas, L. J. If-defense: 3d adversarial point cloud defense via implicit function based restoration. *arXiv preprint arXiv:2010.05272*, 2020.
- Xiang, C., Qi, C. R., and Li, B. Generating 3d adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9136–9144, 2019.
- Xu, X. and Lee, G. H. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13706–13715, 2020.
- Yan, X., Zheng, C., Li, Z., Wang, S., and Cui, S. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5589–5598, 2020.
- Yang, J., Zhang, Q., Fang, R., Ni, B., Liu, J., and Tian, Q. Adversarial attack and defense on point sets. *arXiv preprint arXiv:1902.10899*, 2019a.
- Yang, J., Zhang, Q., Ni, B., Li, L., Liu, J., Zhou, M., and Tian, Q. Modeling point clouds with self-attention and gumbel subset sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3323–3332, 2019b.
- Yi, L., Kim, V. G., Ceylan, D., Shen, I.-C., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A., and Guibas, L. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016.
- Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6023–6032, 2019.
- Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. In *ICLR*. OpenReview.net, 2018.
- Zhao, H., Jiang, L., Fu, C.-W., and Jia, J. Pointweb: Enhancing local neighborhood features for point cloud processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5565–5573, 2019.

- Zhao, Y., Wu, Y., Chen, C., and Lim, A. On isometry robustness of deep 3d point cloud models under adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1201–1210, 2020.
- Zheng, T., Chen, C., Yuan, J., Li, B., and Ren, K. Pointcloud saliency maps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1598–1606, 2019.
- Zhou, H., Chen, K., Zhang, W., Fang, H., Zhou, W., and Yu, N. Dup-net: Denoiser and upsampler network for 3d adversarial point clouds defense. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1961–1970, 2019.
- Zhou, H., Chen, D., Liao, J., Chen, K., Dong, X., Liu, K., Zhang, W., Hua, G., and Yu, N. Lg-gan: Label guided adversarial network for flexible targeted attack of point cloud based deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10356–10365, 2020.