PointPatchMix: Point Cloud Mixing with Patch Scoring

Yi Wang[1*], Jiaze Wang[2*], Jinpeng Li[2], Zixu Zhao[2], Guangyong Chen[3], Anfeng Liu[1†], Pheng-Ann Heng[2]

[1]Central South University, Hunan, China
[2]The Chinese University of Hong Kong, Hong Kong SAR, China
[3]Zhejiang Lab, Zhejiang, China
csu-wy@csu.edu.cn, jzwang@link.cuhk.edu.hk, afengliu@mail.csu.edu.cn

Abstract

Data augmentation is an effective regularization strategy for mitigating overfitting in deep neural networks, and it plays a crucial role in 3D vision tasks, where the point cloud data is relatively limited. While mixing-based augmentation has shown promise for point clouds, previous methods mix point clouds either on block level or point level, which has constrained their ability to strike a balance between generating diverse training samples and preserving the local characteristics of point clouds. The significance of each part component of the point clouds has not been fully considered, as not all parts contribute equally to the classification task, and some parts may contain unimportant or redundant information. To overcome these challenges, we propose PointPatchMix, a novel approach that mixes point clouds at the patch level and integrates a patch scoring module to generate content-based targets for mixed point clouds. Our approach preserves local features at the patch level, while the patch scoring module assigns targets based on the content-based significance score from a pre-trained teacher model. We evaluate PointPatchMix on two benchmark datasets including ModelNet40 and ScanObjectNN, and demonstrate significant improvements over various baselines in both synthetic and real-world datasets, as well as few-shot settings. With PointMAE as our baseline, our model surpasses previous methods by a significant margin. Furthermore, our approach shows strong generalization across various point cloud methods and enhances the robustness of the baseline model. Code is available at https://jiazewang.com/projects/pointpatchmix.html.

Introduction

The precise classification of point clouds has emerged as a pivotal and intricate research focus in the field of computer vision, boasting extensive applications in real-world scenarios such as virtual reality, augmented reality, and autonomous navigation. Deep Neural Networks (DNNs) have made remarkable progress in this area in recent years (Qi et al. 2017a,b; Zhao et al. 2021; Pang et al. 2022; Yu et al. 2022). However, the scarcity of point cloud training data inherently exposes DNNs to overfitting risk, attributed to their inclination to model approximation based on the given data distribution. To alleviate this challenge, many works (Kim et al. 2021; Rao et al. 2021; Lee et al. 2022) have put efforts into point cloud data augmentation techniques which serve to enrich the variety within the training data, thereby enhancing the generalizability of point cloud classifiers.

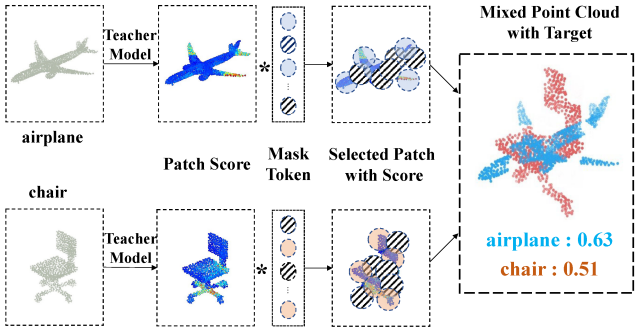

Figure 1: Illustration of generating mixed data using PointPatchMix. Given two point clouds, PointPatchMix processes them at the patch level, with each patch comprising 32 points. A pre-trained teacher model scores each patch based on self-attention mechanism. Then, the mixed point cloud consists of patches selected by mask tokens and the new ground truth is generated.

Many techniques have been proposed to augment point cloud data, such as handcrafted approaches (Ren, Pan, and Liu 2022; Qi et al. 2017a,b; Goyal et al. 2021), and automatically generated approaches. (Li et al. 2020; Zhang et al. 2022a; Chen et al. 2020; Lee et al. 2021). Notably, Among them, mixing-based augmentation (Zhang et al. 2022a; Chen et al. 2020; Lee et al. 2021) has shown great potential for point clouds. In this paradigm, mixed point clouds are endowed with labels derived from their original counterparts. For instance, PointMixup (Chen et al. 2020) defines data augmentation between point clouds as a shortest path linear interpolation and generates new samples by optimally assigning a path function. Similarly, PointCutMix (Zhang et al. 2022a) determines optimal correspondences between two point clouds and create fresh training instances through point replacement.

Despite the significant advancements in mixing-based

*These authors contributed equally.
†Corresponding Author.

techniques, there still remain some limitations that demand careful consideration. Firstly, prior methods such as PointMixup (Chen et al. 2020) and PointCutMix (Zhang et al. 2022a) manifest limitations in their generation of mixed point clouds, operating predominantly at either the block or point levels. Although point-level mixing can increase the diversity of training samples, it may not adequately preserve the local characteristics of the point cloud. While block-level mixing may limit the diversity of the generated point clouds when retaining local features. Striking a balance between diversity and accuracy can therefore be challenging. Furthermore, existing methods generate targets for mixed point clouds through a simple linear combination based on the point cloud sample ratio. However, it may not be optimal for many scenarios where the importance of each patch in the point cloud may significantly differ. Therefore, it is imperative to develop an augmentation method at the patch level that generates content-based targets.

To overcome these challenges, we propose PointPatchMix, a novel point cloud mixing method that operates at the patch level and integrates a patch scoring module. As shown in Figure 1, our patch-level mixing approach generates diverse training samples while preserving the local features of the point clouds, making it an ideal choice for the Transformer-based methods which employs patch embeddings as the basic input units. Additionally, we observe that not all point patches contribute equally to the final classification scores, and some may contain unimportant or redundant information. To address this, we generate significance scores for each patch using a pre-trained teacher network, which considers the value of the information contained in each patch. We conduct extensive experiments on ModelNet40 and ScanObjectNN, and evaluate PointPatchMix with various baselines. With Point-MAE (Pang et al. 2022) as the baseline, our model achieves 86.3% accuracy on ScanObjectNN and 94.1% accuracy on ModelNet40, which outperforms previous methods by a notable margin. PointPatchMix also demonstrates remarkable generalization across diverse architectures, enabling the state-of-the-art architecture, PointGPT (Chen et al. 2023a), to achieve a remarkable accuracy rate of 88.3% on ScanObjectNN.

Our main contributions can be summarized as follows:

- We introduce PointPatchMix, an innovative patch-level augmentation technique that generates diverse and authentic augmented training samples.
- A patch scoring module is proposed to generate content-based targets for the mixed point clouds, which allows for more accurate ground truths for classification.
- Our results demonstrate that PointPatchMix achieves significant improvements in synthetic datasets, real-world datasets and few-shot settings.

Related Works

Transformer with self-attention on point cloud. The transformer model (Vaswani et al. 2017), leveraging the self-attention mechanism, has exhibited remarkable success in the field of NLP (Brown et al. 2020; Lee and Toutanova 2018; Radford et al. 2019; Raffel et al. 2020), surpassing Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to emerge as a paramount feature processor (Chen et al. 2023b; Zhao et al. 2023; Wang, Chen, and Dou 2021). This achievement has been extended to the two-dimensional image domain with ViT (Dosovitskiy et al. 2020; Yuan et al. 2021; Wang et al. 2021b,c; Zhao et al. 2021; Fan and Xiao 2022; Fan et al. 2023). In the point cloud, the transformer architecture is also constantly evolving and improving (Liu et al. 2021). The Point transformer (Zhao et al. 2021) leverages localized self-attention to enhance the applicability of the transformer within expansive and intricate 3D scenarios. Recently, Point-MAE (Pang et al. 2022) demonstrates the significant potential of transformer in processing point cloud for self-supervised learning by using masked self-coding scheme and displaying high generalization ability in downstream tasks.

Data augmentation on point cloud. Many of the previous works (Phan et al. 2018; Qi et al. 2017a,b; Hu et al. 2020) have mostly applied simple data augmentation at the point level, such as random jitter, rotation, and scaling. Li et al. (Li et al. 2020) proposed PointAugment to optimize both enhancer and classifier networks, allowing the framework to generate more complex point cloud samples. PointMixUp, proposed by Chen et al. (Chen et al. 2020), extends Mixup into point clouds to fuses two samples using optimal linear interpolation. To upload the structural integrity of point clouds, Lee et al. (Lee et al. 2021) proposed RsMix, which inherits the advantage of CutMix. It utilizes rigid transformations to mix two samples while maintaining the shape of the original data. PointWOLF (Kim et al. 2021) proposes a multiple weighted local transformations method for data augmentation. PointMixSwap (Umam et al. 2022) employs an attention-based method to diversify 3D point clouds by swapping corresponding divisions across multiple point clouds. With the growing Transformer-based techniques in point clouds (Pang et al. 2022; Zhang et al. 2022b; Chen et al. 2023a), patch-level local point clouds are more effective in enhancing model performance (Sheshappanavar, Singh, and Kambhamettu 2021). To meet this requirement, we propose a novel data augmentation called *PointPatchMix*, which involves point cloud mixing at the patch level, offering a more realistic ground truth for augmented data.

Methods

Preliminary

Problem Setting. In a point cloud classification task, the target is to train a function $f : x \rightarrow [0,1]^C$ to map a point cloud x containing N points to a one-hot class label, where C is the number of classes and $x \in R^{3\times N}$. The optimal parameters of the network can be learned by minimizing the loss between the ground truth y and prediction $f(x)$.

Revisiting PointCutMix. PointCutMix is proposed as a solution to address the limited scale of point cloud datasets by mixing pairs of samples with a random binary mask. PointCutMix creates a new training point cloud $(\tilde{x}, \tilde{y})$ based on a pair of point clouds (x_1, y_1) and (x_2, y_2). The combining

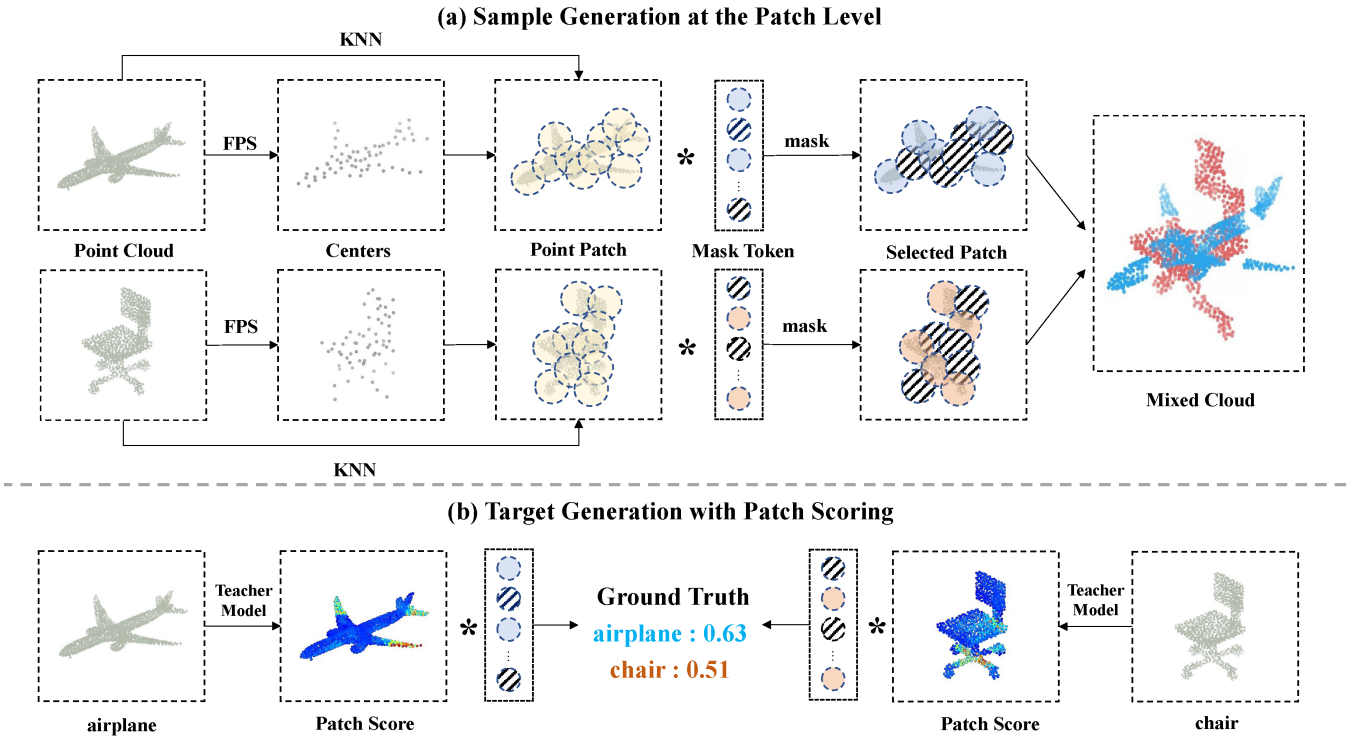

Figure 2: The overall scheme of the PointPatchMix. (a) The original point clouds are divided into multiple patches, subsequently undergoing mask token processing and mixing. (b) A pre-trained teacher model assigns each patch with a content-based significance score. The ground truth of the mixed point cloud is ascertained by aggregating the scores of designated patches.

operation of PointCutMix is as follows:

$$\tilde{x} = M \odot x_1 + (1 - M) \odot x_2 \quad (1)$$

$$\tilde{y} = \lambda y_1 + (1 - \lambda) y_2 \quad (2)$$

$$\lambda = \frac{\sum M}{N} \quad (3)$$

where $M \in \{0, 1\}^N$ denotes which sample the point belongs to, $\odot$ denotes element-wise multiplication, and λ is sampled from a beta distribution $Beta(\beta, \beta)$, which means $\lfloor \lambda N \rfloor$ points are selected from x_1 and $N - \lfloor \lambda N \rfloor$ points are selected from x_2. PointCutMix applies a similar approach to CutMix and PointMixup by creating a mixed target for the generated point clouds through a linear combination of y_1 and y_2 with λ. We argue that block-level mixing in PointCutMix (PointCutMix-K) may not be the optimal choice for mixing-based augmentation. This is because it randomly samples one point and its nearest neighbors to mask, which restricts the diversity of the generated point clouds. On the other hand, point-level mixing (PointCutMix-R) randomly selects points and may not retain the local characteristics of the point cloud. Hence, it is necessary to balance the richness of the generated point cloud while preserving its local characteristics. Drawing inspiration from the architecture of Transformer (Han et al. 2021), we propose patch-level augmentation as a viable solution for mixing-based augmentation. In addition, the label of the mixed point cloud in PointCutMix is generated through a linear combination of the labels of the original pairs, with the mixing ratio λ estimated solely based on the mask size. However, this simplistic approach may not be suitable for many scenarios, as the importance of each patch in the point cloud may vary significantly. Therefore, it is crucial to develop an augmentation method at the patch level that generates content-based target scores.

Point Patch Scoring

To address the limitations of previous methods, we propose a point patch scoring module which can generate content-based targets. Our intuition is that not all patches in point clouds contribute equally to classification accuracy, and we aim to balance the richness of the generated point clouds with their local characteristics by focusing on patch-level mixing. Consequently, a crucial aspect of our approach is determining each patch's significance score. A natural approach is to utilize a pre-trained teacher network based on the Transformer architecture (Pang et al. 2022) to generate the patch significance score.

In a typical self-attention layer of a Transformer (Vaswani et al. 2017), the input tokens $I \in R^{N \times d}$ are used to compute the queries $Q \in R^{N \times d}$, keys $K \in R^{N \times d}$, and values $V \in R^{N \times d}$. The attention matrix A can be obtained by the dot product of queries and keys, which is then scaled by $\sqrt{d}$:

$$A = Softmax(QK^T/\sqrt{d}) \quad (4)$$

Each row of matrix A sums up to 1 due to the Softmax

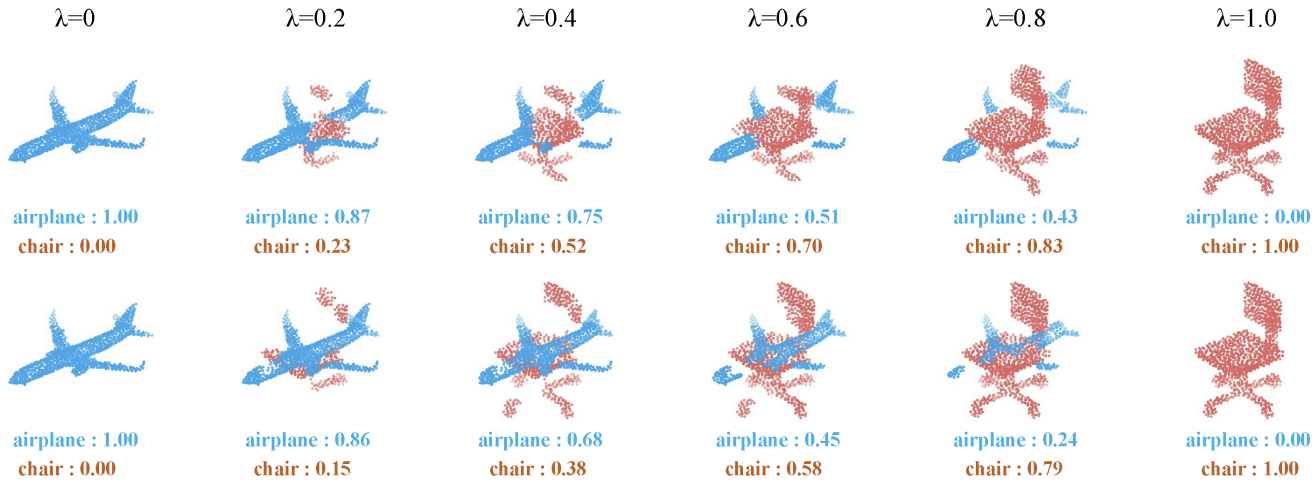

Figure 3: The visualization of the mixed samples between a plane and a chair under different replacement ratios λ. Top: block-level mixing. Bottom: patch-level mixing.

function. Then the output tokens O are computed by the combination of values weighted by the attention weights:

$$O = AV \tag{5}$$

Each row of matrix A corresponds to the attention weights associated with a particular output token, indicating the relative contributions of all input tokens to that specific output token. Specifically, the attention weights in row $A_{1,:}$ represent the classification token, whereby $A_{1,j}$ signifies the relevance of input token j to the output classification token. To facilitate pruning of the attention matrix A, we utilize the weights $A_{1,2}, ..., A_{1,N+1}$ as significance scores while disregarding $A_{1,1}$ since we retain the classification token. As the output tokens O are dependent on both attention matrix A and values V, we incorporate the norm of V_j when determining the significance score of token j. This is motivated by the notion that values with a norm near zero have a negligible impact and thus, their corresponding tokens are deemed less significant. As such, the significance score of patch token j is calculated as follows:

$$S_j = \frac{A_{1,j} \times \|V_j\|}{\sum_{i=2} A_{1,i} \times \|V_i\|} \tag{6}$$

where $i, j \in 2...N+1$. For a multi-head attention layer, we compute the significance scores for each head separately and aggregate them by taking the sum over all heads.

Point Patch Mixing

Optimal patch assignment. Mixing patches at the patch level requires establishing a one-to-one relationship between the patches of two input point clouds. In the image domain, pixels are arranged in a grid formation, and two images can be naturally matched according to their coordinates by resizing or cropping them to the same size. However, point clouds lack a predetermined order and are permutation-invariant, making it essential to define the correspondence between patches based on rules other than their position.

Following PointMixup (Chen et al. 2020) and PointCutMix (Zhang et al. 2022a), we adopt the Earth Mover's Distance (EMD) function to establish an optimal assignment between two point clouds, due to its efficacy in accounting for the geometric relationship, local details, and density distributions of two point clouds. The point clouds are denoted as x_1 and x_2. The EMD calculates the minimum total displacement required for matching each point in x_1 to the corresponding point in x_2. The assignment function θ^* in the EMD can be calculated as:

$$\theta^* = argmin_{\theta \in \Theta} \sum_i \left\| x_{1,i} - x_{2,\theta(i)} \right\|_2 \tag{7}$$

where Θ gives a one-to-one correspondences between two point clouds at the point level. And getting the assignment function at the point level. We can easily extend it into the patch level.

$$\theta_P^* = argmin_{\theta \in \Phi} \sum_p \sum_{i \in p} (\left\| x_{1,i} - x_{2,\theta(i)} \right\|_2) \tag{8}$$

where Φ give a one-to-one correspondeces between two point clouds at the patch level, p is the number of patches. We can use the total points to calculate the patch correspondences with the lowest EMD at the patch level by Equation 8. What's more, we find that directly using the center point position of each patch to represent the whole patch and Equation 7 can obviously save the preprocessing time and get a similar performance. Thus we use the center points of each patch to calculate the optimal patch assignment.

Mixing algorithm. Figure 2 presents an overview of our PointPatchMix. Firstly, we divide the input point cloud x into patches x_p. Next, we generate a patch-level random mask M^t based on the mixing ratio λ. With the significance score from the pre-trained teacher model, the mixed new training sample f is then constructed as follows:

$$\tilde{x}^p = M^t \odot x_1^p + (1 - M^t) \odot x_2^p \tag{9}$$

$$\tilde{y} = \sum_{j=2} M_j^t \odot S_{1j} + \sum_{j=2} (1 - M_j^t) \odot S_{2j} \tag{10}$$

Methods	OBJ-BG	OBJ-ONLY	PB-T50-RS
PointNet	73.3	79.2	68
SpiderCNN	77.1	79.5	73.7
PointNet++	82.3	84.3	77.9
DGCNN	82.8	86.2	78.1
PointCNN	86.1	85.5	78.5
BGA-PN++	-	-	80.2
GBNet	-	-	80.5
PRANet	-	-	81
Transformer-OcCo	84.8	85.5	78.8
Point-BERT	87.4	88.1	83
PointNeXt	-	-	87.7
Point-M2AE	91.2	88.8	86.4
Point-MAE	90.0	88.2	85.1
+PointCutMix	90.5	88.9	85.4
+PointPatchMix	**90.9**	**91.0**	**86.3**
PointGPT	91.6	90.0	86.9
+PointCutMix	91.7	90.2	87.5
+PointPatchMix	**91.9**	**90.7**	**88.3**

Table 1: Comparision with state-of-the-art methods on ScanObjectNN. We report the classification accuracy (%) on three splits of ScanObjectNN.

Methods	Augmentation	Accuracy
PointNet	JRST	89.2
PointNet++	JRST	90.7
PointCNN	T	92.5
KPConv	color drop	92.9
DGCNN	ST	92.9
RS-CNN	ST	92.9
PCT	point drop, T	93.2
PVT	JRST	93.6
PointTransformer	color auto-contrast	93.7
PointMLP	T	94.5
+PointCutMix	PointCutMix	94.5
+PointPatchMix	PointPatchMix	**94.7**

Table 2: Comparision with state-of-the-art supervised methods on ModelNet40. JRST represents jitter, random rotation, random scaling, and translation respectively.

where $j \in 2...N+1$, $\odot$ denotes element-wise multiplication, M_j^t denotes the j-th patch token of the mask M^t, S_{1j} and S_{2j} are the j-th patch token of the significance score of x_1 and x_2 respectively. The visualization of mixed samples under different replacement ratios are shown in Figure 3.

Experiments

Datasets

We conduct extensive experiments on both synthetic and real-world datasets in point cloud shape classification to evaluate the effectiveness of PointPatchMix, i.e., ModelNet40 and ScanObjetNN .

ModelNet40. It is a widely-used clean point cloud object dataset for classification tasks, comprising 12,311 samples spanning 40 object categories. We follow the standard pattern, using 9843 samples for training and 2468 samples for testing. During the training process, we exclusively utilize the proposed hybrid method for data augmentation. For a fair comparison, we adopt the standard voting methods used in prior work during the testing process.

ScanObjectNN. It is a point-cloud object dataset derived from the real world, which contains about 15,000 samples of 15 object categories. These samples are obtained from real scenes, often with occlusion and noise, and therefore extremely challenging. No voting method is used during testing on ScanObjectNN.

Methods	Augmentation	Accuracy
OcCo	ST	93.0
STRL	T	93.1
IAE	ST	93.7
Transformer-OcCo	ST	92.1
Point-BERT	resampling	93.2
Point-MAE	ST	93.8
+PointCutMix	PointCutMix	93.9
+PointPatchMix	PointPatchMix	**94.1**

Table 3: Comparision with state-of-the-art self-supervised methods on ModelNet40.

Experimental Details

To ensure that PointPatchMix is a general data augmentation method, we select multiple popular architectures in the current point cloud field to evaluate its availability and effectiveness. In the specific experimental configuration, all networks are uniformly provided 1024 points for training learning with 300 epochs and a batch size of 32. Meanwhile, we maintain the same configuration as possible with the original published paper to facilitate a fair comparison with the baseline. For PointNet, PointNet++, we use the Adam optimizer with an initial learning rate of 0.001 and a decay rate of 0.5 per 20 cycles. For Point-MAE, we use the AdamW optimizer with an initial learning rate of 0.001 and a weight decay of 0.05. A cosine annealing strategy is used to attenuate the learning rate. We adopt Point-MAE as our teacher model to generate the patch token scores and save the scores offline to improve training efficiency.

Experimental Results

In this subsection, we utilize Point-MAE, a robust classification model, as our baseline. PointPatchMix achieves impressive improvements in various scenarios, including real-world data, synthetic data, and few-shot learning settings.

Comparison with state-of-the-art methods on ScanObjectNN. Table 1 demonstrates the classification results on ScanObjectNN. Notably, the Point-MAE with PointPatchMix exhibits superior performance over the baseline, showcasing significant improvements of 0.9% on OBJ-BG, 2.8% on OBJ-ONLY, and 1.2% on PB-T50-RS, respectively. Furthermore, we investigate the potential of Point-MAE in combination with the advanced PointGPT architecture as the baseline. This analysis demonstrates additional performance enhancements, thereby substantiating the efficacy of the PointPatchMix within this context.

Comparison with state-of-the-art methods on ModelNet40. Table 3 presents the classification results on ModelNet40. All the methods are given 1024 points that only contain coordinate information. Our PointPatchMix achieves 94.1% accuracy and even outperforms Point-MAE with 8192 points as input. We also adopt PointMLP as our

Methods	5-way,10-shot	5-way,20-shot	10-way,10-shot	10-way,20-shot
PointNet	52.0 ± 3.8	57.8 ± 4.9	46.6 ± 4.3	35.2 ± 4.8
PointNet + OcCo	89.7 ± 1.9	92.4 ± 1.6	93.9 ± 1.8	89.7 ± 1.5
PointNet + CrossPoint	90.9 ± 4.8	93.5 ± 4.4	83.6 ± 4.7	90.2 ± 2.2
DGCNN	31.6 ± 2.8	40.8 ± 4.6	19.9 ± 2.1	16.9 ± 1.5
DGCNN + OcCo	90.6 ± 2.8	92.5 ± 1.9	82.9 ± 1.3	86.5 ± 2.2
DGCNN + CrossPoint	92.5 ± 3.0	94.9 ± 2.1	83.6 ± 5.3	87.9 ± 4.2
Transformer-rand	87.8 ± 5.2	93.3 ± 4.3	84.6 ± 5.5	89.4 ± 6.3
Transformer-OcCo	94.0 ± 3.6	95.9 ± 2.3	89.4 ± 5.1	92.4 ± 4.6
Point-BERT	94.6 ± 3.1	96.3 ± 2.7	91.0 ± 5.4	92.7 ± 5.1
Point-MAE	96.3 ± 2.5	97.8 ± 1.8	92.6 ± 4.1	95.0 ± 3.0
+PointCutMix	96.4 ± 2.0	97.8 ± 2.3	92.6 ± 3.8	95.1 ± 3.6
+PointPatchMix	$\mathbf{96.6 \pm 2.2}$	$\mathbf{97.8 \pm 1.7}$	$\mathbf{92.7 \pm 3.3}$	$\mathbf{95.2 \pm 2.7}$
PointGPT	96.8 ± 2.0	98.6 ± 1.1	92.6 ± 4.6	95.2 ± 3.4
+PointCutMix	96.8 ± 2.2	98.6 ± 1.1	92.7 ± 3.1	95.4 ± 3.7
+PointPatchMix	$\mathbf{97.0 \pm 2.2}$	$\mathbf{98.7 \pm 1.1}$	$\mathbf{93.0 \pm 3.7}$	$\mathbf{95.7 \pm 3.1}$

Table 4: Few-shot classification on ModelNet40. We report the accuracy with standard of 10 independent experiments.

Method	PointNet	PointNet++	Transformer
baseline	89.2	90.7	91.4
PointMixup	89.9	92.7	92.1
RSMix	89.7	92.1	92.4
PointMixSwap	89.7	92.3	92.5
PointCutMix	89.6	92.3	92.5
PointPatchMix	**90.1**	**92.9**	**93.3**

Table 5: Comparison with other mixing-based methods. We report the classification accuracy(%).

baseline and PointPatchMix further improve the accuracy to 94.7%, thus the effectiveness of our PointPatchMix for point-based architecture can be proved.

Few-shot Learning. In order to assess the generalization ability of PointPatchMix with limited training data, we conduct the few-shot classification task on ModelNet40. Following standard practice, we performed N-way K-shot experiments on PointPatchMix 10 times, randomly selecting N classes from ModelNet40 and sampling K objects from each class. Table 4 demonstrates PointPatchMix outperforms previous methods in all few-shot settings, indicating its strong capacity with limited training data.

Ablation Studies

To evaluate the contribution of each component in PointPatchMix, we conduct ablation studies on ModelNet40. We mainly use the Transformer proposed in Point-MAE(Pang et al. 2022) without pre-training as our baseline model for evaluation, which archives 91.4% on ModelNet40.

Comparison with other mixing-based methods on ModelNet40. We compare PointPatchMix with other mixing-based methods in Table 5. The results depicted in the table unequivocally demonstrate the superior performance of PointPatchMix across architectures such as PointNet, PointNet++, and Transformer. We can observe that other mixing-based methods have limited performance gains on Transformer compared to PointNet and PointNet++. On the contrary, our proposed PointPatchMix improves our baseline by 1.9% and surpasses PointCutMix by 0.8% on Transformer, which validates that PointPatchMix fits well for Transformer-based architectures.

Mixing Level	Target Generation	Patch Assignment	Beta Value	Accuracy
-	-	-	-	91.4
Block	Linear	All	1	91.7
Patch	Linear	All	1	92.4
Patch	Score	All	1	92.9
Patch	Score	Center	1	92.9
Patch	Score	Center	0.5	92.8
Patch	Score	Center	1.5	**93.3**
Patch	Score	Center	2	93.1

Table 6: Ablation Studies. We conducted experiments in a total of four perspectives: mixing level, target generation, patch assignment, and the influence of β, and report the classification accuracy (%) of Transformer on ModelNet40.

Mixing Level. In our ablation study, we compared patch-level mixing with block-level mixing and observed that patch-level mixing outperforms block-level mixing by 0.7% on ModelNet40, as shown in Table 6. This indicates that selecting points at the patch level generates more diverse training data while preserving the local features of the point clouds. Hence, we chose patch-level mixing as our main approach for PointPatchMix.

Target Generation. To validate the effectiveness of our proposed point patch scoring module, we compared it with the approach of using a linear combination to generate the targets based on the number of selected patches for each point cloud. Our experimental results clearly demonstrate that our content-based point patch scoring method outperforms the linear combination method, indicating that our approach can better capture the content information of each object and generate more accurate targets.

Optimal patch assignment points. We also explore two

Figure 4: Qualitative examples of PointPatchMix. For each set of samples, the mixed point cloud (right) consists of blue and red patches, where the blue patches are randomly selected from the left object and the red patches are the corresponding complementary part from the middle object. At the bottom of the point clouds, the blue and red numbers represent the scores of these patches, while the black numbers are the percentage of the number of points.

Transforms	T	+PCM	+ PPM
-	91.4	92.5 (1.1↑)	**93.3 (1.9↑)**
Noise σ=0.01	91.0	91.8 (0.8↑)	**92.3 (1.3↑)**
Noise σ=0.03	75.8	77.8 (3.0↑)	**79.7 (3.9↑)**
Z-rotation[-30 30]	80.2	84.2 (4.0↑)	**85.2 (5.0↑)**
X-rotation[-30 30]	81.7	87.2 (5.5↑)	**87.6 (5.9↑)**
Y-rotation[-30 30]	87.5	90.0 (2.5↑)	**90.5 (3.0↑)**
Scale (0.6)	68.3	80.6 (12.1↑)	**81.7 (13.4↑)**
Scale (2.0)	42.3	44.8 (2.5↑)	**47.4 (5.1↑)**
DropPoint(0.2)	91.1	92.1 (1.0↑)	**92.5 (1.4↑)**

Table 7: Robustness analysis, where T, PCM, PPM represent for Transformer, PointCutMix and PointPatchMix respectfully. We report the classification accuracy (%) with Transformer to four noisy environments: jitter, rotation, scaling, and DropPoint.

different ways of patch assignment, and we find that there are no obvious differences between using all points and using only center points for optimal patch assignment, while center points based patch assignment requires less preprocessing time. Thus using center points for optimal patch assignment is a more efficient and practical option.

Influence of β. We investigate the influence of β which is used to sample the data from a beta distribution. Our model achieves best when $\beta = 1.5$. Thus we set $\beta = 1.5$ as default for all experiments.

Robustness. We tested the robustness (Sun et al. 2022) of PointPatchMix with Transformer to four noisy environments: jitter, rotation, scaling, and DropPoint, in order to verify that our method makes the model robust to noise. As we can see from Table 7, PointPatchMix improves the baseline model notably and outperforms PointCutMix on all the test settings, which verified that PointPatchMix can successfully improve the robustness ability of the baseline model.

Qualitative examples of PointPatchMix. We present additional examples of PointPatchMix for qualitative analysis in Figure 4. The diversity of target values and patch ratios in the generated point clouds highlights the effectiveness of our proposed patch scoring module. By generating targets based on patch contents and emphasizing important regions during training, PointPatchMix enables the network to focus on informative areas, resulting in improved point cloud classification performance.

Conclusion

Data augmentation is a critical and challenging task in point cloud processing. To address this issue, we propose a novel mixing-based data augmentation method, PointPatchMix. This approach generates diverse training samples while preserving the local characteristics of point clouds. Our patch-level mixing strikes a balance between block-level and point-level mixing methods and outperforms their performance. Furthermore, our patch scoring module assigns more realistic content-based targets to the mixed point clouds. Experimental results demonstrate that PointPatchMix performs competitively in various settings, including real-world point cloud classification, synthetic point cloud classification, and few-shot learning settings. It also demonstrates strong generalization across multiple architectures and enhances the baseline model's robustness.

Our study focuses primarily on point cloud classification, which is crucial in 3D scene understanding. However, point cloud data has various applications in other domains, such as scene and part segmentation. Thus, a promising future direction would be to broaden the scope of our investigation to encompass these areas. This effort could contribute to the development of more comprehensive and sophisticated point augmentation methods for point cloud analysis.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (62072475). The work described in this paper was supported in part by the following grant from the Research Grants Council of the Hong Kong SAR, China (Project No. T45-401/22-N).

References

Afham, M.; Dissanayake, I.; Dissanayake, D.; Dharmasiri, A.; Thilakarathna, K.; and Rodrigo, R. 2022. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9902–9912.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Chen, G.; Wang, M.; Yang, Y.; Yu, K.; Yuan, L.; and Yue, Y. 2023a. PointGPT: Auto-regressively Generative Pre-training from Point Clouds. *arXiv preprint arXiv:2305.11487*.

Chen, H.; Wang, J.; Shao, K.; Liu, F.; Hao, J.; Guan, C.; Chen, G.; and Heng, P.-A. 2023b. Traj-MAE: Masked Autoencoders for Trajectory Prediction. *ICCV*.

Chen, Y.; Hu, V. T.; Gavves, E.; Mensink, T.; Mettes, P.; Yang, P.; and Snoek, C. G. 2020. Pointmixup: Augmentation for point clouds. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, 330–345. Springer.

Cheng, S.; Chen, X.; He, X.; Liu, Z.; and Bai, X. 2021. Pranet: Point relation-aware network for 3d point cloud analysis. *IEEE Transactions on Image Processing*, 30: 4436–4448.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fan, J.; and Xiao, C. 2022. Generalized data distribution iteration. *arXiv preprint arXiv:2206.03192*.

Fan, J.; Zhuang, Y.; Liu, Y.; Jianye, H.; Wang, B.; Zhu, J.; Wang, H.; and Xia, S.-T. 2023. Learnable behavior control: Breaking atari human world records via sample-efficient behavior selection. In *International Conference on Learning Representations*.

Goyal, A.; Law, H.; Liu, B.; Newell, A.; and Deng, J. 2021. Revisiting point cloud shape classification with a simple and effective baseline. In *International Conference on Machine Learning*, 3809–3820. PMLR.

Guo, M.-H.; Cai, J.-X.; Liu, Z.-N.; Mu, T.-J.; Martin, R. R.; and Hu, S.-M. 2021. Pct: Point cloud transformer. *Computational Visual Media*, 7: 187–199.

Han, K.; Xiao, A.; Wu, E.; Guo, J.; Xu, C.; and Wang, Y. 2021. Transformer in transformer. *Advances in Neural Information Processing Systems*, 34: 15908–15919.

Hu, L.; Qin, M.; Zhang, F.; Du, Z.; and Liu, R. 2020. RSCNN: a CNN-based method to enhance low-light remote-sensing images. *Remote Sensing*, 13(1): 62.

Huang, S.; Xie, Y.; Zhu, S.-C.; and Zhu, Y. 2021. Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6535–6545.

Kim, S.; Lee, S.; Hwang, D.; Lee, J.; Hwang, S. J.; and Kim, H. J. 2021. Point cloud augmentation with weighted local transformations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 548–557.

Lee, D.; Lee, J.; Lee, J.; Lee, H.; Lee, M.; Woo, S.; and Lee, S. 2021. Regularization strategy for point cloud via rigidly mixed sample. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15900–15909.

Lee, J.; and Toutanova, K. 2018. Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Lee, S.; Jeon, M.; Kim, I.; Xiong, Y.; and Kim, H. J. 2022. Sagemix: Saliency-guided mixup for point clouds. *Advances in Neural Information Processing Systems*, 35: 23580–23592.

Li, R.; Li, X.; Heng, P.-A.; and Fu, C.-W. 2020. Pointaugment: an auto-augmentation framework for point cloud classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6378–6387.

Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; and Chen, B. 2018. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31.

Liu, Y.; Fan, B.; Xiang, S.; and Pan, C. 2019. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8895–8904.

Liu, Y.; Zhang, Y.; Wang, Y.; Hou, F.; Yuan, J.; Tian, J.; Zhang, Y.; Shi, Z.; Fan, J.; and He, Z. 2021. A survey of visual transformers. *arXiv preprint arXiv:2111.06091*.

Ma, X.; Qin, C.; You, H.; Ran, H.; and Fu, Y. 2022. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. *arXiv preprint arXiv:2202.07123*.

Pang, Y.; Wang, W.; Tay, F. E.; Liu, W.; Tian, Y.; and Yuan, L. 2022. Masked autoencoders for point cloud self-supervised learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, 604–621. Springer.

Phan, A. V.; Le Nguyen, M.; Nguyen, Y. L. H.; and Bui, L. T. 2018. Dgcnn: A convolutional neural network over large-scale labeled graphs. *Neural Networks*, 108: 533–543.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

Qian, G.; Li, Y.; Peng, H.; Mai, J.; Hammoud, H.; Elhoseiny, M.; and Ghanem, B. 2022. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 35: 23192–23204.

Qiu, S.; Anwar, S.; and Barnes, N. 2021. Geometric back-projection network for point cloud classification. *IEEE Transactions on Multimedia*, 24: 1943–1955.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1): 5485–5551.

Rao, Y.; Liu, B.; Wei, Y.; Lu, J.; Hsieh, C.-J.; and Zhou, J. 2021. Randomrooms: Unsupervised pre-training from synthetic shapes and randomized layouts for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3283–3292.

Ren, J.; Pan, L.; and Liu, Z. 2022. Benchmarking and analyzing point cloud classification under corruptions. In *International Conference on Machine Learning*, 18559–18575. PMLR.

Sheshappanavar, S. V.; Singh, V. V.; and Kambhamettu, C. 2021. Patchaugment: Local neighborhood augmentation in point cloud classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2118–2127.

Sun, J.; Zhang, Q.; Kailkhura, B.; Yu, Z.; Xiao, C.; and Mao, Z. M. 2022. Benchmarking robustness of 3d point cloud recognition against common corruptions. *arXiv preprint arXiv:2201.12296*.

Thomas, H.; Qi, C. R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; and Guibas, L. J. 2019. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6411–6420.

Umam, A.; Yang, C.-K.; Chuang, Y.-Y.; Chuang, J.-H.; and Lin, Y.-Y. 2022. Point mixswap: Attentional point cloud mixing via swapping matched structural divisions. In *European Conference on Computer Vision*, 596–611. Springer.

Uy, M. A.; Pham, Q.-H.; Hua, B.-S.; Nguyen, T.; and Yeung, S.-K. 2019. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1588–1597.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, H.; Liu, Q.; Yue, X.; Lasenby, J.; and Kusner, M. J. 2021a. Unsupervised point cloud pre-training via occlusion completion. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9782–9792.

Wang, J.; Chen, K.; and Dou, Q. 2021. Category-level 6D object pose estimation via cascaded relation and recurrent reconstruction networks. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4807–4814. IEEE.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2021b. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, 568–578.

Wang, W.; Yao, L.; Chen, L.; Lin, B.; Cai, D.; He, X.; and Liu, W. 2021c. Crossformer: A versatile vision transformer hinging on cross-scale attention. *arXiv preprint arXiv:2108.00154*.

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1912–1920.

Xu, Y.; Fan, T.; Xu, M.; Zeng, L.; and Qiao, Y. 2018. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In *Proceedings of the European conference on computer vision (ECCV)*, 87–102.

Yan, S.; Yang, Z.; Li, H.; Guan, L.; Kang, H.; Hua, G.; and Huang, Q. 2022. Implicit autoencoder for point cloud self-supervised representation learning. *arXiv preprint arXiv:2201.00785*.

Yu, X.; Tang, L.; Rao, Y.; Huang, T.; Zhou, J.; and Lu, J. 2022. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19313–19322.

Yuan, L.; Chen, Y.; Wang, T.; Yu, W.; Shi, Y.; Jiang, Z.-H.; Tay, F. E.; Feng, J.; and Yan, S. 2021. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*, 558–567.

Zhang, C.; Wan, H.; Liu, S.; Shen, X.; and Wu, Z. 2021. Pvt: Point-voxel transformer for 3d deep learning. arXiv 2021. *arXiv preprint arXiv:2108.06076*.

Zhang, J.; Chen, L.; Ouyang, B.; Liu, B.; Zhu, J.; Chen, Y.; Meng, Y.; and Wu, D. 2022a. Pointcutmix: Regularization strategy for point cloud classification. *Neurocomputing*, 505: 58–67.

Zhang, R.; Guo, Z.; Gao, P.; Fang, R.; Zhao, B.; Wang, D.; Qiao, Y.; and Li, H. 2022b. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. *Advances in neural information processing systems*, 35: 27061–27074.

Zhao, H.; Jiang, L.; Jia, J.; Torr, P. H.; and Koltun, V. 2021. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, 16259–16268.

Zhao, Z.; Wang, J.; Horn, M.; Ding, Y.; He, T.; Bai, Z.; Zietlow, D.; Simon-Gabriel, C.-J.; Shuai, B.; Tu, Z.; et al. 2023. Object-centric multiple object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16601–16611.