Point MixSwap: Attentional Point Cloud Mixing via Swapping Matched Structural Divisions

Ardian Umam[1*], Cheng-Kun Yang[2*], Yung-Yu Chuang[2], Jen-Hui Chuang[1], and Yen-Yu Lin[1,3]

[1]National Yang Ming Chiao Tung University, Taiwan
ardianumam.ee09@nycu.edu.tw, jchuang@cs.nctu.edu.tw, lin@cs.nycu.edu.tw
[2]National Taiwan University, Taiwan
d08922002@csie.ntu.edu.tw, cyy@csie.ntu.edu.tw
[3]Academia Sinica, Taiwan

Abstract. Data augmentation is developed for increasing the amount and diversity of training data to enhance model learning. Compared to 2D images, point clouds, with the 3D geometric nature as well as the high collection and annotation costs, pose great challenges and potentials for augmentation. This paper presents a 3D augmentation method that explores the structural variance across multiple point clouds, and generates more diverse point clouds to enrich the training set. Specifically, we propose an attention module that decomposes a point cloud into several disjoint point subsets, called divisions, in a way where each division has a corresponding division in another point cloud. The augmented point clouds are synthesized by swapping matched divisions. They exhibit high diversity since both intra- and inter-cloud variations are explored, hence useful for downstream tasks. The proposed method for augmentation can act as a module and be integrated into a point-based network. The resultant framework is end-to-end trainable. The experiments show that it achieves state-of-the-art performance on the ModelNet40 and ModelNet10 benchmarks. The code for this work is publicly available.[1]

1 Introduction

Recent advance in deep neural networks (DNN) has been made for 3D point cloud analysis ranging from classification [15,16,27], segmentation [25,33] to detection [14,17]. However, the issue of data hungry in DNN becomes even worse for point clouds due to the high collection and annotation costs [28,29]. Existing point cloud datasets are typically limited in both object quantity and category diversity. For example, ModelNet40 [26] (12K objects of 40 categories) and ScanObjectNN [21] (15K objects of 15 categories), two benchmarks for point cloud classification, are much smaller than image classification benchmarks, such as the ImageNet [10] dataset (1.4M images of 1K categories). Limited training data often make 3D point cloud networks suffer from overfitting and poor generalization to unseen data.

[*] The authors have equal contribution to this work
[1] The source code is available at: https://github.com/ardianumam/PointMixSwap

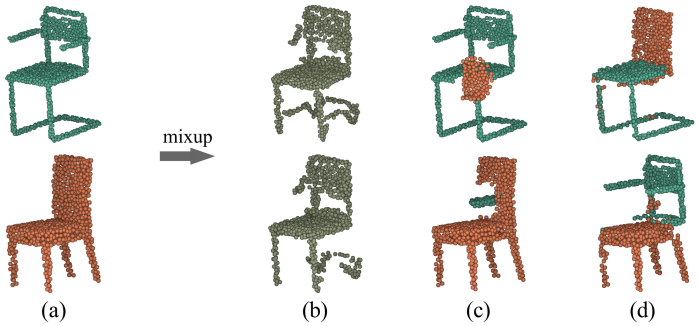

Fig. 1: Given (a) two source point clouds, the augmented samples are synthesized by (b) PointMixup [3], (c) RSMix [11], and (d) our method. The augmented point clouds by our method are diverse in the sense that the structural variance across different point clouds is utilized for synthesis, which is achieved by swapping the matched structural divisions. Colors show the identities of source points. Note that points generated by PointMixup do not have corresponding points in the source clouds, thereby drawn in another color.

Data augmentation aims to increase the size and diversity of training data and can alleviate the unfavorable effects caused by the lack of annotated data. Compared to 2D images, 3D point clouds with rich geometric shapes and deformations offer great potentials for developing structural data augmentation techniques, which have been relatively underutilized. As an effective data augmentation technique, *mixup* [32] has made significant progress on 2D image augmentation. It targets at expanding the data distribution based on the assumption that a linearly interpolated data sample also leads to linearly interpolated label. However, the literature about point cloud mixup is rare. The permutation invariant property of 3D point clouds results in no point-to-point correspondences across clouds. It follows that linear interpolation commonly used in mixup method is not applicable.

To address this issue, PointMixup [3] computes the shortest paths to match points across clouds, and then applies linear interpolation to the coordinates of matched points. Meanwhile, RSMix [11] mixes two point cloud samples by replacing a specific part of one sample with a shape-preserved part from another sample. The synthesized point clouds by PointMixup suffer from the geometric shape distortion problem, while RSMix generates discontinuous and less realistic samples, especially in the areas with points from different clouds. We observe that point clouds of the same class are usually composed of matchable components across different clouds. A chair, for example, is composed of legs, a cushion, and a back. These matched components in different clouds exhibit structural variability, which can be used to generate more diverse and realistic mixup samples, an aspect which is not explored in PointMixup and RSMix.

To this end, we present *Point MixSwap* that considers intra-class mixup and can synthesize diverse point clouds by swapping similar parts of source

point clouds. Take the chair category in Figure 1 as an example. Despite the rich variations in style and shape, most chairs can be decomposed into several matched and semantically meaningful parts, such as chair leg, cushion, and back. Synthesizing new point clouds by swapping the matched parts alleviates the issue of geometric distortion while making these new clouds more diverse, as shown in Figure 1.

Specifically, our goal is to divide a point cloud, a set of points, into a few disjoint and meaningful subsets, called *divisions* in this work, in a way where each division has the corresponding division in another point cloud of the same class. To this end, we introduce an encoder-decoder module. The encoder is applied to each cloud with its points as tokens, and captures both short- and long-range dependency. Inspired by [1], the decoder takes as input both *division queries* and the point-specific outputs of the encoder. Suppose the predefined number of divisions is R. There will be R division queries, one for each division. Via proper designs, the R division queries in the decoder can divide each point cloud into R divisions, with each covering similar points that are attended by the same division query. In addition, divisions which are from different point clouds but are associated with the same query are considered matched. In this way, not only intra-cloud division decomposition but also inter-cloud division variance are utilized for mixup.

This work makes the following contributions. First, we introduce an effective technique that explores structural variance for point clouds of the same class for synthesizing diverse point clouds by swapping matched divisions. Second, a novel encoder-decoder architecture is introduced to decompose a point cloud into semantically meaningful divisions with cross-cloud correspondences. Third, the synthesized point clouds lead to significant improvement for classification, reaching the state-of-the-art performance, and shape retrieval.

2 Related Work

Data augmentation on 2D and 3D data. Various methods have been proposed for data augmentation on 2D images, ranging from conventional approaches, such as random crop and color jittering [10,18,20] to advanced ones, such as AutoAugment [5,6] and generative adversarial networks (GAN) based methods [19,35,36]. In contrast, literature on 3D point cloud augmentation is relatively scarce [4,9,12]. Li *et al.*[12] propose the augmentor network to derive a rotation matrix and a point-wise translation to transform the point clouds in the batch. Choi *et al.* [4] come up with part-aware data augmentation for point cloud object detection. Given 3D object bounding boxes, they set the number of divisions and apply separate operations, such as random drop and random jittering. As their divisions are predefined, such an approach cannot ensure a consistent division meaning and its correspondence across point clouds within a class, which is a key factor in motivating our MixSwap.

Data augmentation via mixup. Existing methods [8,24,30,31,32] make significant progress on mixup for generating 2D images. For example, Kim *et al.* [8]

consider saliency maps in the process of mixup, ensuring augmented data with sufficient information. Yun *et al*. [31] perform random cut in an image and replace the cut region with a patch from another image. Nonetheless, these methods are designed for 2D images and are inapplicable to data in geometric domains, including point clouds.

PointMixup [3] generalizes the idea of mixup to 3D data by seeking the optimal *interpolant* defined by the shortest path interpolation. Nonetheless, the interpolants, being locally generated virtual samples, suffer from the structural or shape distortion. Although this issue has been partly addressed in RSMix [11], where a subset of a point cloud is replaced by a subset of another cloud, the resultant augmented clouds preserve geometric structure within individual subsets, but with less realistic global appearance, as shown in Figure 1(c), especially in the boundary of different subsets. In addition, none of these two methods have explored the structural variance within the point clouds of the same class. On the other hand, our method focuses on synthesizing point clouds by developing an encoder-decoder module, which carries out intra-cloud division decomposition and leverages inter-cloud division variance to enrich mixup samples.

Point cloud structure division. Parsing point clouds into semantic parts reveals crucial information for point cloud analysis. Chen *et al*. [2] encode the shape structure intrinsically for 3D points in an unsupervised manner. Zhu *et al*. [34] develop an adaptive learning module for shape co-segmentation using the group consistency loss and an additional shape part dataset. However, these methods usually derive one model only for each category. In contrast, our method can decompose a pair of point clouds of the same class into geometrically consistent and matched divisions. Furthermore, our method does not require part-wise annotations and, more importantly, is applicable to point clouds of an arbitrary class by using a single model. In light of the differences between shape co-segmentation and our technique, our aim is to create consistent divisions within samples in order to improve augmentation. As such, perfect decomposition is not a requirement for the proposed method.

3 Proposed Method

3.1 Overview

We are given a training set of point clouds of C categories, $\mathcal{D} = \{(P_n, \mathbf{y}_n)\}$. Without loss of generality, we assume that the number of points in each cloud is M, *i.e.*, $P_n = \{\mathbf{p}_n^m\}_{m=1}^M$, where point $\mathbf{p}_n^m \in \mathbb{R}^3$ is represented by its 3D coordinate and $\mathbf{y}_n \in \{0,1\}^C$ is a C-dimensional binary vector indicating the category of P_n. The downstream task in this work is to train a function that is capable of mapping a point cloud to its class label, *i.e.*, point cloud classification.

This work proposes a data augmentation that utilizes structural variance posed within point clouds of the same class to synthesize training data by decomposing each point cloud into structural divisions and enriching the source dataset via cross-cloud combinations of these structural divisions. The proposed

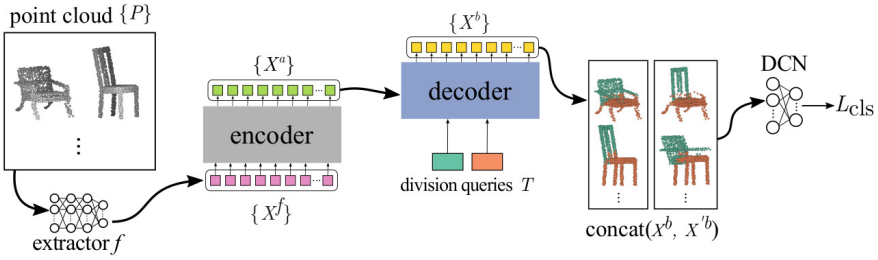

Fig. 2: **Overview of the proposed network architecture.** Point MixSwap leverages the attention mechanism to identify structural divisions for point clouds of the same category via an encoder-decoder module. This module is integrated into a point cloud framework. It receives the per-point features compiled by a feature extraction backbone. It generates augmented point clouds, which serve as the input to the downstream classifier network (DCN). The whole network can be end-to-end trained via the objective function of DCN.

method is depicted in Figure 2, which considers *intra-class mixup*. In mini-batch optimization, each batch consists of N point clouds of an arbitrary class, *i.e.*, $\{P_n\}_{n=1}^N$. After applying a feature extractor f, the features $\{X_n^f\}_{n=1}^N$ are obtained, where $X_n^f = \{\mathbf{x}_n^{m,f}\}_{m=1}^M$, with $\mathbf{x}_n^{m,f} \in \mathbb{R}^E$ represents the per-point feature vector of embedding size E.

An encoder-decoder architecture is introduced to discover the divisions that are geometrical parts shared across point clouds. The per-point features of point cloud P, *i.e.*, X^f, are fed into the encoder to produce its self-attention features X^a. Let R denote the predefined number of divisions that R *division queries*, $T \in \mathbb{R}^{R\times E}$, are created. The decoder takes as input both division queries T and the output of the encoder for the point cloud X^a, before the division-point attention features X^b is generated for the downstream classification task. In the process, the R division queries jointly decompose point cloud P_n into R disjoint subsets. After division swapping, the augmented point clouds are generated to facilitate the downstream classifier network.

Figure 2 shows an example with $R = 2$, where coloring is used to illustrate the mapping between divisions and points. The details about the encoder-decoder module and division swapping are provided in the following.

3.2 Encoder-decoder architecture

The encoder-decoder architecture, *i.e.*, transformer [23], can offer an effective way to capture the correlation across samples. The encoder is composed of several self-attention layers to capture long-range dependency and improve the features given the extractor f. As for the decoder, we are inspired by DETR [1], where the learned positional embedding can be utilized as anchor boxes for object detection, and extend the idea to 3D point clouds to capture the similar

geometrical parts shared across point clouds. Specifically, we create R *division queries* initialized using the Xavier method [7], where R is the number of divisions for decomposition. The decoder takes the self-attention features X^a from encoder and division queries as input, and produces the division-point attention features X^b. After optimization, these R division queries represent the R divisions shared across point clouds.

In the original decoder layer [23], the number of output feature vectors is the same as the number of the queries. Since we pass the division queries $T \in \mathbb{R}^{R\times E}$ as tokens into the decoder, the generated feature vectors are query-specific, instead of point-specific, which is ineffectual for classification. Hence, we introduce a designed decoder, which is composed of two coupled cross-attention layers, as illustrated in Figure 3. In the first layer, the division queries T serve as the queries while the point-specific features X^a act as the key-value pairs. Their roles are switched in the second layer and jointly produce point-specific features X^b, where $X^b \in \mathbb{R}^{M\times E}$.

Following by the common practice in [23], the intermediate attention features X^o in Figure 3 are computed as follows

$$Attention(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{E}})V = SV = X^o, \tag{1}$$

where $Q = TQ_w$, $K = X^aK_w$, and $V = X^aV_w$, while Q_w, K_w, and V_w are three matrices for linear projection. The softmax operation is applied along the last dimension. The generated features X^o softly attend to all points and are then passed into the second attention layer.

In Eq. 1, the point-division attention matrix $S \in \mathbb{R}^{R\times M}$, which we call it the *division map*, is obtained and will be utilized for point cloud decomposition. In contrast to the first cross-attention layer, we use self-attention features X^a as queries and X^o from the first layer as key-value pairs. Through the similar cross-attention operation used in the first layer, we obtain the features X^b, which encode the correlation between the division queries T and the per-point features X^a, with residual learning adopted in the two coupled cross-attention layers.

The division-point attention features X^b can be fed into the downstream classifier network (DCN) for training. The whole network is end-to-end trainable in accordance with the task of classification. In this way, these division queries and the division map S are learned to minimize the cross-entropy loss,

$$L_{cls} = -\sum_{n=1}^{N}\sum_{c=1}^{C}\mathbf{y}_{n,c}\log(\hat{\mathbf{y}}_{n,c}), \tag{2}$$

where $\mathbf{y}_n$ is one-hot encoded label vector of point cloud P_n, $\hat{\mathbf{y}}_n$ is predicted probability distribution, and C is the number of classes in the training set.

The structural division can be inferred from division map S in Eq. 1. Specifically, the m-th point is assigned to division $d(m)$ if

$$d(m) = \underset{r}{\text{argmax}}(S(r, m)). \tag{3}$$

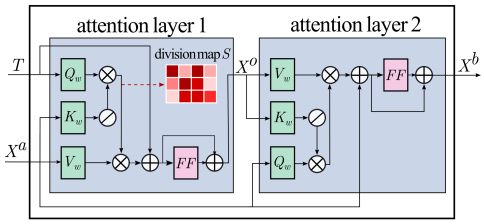

Fig. 3: Architecture of the coupled decoder layers in Point MixSwap. Symbols $\otimes$, $\oplus$, and $\oslash$ denote matrix multiplication, element-wise sum, and matrix transposition, respectively. The pink boxes (FF) represent multilayer perceptron.

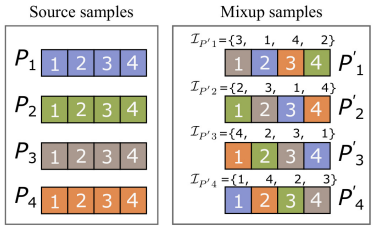

Fig. 4: Division mixswap is depicted to synthesize R new mixup point clouds with complete, non-repeating divisions from R source point clouds. An example of $R = 4$ is given in the figure.

In this way, the division map S can be considered as the division segmentation map and is used to retrieve the R structural divisions for each point cloud. Furthermore, since the division queries are shared for all samples, every r-th division query attends similar subsets of points across point clouds. As a result, a division in a point cloud has its corresponding division in each of other point clouds.

3.3 Division mixswap

New mixup point clouds are synthesized by swapping matched divisions, and each of them contains non-repeating divisions. Specifically, to synthesize R new point clouds $\{P'_r\}_{r=1}^{R}$, we randomly pick R source point clouds in a batch, $\{P_r\}_{r=1}^{R}$. Figure 4 illustrates how our mixswap synthesizes R new point clouds from R source point clouds with $R = 4$. The first mixing index array $\mathcal{I}_{P'1}$ is defined as a random permutation vector of integer numbers ranging from 1 to R, and the following mixing index arrays are specified as one time cyclic rotation of their previous one. Using these mixing index arrays, new mixup point clouds are then synthesized, where the r-th element in the mixing index array gives the source of the r-th division. Since the divison S may decompose point clouds into division of diverse sizes, therefore, we further sample each of the synthesized point clouds into a fixed number of points.

$$P^{'} = \Gamma^{M}\left(\text{concat}(\{\mathbb{S}_r \odot P_{\mathcal{I}(r)}\}_{r=1}^{R})\right). \tag{4}$$

where $\mathbb{S}_r$ denotes a binary mask used to select all points of the r-th division, Γ^M denotes a sampling operation which returns M points, and $\odot$ represents element-wise multiplication. Mixup point clouds acquired via Eq. 4 keep the orderless property, thus consistent with point cloud data.

Mixup operation can be carried out in both the input level, *i.e.*, performed among point cloud samples P_n, and in the feature level, *i.e.*, performed among

point features X^b. The former requires aligned training sets whose point clouds have the same pose in order to achieve its optimal improvement gain. Meanwhile, considering point features are computed from the original point clouds, the latter is more robust to unaligned training sets, which can be performed by replacing P with X^b in Eq. 4. In the following, we discuss the cases where our method is applied to point clouds with unaligned poses.

3.4 Alignment mechanism

As shown in Figure 2, our method adopts an existing point cloud feature extractor. Most extractors such as [15,25] are designed to work with unaligned point clouds and can implicitly address pose variations with some specified mechanism, such as T-Net in PointNet [15]. As a result, the resultant features are somewhat robust to variations. To further improve the performance on unaligned cases, we present a mechanism, called principal axis alignment (PAA), to pre-process the given point clouds. We compute the largest principal axes of each point cloud. In a batch of point clouds, one is randomly chosen as the reference, while the rest are aligned to the reference according to the principal axes.

3.5 Implementation details

We train the network with 500 epochs, where the first warm-up 20 epochs are run without executing Point MixSwap, to stabilize the learning of division queries. For DGCNN [25], the SGD solver is adopted with a momentum of 0.9 and a learning rate of 0.001 scheduled using the cosine annealing strategy [13]. For PointNet [15], the Adam optimizer is employed with an initial learning rate of 0.001 and is gradually reduced with a decay rate of 0.5 every 20 epochs. Unless further specified, we set the number of divisions to three, $R = 3$, and use feature-level augmentation in the experiments.

Limitations. The proposed Point MixSwap works for point clouds of the same category. It is not applicable to point clouds of different categories.

4 Experimental Results

4.1 Datasets

We evaluate the proposed Point MixSwap on the ModelNet40 (**M40**) [26], ModelNet10 (**M10**) [26], and ScanObjectNN (**SON**) [22] datasets, which are widely used for point cloud recognition. The OBJ_ONLY version is adopted for SON. M40 and M10 are synthetic datasets, while SON is a real-world dataset. Following previous works [15,16,25], we uniformly sample 1,024 points on the mesh faces according to the face areas and then normalize them into a unit sphere. We discard the normals of these samples and only use their 3D point coordinates.

We evaluate the proposed method on the reduced datasets, to investigate the effectiveness of our method when less training data are available. The dataset size is reduced to 20% and 50% with stratified sampling.

Table 1: Accuracy scores of the proposed Point MixSwap on 20%, 50%, and 100% of the ModelNet40 (M40) and ModelNet10 (M10) datasets.

Method	Rate 20%		Rate 50%		Rate 100%	
	M40	M10	M40	M10	M40	M10
PointNet	82.1	89.4	85.9	92.7	88.6	93.2
PointNet + Ours	**86.3** (4.2↑)	**91.3** (1.9↑)	**88.7** (2.8↑)	**93.6** (0.9↑)	**90.2** (1.6↑)	**93.9** (0.7↑)
DGCNN	87.5	93.2	91.5	94.3	92.7	94.8
DGCNN + Ours	**91.3** (3.8↑)	**94.6** (1.4↑)	**92.8** (1.3↑)	**94.9** (0.6↑)	**93.5** (0.8↑)	**96.0** (1.2↑)

Table 2: Accuracy scores of the proposed Point MixSwap on 20%, 50%, and 100% of the rotated ModelNet40 (RM40) and ScanObjectNN (SON) datasets.

Method	Rate 20%		Rate 50%		Rate 100%	
	RM40	SON	RM40	SON	RM40	SON
PointNet	82.0	62.5	85.5	71.3	88.5	76.2
PointNet + Ours	85.2 (3.2↑)	66.1 (3.6↑)	87.7 (2.2↑)	74.0 (2.7↑)	89.5 (1.0↑)	78.8 (2.6↑)
PointNet + Ours + PAA	**86.2** (4.2↑)	**67.0** (4.5↑)	**87.9** (2.4↑)	**74.3** (3.0↑)	**89.7** (1.2↑)	**78.9** (2.7↑)
DGCNN	87.0	73.7	90.3	81.6	91.5	86.2
DGCNN + Ours	89.3 (2.3↑)	76.3 (2.6↑)	91.1 (0.8↑)	84.1 (2.5↑)	**92.3** (0.8↑)	88.6 (2.4↑)
DGCNN + Ours + PAA	**90.1** (3.1↑)	**76.8** (3.1↑)	**91.3** (1.0↑)	**84.8** (3.2↑)	**92.3** (0.8↑)	**89.0** (2.8↑)

4.2 Shape classification

To evaluate our method, we consider PointNet [15] and DGCNN [25] as the backbones for feature extraction, and report the performance of the models trained with (ours) and without (baseline) the proposed Point MixSwap. We first evaluate the proposed method on M40 and M10, where most of the samples are well aligned. As demonstrated in Table 1, Point MixSwap consistently boosts the accuracy regardless of the backbone networks and training data sizes. With only 50% of the training set, it achieves slightly better performance than the baseline model trained on the full dataset, in all backbones and datasets. While at the reduction rate of 20%, the accuracy is also comparable to the baseline with 50% of training data. The results reveal the effectiveness of Point MixSwap to work with different point cloud classification network architectures. More experiments on different backbones can be found in the supplementary material.

To further demonstrate the generality of the proposed method to unaligned and real-world datasets, we evaluate the proposed method on the rotated ModelNet40 (RM40), where random rotation is applied to each point cloud of the training and testing sets, and the unaligned dataset, SON. The proposed method is evaluated with and without using the proposed principal axis alignment (PAA). Table 2 summarizes the results. In all settings, the proposed method without alignment produces notable improvement compared to the baseline although the source samples are unaligned. This is because the adopted backbones are developed to address pose variations. Also, each derived division query attends to point tokens described by per-point local features, and tolerates a certain de-

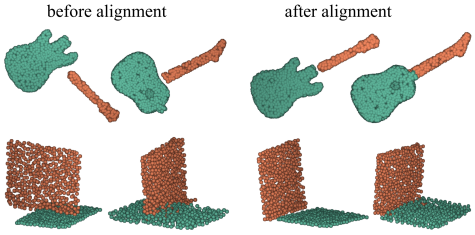

Fig. 5: Mixup samples generated using Point Mixup before and after applying alignment.

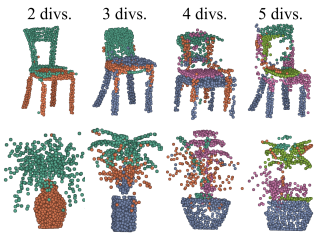

Fig. 6: Mixswap samples with different numbers of divisions.

gree of unalignment. With the alignment mechanism PAA, the proposed method yields further and consistent improvement in all settings. Visualization of some mixup samples before and after applying alignment is given in Figure 5.

4.3 Competing methods and comparisons

We compare the proposed method with the state-of-the-art point cloud augmentation methods on the reduced and full training datasets. The competing methods include PointMixup [3], PointAugment [12], RSMix [11] and PointWOLF [9]. For the accuracy scores already reported in the original papers, we take the numbers directly from the papers. For those that are not given in the papers, particularly for those by PointAugment and RSMix on the reduced training sets, we run their official released codes for obtaining the accuracy scores. In addition, we note that PointAugment's performance is unstable on ModelNet10; Thus, we run the official codes several times and report the average accuracy scores instead of the ones from the paper.

Table 3(a) reports the accuracy scores of all compared methods on both M40 and M10, with 20% of the dataset and the full dataset. The proposed Point MixSwap outperforms the state-of-the-art methods in all settings. In addition, our method shows a good performance gain when the training data is insufficient, 20% of the dataset in this case. It shows that the proposed method is effective for data augmentation.

4.4 Ablation study and analysis

We perform ablation studies to evaluate the impacts of the proposed components and present performance analysis. Here, the experiments are conducted on 20% of the training sets.

Contributions of components. To evaluate the effectiveness of the proposed method, we first report the performance of the baseline by training without using any data augmentations. Here, DGCNN is adopted as the baseline. Then we evaluate the contribution of the conventional data augmentation (CDA) and the proposed Point MixSwap. The adopted CDA comprises random scaling, random

Table 3: (a) Comparisons with existing methods on 20% and 100% of M40 and M10. (b) Accuracy scores of Point MixSwap with different numbers of divisions and in different mixup levels on 20% of three datasets.

Method	Rate 20%		Rate 100%	
	M40	M10	M40	M10
DGCNN	87.5	93.2	92.6	94.8
DGCNN + PointMixup [3]	89.0	93.8	93.1	95.1
DGCNN + PointAugment [12]	88.6	92.8	93.4	95.2
DGCNN + RSMix [11]	90.1	93.7	**93.5**	95.9
DGCNN + PointWOLF [9]	89.3	93.5	93.2	95.1
DGCNN + Ours	**91.3**	**94.6**	**93.5**	**96.0**

(a)

Divisions	Level	M40	M10	SON
2	Input	91.0	94.6	75.9
	Feature	91.1	**94.7**	76.2
3	Input	91.2	94.5	76.1
	Feature	**91.3**	94.6	**76.3**
4	Input	91.0	94.4	75.7
	Feature	91.2	94.6	76.1
5	Input	91.0	94.3	75.5
	Feature	91.2	94.6	76.0

(b)

Table 4: Accuracy scores by using the baselines, different variants of our method, and three trivial division methods.

CDA	Point MixSwap			Trivial division			Accuracy		
	Enc-dec	Input-level	Feature-level	Hor.	Ver.	Random	M40	M10	SON
							87.5	93.2	73.0
✓							88.7 (1.2 ↑)	93.5 (0.3)	73.7 (0.7)
✓	✓						88.8 (1.3 ↑)	93.6 (0.4 ↑)	73.6 (0.6 ↑)
	✓	✓					89.5 (2.0 ↑)	94.0 (0.8 ↑)	74.7 (1.7 ↑)
	✓		✓				89.7 (2.2 ↑)	94.2 (1.0 ↑)	75.0 (2.0 ↑)
	✓	✓	✓				89.5 (2.0 ↑)	94.1 (0.9 ↑)	74.9 (1.9 ↑)
✓	✓	✓					91.1 (3.6 ↑)	94.5 (1.3 ↑)	76.1 (3.1 ↑)
✓	✓		✓				**91.3** (3.8 ↑)	**94.6** (1.4 ↑)	**76.3** (3.3 ↑)
✓	✓	✓	✓				91.2 (3.7 ↑)	94.5 (1.3 ↑)	76.1 (3.1 ↑)
✓				✓			89.2 (1.7 ↑)	93.7 (0.5 ↑)	73.9 (0.9 ↑)
✓					✓		89.0 (1.5 ↑)	93.6 (0.4 ↑)	73.8 (0.8 ↑)
✓						✓	88.9 (1.4 ↑)	93.4 (0.2 ↑)	73.8 (0.8 ↑)

translation and random drop, following [11]. Moreover, we perform the mixup operation at the input level, the feature level, or both, to see the performance with different component combinations. In addition, to check if the accuracy improvement comes from Point MixSwap rather than trivial data decomposition and reconstruction, three simple division approaches are investigated. In the first two approaches, we uniformly divide a point cloud horizontally and vertically, respectively. The third approach uses random division.

Table 4 reports the results of the ablation studies. First, we compare the performance of data augmentation by using CDA, input-level and feature-level Point MixSwap. Both input-level and feature-level Point MixSwap achieve notably higher accuracy than CDA. To further investigate the source of the performance gain, we combine Point MixSwap with CDA, but neither input-level nor feature-level mixup is enabled. In this case, the difference from the CDA-only configuration lies in the attention mechanism enabled by the encoder-decoder

blocks (Enc-dec) to process the features. Table 4 shows that, on M40, the Point MixSwap+CDA configuration (88.8) yields slightly better performance than the CDA configuration (88.7). Similar trend is also encountered on the M10 and SON datasets. It indicates that the major source of performance gain is not the attention mechanism, but the effective divisions derived by the proposed encoder-decoder block for mixup augmentation.

Second, we investigate the impacts of performing mixup at the input level and the feature level. In Table 4, the feature-level mixup achieves relatively higher accuracy than the input-level mixup, *i.e.*, 89.7 versus 89.5 on M40, 94.2 versus 94.0 on M10, and 75.0 versus 74.7 on SON. Combining both of them yields slightly lower accuracy than using the feature-level mixup alone. Third, we combine Point MixSwap with CDA. The result demonstrates that Point MixSwap can be complementary to other augmentation methods, and can work together with other types of data augmentation for further performance enhancement. Specifically, the combination of feature-level Point MixSwap and CDA achieves significant performance gains compared to the baseline.

Finally, we consider the performance by using the three trivial division approaches. In Table 4, notably inferior improvements are obtained by using the three division approaches compared to the proposed Point MixSwap. The results indicate that accuracy improvements are not due to trivial data decomposition and reconstruction for augmentation, but rather to the effective divisions derived by using the proposed method.

Analysis on the number of divisions. We analyze the impact of division numbers on Point MixSwap. Table 3(b) reports the results by setting the division number to 2, 3, 4, and 5, respectively. For each number, we measure the accuracy with mixup at the input level and feature level. It can be observed that feature-level mixup yields better performance for all division numbers. This is reasonably well grounded because point features are computed in the original samples, *i.e.*, before mixup is performed. Meanwhile, for the input-level mixup, the performance could degrade with a higher division number. Figure 6 visualizes some mixswap results with different numbers of divisions.

4.5 Qualitative results of mixup samples

Figure 7 shows the synthesized examples via the proposed Point MixSwap. We set the number of division queries to 2 and 3, to generate new samples shown in Figure 7(a) and Figure 7(b), respectively. For each setting, the second column depicts the generated mixswap samples from the source sample pair/triplet given in the first column. Note that for the guitar and bed categories, different poses of source samples are provided, and the generated mixup samples after applying alignment mechanism are shown, in which the first source sample is set as the reference. The generated mixup samples before alignment mechanism is applied, can be found in the supplementary material.

In Figure 7(a), Point MixSwap successfully identifies chair leg and back as the two major structural divisions in the chair category, and poses a consistent

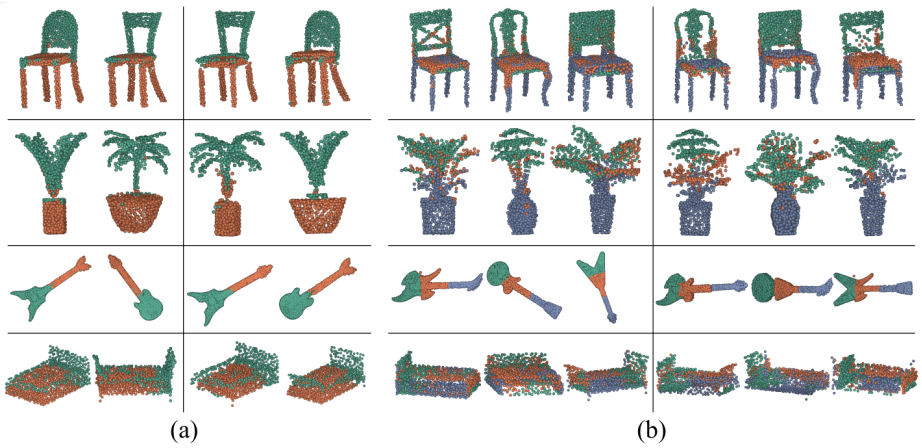

Fig. 7: Mixup examples generated by Point MixSwap. (a) Four examples with two divisions. For a pair of source point clouds on the left, we show the generated mixup samples. (b) Four examples with three divisions. For an input triplet on the left, the three generated mixup samples are displayed. Points are colored according to their divisions.

correspondence across samples, where points of the same division are colored with the same color. For the plant, guitar and bed categories, the two major structural divisions are also consistently identified across samples. Hence, the generated mixup samples accomplished via Point MixSwap show not only diverse geometric shapes but also structure-preserved characteristics.

When the number of divisions is set to 3, the chair is segmented by its leg, cushion and back, as the three structural divisions. As for the plant, guitar and bed categories, they have meaningful divisions as shown in the figure. A higher division number enables the mixup process to possibly generate more diverse samples in the sense that each new sample can be synthesized with more structural divisions from more different samples in the input set. We further discuss the case where the given division number exceeds the number of structural divisions posed by certain categories. Take the plant category as an example. According to the geometrical structure, each sample naturally poses two structural divisions, the pot and plant. Given three division queries, two of these tokens attend to similar structural divisions, the plant part in this case, as depicted in Figure 7(b) with green and brown colors. Nonetheless, our method can still generate diverse and structure-preserved mixup samples by utilizing the division cross-correspondence. More visualization examples of other categories can be found in the supplementary material.

Table 5: Shape retrieval performance in mAP (%) of different data augmentation methods on the M40 dataset.

Backbone	CDA	PointAugment	Ours
PointNet	70.5	75.8 (5.3 ↑)	**78.4** (7.9 ↑)
DGCNN	85.3	89.0 (3.7 ↑)	**90.6** (5.3 ↑)

4.6 Shape retrieval

To demonstrate the advantage of the proposed method to another downstream task, following PointAugment [12], we also examine the proposed method for shape retrieval which retrieves the most similar shape based on cosine similarity of the global features. We regard every sample in the testing set as a query shape, and the retrieval performance in mean average precision (mAP) is reported on the M40 dataset, as shown in Table 5. The proposed method produces significant improvement margin compared to CDA in both PointNet and DGCNN, while a notable margin is also observed compared to PointAugment.

5 Conclusion

This paper proposes Point MixSwap, a novel data augmentation technique for 3D point clouds. It is developed to exploit structural variations among point clouds of the same class to synthesize diverse and structure-preserved augmented samples. Point MixSwap introduces an intuitive idea of data augmentation by decomposing a point cloud into several disjoint divisions. Each division has a consistently corresponding division in other point clouds. Thus, augmented mixup data can be synthesized by swapping one or more matched divisions among the source point clouds. As a mixup augmentation technique, Point MixSwap is guided by an attention mechanism, and to the best of our knowledge, it is the first augmentation technique that utilizes an attention mechanism to explore matchable divisions across source data. Point MixSwap is end-to-end trainable and can be employed by any point-based networks. Comprehensive experiments demonstrate the effectiveness of Point MixSwap on boosting the model accuracy, especially when only limited data are available.

Acknowledgments. This work was supported in part by the Ministry of Science and Technology (MOST) under grants 109-2221-E-009-113-MY3, 111-2628-E-A49-025-MY3, 111-2634-F-007-002, 110-2634-F-002-050, 110-2634-F-002-051, 110-2634-F-006-022 and 110-2634-F-A49-006. This work was funded in part by Qualcomm through a Taiwan University Research Collaboration Project and by MediaTek. We thank the National Center for High-performance Computing (NCHC) of National Applied Research Laboratories (NARLabs) in Taiwan for providing computational and storage resources.

References

1. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020)
2. Chen, N., Liu, L., Cui, Z., Chen, R., Ceylan, D., Tu, C., Wang, W.: Unsupervised learning of intrinsic structural representation points. In: CVPR (2020)
3. Chen, Y., Hu, V.T., Gavves, E., Mensink, T., Mettes, P., Yang, P., Snoek, C.G.: Pointmixup: Augmentation for point clouds. In: ECCV. pp. 330–345 (2020)
4. Choi, J., Song, Y., Kwak, N.: Part-aware data augmentation for 3d object detection in point cloud (2021)
5. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: CVPR (2019)
6. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.: Randaugment: Practical automated data augmentation with a reduced search space. In: NIPS
7. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. pp. 249–256. JMLR Workshop and Conference Proceedings (2010)
8. Kim, J.H., Choo, W., Song, H.O.: Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In: ICLR (2020)
9. Kim, S., Lee, S., Hwang, D., Lee, J., Hwang, S.J., Kim, H.J.: Point cloud augmentation with weighted local transformations. In: ICCV (2021)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks
11. Lee, D., Lee, J., Lee, J., Lee, H., Lee, M., Woo, S., Lee, S.: Regularization strategy for point cloud via rigidly mixed sample. In: CVPR (2021)
12. Li, R., Li, X., Heng, P.A., Fu, C.W.: Pointaugment: an auto-augmentation framework for point cloud classification. In: CVPR (2020)
13. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)
14. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. In: ICCV (2019)
15. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: CVPR (2017)
16. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. NIPS (2017)
17. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H.: Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: CVPR (2020)
18. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR (2014)
19. Sixt, L., Wild, B., Landgraf, T.: Rendergan: Generating realistic labeled data. Frontiers in Robotics and AI (2018)
20. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: ICCV (2015)
21. Uy, M.A., Pham, Q.H., Hua, B.S., Nguyen, D.T., Yeung, S.K.: Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In: ICCV (2019)
22. Uy, M.A., Pham, Q.H., Hua, B.S., Nguyen, T., Yeung, S.K.: 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. pp. 1588–1597 (2019)

23. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NIPS (2017)
24. Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., Bengio, Y.: Manifold mixup: Better representations by interpolating hidden states. In: ICML (2019)
25. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. TOG (2019)
26. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: CVPR (2015)
27. Xiang, T., Zhang, C., Song, Y., Yu, J., Cai, W.: Walk in the cloud: Learning curves for point clouds shape analysis. In: ICCV (2021)
28. Yang, C.K., Chuang, Y.Y., Lin, Y.Y.: Unsupervised point cloud object co-segmentation by co-contrastive learning and mutual attention sampling. In: ICCV (2021)
29. Yang, C.K., Wu, J.J., Chen, K.S., Chuang, Y.Y., Lin, Y.Y.: An mil-derived transformer for weakly supervised point cloud segmentation. In: CVPR (2022)
30. Yoo, J., Ahn, N., Sohn, K.A.: Rethinking data augmentation for image super-resolution: A comprehensive analysis and a new strategy. In: CVPR (2020)
31. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: ICCV (2019)
32. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. ICLR (2018)
33. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: ICCV (2021)
34. Zhu, C., Xu, K., Chaudhuri, S., Yi, L., Guibas, L.J., Zhang, H.: Adacoseg: Adaptive shape co-segmentation with group consistency loss. In: CVPR (2020)
35. Zhu, X., Liu, Y., Li, J., Wan, T., Qin, Z.: Emotion classification with data augmentation using generative adversarial networks. In: KDD (2018)
36. Zhu, Y., Aoun, M., Krijn, M., Vanschoren, J., Campus, H.T.: Data augmentation using conditional generative adversarial networks for leaf counting in arabidopsis plants. In: BMVC (2018)