ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10095576

PCSALMIX: GRADIENT SALIENCY-BASED MIX AUGMENTATION FOR POINT CLOUD CLASSIFICATION

Tao Hong * *Zeren Zhang* * *Jinwen Ma* ✉

School of Mathematical Sciences, Peking University, Beijing 100871, China

ABSTRACT

Point cloud classification has sparked many researchers' interest for its cornerstone role in 3D applications. Inheriting the CutMix series augmentation that performs well in 2D images, PointCutMix and RSMix are proposed to generate new samples for 3D point clouds, by replacing partial points of one cloud with those of another. However, the selection of mixed regions is all built on randomness, ignoring the significance of point clouds' saliency. To address this deficiency, we propose **PCSalMix**: a novel **Sal**iency-based **Mix** augmentation for **P**oint **C**loud classification. The gradient of classification network on inputs is a natural tool to locate the saliency. Based on this discovery, we extract points with larger gradient values to make more representative samples. Afterward, the soft labels are weighted more accurately by accumulated gradients rather than count ratios of points. The experimental results verify the outperformance of our method on ModelNet40 and ModelNet10 benchmarks in terms of accuracy and robustness against adversarial attacks.

***Index Terms*—** Point Cloud Classification, Mix Augmentation, Gradient Saliency

1. INTRODUCTION

Recently, the exploration of point clouds has drawn lots of attention due to their value in many applications such as autonomous driving. Compared with 2D images, 3D point clouds are more challenging especially in two aspects: (1) Unordered: a set of points without a specific order. (2) Invariant: learned representation should be invariant to certain transformations. Focusing on the foundational classification of point clouds, it has gone through a fast development: from classical PointNet [1] to PointNet++ [2] and DGCNN [3] *etc.*

Except for the evolution of network models, data augmentations are commonly adopted strategies to enhance the representation capability of models due to the scarcity of point cloud data. Among them, there are some conventional data augmentations (ConvDA) such as rotation, scaling and jittering [4]. Besides, a series of mix data augmentations (MixDA) is inherited from 2D images: PointMixup [5], PointCutMix [6] and RSMix [7]. As the names indicate, they are extended works of Mixup [8] and CutMix [9] in 2D images, which interpolate or splice between two images to generate a new mixed image. Considering the disorder of point clouds, the key is to first match two samples by subset replacement or assignment metrics such as Earth Mover's Distance (EMD).

Although effective, the above CutMix series all face the same problem: the mixed part is randomly selected so that some mixed samples may be invalid. For example, cutting a background region of one image to paste to another doesn't generate a meaningful image. Therefore, several saliency-based MixDAs are successively proposed such as PuzzleMix [10] and SaliencyMix [11], which try to select the salient regions to conduct mixDA. However, these works either require external tools to locate saliency or rely on additional policies to generate images. From 2D images to 3D point clouds, the large gap between their data forms determines that the non-trivial generalization of saliency-based MixDA is a challenge.

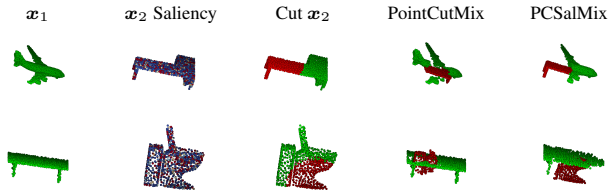

Fig. 1. Visual examples of mixed point clouds. Our PCSalMix generates more solid samples than PointCutMix. In subgraph of $\boldsymbol{x}_2$ *Saliency*, red points represent greater saliency.

Looking back at point clouds, several works [12, 13] are proposed to detect salient points, but it is still inconvenient to directly integrate with MixDA. From our point of view, the gradients of neural network models can properly locate the most distinguishable part of one sample. Recall the definition of gradient:

$$\nabla \boldsymbol{x} = \frac{\partial \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y})}{\partial \boldsymbol{x}} = \lim_{\Delta \boldsymbol{x} \to 0} \frac{\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \Delta \boldsymbol{x}), \boldsymbol{y}) - \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y})}{\Delta \boldsymbol{x}} \tag{1}$$

where ℓ, $\boldsymbol{x}$, $\boldsymbol{y}$, f and θ are the loss function, input sample, label, network and its parameters, respectively. $\nabla \boldsymbol{x}$ itself is essentially the sensitivity of loss function to the disturbance $\Delta \boldsymbol{x}$ on input sample. It is exactly tied to the salient regions we are searching for.

Inspired by the above discovery, for **P**oint **C**loud classification, we propose a simple but effective gradient **Sal**iency-based **Mix** strategy to fulfill data augmentation, named **PCSalMix**. Specifically, we only need the gradients of inputs to locate the salient points with larger gradient response values, then mix operation can be carried out naturally, generating more solid and sound samples as shown in Fig. 1. Except for inputs, the generation of soft labels is also a crucial term in MixDAs. As far as we know, all the existing MixDAs generate the soft label by calculating the count ratios of mixed points. Enlightened by the insight that different points in one point cloud account for different importance, we propose to generate the weight of mixed soft label by accumulating gradients of points, which coincides exactly with our definition of saliency and is shown more accurately than the weight calculated via the number ratio. The overview of PCSalMix is shown in Fig. 2.

PCSalMix only needs the gradients of inputs that are off-the-shelf during the gradient backpropagation of training networks. This

* Equal contribution.

Supported by the Natural Science Foundation of China: grant 62071171.

advantage determines its generality, such as being agnostic to different data forms. Compared with previous PointCutMix and RSMix, we only need to adjust the selection of cut point center from randomness to high gradient values, like a plug-and-play improvement. The superiority of our method is verified by the experiments on classification benchmarks: ModelNet40 and ModelNet10 [14]. PCSalMix outperforms current state-of-the-art MixDAs such as PointCutMix in accuracy, including robustness against adversarial attacks. Our main contributions are summarized as follows:

• We propose a systematic gradient saliency-based mix augmentation for point cloud classification, which is simple and general due to its perfect compatibility with networks.

• We propose a new gradient-based paradigm to generate soft labels, getting a more accurate judgement.

• With the exploration of saliency, the effectiveness of PCSalMix is fully embodied in the experimental outperformance.

2. RELATED WORK

Developments of Point Cloud Classification In the deep learning history of point cloud classification, PointNet [1] is a groundbreaking work that independently learns on each point and gathers the final features for a global representation, yet ignores local features. To this end, PointNet++ [2] introduces a hierarchical structure on a nested partitioning of the point set. Furthermore, RS-CNN [15] proposes an irregular Relation-Shape CNN while DGCNN [3] proposes a Dynamic Graph CNN containing EdgeConv to model point clouds.

Mix Series Data Augmentation Data augmentation is a widely adopted technology to enhance the generalization of neural networks. For 2D images, MixDAs refer to mixing different images in one batch and summing labels of original images according to their individual mixed proportions. We can divide MixDAs into two types: randomly mix series including Mixup [8], CutMix [9] and Manifold Mixup [16]; saliency-based mix series including PuzzleMix [10], SaliencyMix [11], *etc.* As the names indicate, PointMixup [5] and PointCutMix [6] extend Mixup and CutMix to 3D point clouds with some modifications. The key challenge is the disorder of point clouds, so EMD is utilized to assign two samples first. Besides, RSMix [7] proposes a Rigid Subset Mix: replacing part of a sample with a shape-preserved subset from another. On the other hand, PointAugment [17] formulates a learnable function with a shape-wise transformation and a point-wise displacement, to alternately optimize with the classifier. What's more, new augmentations of point clouds are successively proposed for semantic segmentation [18] and object detection [19].

Point Cloud Saliency Visual saliency describes the human attention distribution for a given scene. To detect saliency points of clouds, Ding *et al.* proposed an optimization framework to integrate both the local distinctness and the global rarity values to obtain final saliency [12]. Zheng *et al.* proposed saliency maps by assigning each point a score reflecting its contribution to the model-recognition loss [13]. Earlier work on point cloud saliency can be referred to [20, 21].

3. PROPOSED APPROACH

3.1. Preliminary

For point cloud classification, let $\boldsymbol{x} \in \mathcal{X}$ denote the training sample and $\boldsymbol{y} \in \mathcal{Y}$ denote the one-hot label, then their dimensions are $\boldsymbol{x} \in \mathbb{R}^{N\times C}$ and $\boldsymbol{y} \in \mathbb{R}^K$, where N is the point number of one point cloud, C is the number of channels (C will be set to 3 below, *i.e.*, three-dimensional coordinates), and K is the number of categories.

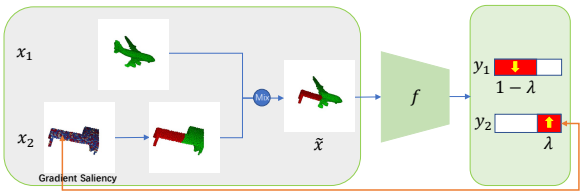

Fig. 2. The overview of our proposed approach PCSalMix. All the symbols are consistent with those described in Sec. 3.

That is to say, $\boldsymbol{x} = \{\boldsymbol{x}_i\}_{i=1}^N, \boldsymbol{x}_i \in \mathbb{R}^3$. Then classification network is to learn a mapping $f : \mathcal{X} \mapsto \mathcal{Y}$ with parameters $\boldsymbol{\theta}$.

Denote two point clouds as $\boldsymbol{x}_1 = \{\boldsymbol{x}_{1,i}\}_{i=1}^N, \boldsymbol{x}_2 = \{\boldsymbol{x}_{2,j}\}_{j=1}^N$ and corresponding labels as $\boldsymbol{y}_1, \boldsymbol{y}_2$. In analogy to the MixDA of images, we first need to match unordered points between two point clouds. The commonly used match principle EMD solves the assignment problem: $\phi^* = \arg\min_{\phi\in\Phi} \sum_{i=1}^N \left\|\boldsymbol{x}_{1,\phi(i)} - \boldsymbol{x}_{2,i}\right\|_2$, where $\Phi = \{\{1, \cdots, N\} \mapsto \{1, \cdots, N\}\}$ is the set of possible bijection assignments and $\|\cdot\|_2$ is L2 norm of vector. Given the optimal ϕ^*, EMD is defined as $d_{\text{EMD}} = \frac{1}{N}\sum_{i=1}^N \left\|\boldsymbol{x}_{1,\phi^*(i)} - \boldsymbol{x}_{2,i}\right\|_2$.

Let $\boldsymbol{x}_2$ remain unchanged and denote assigned $\boldsymbol{x}_1$ as $\hat{\boldsymbol{x}}_1$, then PointCutMix can be expressed as:

$$\tilde{\boldsymbol{x}} = (\boldsymbol{I} - \boldsymbol{B}) \cdot \hat{\boldsymbol{x}}_1 + \boldsymbol{B} \cdot \boldsymbol{x_2} \tag{2}$$

$$\tilde{\boldsymbol{y}} = (1 - \lambda)\boldsymbol{y}_1 + \lambda \boldsymbol{y}_2 \tag{3}$$

where $\boldsymbol{B} = \text{diag}\{b_1, b_2, \cdots, b_N\}, b_i \in \{0, 1\}$ is a diagonal binary mask indicating where to drop out and fill in from two samples, $\boldsymbol{I}$ is an N-order identity matrix, and $\lambda \in [0, 1]$ denotes mixed ratio which is sampled from Beta Distribution $\text{Beta}(\beta, \beta)$ (generally, $\beta = 1$). PointMixup can also be reduced to the above paradigm, where $\boldsymbol{B}$ degenerates to a diagonal mask with constant coefficient λ.

3.2. Gradient-based Saliency Region Location

After assignment, MixDAs mean selecting a subset $\boldsymbol{x}_2^s$ of $\boldsymbol{x}_2$ to replace the corresponding subset $\hat{\boldsymbol{x}}_1^s$ of $\hat{\boldsymbol{x}}_1$. Then the mixed sample is a point set $(\hat{\boldsymbol{x}}_1 / \hat{\boldsymbol{x}}_1^s) \cup \boldsymbol{x}_2^s$. Denoting the number of subset points as N_2, PointCutMix provides two selection modes: random sampling and k-Nearest Neighbor (kNN) sampling, named PointCutMix-R and PointCutMix-K, respectively. We denote these two subsets as $\boldsymbol{x}_2^s \sim \text{Rand}(N_1)$ and $\boldsymbol{x}_2^s \sim \text{kNN}(i_0, N_1)$ for simplification, where i_0 is the index of initial center point of kNN. Intuitively, the latter mode is relatively more regular, like a natural combination of two object parts. But the center of kNN samples $\boldsymbol{x}_{2,i_0}$ is completely random and thereby neglects the saliency information of point clouds.

Reviewing Eqn. (1), we take the absolute value of backpropagated gradient of point cloud $\boldsymbol{x}$ as

$$\boldsymbol{G} = \left|\frac{\partial \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y})}{\partial \boldsymbol{x}}\right| \tag{4}$$

where $\boldsymbol{G} \in \mathbb{R}^N$ is averaged over dimension C. Then we locate the point with maximum gradient value as the sampled center:

$$i_0 = \arg\max_i \ \boldsymbol{G}_i, \ i = 1, 2, \cdots, N \tag{5}$$

Notably, to make the gradient-based location more accurate, we do not apply the mix augmentation until after training several epochs, which can be called a *calibration* for gradients. Furthermore, to reduce the disturbance of gradient noise, we improve

the selection source of cut center from top-1 to top-k gradients: $i_0 \in \{\arg\text{top-}k_i\ \boldsymbol{G}_i\}$.

Apart from PointCutMix, RSMix also follows the mix process of cutting and replacement. For two rigid subsets of two point clouds, $\boldsymbol{x}_1^{rs}$ and $\boldsymbol{x}_2^{rs}$, the latter is utilized to replace the former. Though RSMix assigns two point clouds by rigid subset, the center location of a rigid subset is still random. So we can also improve it to extract saliency region of $\boldsymbol{x}_2$ by gradient to locate $\boldsymbol{x}_2^{rs}$.

3.3. Gradient-based Attentive Label Weight

As far as we know, the mixed weight λ of point clouds is reflected in the number ratio of mixed points. After taking λ, the number of cut subset is taken as $N_1 = \lfloor \lambda * N \rfloor$. Thus the final weight of a soft label is the inverse mapping: $\lambda = N_1/N$.

We want to emphasize that in one point cloud, different points contain different saliency information, so they are supposed to have different dominance on λ. Then how can we assign such quantified weight to bridge the gap between the sample and label spaces? The gradient matrix $\boldsymbol{G}$ exactly provides the tool. We modify the label weight from original counting by number to an attentive valuation:

$$\lambda = \frac{\sum\limits_{\boldsymbol{x}_{2,i} \in \boldsymbol{x}_2^s} g_{2,i}}{\sum\limits_{\boldsymbol{x}_{2,i} \in \boldsymbol{x}_2^s} g_{2,i} + \sum\limits_{\hat{\boldsymbol{x}}_{1,i} \in \hat{\boldsymbol{x}}_1} g_{1,i} - \sum\limits_{\hat{\boldsymbol{x}}_{1,i} \in \hat{\boldsymbol{x}}_1^s} g_{1,i}}, \tag{6}$$

where $g_{1,i} \in \boldsymbol{G}_1, g_{2,i} \in \boldsymbol{G}_2$ represent the gradients of $\hat{\boldsymbol{x}}_1$ and $\boldsymbol{x}_2$.

For different samples in one batch, λ in Eqn. (6) can not be guaranteed the same value, which might make training process difficult to converge. To deal with this problem, we then convert the implementation of softening labels by modifying the loss calculation from $(1-\lambda) \cdot \ell(f_{\boldsymbol{\theta}}(\widetilde{\boldsymbol{x}}), \boldsymbol{y}_1) + \lambda \cdot \ell(f_{\boldsymbol{\theta}}(\widetilde{\boldsymbol{x}}), \boldsymbol{y}_2)$ to $\ell(f_{\boldsymbol{\theta}}(\widetilde{\boldsymbol{x}}), \widetilde{\boldsymbol{y}})$.

It's worth noting that although the sampling way $\boldsymbol{x}_2^s \sim \mathrm{Rand}(N_1)$ of PointCutMix-R is not proper to impose our gradient-based saliency location, improving the value of λ with gradient-based attentive weight still get a promotion. To summarize, we name our PCSalMix as **PCSalMix-R** and **PCSalMix-K** respectively, corresponding to the random and kNN sampling way.

4. EXPERIMENTAL RESULTS

4.1. Datasets and Implementation Details

We evaluate PCSalMix on ModelNet40 and ModelNet10, the two widely used benchmark datasets for point cloud classification. ModelNet40 consists of $12,311$ CAD models from 40 man-made object categories, and ModelNet10 is a subset of it which includes 4,899 samples from 10 categories.

As for the networks, we mainly adopt the representative PointNet, PointNet++ and DGCNN, inherited from PointCutMix *etc.*, to conduct comparative experiments. All the hyper-parameter settings follow the original networks (*e.g.*, for DGCNN, the learning rate is 0.1 and follows the cosine decay *etc.*). For ConvDAs, we adopt scaling (0.8-1.25) and shifting (range$=0.1$) when compared to PointCutMix. And without otherwise specified, we take 5-epoch calibration, top-40 gradients and 0.5 mix probability. All experiments are conducted on NVIDIA A100 GPUs in PyTorch framework.

Note that there are two different splits of handling point clouds: pre-aligned (*-A) and unaligned (*-U); single-view and multi-view. Alignment is defined with horizontally rotated point clouds for training, while views are judged by whether objects are evaluated from different angles or not. We have explored and contrasted the results of previous works in detail to ensure fair and solid comparisons.

Table 1. ModelNet40 classification results (accuracy, %) with different models. † indicates that the second result belongs to reproduced PointCutMix and the last 2 rows belong to our PCSalMix. Other results are cited from the reported papers. (The same as below.)

Method	PointNet	PointNet++	DGCNN
Baseline-U	89.2	90.7	92.3
Baseline-A	-	91.9	92.7
PointMixup-U	89.9	91.7	-
PointMixup-A	-	92.7	92.9
PointAugment	90.9	92.9	93.4
PointCutMix-R†	- / 89.42	92.8 / 92.99	92.8 / 92.77
PointCutMix-K†	- / 89.55	93.4 / 93.23	93.1 / 93.15
PointCutMix-S	-	93.4	93.2
PCSalMix-R	90.48	93.11	93.19
PCSalMix-K	**90.92**	**93.56**	**93.52**

Table 2. ModelNet10 classification results with different models.

Method	PointNet	PointNet++	DGCNN
Baseline	-	93.3	94.8
PointCutMix-K†	93.83	95.04	94.93
PCSalMix-K	**93.94**	**95.59**	**95.37**

4.2. Point Cloud Classification

The comprehensive classification results of ModelNet40 are shown in Table 1. The superiority of our PCSalMix over previous MixDAs such as PointCutMix is reflected both on random and kNN sampling. Note that PointCutMix-S has tried to explore mixing with saliency while it brings little performance change. And PointAugment demands additional network structure beyond the classifier, it's not a fair and favorable comparison with it. The effectiveness of PCSalMix remains agnostic to different network models. PointNet performs MLP for all points in one cloud together, which makes it difficult to distinguish the replaced region. Yet our PCSalMix still brings a significant boost to its performance. Since ModelNet10 is a subset of ModelNet40, we just conduct the key experiments on ModelNet10, which are shown in Table 2 and share the same behavior.

In addition to PointCutMix, we try to conduct improved experiments on RSMix. However, we are unable to reproduce similar effects reported in the original paper. For example, sometimes RSMix does not even perform better than ConvDAs. But the effectiveness of PCSalMix strategy still works: utilizing DGCNN with ConvDAs of scaling and dropping, PCSalMix gets an accuracy of 93.68%, better than the reported 93.5%. As stated in Sec. 3.2, the center of cut subset $\boldsymbol{x}_2^{rs}$ is located by gradient saliency (grad_2). When locating the center of replaced subset $\boldsymbol{x}_1^{rs}$, random sampling (rand_1, the same as original paper) performs better than also by gradient (grad_1), since this makes the preserved points $\boldsymbol{x}_1/\boldsymbol{x}_1^{rs}$ relatively more salient.

4.3. Robustness against Adversarial Attack

In addition to classification accuracy, we test models' robustness against point dropping attack [13]. We conduct experiments with the following steps: attacking base test samples (from IF-Defense [22] codebase) with provided pre-trained models to generate adversarial samples; testing on adversarial samples with trained models (trained on base training samples with different augmentations). In other words, we do not impose additional defense means during training as IF-Defense. Though comparing with IF-Defense is un-

Table 3. Robustness on ModelNet40 against point dropping attacks. The best and second-place results for each row are red and bold. *No* and *IF* denote No-Defense and IF-Defense (* represents the reported results), while *Cut* and *Sal* denote PointCutMix and our PCSalMix.

Attack	Model	No	IF*	Cut-R	Sal-R	Cut-K	Sal-K
W/O	PointNet	88.41	87.64	88.45	**89.59**	88.90	**90.03**
	PointNet++	89.02	89.02	91.45	91.73	**91.82**	**92.38**
	DGCNN	89.79	89.22	91.61	91.73	**91.86**	**92.10**
Drop200	PointNet	49.43	66.94	76.74	76.01	**77.67**	**78.97**
	PointNet++	71.47	79.09	87.03	**88.41**	87.88	**88.37**
	DGCNN	58.71	73.30	85.62	**85.74**	85.17	**87.84**
Drop100	PointNet	68.35	77.76	83.43	**83.91**	83.79	**86.06**
	PointNet++	80.55	84.56	90.44	**90.56**	90.03	**90.76**
	DGCNN	76.42	83.43	**89.26**	89.18	88.94	**89.95**

Table 4. ModelNet40 accuracy of PCSalMix-K with different top-k gradients as the support set of cut center.

Top-k	1	5	20	40	80
DGCNN	93.15	93.19	93.23	**93.52**	93.07

favorable, we still list its results for reference. From Table 3, we can observe the superiority of PCSalMix's robustness against the dropping attacks: PCSalMix behaves better than PointCutMix, and kNN sampling behaves better than random sampling. PCSalMix's robustness is largely due to the learning of key points (saliency regions with high gradients) during training (different points contribute different importance), so that models can still focus on essential features to classify after dropping some points.

4.4. Exploration Details

We illustrate some PCSalMix details here. First, for the ablation study of gradient-based weight, we can refer to the comparison between PCSalMix-R and PointCutMix-R in Table 1, where ours performs better benefiting from the attentive weight. As for the support set of cut region's center, the value k of top-k gradients is investigated in Table 4. To a certain extent, a larger k can reduce the noise of gradients and make richer cut regions. Following this idea, we try to replace points with top-k gradients directly ($k = \lfloor \lambda * N \rfloor$) instead of by kNN sampling around a center, yet the performance has deteriorated. Next, the effects of mix probability p are shown in Table 5. It is best to retain a certain proportion of original samples during MixDAs. The accuracy of PointNet decreases more significantly at $p = 1$, because this network relatively ignores the local representation. And the mapping ways of PCSalMix can be abbreviated as *salient-to-assigned* based on PointCutMix and *salient-to-random* based on RSMix. Besides, we want to argue that with EMD assignment, our MixDA also basically maintains the rigid body structure, like RSMix. In addition to random and kNN sampling, we try to borrow sphere sampling from RSMix, but it does not bring gains. Moreover, we've mentioned the advantages of the generality of our approach across different data forms. On the 2D image dataset ImageNet [23] with ResNet-50 [24], our gradient saliency-based method improves the accuracy from 77.08% of CutMix to 78.21%.

When it comes to time efficiency, PCSalMix achieves better performance without taking too much extra time, as shown in Table 6. Increased time is mainly consumed in the double backpropagation in one-batch training, *i.e.*, one time with original samples for saliency location while another time with mixed samples for real training. The first-time gradient can be utilized for training models either, so that we can update the network parameters twice. In this way, it is equivalent to training twice the sample size ideally if we change p from 0.5 to 1, then the training epochs can be reduced, even to half.

Table 5. ModelNet40 accuracy of PCSalMix-K with different p.

Mix prob. p	0	0.25	0.5	0.75	1
PointNet	89.20	90.84	**90.92**	89.91	89.75
DGCNN	92.70	92.99	**93.52**	93.31	93.27

Table 6. Efficiency comparison: average training time per epoch on ModelNet40 with PointNet ($p = 1$ except for the baseline, 1 GPU).

Method	Baseline	PointCutMix-K	PCSalMix-K
Time (s)	41.32	45.27	46.49

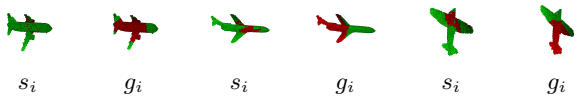

Fig. 3. Comparison of different saliency region location methods: based on s_i or our g_i.

4.5. Saliency Detection and Visual Analysis

In addition to directly using gradients as the saliency score for point clouds, our motivation can be extended to combine with any method that can detect saliency regions. In [13], Zheng *et al.* defined the saliency map of point $\boldsymbol{x}_i$ as $s_i = -\frac{\partial \ell}{\partial r_i} r_i^{1+\alpha}$, where $r_i = \sqrt{\sum_{j=1}^{3} (\boldsymbol{x}_{ij} - \boldsymbol{x}_{cj})^2}$, $\boldsymbol{x}_c$ is the center of a point cloud, and $\alpha > 0$. Based on this saliency map, they proposed an iterative point-drop algorithm to dynamically narrow the saliency region. The computation of saliency regions by s_i is much more expensive than if we directly use gradient g_i as the saliency score to locate regions with sampling, since the former has to perform gradient backpropagation multiple times. Naturally, we can replace our g_i-based location with s_i-based location, while the other steps are the same. However, the ModelNet40 accuracy of PointNet decreases from 90.92% by g_i to 88.65% by s_i. This is most likely due to the inaccurate detection of saliency regions. As shown in Fig. 3, the s_i-based method only detects the cargo hold of the aircraft, while ours detects key parts, such as the tail and rudder. Another saliency extraction method proposed by Ding *et al.* [12] fails to be implemented, due to its not being open source. For more visualization examples, please refer to Fig. 1.

5. CONCLUSION

We propose an effective and general gradient saliency-based mix strategy for 3D point cloud classification: PCSalMix. We locate saliency regions through neural network's gradients to impose cut and replacement operation, and adopt the attentive gradient values to set more accurate weights for soft labels. The effectiveness of our PCSalMix is fully demonstrated by experiments. In future work, it is worth applying saliency-based mix augmentation to downstream tasks of point clouds such as segmentation and object detection.

6. REFERENCES

[1] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[2] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[3] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon, "Dynamic graph cnn for learning on point clouds," *Acm Transactions On Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.

[4] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui, "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*, 2020, pp. 5589–5598.

[5] Yunlu Chen, Vincent Tao Hu, Efstratios Gavves, Thomas Mensink, Pascal Mettes, Pengwan Yang, and Cees GM Snoek, "Pointmixup: Augmentation for point clouds," in *European Conference on Computer Vision*. Springer, 2020, pp. 330–345.

[6] Jinlai Zhang, Lyujie Chen, Bo Ouyang, Binbin Liu, Jihong Zhu, Yujin Chen, Yanmei Meng, and Danfeng Wu, "Pointcutmix: Regularization strategy for point cloud classification," *Neurocomputing*, vol. 505, pp. 58–67, 2022.

[7] Dogyoon Lee, Jaeha Lee, Junhyeop Lee, Hyeongmin Lee, Minhyeok Lee, Sungmin Woo, and Sangyoun Lee, "Regularization strategy for point cloud via rigidly mixed sample," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15900–15909.

[8] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz, "mixup: Beyond empirical risk minimization," in *The International Conference on Learning Representations(ICLR)*, 2018.

[9] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6023–6032.

[10] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song, "Puzzle mix: Exploiting saliency and local statistics for optimal mixup," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5275–5285.

[11] AFM Uddin, Mst Monira, Wheemyung Shin, TaeChoong Chung, Sung-Ho Bae, et al., "Saliencymix: A saliency guided data augmentation strategy for better regularization," *arXiv preprint arXiv:2006.01791*, 2020.

[12] Xiaoying Ding, Weisi Lin, Zhenzhong Chen, and Xinfeng Zhang, "Point cloud saliency detection by local and global feature fusion," *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5379–5393, 2019.

[13] Tianhang Zheng, Changyou Chen, Junsong Yuan, Bo Li, and Kui Ren, "Pointcloud saliency maps," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1598–1606.

[14] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[15] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan, "Relation-shape convolutional neural network for point cloud analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8895–8904.

[16] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio, "Manifold mixup: Better representations by interpolating hidden states," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6438–6447.

[17] Ruihui Li, Xianzhi Li, Pheng-Ann Heng, and Chi-Wing Fu, "Pointaugment: an auto-augmentation framework for point cloud classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6378–6387.

[18] Alexey Nekrasov, Jonas Schult, Or Litany, Bastian Leibe, and Francis Engelmann, "Mix3d: Out-of-context data augmentation for 3d scenes," in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 116–125.

[19] Jaeseok Choi, Yeji Song, and Nojun Kwak, "Part-aware data augmentation for 3d object detection in point cloud," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 3391–3397.

[20] Oytun Akman and Pieter Jonker, "Computing saliency map from spatial information in point cloud data," in *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer, 2010, pp. 290–299.

[21] Haifeng Yu, Rui Wang, Junli Chen, Liang Liu, and Wanggen Wan, "Saliency computation and simplification of point cloud data," in *Proceedings of 2012 2nd International Conference on Computer Science and Network Technology*. IEEE, 2012, pp. 1350–1353.

[22] Ziyi Wu, Yueqi Duan, He Wang, Qingnan Fan, and Leonidas J Guibas, "If-defense: 3d adversarial point cloud defense via implicit function based restoration," *arXiv preprint arXiv:2010.05272*, 2020.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Sun Jian, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.