



OPEN PCAAlign: a general data augmentation framework for point clouds

Chen Zhang^{1,4}, Abiao Li^{2,5}, Dan Zhang^{1,4,5}✉ & Chenlei Lv³✉

With the advancement of 3D scanning technologies and deep learning theories, point cloud-based deep learning networks have gained considerable attention in the fields of 3D vision and computer graphics. Leveraging the rich geometric information present in 3D point clouds, these networks facilitate more accurate feature learning tasks. However, existing networks often suffer from generalization defects caused by variations in pose and inconsistent representations of training data. In this paper, we propose a novel data augmentation framework to overcome these limitations. Our approach utilizes principal component analysis (PCA) to generate four aligned copies of a point cloud. These copies are then input into a multi-channel structure, which is compatible with popular backbones of point cloud-based deep networks. Finally, the outputs of the multi-channel structure are merged to generate rotation-invariant feature learning results. Experimental evaluations demonstrate the efficacy of our framework, showcasing significant improvements in various existing point cloud-based deep learning methods. Notably, our method exhibits enhanced robustness in classification tasks, particularly when dealing with point clouds containing random pose variations and non-uniform densities. Project link: <https://github.com/LAB123-tech/PCAAlign>.

In recent years, the utilization of 3D point clouds has become increasingly popular across various fields, such as robotics, autonomous driving, and augmented/virtual reality. Compared to 2D images, 3D point clouds encompass comprehensive geometric information, enabling more precise feature analysis and semantic learning. Particularly when combined with deep learning frameworks, a multitude of research studies based on point clouds have achieved significant advancements in related applications. These studies leverage statistical knowledge derived from extensive datasets in an intelligent manner. They offer deep feature-based functionalities for tasks that cannot be addressed by conventional geometric methods, including semantic segmentation, up-sampling in the absence of accurate local neighborhood information, completion of missing areas. However, the scale of existing 3D point cloud datasets are relatively small with limited number of semantic categories due to the high cost of acquiring 3D point clouds. The drawback makes it difficult to design deep network architectures with generalization learning ability for training point clouds. To solve the problem, point cloud-based data augmentation is proposed.

The main purpose of data augmentation is to address the limitations of the existing training dataset, which include insufficient samples, sensitivity to similar transformations (such as translation, scaling, and rotation), and data bias, among others. For example, when adjusting the density of a point cloud based on its point distribution or transforming its pose, the semantic label of the new copy remains unchanged. These new copies expand the training dataset and can be considered as targeted data augmentation. Some classic point cloud-based deep networks incorporate default data augmentation techniques, such as random rotation and associated normal vector, to improve performance. Another type of data augmentation aims to enhance the network itself by analyzing the prediction deviations between the existing training set and the test set. Due to the limited scale of the dataset, the probability distribution of the training dataset may not fully cover the test set, leading to performance deficiencies in the trained network. The second type of data augmentation addresses this issue by establishing a reinforcement learning framework. This reinforcement is compatible with any neural network architecture.

¹School of Computer Science, Qinghai Normal University, Xining 81017, People's Republic of China. ²School of Information Technology, Jiangxi University of Finance and Economics, Nanchang 330013, People's Republic of China. ³College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, People's Republic of China. ⁴State Key lab of Tibetan Intelligent Information Processing and Application(Co-established by province and ministry), Xining 81017, People's Republic of China. ⁵These authors contributed equally: Abiao Li and Dan Zhang. ✉email: danz@mail.bnu.edu.cn; chenleilv@mail.bnu.edu.cn

However, this data augmentation approach may encounter over-fitting issues since the deep neural network becomes attached to the test samples, and no new point clouds are introduced.

In this paper, we provide a new general data augmentation framework for point cloud-based deep learning. It contains two parts: aligned point clouds generation and multi-channel structure for semantic feature learning. The aligned point clouds generation is to produce regular copies from the original point cloud by PCA. Without additional feature analysis, the influence of different poses can be removed from the dataset, which is helpful for the following step to handle random placements of point clouds. Based on the aligned point clouds, the multi-channel structure provides a universal framework to train the deep features and is compatible with any existing deep neural networks. By sharing multiple parameters, the structure can assist selected backbone networks in obtaining better feature learning ability, while being robust to different poses and improving semantic analysis accuracy. The pipeline is shown in Fig. 1. The contributions of our method can be concluded as:

- We present a PCA-based point cloud generation method that produces new copies of the original point cloud with aligned poses. It avoids complex feature coding processes while achieving rotation-invariant properties in the following analysis.
- We design a multi-channel structure to enhance the feature learning ability for any point cloud-based deep neural network. Benefiting from the aligned point clouds and decoupling combination way, the structure is invariant to different rotations of point clouds and compatible with various types of deep neural networks.
- We provide a comprehensive validation of data augmentation with various mainstream point cloud-based deep learning methods. The related analysis provides the most detailed validation data so far to demonstrate the impact of data augmentation in semantic analysis task.

Related works

Point cloud deep learning network

Following the development of deep learning, many researchers attempted to transfer related technologies for point cloud-based semantic analysis. Qi et al.¹ proposed the prior work (PointNet) to implement point cloud classification and semantic segmentation. It introduces multilayer perceptron (MLP) to encode point-based features and max-pooling mechanisms to handle the disorder problem. Following the achievements, they enhanced the framework named PointNet++² with the local geometric analysis. Afterward, researchers employed different backbone networks to improve the performance for similar tasks. Ben et al.³ utilized a point network named 3DmFV based on the convolutional neural networks (CNN) and 3D Fisher vector representation to implement point cloud-based classification and segmentation. Li et al.⁴ presented another CNN-based point cloud deep learning framework named PointCNN. It serializes points according to feature weights for geometric feature keeping. Thomas et al.⁵ and Wu et al.⁶ employed the same backbone of CNN to improve the performance for point cloud semantic analysis. Wang et al.⁷ proposed a new point cloud-based deep learning backbone: DGCNN, which is based on the dynamic graph CNN. It further enhances the learning ability of local geometric features. With the widespread application of attention mechanisms in deep learning⁸, many scholars have begun to introduce relevant frameworks to enhance the semantic analysis ability for point clouds. Zhao et al.⁹ and Guo et al.¹⁰ employed a transformer structure: PCT, which utilizes the self-attention mechanism for point cloud-based feature learning. Subsequently, a series of improvement works^{11–14} were proposed to enhance the point cloud analysis performance based on the transformer structure.

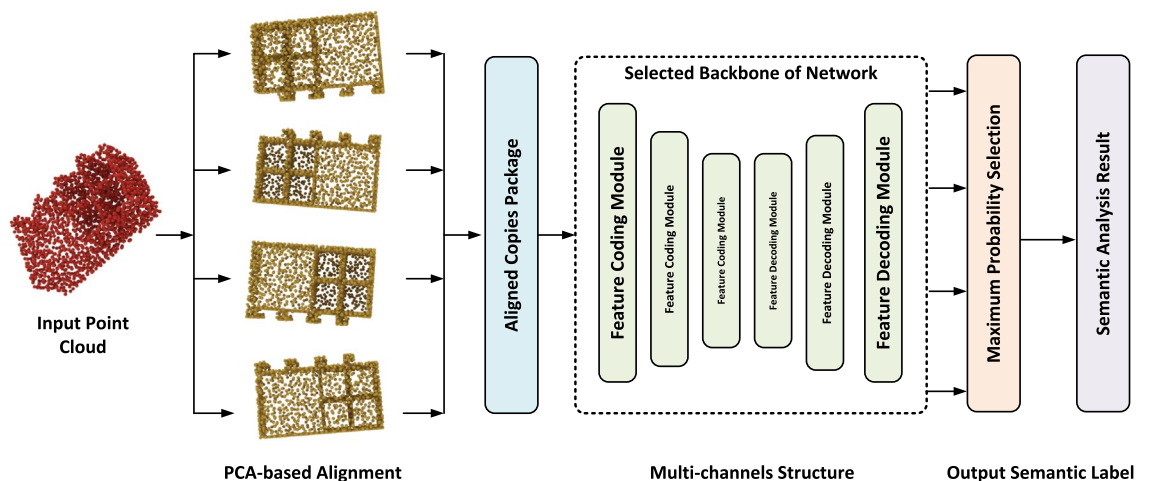


Fig. 1. Pipeline of PCAlign. The part marked by dashed lines can be replaced by any mainstream point cloud-based deep neural network.

Data augmentation for point cloud deep learning network

Although the new architectures for point cloud-based deep learning have achieved significant performance improvements, they are still limited by the bias of the training dataset, including scale, quality, and random pose. To address the impact of limited samples on the network's ability to learn semantic features, research works of data augmentation have been proposed. Sun et al.¹⁵ proposed a rotation-invariant representation for point clouds which is to reduce the influences of different poses in the dataset. Li et al.¹⁶ utilized a similar method to select regular poses of point clouds. Xiao et al.¹⁷ proposed a data augmentation framework that can mitigate the data constraint effectively across different perception tasks and scenarios. Leng et al.¹⁸ propose a data augmentation method to leverage unlabeled data to enrich the training data.

To extend the scale of the dataset, Chen et al.¹⁹ designed an interpolation method that can generate new models from existing samples. Choi et al.²⁰ divided objects into partitions with different regulations to enrich the dataset. Shivanand et al.²¹ changed local neighborhoods to implement data augmentation in point cloud-based deep learning tasks. Mei et al.²² attempted to analyze point-based soft clustering to implement a data augmentation-free unsupervised learning for point clouds. Some data augmentation methods utilize an iterative loop mechanism that feeds back the results from the testing set into the training process to implement sample augmentation. Li et al.²³ proposed a new auto-augmentation framework that automatically optimizes and augments point cloud samples to enrich the data diversity. The framework is sample-aware and takes an adversarial learning strategy to jointly optimize an augmented network and a classifier network to produce augmented samples. Zhang et al.²⁴ employed a bilevel optimization to establish an augmentor that minimizes a base model's loss on a validation set.

In this paper, we propose a general data augmentation framework that is to enhance training data by alignment. Different from traditional methods, our framework does not perform complex feature extraction for rotation-invariant representation, but rather directly generates aligned copies without pose selection. Combined with a multi-channel structure, our framework can flexibly connect to any backbone network and conduct parameter-shared multi-path learning to implement data augmentation. In the following parts, we discuss the implementation details.

PCA-based alignment Pre-processing

As mentioned before, the non-uniform density takes uncontrollable factors for point cloud analysis. In addition, inconsistent point numbers and random outliers have negative effects on the deep feature analysis of point clouds. Therefore, pre-processing should be employed to improve the quality of the input point cloud. The traditional solutions use farthest point sampling (FPS) to uniform density and point number. The drawback is that the sampling doesn't consider manifold distribution which reduces the accuracy of geometric consistency. The computational efficiency is relatively low when the point number of the input point cloud reaches the order of 100k. Considering the above factors, we employ the approximate intrinsic simplification (AIVS)²⁵ to implement uniform processing. As our method requires eigenvector-based point cloud alignment, random outliers should be removed to avoid errors. We use an outlier removal method mentioned in²⁶ to delete outliers.

Alignment

Based on the pre-processing point clouds, we implement PCA-based alignment that is to generate aligned point cloud copies. It has been introduced that the rotation of the point cloud should not change the related shape feature²⁷. However, the existing deep neural networks are sensitive to different poses of point clouds which reduces the robustness of feature learning. Some rotation-invariant feature representations are proposed to solve the problem. Such representations increase the complexity and unpredictability of the training process. Other solutions attempt to search for a unique pose to align point clouds. In our view, it is unnecessary as there are inherent ambiguities in the alignment posture of different objects. We provide a more concise method to achieve rotation-invariant properties. It computes eigenvectors from the point cloud at first. Then, four aligned point cloud copies are generated based on the eigenvectors. The four copies take all canonical poses, which reduces the influences of different rotations significantly.

Firstly, we compute the initial align axes by PCA for alignment. Let P to be an input point cloud. The covariance matrix C from P is represented as

$$C = \mathbf{X}\mathbf{X}^T, \quad \mathbf{X} = [(p_1 - \bar{p}), \dots, (p_n - \bar{p})], \quad (1)$$

where p_i is a point of P , \bar{p} is the center of P . Based on the C , we implement PCA to achieve eigenvectors and eigenvalues, represented as

$$\mathbf{E}\mathbf{\Lambda}\mathbf{E}^T = C, \quad \mathbf{E} = [e_1, e_2, e_3], \quad \mathbf{\Lambda} = [\lambda_1, \lambda_2, \lambda_3], \quad (2)$$

where e_i represents the eigenvector and λ_i represents the eigenvalue. We achieve eigenvectors e_{max} and e_{min} corresponding to the maximum eigenvalue λ_{max} and minimum one λ_{min} . The two eigenvectors are regarded as the initial align axes.

It should be noticed that each aligned axis has two directions. Then, there are four possible combinations by e_{max} and e_{min} corresponding to four aligned poses. According to the combinations, four local coordinate systems can be established for aligned point cloud copies generation. The local coordinate systems $Local_{cord}$ can be represented as

$$Local_{cord} = \begin{cases} [e_{max}^+, e_{min}^+, e_{max}^+ \times e_{min}^+] \\ [e_{max}^+, e_{min}^+, e_{max}^+ \times e_{min}^+] \\ [e_{max}^-, e_{min}^-, e_{max}^- \times e_{min}^-] \\ [e_{max}^-, e_{min}^-, e_{max}^- \times e_{min}^-] \end{cases}, \quad (3)$$

where symbols $^+$ and $^-$ represent directions, the third axis is computed by the cross product of oriented axes. An instance with direction labels is shown in Fig. 2. Based on the four local coordinate systems, an input point cloud can be transferred into four aligned point cloud copies. In Fig. 3, we show some instances. Based on the PCA-based alignment, the input point cloud with random poses can be aligned into four copies with aligned poses. The influence of different poses is reduced significantly. Although there are no one-to-one correspondences for the four aligned copies, the rotation-invariant property can be guaranteed in the following calculation.

The generated aligned copies control the rotation ranges of point clouds, which provide an initial pose alignment. However, the copies don't provide strict correspondences between different point clouds. It means that the copies still do not provide rotational invariance. To solve the problem, we design a multi-channel structure that packs the aligned copies of point clouds as a single input unit. The structure employs a maximum probability selection module to achieve rotational robustness. It doesn't require additional feature coding that avoids compatibility issues and geometric information loss. The implementation details are introduced below.

Firstly, the aligned copies $\{P_A\}$ from P are input into the structure that is shown in Fig. 4. $\{P_A\}$ is packaged into a tensor ($4 \times n \times 3$) that is four times the size of P ($n \times 3$). Next, the packaged tensor is used to train a deep network based on a selected backbone. First dimensional data of the tensor are used to obtain an output that is four times to the original output through the network. This process can be considered as independently network training four copies while the parameters are shared at the same time. Then, we achieve the result that corresponds to the four aligned copies. The result is a mixed vector that is four times larger than output vector of original network.

As mentioned before, there is no one-to-one correspondence between aligned copies at first. The order of the copies still influences the output mixed vector of the structure. To solve the problem, we introduce a maximum probability selection module to achieve rotation-invariant results from the output mixed vector. The maximum

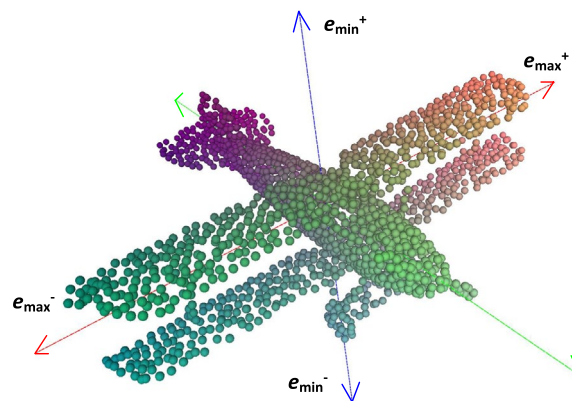


Fig. 2. An instance of local coordinate system.

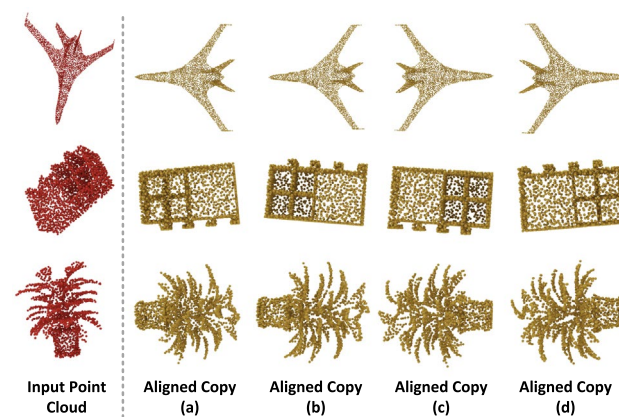


Fig. 3. Aligned point cloud copies (a)–(d) generated by PCA.

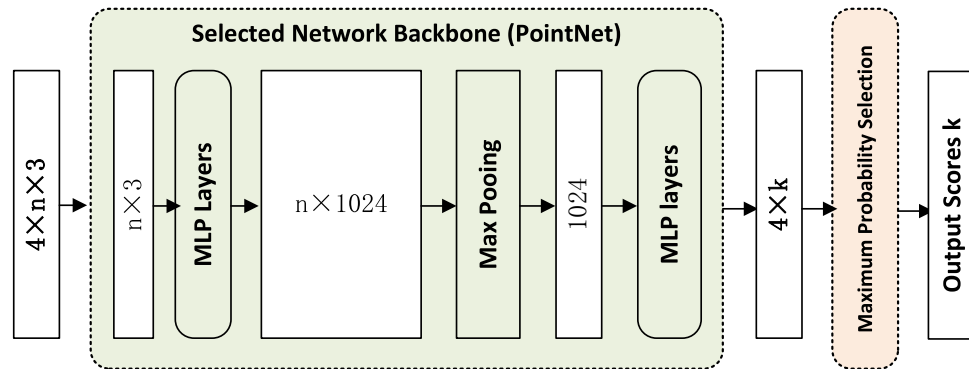


Fig. 4. Visualization of multi-channel structure. The PointNet¹ is selected to be the network backbone as an instance. The final output score vector is selected from the mixed vectors that have maximum probability value.

probability selection module implements one hot encoding for output probability vectors from four channels. Then, we select the component with the highest value of one hot encode as the reference value. According to the reference value, we re-examine each channel's probability distribution and identify the probability vector with the highest probability on the corresponding reference component for each channel. Finally, the examined probability vector is the output. An instance is shown in Fig. 5. Such a design can suppress the maximum probability of error in a specific channel's impact on the determination result, which ensures the final result complies with statistical analysis. Note that if the probabilities are accumulated instead of the maximum, if one of the four copies has a high probability of making an incorrect prediction, then even if there are multiple correct predictions in other point cloud copies, the final prediction will still be incorrect. Therefore, here we use the maximum probability instead of a simple summation, and the results in Table 8 also verify the above theory.

It should be noticed that some deep neural networks are bound with random rotation modules to provide default data augmentation. Such modules reduce the function of aligned copies. In practice, we remove the module from the backbone to regular the input. Then, the optimal pose of the original input point cloud for semantic analysis can be filtered through the maximum probability selection. Combined with the PCA-based alignment and the multi-channel structure, PCAlign searches for the result with the highest semantic matching degree from four aligned copies while eliminating pose interference and improving learning accuracy. In the experimental section, we will comprehensively validate the PCAlign.

Experiments

In this part, we evaluate the performance of PCAlign in point cloud-based semantic classification tasks. We use a cloud computing platform to provide an efficient and fair experimental environment, which is equipped with Intel i9 3.0 GHz and A100 as CPU and GPU units. The operation system is Windows 11 and the deep learning platform is established based on pycharm and pytorch. The test datasets are ModelNet40²⁸ and ShapeNet²⁹ which have been used in many related research works. The point cloud learning networks used for enhancement include: Point¹, Point++², DGCNN⁷, PCT¹⁰. To reveal the performance of PCAlign, we conduct tests on ModelNet40 to investigate its impact on classification tasks under various variations and feature enhancements, including random rotation, resampling, and normal vector augmentation. We also evaluate the improvement of PCAlign in part segmentation task based on ShapeNet models. Finally, we provide a comprehensive analysis to illustrate the operation mechanism of PCAlign.

We used the Adam optimizer with momentum and weight decay values set to 0.9 and 0.0001, respectively. We warm up the network for 10 epochs and employ a cosine learning rate schedule for the remaining epochs, decreasing the learning rate to 0.000001 at the final epoch. For the ModelNet dataset, we trained for 150 epochs, starting with an initial learning rate of 0.5. For the ShapeNetPart dataset, we trained for 200 epochs, beginning with an initial learning rate of 0.05. Since we used multiple point cloud copies, the batch size was set to 24.

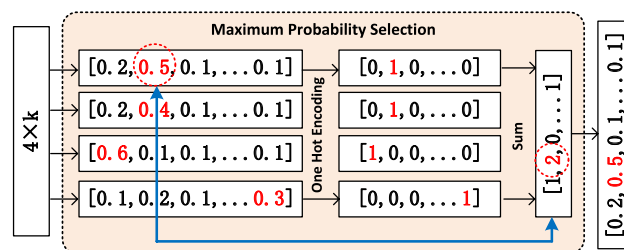


Fig. 5. Instance of maximum probability selection module.

Evaluation for classification

As mentioned before, the PCAlign can combine any deep neural network to obtain semantic analysis results. To evaluate the performance of PCAlign in the semantic classification task, we use four mainstream networks to provide quantitative results. To achieve fair experimental data, we conducted new parameter training on all network architectures with a fixed epoch number (200). In Table 1, we report the results of the classification accuracy by different methods. Benefited from the PCAlign, most of networks achieve improvements of performance. It demonstrates the excellent generalization of PCAlign for enhancement of feature learning capability. To be a general data augmentation framework, PCAlign needs to demonstrate its superior performance compared to other data augmentation approaches. For the purpose, we compare different data augmentation frameworks with PointNet++ and PCT for classification tasks, including PointAugment²³, PointWOLF³⁰, PolarMix¹⁷ and Psedoaugment¹⁸. In Table 2, we report the quantitative results. It is clear that PCAlign achieves better results.

As can be seen from the time complexity in Tables 1 and 2, the parameter count remains consistent with the original algorithm. Because the data augmentation algorithm designed in this paper does not add any extra parameters. The FLOPs have increased fourfold. Because the input samples have increased fourfold, the FLOPs have also increased fourfold. However, there is no significant increase in time. This is because during network inference, the four point cloud copies can be processed in parallel. The only additional time cost is due to the maximum probability selection module. However, the additional time cost for inference is also very small.

However, we must acknowledge that the algorithm proposed in this paper incurs a significant increase in time cost during training. Other data augmentation methods have increased the parameter count of the baseline algorithm. The parameter count remains consistent with the original algorithm. In terms of FLOPs, other algorithms only increase a little. In terms of time, other algorithms only add a slight increase as well.

Evaluations for random rotations

The ModelNet40 takes default pose alignment that weakens the advantage of rotation-invariant property implemented by PCAlign. Most of the raw collected point clouds lack unified perspectives, especially for dataset with significant semantic differences across categories. To obtain a more objective measurement for unpredictable poses of point clouds, we add random rotations into the test dataset and implement related feature training for classification task. In Fig. 6, we show some instances of point clouds with random rotations and related aligned copies. PCAlign provides stable pose control. In Table 3, we report the quantitative results of different methods. Comparing with previous experimental data in Table 1, all reference methods show significant performance degradation for the previously reported data when dealing with data that has random rotations. Benefited from

Method/classification	mACC (%)	OA (%)	# Params	Flops/sample	Time (MS)
PointNet ¹	87.3	90.6	3.472M	0.44G	6.8
PointNet++ ²	89.8	92.3	1.745M	4.09G	53.8
DGCNN ⁷	89.7	92.6	1.815M	2.45G	35.4
PCT ¹⁰	89.5	92.5	9.154M	17.15G	66.7
PCAlign + PointNet ¹	88.7 (↑ 1.5)	91.2 (↑ 0.6)	3.472M	1.803G	10.1
PCAlign + PointNet++ ²	90.7 (↑ 0.9)	92.8 (↑ 0.5)	1.745M	16.23G	64.6
PCAlign + DGCNN ⁷	89.6 (↓ 0.1)	92.6 (↑ 0.0)	1.815M	9.63G	47.2
PCAlign + PCT ¹⁰	91.1 (↑ 1.9)	93.8 (↑ 1.3)	9.154M	67.92G	78.9

Table 1. Comparisons mACC, OA and the computational complexity of different deep neural networks with and without PCAlign in classification task based on ModelNet40.

Method/classification	mACC (%)	OA (%)	# Params	Flops/sample	Time (MS)
PointNet++ ²	89.8	92.3	1.745M	4.09G	53.8
PCT ¹⁰	89.5	92.5	9.154M	17.15G	66.7
PointWOLF ³⁰ + PointNet++ ²	90.1	92.4	2.115M	5.32G	59.6
PointWOLF ³⁰ + PCT ¹⁰	90.3	92.8	9.538M	18.23G	71.3
PointAugment ²³ + PointNet++ ²	90.5	92.6	2.354M	5.69G	61.3
PointAugment ²³ + PCT ¹⁰	90.7	93.2	10.032M	18.65G	74.5
PolarMix ¹⁷ + PointNet++ ²	90.2	91.4	1.745M	5.74G	65.3
PolarMix ¹⁷ + PCT ¹⁰	90.3	93.1	9.154M	18.25G	79.5
Psedoaugment ¹⁸ + PointNet++ ²	90.4	91.5	2.425M	6.571G	70.2
Psedoaugment ¹⁸ + PCT ¹⁰	90.5	93.3	10.834M	19.65G	83.6
PCAlign + PointNet++ ²	90.7	92.8	1.745M	16.23G	64.6
PCAlign + PCT ¹⁰	91.1	93.8	9.154M	67.92G	78.9

Table 2. Comparisons of different data augmentation framework in classification task based on ModelNet40.

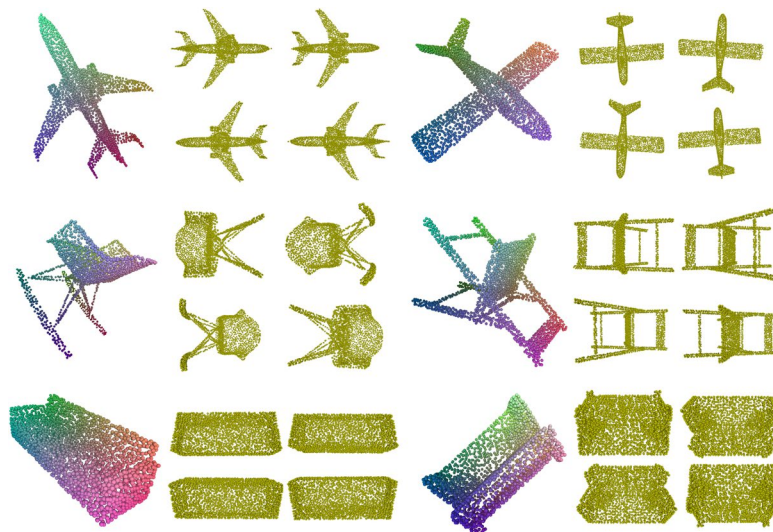


Fig. 6. Visualization of point clouds with random rotations and related aligned copies.

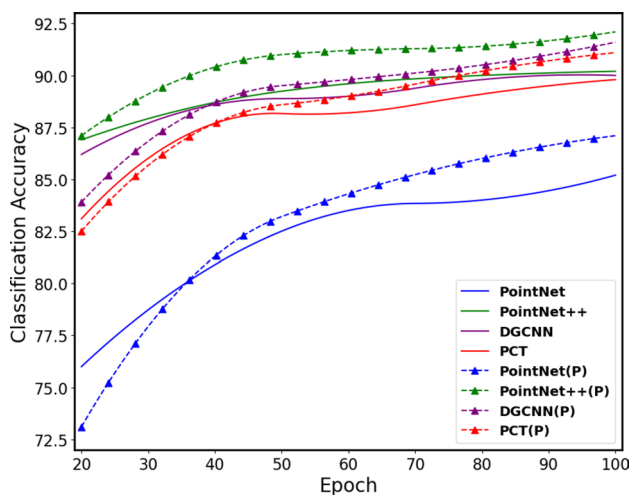


Fig. 7. Visualization of classification accuracy changes according to the network iterations for different methods. The label (P) means that the method is enhanced by PCAlign.

Method/classification	mACC (%)	OA (%)
PointNet ¹	82.5	86.4
PointNet++ ²	87.8	90.2
DGCNN ⁷	87.3	90.5
PCT ¹⁰	87.9	90.6
PCAlign + PointNet ¹	88.7 (↑ 6.2)	91.2 (↑ 4.8)
PCAlign + PointNet++ ²	90.6 (↑ 2.8)	92.6 (↑ 2.4)
PCAlign + DGCNN ⁷	89.6 (↑ 2.3)	92.6 (↑ 2.1)
PCAlign + PCT ¹⁰	90.7 (↑ 2.8)	93.6 (↑ 3.0)

Table 3. Comparisons of different deep neural networks with and without PCAlign in classification task based on ModelNet40. Random rotations are added into the point clouds before training which is used to estimate the pose influence.

the PCA-based alignment, PCAlign provides rotational robustness that ensures training process is not affected by various poses. The classification results based on PCAlign are same between Tables 1 and 3.

In fact, the precise control of posture provided by PCAlign not only offers rotational robustness, but also aids in the convergence of final deep network parameter optimization. The reason is that the feature encoding on aligned copies helps capture significant geometric information related to the semantic information, while simultaneously eliminating semantic ambiguity caused by different poses. To demonstrate the hypothesis, we present the classification accuracy as a function of the number of network iterations in Fig. 7. It proves that PCAlign is helpful for convergence.

Evaluations for random resampling

It has been discussed that the local neighborhoods with different point distributions or densities take an important influence on data augmentation. The reason is that the neighbor structures decide the feature coding paths in most of deep neural networks. In general, the coding paths are constructed by K-nearest neighbor searching and farthest point sampling. Significantly, non-uniform distributions change the k neighbors for points. In the previous test, the point clouds were pre-processed by uniform simplification to optimize the point distributions. To evaluate the influence of non-uniform distributions in point-based semantic analysis, we provide a quantitative analysis for point clouds with different point distributions. We use a random resampling to select points from the point cloud, which doesn't consider densities. In Fig. 8, we compare two kinds of resampling results. The random resampling changes the point distributions in different local regions. Based on the changed point clouds, we retrain networks and report new classification results in Table 4. Experimental data directly reflects

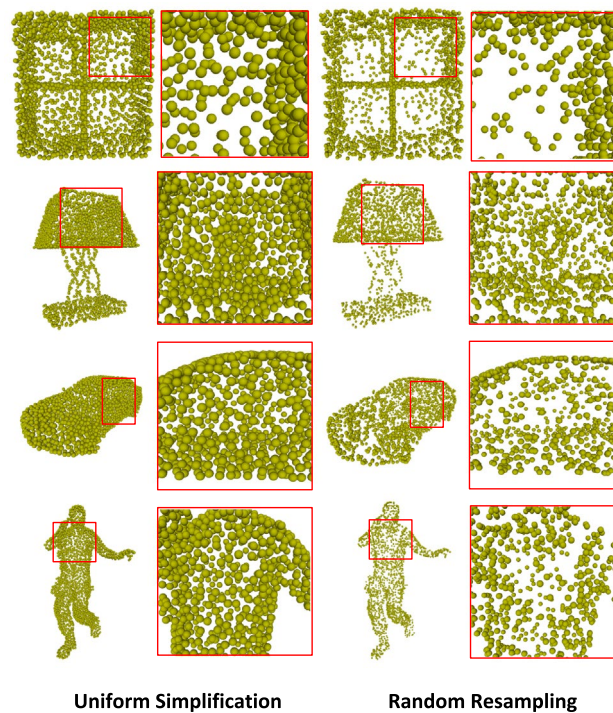


Fig. 8. Visualization of random resampling for point clouds.

Method/classification	mACC (%)	OA (%)
PointNet ¹	86.0	89.9
PointNet++ ²	90.2	92.4
DGCNN ⁷	88.7	92.1
PCT ¹⁰	90.0	92.7
PCAlign + PointNet ¹	87.5 (↑ 1.5)	90.4 (↑ 0.5)
PCAlign + PointNet++ ²	90.6 (↑ 0.4)	92.8 (↑ 0.4)
PCAlign + DGCNN ⁷	89.4 (↑ 0.7)	92.5 (↑ 0.4)
PCAlign + PCT ¹⁰	90.5 (↑ 0.5)	92.7 (↑ 0.0)

Table 4. Comparisons of different deep neural networks with and without improvement of PCAlign in classification task based on ModelNet40. Random resampling is added into the point clouds before training which is used to evaluate the influence of point distributions.

the sensitivity of the related neural network to point distributions in local neighborhoods. Overall, PCAlign can achieve more stable results.

Evaluations for normal vector enhancement

In the latest experiments and engineering practices, researchers found that introducing normal vectors as input data can effectively enhance the feature learning ability of deep neural networks. By only learning the coordinates of points, the network naturally becomes sensitive to poses. Once normal vectors are introduced, more local geometric information is incorporated into the feature encoding process, which significantly enhances the encoding capability of related deep neural networks for point cloud geometric features. To evaluate the influence of normal vector enhancement, we report the classification accuracy of different methods in Table 5. It can be observed that the performance of the majority of methods has been improved. Due to the PCA-based alignment, PCAlign effectively establishes a global normal alignment for point clouds. It limits the performance improvement of normal vector enhancement for PCAlign.

Evaluations for part segmentation

For improvement of classification, aligning poses of PCAlign is a straightforward and intuitive approach. Indeed, it can particularly enhance object recognition accuracy in cases where there are significant pose variations between point clouds. To further validate PCAlign's generalization in deep learning tasks, we evaluate its performance in the task of part segmentation. In Table 6, we report the improvement of PCAlign for PCT¹⁰ and PG³¹. The classification accuracy of most categories can be improved. We also compare different data augmentation frameworks with PCT for part segmentation task. In Fig. 9, some instances are shown. In Table 7, we report the quantitative results. The PointAugment²³ and PointWOLF³⁰ attempt to change the point positions, which take some diversity of semantic features. However, such diversity cannot improve the accuracy of local feature

Method/classification	mACC (%)	OA (%)
PointNet ¹	89.5	91.7
PointNet++ ²	91.0	92.9
DGCNN ⁷	88.8	91.7
PCT ¹⁰	90.0	92.8
PCAlign + PointNet ¹	89.1 (↓ 0.4)	91.7
PCAlign + PointNet++ ²	90.7 (↓ 0.3)	92.8 (↓ 0.1)
PCAlign + DGCNN ⁷	89.6 (↑ 0.6)	91.8 (↑ 0.1)
PCAlign + PCT ¹⁰	91.1 (↑ 1.1)	93.8 (↑ 1.0)

Table 5. Comparisons of different deep neural networks with and without improvement of PCAlign in classification task based on ModelNet40. Normal vectors are bound as the input regarded as the default data augmentation.

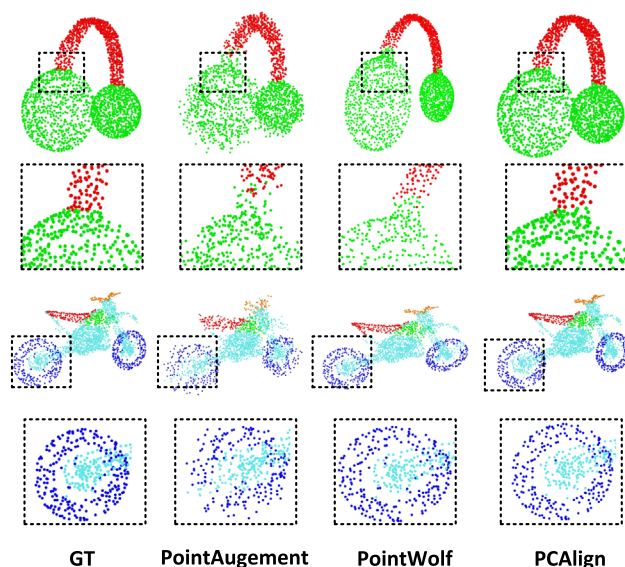


Fig. 9. Visualization of part segmentation of different data segmentation frameworks.

Method	ins. avg.	air.	bag	cap.	car	cha.	ear.	gui.	kni.	lam.	lap.	mot.	mug	pis.	roc.	ska.
PCT	86.1	83.5	83.4	87.5	80.5	90.3	78.2	92.2	88.1	82.6	96.2	77.5	95.8	85.4	64.6	83.3
PCAlign+PCT	86.7↑ (0.6)	84.7	84.5	84.1	79.9	90.5	81.6	89.8	88.9	85.8	96.7	80.2	96.0	86.1	65.3	82.1
PointGT	85.8	84.3	84.5	88.3	80.9	91.4	78.1	92.1	88.5	85.3	95.9	77.1	95.1	84.7	63.3	75.6
PCAlign+PointGT	86.5↑ (0.7)	84.8	84.9	88.1	81.1	91.2	81.1	92.3	88.7	85.3	96.1	78.3	94.8	85.1	64.0	79.5

Table 6. Comparisons of different deep neural networks with and without improvement of PCAlign in part segmentation task based on ShapeNet.

Method	ins. avg.	air.	bag	cap.	car	cha.	ear.	gui.	kni.	lam.	lap.	mot.	mug	pis.	roc.	ska.
PCT	86.1	83.5	83.4	87.5	80.5	90.3	78.2	92.2	88.1	82.6	96.2	77.5	95.8	85.4	64.6	83.3
PointWOLF ³⁰ +PCT	86.4↑ (0.3)	84.2	84.3	84.4	80.7	90.7	81.4	92.1	88.7	85.9	96.2	78.8	95.9	85.9	64.4	81.4
PointAugment ²³ +PCT	86.3↑ (0.2)	84.1	84.3	84.6	79.5	90.4	81.2	92	87.5	85.3	96.1	78.5	96.1	85.4	64.1	83.5
PCAlign+PCT	86.7↑ (0.6)	84.7	84.5	84.1	79.9	90.5	81.6	89.8	88.9	85.8	96.7	80.2	96.0	86.1	65.3	82.1

Table 7. Comparisons of different data augmentation framework in part segmentation task based on ShapeNet. The backbone is PCT¹⁰.

Samples	1	2	3	4	5	6	sum
Cat.mIoU	82.1	83.4	84.1	84.8	84.6	84.5	83.9
Ins.mIoU	83.7	85.5	86.3	86.7	86.5	86.4	86.8

Table 8. Investigation of different number of copies.

detection, especially for the joints of components. PCAlign achieves better results as a general data augmentation framework.

Evaluation of ablation experiments

To demonstrate the effectiveness of copy settings, we conducted ablation studies by varying the number of copies sampled from generated samples. As shown in Table 8, the performance is worst with 1 copy. At 2 copies, performance improves the most as the voting mechanism takes effect. With 3 copies, performance further improves, reaching its peak at 4 copies. This is because increased votes help correct errors with accurate predictions from other points in the point cloud. However, with 5 and 6 copies, excessive copies introduce more errors, leading to biased predictions and significantly longer training times. Summing over four copies and taking the maximum value is crude and further degrades performance due to incorrect predictions. However, averaging performs better than a single copy, demonstrating this module's effectiveness in our study.

We conducted some experiments, including averaging, summing, and performing convolution operations, with the results presented in Table 9. Both averaging and summing yield similar results, significantly lower than those achieved using our proposed module. These methods are too simplistic to filter out the correct point cloud copies, resembling traditional data augmentation by rotating data, but involve averaging or summing point cloud feature copies. Convolution slightly improves performance over averaging and summing due to the increased use of network parameters, though this improvement is minor. In contrast, our module achieves peak performance by effectively identifying the correct copies among the 4 point cloud copies.

Analysis

Based on the experimental data, the performance improvement brought by PCAlign is significant. By comparing the data in Tables 1 and 3, it can be observed that the random model posture significantly reduces the accuracy

Models	AVG	SUM	CONV	OURS	Cat.mIoU	Ins.mIoU
A	✓				83.5	85.4
B		✓			83.6	85.5
C			✓		83.8	85.9
D				✓	84.8	86.7

Table 9. Ablation study about the multi-channel structure.

of feature learning in traditional deep networks. PCAlign can avoid the loss of feature learning accuracy, especially when the training data has significant pose variations. For point clouds with non-uniform densities, some methods may experience degraded performance. PCAlign can reverse this degradation and make feature learning of non-uniform point clouds more stable. In Table 10, we show the performance fluctuations for different methods based on the experimental data of Tables 1, 2, 3, 4 and 5. It proves that the PCAlign is able to stabilize the performance of various methods and improve their robustness. For part segmentation task, PCAlign provides improvement for most categories based on quantitative analysis in Tables 6 and 7. Compared with other data augmentation frameworks, PCAlign can achieve better performance.

Limitations

PCAlign primarily implements data augmentation through pose alignment, it can achieve significant improvement for training point clouds with random poses, which has been proved in Table 3. However, it cannot improve the classification performance by adding input features such as normal vectors. The experimental data shown in Table 5 illustrate that normal vectors cannot improve the classification accuracy for methods with PCAlign-based data augmentation. The reason is that the maximum probability selection mandatory select a single pose as the output result which is not good to point clouds that belongs to the same category but exhibits different distributions. The supplementary role of the normal vector is diminished precisely due to the mandatory pose selection. Another limitation is that PCAlign removes the random rotation module before training which may lead to performance degradation. Some networks can achieve stronger feature learning capabilities with improved robustness through the analysis of local neighborhoods accompanied by random rotation module. In Table 11, we show the classification accuracy of PointMLP³² with and without PCAlign. Due to the removal of the random rotation module, there has been a certain degradation in performance even the poses of point clouds are aligned.

Conclusion

In this paper, we provide a general data augmentation framework, PCAlign, to achieve the rotation-invariant property for point cloud-based semantic analysis. The PCAlign uses PCA-based alignment to generate aligned copies from the input point cloud. The unpredictable pose is controlled within a smaller range which reduces the influence of random rotations. With the multi-channel structure, PCAlign can achieve more accurate semantic analysis results while keeping strict rotational robustness. It doesn't require complex feature coding and has good compatibility with mainstream backbone networks. In general, PCAlign enhances the training samples by alignment, resulting in improved rotational robustness and feature analysis performance for the connected network. In future work, we will explore a general feature enhancement strategy to improve the performance of PCAlign.

Method/classification	mACC (%)	OA (%)
PointNet	85.2±2.7	88.9±2.5
PCAlign+PointNet	88.3±0.8	90.9±0.5
PointNet++ ²	89.2±1.4	91.6±1.4
PCAlign+PointNet++ ²	90.8±0.2	92.8±0.2
DGCNN ⁷	88.5±1.2	91.7±1.2
PCAlign+DGCNN ⁷	89.5±0.1	92.6±0.1
PCT	89.1±1.2	91.9±1.3
PCAlign+PCT	90.8±0.3	93.3±0.5

Table 10. Performance fluctuations of classification accuracy by different methods. The average values of methods enhanced by PCAlign are larger than original ones. The range of performance fluctuation (\pm) is controlled by PCAlign significantly, which means that the robustness is improved.

Dataset	Method/classification	mACC (%)	OA (%)
Original	PointMLP	91.0	93.2
	PCAlign+PointMLP	90.2(↓ 0.8)	92.5(↓ 0.6)
Random rotations	PointMLP	90.4	93.0
	PCAlign+PointMLP	90.2(↓ 0.2)	92.5(↓ 0.5)
Random resampling	PointMLP	90.8	93.2
	PCAlign+PointMLP	90.1 (↓ 0.7)	92.3(↓ 0.9)
Normal vector enhancement	PointMLP	91.0	93.1
	PCAlign+PointMLP	88.1(↓ 2.9)	91.4(↓ 1.7)

Table 11. Comparisons of PointMLP with and without PCAlign in different test datasets with mentioned variations on ModelNet40.

Data availability

The datasets used during the current study are available from the corresponding author upon reasonable request.

Received: 11 April 2024; Accepted: 5 September 2024

Published online: 12 September 2024

References

1. Qi, C. R., Su, H., Mo, K. & Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660 (2017).
2. Qi, C. R., Yi, L., Su, H. & Guibas, L. J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems* **30** (2017).
3. Ben-Shabat, Y., Lindenbaum, M. & Fischer, A. 3D point cloud classification and segmentation using 3D modified Fisher vector representation for convolutional neural networks. *IEEE Robot. Auto. Lett.* **3**, 3145–3152 (2018).
4. Li, Y. *et al.* Pointcnn: Convolution on x-transformed points. In *Advances in neural information processing systems*, vol. 31 (2018).
5. Thomas, H. *et al.* Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6411–6420 (2019).
6. Wu, W., Qi, Z. & Fuxin, L. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 9621–9630 (2019).
7. Wang, Y. *et al.* Dynamic graph CNN for learning on point clouds. *ACM Transactions On Graphics (TOG)* **38**, 1–12 (2019).
8. Vaswani, A. *et al.* Attention is all you need. In *Advances in neural information processing systems*, vol. 30 (2017).
9. Zhao, H., Jiang, L., Jia, J., Torr, P. H. & Koltun, V. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, 16259–16268 (2021).
10. Guo, M.-H. *et al.* Pct: Point cloud transformer. *Computational Visual Media* **7**, 187–199 (2021).
11. Han, X.-F., Jin, Y.-F., Cheng, H.-X. & Xiao, G.-Q. Dual transformer for point cloud analysis. *IEEE Trans. Multimedia* (2022).
12. Lai, X. *et al.* Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8500–8509 (2022).
13. Hui, L., Yang, H., Cheng, M., Xie, J. & Yang, J. Pyramid point cloud transformer for large-scale place recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6098–6107 (2021).
14. Yu, X. *et al.* 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 (2022).
15. Sun, X., Lian, Z. & Xiao, J. Srinet: Learning strictly rotation-invariant representations for point cloud classification and segmentation. In *Proceedings of the 27th ACM International Conference on Multimedia*, 980–988 (2019).
16. Li, F., Fujiwara, K., Okura, F. & Matsushita, Y. A closer look at rotation-invariant deep point cloud analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16218–16227 (2021).
17. Xiao, A. *et al.* Polarmix: A general data augmentation technique for lidar point clouds. *Adv. Neural Informat. Process. Syst.* **35**, 11035–11048 (2022).
18. Leng, Z. *et al.* Pseudoaugment: Learning to use unlabeled data for data augmentation in point clouds. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI*, 555–572 (2022).
19. Chen, Y. *et al.* 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 (2020).
20. Choi, J., Song, Y. & Kwak, N. Part-aware data augmentation for 3d object detection in point cloud. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3391–3397 (2021).
21. Sheshappanavar, S. V., Singh, V. V. & Kambhamettu, C. Patchaugment: Local neighborhood augmentation in point cloud classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2118–2127 (2021).
22. Mei, G. *et al.* Data augmentation-free unsupervised learning for 3d point cloud understanding. *The 33rd British Machine Vision Conference Proceedings* (2022).
23. Li, R., Li, X., Heng, P.-A. & Fu, C.-W. Pointaugment: an auto-augmentation framework for point cloud classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6378–6387 (2020).
24. Zhang, W., Xu, X., Liu, F., Zhang, L. & Foo, C.-S. On automatic data augmentation for 3d point cloud classification. arXiv preprint [arXiv:2112.06029](https://arxiv.org/abs/2112.06029) (2021).
25. Lv, C., Lin, W. & Zhao, B. Approximate intrinsic voxel structure for point cloud simplification. *IEEE Trans. Image Process.* **30**, 7241–7255 (2021).
26. Lv, C., Lin, W. & Zhao, B. Intrinsic and isotropic resampling for 3d point clouds. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**, 3274–3291 (2022).
27. Lv, C., Lin, W. & Zhao, B. Kss-icp: Point cloud registration based on Kendall shape space. *IEEE Trans. Image Process.* **32**, 1681–1693 (2023).
28. Wu, Z. *et al.* 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1912–1920 (2015).
29. Chang, A. X. *et al.* Shapenet: An information-rich 3d model repository. arXiv preprint [arXiv:1512.03012](https://arxiv.org/abs/1512.03012) (2015).
30. Kim, S. *et al.* Point cloud augmentation with weighted local transformations. In *Proceedings of the IEEE/CVF international conference on computer vision*, 548–557 (2021).
31. Li, Z. *et al.* Geodesic self-attention for 3d point clouds. *Adv. Neural. Inf. Process. Syst.* **35**, 6190–6203 (2022).
32. Ma, X., Qin, C., You, H., Ran, H. & Fu, Y. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. In *International Conference on Learning Representations* (2022).

Acknowledgements

This work was supported in part by a grant from National Nature Science Foundation of China (62102213, 62172247, 62262056); Key R&D and Transformation Plan of Qinghai Province (No.2022-QY-203).

Author contributions

C.Z. and A.L. conceived the experiment(s), D.Z. and C.L. conducted the experiment(s), C.D., D.Z. and A.L. analysed the results. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to D.Z. or C.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2024