PaRot: Patch-Wise Rotation-Invariant Network via Feature Disentanglement and Pose Restoration

Dingxin Zhang*, Jianhui Yu*, Chaoyi Zhang, Weidong Cai
School of Computer Science, University of Sydney, Australia
dzha2344@uni.sydney.edu.au, {jianhui.yu, chaoyi.zhang, tom.cai}@sydney.edu.au

Abstract

Recent interest in point cloud analysis has led rapid progress in designing deep learning methods for 3D models. However, state-of-the-art models are not robust to rotations, which remains an unknown prior to real applications and harms the model performance. In this work, we introduce a novel **Pa**tch-wise **Rot**ation-invariant network (PaRot), which achieves rotation invariance via feature disentanglement and produces consistent predictions for samples with arbitrary rotations. Specifically, we design a siamese training module which disentangles rotation invariance and equivariance from patches defined over different scales, e.g., the local geometry and global shape, via a pair of rotations. However, our disentangled invariant feature loses the intrinsic pose information of each patch. To solve this problem, we propose a rotation-invariant geometric relation to restore the relative pose with equivariant information for patches defined over different scales. Utilising the pose information, we propose a hierarchical module which implements intra-scale and inter-scale feature aggregation for 3D shape learning. Moreover, we introduce a pose-aware feature propagation process with the rotation-invariant relative pose information embedded. Experiments show that our disentanglement module extracts high-quality rotation-robust features and the proposed lightweight model achieves competitive results in rotated 3D object classification and part segmentation tasks.

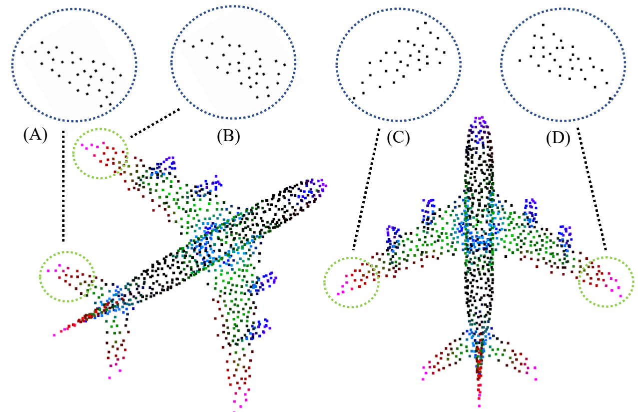

Figure 1: Illustration of pose information loss when generating patch-wise rotation-invariant features. We visualise our learned patch-wise rotation-invariant features of an air plane instance under two different orientations. As it can be seen, for B and C that have exactly same geometric shapes and different poses (i.e. orientation and global position), the same features are generated. However, the features for patches on tail (A), left wing (B, C) and right wing (D) will be very similar. The information for distinguishing their difference between are lost.

Introduction

Point cloud analysis has recently drawn much interest from researchers. As a common form of 3D representations, point clouds are applied in areas such as unmanned driving and 3D face recognition. Recent deep learning models (Qi et al. 2017a,b) show great potential on well aligned point clouds for classification and segmentation. However, 3D objects are normally rotated and orientation angles are unknown in real scenarios, which can largely impact the deep learning models that are sensitive to rotations. Therefore, making the model invariant to rotations becomes an important research topic.

Pioneering work has attempted to obtain rotation robustness by transforming the shape into a canonical pose (Qi et al. 2017a; Jaderberg et al. 2015), which cannot achieve consistent invariance to rotation. Recent works construct rotation-invariant representations from local geometry as model input (Zhang et al. 2019; Kim, Park, and Han 2020), which achieves consistent behavior under random rotations. However, the rotation-invariant representations lose intrinsic pose information (i.e., orientation and position) as illustrated in Fig. 1. To solve this problem, Li et al. (2021b) and Zhao et al. (2022) generate rotation-invariant features from the global 3D shape and restore the global pose information by simply concatenating features at both local and global scales, without fully exploring how the pose information can be used more effectively. Moreover, these methods achieve rotation invariance based on handcrafted features invariant to rotation, which could limit the model performance.

In this work, we borrow the idea of Zhao et al. (2022) where we obtain rotation-invariant information from both local and global scales. Specifically, we capture point patches

*These authors contributed equally.

from the local geometry and global shape. We then apply a feature disentanglement module, where pairs of rotations are introduced to each point patch, to extract rotation-invariant shape content and rotation-equivariant orientation information via a siamese training process. In this way, the invariant features are dynamically generated, which enhances the feature representation. To recover the pose information, we firstly define a geometric relation between two patches by computing the positional and orientational differences. We then propose a **Pa**tch-wise **Rot**ation-invariant network (PaRot), which takes as input the geometric relation and rotation-invariant shape contents encoded from local patches, for an intra-scale aggregation process. In addition, an inter-scale process is considered, which takes the geometric relation and features encoded across local and global patches, for global context exploiting. Moreover, we follow PointNet++ (Qi et al. 2017b) for segmentation by using and modifying the feature propagation module at a minimal cost. Specifically, we propose a pose-aware feature propagation module, where the previously static distance-based feature interpolation is replaced by a learnable process, with encoded geometric relations to preserve the rotation-invariant relative pose information. More importantly, extensive experiments on different benchmarks present the superiority of our method.

The contributions of this work are summarised as follows: (1) We propose a siamese training module by introducing pairs of rotations to disentangle patch-wise learnable high-quality rotation-invariant shape content feature and rotation-equivariant orientation feature; (2) We define a rotation-invariant geometric relation representation to restore relative pose information between patches to guide the inter-scale and intra-scale learning; (3) We design a relative pose-aware feature propagation method for more accurate rotation-invariant segmentation.

Related Work

Deep Learning on Point Clouds. Previous deep learning methods for 3D point clouds capitalise on the advanced development of 2D convolutional neural networks. Recent works directly consume point set data by extracting point-wise feature. PointNet (Qi et al. 2017a) presents a groundbreaking structure, utilising MLPs to learn point-wise spatial features and achieve permutation invariance with max pooling. The following works are extended on the basis of PointNet framework, including learning local context to abstracting geometry information from different scales of patches (Qi et al. 2017b; Zhao et al. 2019; Yu et al. 2021), developing convolution operators for better feature extraction (Thomas et al. 2019; Liu et al. 2019), and improving symmetry functions to promote feature aggregation (Xiang et al. 2021; Chen et al. 2022). However, most methods are rotation-sensitive and their performances degrade drastically when input point clouds are rotated arbitrarily.

Rotation Equivariance. One approach to achieving rotation robustness is to ensure the learned features of all intermediate layers rotate correspondingly with the input. Spherical convolution-based methods (Cohen et al. 2018; Esteves et al. 2018; Rao, Lu, and Zhou 2019) transform point clouds into a spherical harmonic domain and apply spherical convolutions to capture roughly rotation-equivariant features. Tensor field-based networks (Poulenard and Guibas 2021; Zhao et al. 2020; Deng et al. 2021) consume and output tensor field features that maintain strictly rotation-equivariant. To ensure rotation invariance, these methods require an extra operation to transform high-level equivariant features into an invariant form, which will introduce information loss during training. In our work, the equivariant orientation features are employed for restoring relative pose information during hierarchical geometric learning to reduce the information loss.

Rotation Invariance. Another approach focuses on learning rotation-invariant features. A common approach is to transform the Cartesian coordinates of point clouds into a handcrafted rotation-invariant representation in the data preprocessing stage. Zhang et al. (2019), Chen et al. (2019), and Xu et al. (2021) design handcrafted features within local patches. These methods eliminate pose information of patches when generating rotation-invariant features and harm the geometry learning process. To address this issue, Zhang et al. (2020), Li et al. (2021b), and Chen and Cong (2022) take relative pose information into consideration when handcrafting representations. The relative pose between neighbouring patches (Chen and Cong 2022) or between local patches and global shapes (Zhang et al. 2020; Li et al. 2021b; Zhao et al. 2022) is embedded into handcrafted features. In our work, the patch-wise rotation-invariant features are abstracted via neural networks and arbitrary rotations. Meanwhile, pose information is preserved by predicting patch-wise orientations and restored by computing intra- and inter-scale geometric relations. Moreover, we embed geometric relations in feature propagation process to enhance the segmentation performance.

Siamese Training. Siamese training enables feature disentanglement, which is an efficient technique in exploring the data variation and similarity. Recent works employ siamese training in disentangling rotation-equivariant and rotation-invariant features (Sun et al. 2021; Gu et al. 2020; Chen, Yang, and Tao 2022; Sajnani et al. 2022). However, those methods mainly focus on registration and reconstruction tasks, and the extracted equivariant features are used for canonicalization. In contrast, equivariant orientation matrices in our method are employed to construct geometric relations for relative pose restoration, so that we can aggregate rotation-invariant features from intra- and inter-scale learning.

Method

Given a point cloud $\mathbf{P} \in \mathbb{R}^{N\times 3}$, a rotation-robust point cloud model $f(\cdot)$ needs to be invariant to any arbitrary rotation $\mathbf{R} \in \mathbb{R}^{3\times 3}$ applied to $\mathbf{P}$ and produces consistent predictions: $f(\mathbf{P}) = f(\mathbf{PR})$. A Patch-wise Rotation-invariant network (PaRot) is introduced to achieve this goal. We first disentangle patch-wise rotation invariance and equivariance from shape descriptors. Comprehensive geometric representations with rotation invariance are extracted intra-scale and inter-scale, with geometric relations embedded to preserve

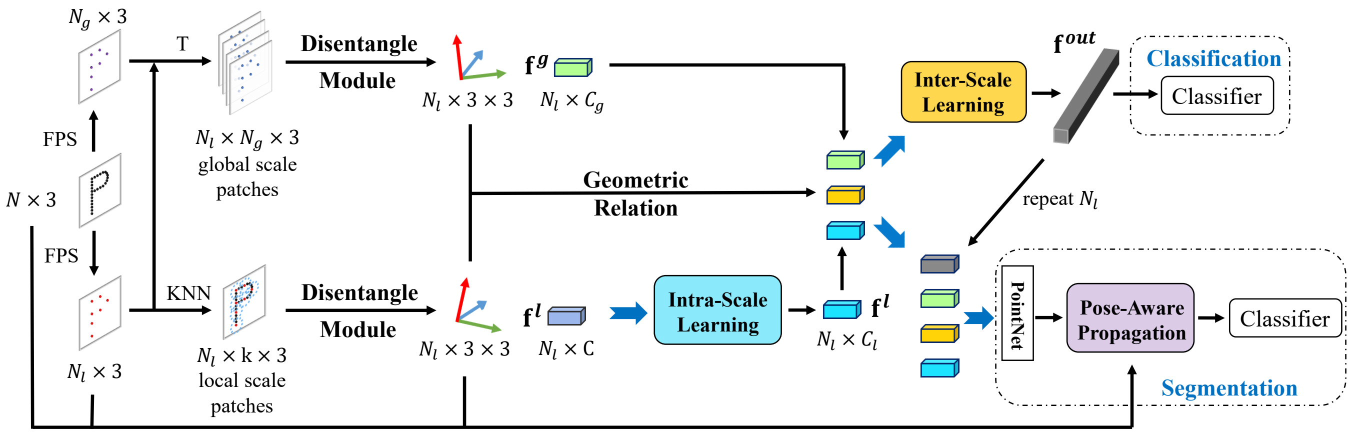

Figure 2: The overall architecture of PaRot for 3D classification and segmentation, where KNN and T denote k nearest neighboring and translation, respectively. We generate local scale patches and global scale patches with FPS, KNN, and translation operations. Then we disentangle patches of different scales into shape content information and orientation information separately. The learned orientations are utilised to determine geometric relations for restoring patch-wise relative pose and guiding intra- and inter-scale invariant features learning and pose-aware feature propagation.

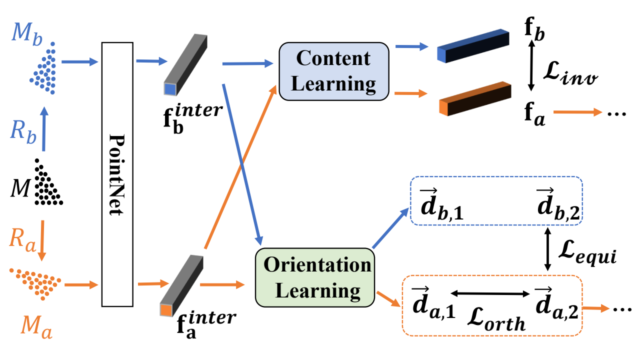

Figure 3: Frameworks of disentanglement module. The input patch is randomly rotated twice and then are independently fed through into the module as branch a and b. The output of branch a is hierarchically encoded while that of branch b is used to assist learning.

pose relations between different patches. Finally, we propose a rotation-invariant feature propagation module with geometric relations to maintain the rotation invariance for semantic point labelling.

Rotation Invariance and Equivariance Disentanglement

Inspired by (Sun et al. 2021), we propose a disentanglement module based on the siamese training pipeline, which decomposes latent shape descriptors into rotation-invariant shape contents for rotation invariance study and rotation-equivariant shape orientations to preserve pose information.

Rotation-Invariant Content Learning. We discuss that for any 3D object under rotations, the shape content remains invariant to random rotations, and we extract such rotation-invariant content features in a patch-wise manner. Particularly, as shown in Fig. 3, we introduce a pair of arbitrary rotations $\mathbf{R}_a$ and $\mathbf{R}_b$ to input point patch $\mathbf{M} \in \mathbb{R}^{n\times3}$, leading to two randomly rotated patches $\mathbf{M}_a$ and $\mathbf{M}_b$. A light-weight PointNet network (Qi et al. 2017a) is employed and shared between $\mathbf{M}_a$ and $\mathbf{M}_b$ to encode the geometric information, leading to two intermediate shape descriptors $\mathbf{f}_a^{inter}$ and $\mathbf{f}_b^{inter}$. Multi-layer perceptrons (MLPs) are thus applied to disentangle rotation-invariant shape contents $\mathbf{f}_a$ and $\mathbf{f}_b$ from latent shape descriptors, and rotation invariance is achieved via minimizing the feature distance between $\mathbf{f}_a$ and $\mathbf{f}_b$. Hence, we define a *rotation-invariant* loss function $\mathcal{L}_{inv}$ to enforce a high degree of similarity between these two features under rotations, which is represented as:

$$\mathcal{L}_{inv} = \|\mathbf{f}_a - \mathbf{f}_b\|_2^2 , \tag{1}$$

where $\|\cdot\|_2^2$ denotes the L2 Norm.

Rotation-Equivariant Orientation Learning. An obvious intuition we discuss here is that for any 3D patch, there exists an intrinsic orientation, essential for recovering the pose and denoted by a rotation matrix, which is naturally equivariant to rotations. To this end, we borrow the idea from FS-Net (Chen et al. 2021) to extract direction vectors from latent shape descriptors. Specifically, as shown in Fig. 4, two perpendicular direction vectors $\vec{\mathbf{d}}_1$ and $\vec{\mathbf{d}}_2$ are predicted for each branch in terms of the latent shape descriptors $\mathbf{F}$, where we drop subscripts (i.e., a and b) indexing the siamese branch when same operations are applied to both branches. To ensure the equivariance, we introduce a *rotation-equivariant* loss function $\mathcal{L}_{equi}$ between the learned direction vectors as follows:

$$\mathcal{L}_{equi} = \left\|\vec{\mathbf{d}}_{a,1} - \mathbf{R}_b^{-1}\mathbf{R}_a\vec{\mathbf{d}}_{b,1}\right\|_2^2 + \left\|\vec{\mathbf{d}}_{a,2} - \mathbf{R}_b^{-1}\mathbf{R}_a\vec{\mathbf{d}}_{b,2}\right\|_2^2 , \tag{2}$$

where $\mathbf{R}_b^{-1}\mathbf{R}_a$ is a rotation matrix transforming $\mathbf{M}_a$ to $\mathbf{M}_b$. In this way, the direction vectors are enforced to contain only orientation information of the local patch, which are later

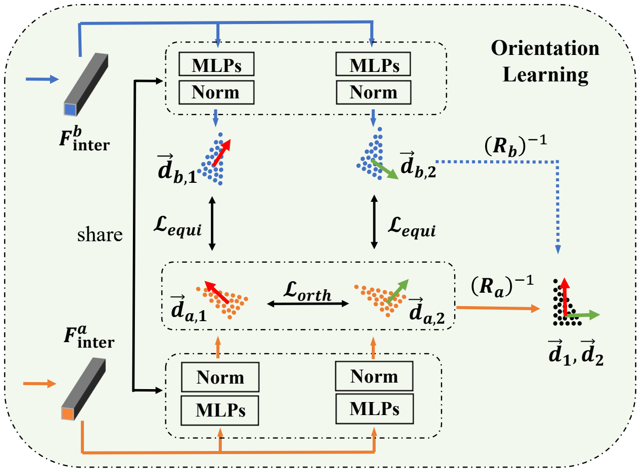

Figure 4: Illustrations of rotation-equivariant orientation learning. The output orientation of branch a is rotated back by $\mathbf{R}_a^{-1}$ which is used as the predicted orientation of the original patch $\mathbf{M}$.

used for pose information embedding. Furthermore, to preserve the orthogonality between $\vec{\mathbf{d}}_1$ and $\vec{\mathbf{d}}_2$ as the column vectors of a rotation matrix, we design a loss function $\mathcal{L}_{orth}$ represented as follows:

$$\mathcal{L}_{orth} = \left\| \vec{\mathbf{d}}_1^\top \vec{\mathbf{d}}_2 \right\|_2^2 . \tag{3}$$

One thing needs to be mentioned is that our orientation matrix is learned based on the random rotated pose of $\mathbf{M}$, which is different from that of the initial input $\mathbf{M}$. As $\mathbf{R}_a$ and $\mathbf{R}_b$ are only introduced for disentanglement, we need to remove their impacts and obtain the initial orientation $\mathbf{O}$ of $\mathbf{M}$ by transforming predicted direction vectors back as: $[\vec{\mathbf{d}}_1, \vec{\mathbf{d}}_2] = \mathbf{R}_a^{-1}[\vec{\mathbf{d}}_{a,1}, \vec{\mathbf{d}}_{a,2}] = \mathbf{R}_b^{-1}[\vec{\mathbf{d}}_{b,1}, \vec{\mathbf{d}}_{b,2}]$.

As $\vec{\mathbf{d}}_1$ and $\vec{\mathbf{d}}_2$ are ensured to be non-parallel, we can simply define a third direction vector $\vec{\mathbf{d}}_3 = \vec{\mathbf{d}}_1 \times \vec{\mathbf{d}}_2$ which is orthogonal to both $\vec{\mathbf{d}}_1$ and $\vec{\mathbf{d}}_2$. We hence denote the orientation matrix learned for initial input $\mathbf{O}$ as the concatenation of direction vectors: $\mathbf{O} = [\vec{\mathbf{d}}_1, \vec{\mathbf{d}}_2, \vec{\mathbf{d}}_3]$, where $[\cdot]$ is concatenation. Finally, as all learnable MLPs are shared across both branches, our learned rotation-invariant shape contents and orientations are the same. Following the conventional implementation of the siamese training procedure (Sun et al. 2021), we utilise the outputs of branch a for the later model inference and only calculate $\mathcal{L}_{orth}$ for branch a.

Intra- and Inter-Scale Rotation Invariance Learning with Geometric Relations

For rotation invariance learning, we propose a geometric relation embedding network to aggregate rotation-invariant features from intra- and inter-scale learning, where geometric relations between patches are computed with patch-wise orientation matrices and global absolute positions to recover relative pose information.

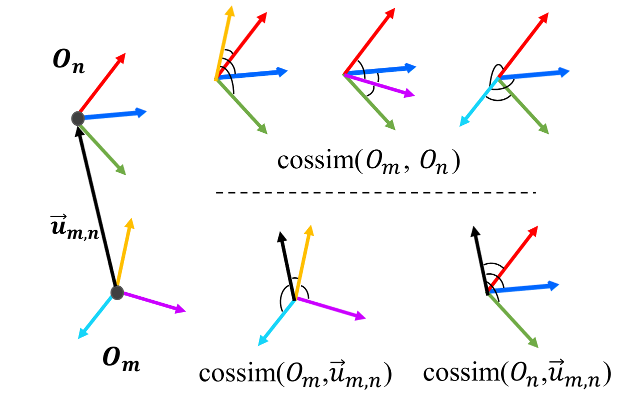

Figure 5: Angles used in geometric relation representation. For two patches $\mathbf{M}_m$ and $\mathbf{M}_n$, we use different colors to illustrate different vectors in their orientation matrices $\mathbf{O}_m$ and $\mathbf{O}_n$.

Geometric Relation Representation. The pose information of patches is critical for geometric feature learning, but it cannot be encoded into a rotation-invariant form, since the orientations and global positions are equivariant to rotations. To maintain such pose information, we have to consider geometric properties between different patches that are invariant to rotations, such that the learned rotation invariance can be maintained. To achieve this goal, we adopt relative geometric relations, i.e., relative angles and distances between patches, to retain the pose information of the original 3D shape during feature learning. Specifically, given two patches $\mathbf{M}_m$ and $\mathbf{M}_n$ with corresponding reference points $\mathbf{p}_m$ and $\mathbf{p}_n$, their orientation matrices $\mathbf{O}_m = [\vec{\mathbf{d}}_1^m, \vec{\mathbf{d}}_2^m, \vec{\mathbf{d}}_3^m]$ and $\mathbf{O}_n = [\vec{\mathbf{d}}_1^n, \vec{\mathbf{d}}_2^n, \vec{\mathbf{d}}_3^n]$ can be learned according to our disentanglement module. Hence, the geometric relation $\mathcal{G}(\mathbf{M}_m, \mathbf{M}_n)$ between the two patches is defined as:

$$\begin{aligned}\mathcal{G}(\mathbf{M}_m, \mathbf{M}_n) = &[\mathrm{dist}(\mathbf{p}_m, \mathbf{p}_n), \mathrm{cossim}(\mathbf{O}_m, \mathbf{O}_n), \\ &\mathrm{cossim}(\mathbf{O}_m, [\vec{\mathbf{u}}_{mn}]), \\ &\mathrm{cossim}(\mathbf{O}_n, [\vec{\mathbf{u}}_{mn}])],\end{aligned} \tag{4}$$

where $\mathrm{dist}(\cdot)$ calculates the Euclidean distance between two points, $\mathrm{cossim}(\cdot)$ computes the cosine similarity between all column vectors of first matrix and all columns vectors of second matrix and $\vec{\mathbf{u}}_{mn} = \mathbf{p}_n - \mathbf{p}_m$ is the vector between the reference points of two patches, pointing from $\mathbf{p}_n$ to $\mathbf{p}_m$. As shown in Fig. 5, $\mathcal{G}(\mathbf{M}_m, \mathbf{M}_n)$ consists of 1 distance parameter, and $9 + 3 + 3 = 15$ angles parameters.

Local-Scale Feature Disentangling. To generate local patches, we sample N_ℓ points using farthest point sampling from the original point set $\mathbf{P}$ as a local query point set $\mathbf{Q} \in \mathbb{R}^{N_\ell \times 3}$. For any point $\mathbf{q}_i \in \mathbf{Q}$ taken as the reference point, k-NN search is performed on $\mathbf{P}$, resulting in a total of N_ℓ patches. Then, we generate a local reference frame for each patch and implement patch-wise translations to ensure the reference point $\mathbf{q}_i$ of patch $\mathbf{M}_i^\ell$ coincide with the refer-

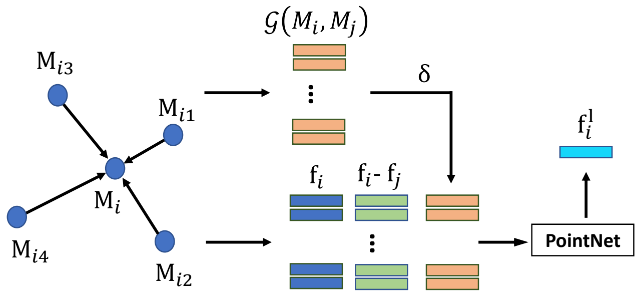

Figure 6: Illustration of the geometric relation embedded intra-scale edge convolution.

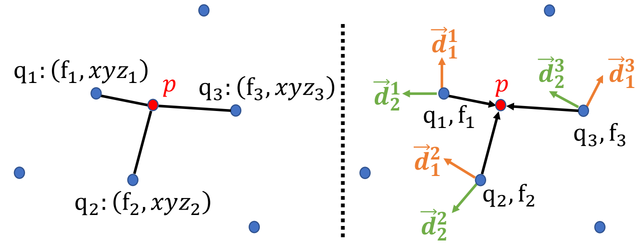

Figure 7: Left: typical feature propagation, the distance based interpolation. Right: relative pose-aware feature propagation process. The third direction $\vec{\mathbf{d}}_3$ of each patch is not shown for simplicity.

ence frame origin point:

$$\mathbf{M}_i^\ell = [q_j - q_i]_{j:j\in\mathcal{N}_{\mathbf{P}}(i)}, \tag{5}$$

where $\mathcal{N}_{\mathbf{P}}(i)$ denotes the neighboring points of $\mathbf{q}_i$ in $\mathbf{P}$. We then apply shape feature disentanglement to extract rotation-invariant content features $\mathbf{f}_i$ and rotation-equivariant orientation matrices $\mathbf{O}_i$ in a patch-wise manner.

Intra-Scale Relative Pose Aware Feature Learning. To capture the geometry of patches within a larger receptive field, we apply the edge convolution (Wang et al. 2019) to further encode $\mathbf{f}_i$ to refine rotation-invariant features. However, as mentioned before, the disentangled rotation-invariant shape feature does not contain pose information, performing convolution only on shape content features will lead to geometric information loss. As shown in Fig. 6, we address this issue by proposing a geometric relation encoding operation with a linear mapping function $\delta(\cdot)$ and embed the encoding into convolution, aggregating a local geometry feature $\mathbf{f}_i^\ell$ as follows:

$$\mathbf{f}_i^\ell = \mathcal{A}_{j:j\in\mathcal{N}(i)}\Big(\text{MLP}\left([\mathbf{f}_j, \mathbf{f}_j - \mathbf{f}_i, \delta(\mathcal{G}(\mathbf{M}_i, \mathbf{M}_j))]\right)\Big), \tag{6}$$

where $\mathcal{A}(\cdot)$ is the max-pooling. In this way, we explicitly embed the relative pose information between different patches into the rotation invariance learning process, leading to a more distinct feature representation.

Global-Scale Feature Disentangling. For global scale feature disentangling, we downsample $\mathbf{P}$ into a new point set $\mathbf{G} \in \mathbb{R}^{N_g\times 3}$ with N_g points. We further use local reference points $\mathbf{q}_i \in \mathbf{Q}$ to generate N_ℓ corresponding global patches. Each global patch $\mathbf{M}_i^g$ will contain all points of $\mathbf{G}$ and the same translation operation in Eq. 5 will be implemented to center the reference point and generate a reference frame. In such a way, we integrate global context information and the corresponding local patch positional information into each global patch. Then we disentangle $\mathbf{M}_i^g$ into global-scale rotation-invariant features $\mathbf{f}_i^g$ and orientation matrices $\mathbf{O}_i^g$, which includes the shape information of global patches and the positional information of reference points. This information can be used in enhancing the feature distinction of rotation-invariant representations.

Besides, it is worth mentioning that there is no need to implement intra-scale learning for global scale patches, since the global patches cover all points, preserving the global context with the input shape.

Inter-Scale Rotation-Invariant Learning. As local and global branches utilise the same downsampled point set $\mathbf{Q}$ as the reference point set, we discuss that local-scale and global-scale features $\mathbf{f}_i^\ell$ and $\mathbf{f}_i^g$ can be fused to combine local geometry and global context information. To this end, we adopt geometric relations between $\mathbf{M}_i^\ell$ and $\mathbf{M}_i^g$ along with $\mathbf{f}_i^\ell$ and $\mathbf{f}_i^g$ as module inputs and directly output a relative pose-aware fused feature:

$$\mathbf{f}_i^{out} = \mathcal{A}_{i:i\in\mathbf{Q}}\Big(\text{MLP}\left([\mathbf{f}_i^\ell, \mathbf{f}_i^g, \delta(\mathcal{G}(\mathbf{M}_i^\ell, \mathbf{M}_i^g))]\right)\Big). \tag{7}$$

When calculating the geometric relation $\mathcal{G}(\mathbf{M}_i^\ell, \mathbf{M}_i^g)$, we define the origin $\vec{\mathbf{0}}$ as reference points of global patches, otherwise reference points of $\mathbf{M}_i^\ell$ and $\mathbf{M}_i^g$ will be the same.

Rotation-Invariant Pose-Aware Feature Propagation Module

Unlike the typical method proposed in PointNet++ (Qi et al. 2017b) (see Fig. 7 (left)), which propagates features from subsampled points to the original points based on point distances and absolute point positions over k nearest neighbors, we propose a pose-aware feature propagation module, which dismisses distance-based interpolation and utilises relative geometric relations for pose information embedding. As shown in Fig. 7 (right), given a point $\mathbf{q}_j \in \mathbf{Q}$ with feature $\mathbf{f}_j^g$ and a point $\mathbf{p}_i \in \mathbf{P}$ with feature $\mathbf{f}_i^p$, we follow Section to embed the feature propagation module with relative geometric relations between $\mathbf{q}_j$ and $\mathbf{p}_i$. Specifically, a direction vector $\vec{\mathbf{u}}_{ij:j\in\mathcal{N}_{\mathbf{Q}}(i)} = \mathbf{q}_j - \mathbf{p}_i$ is defined, pointing from $\mathbf{p}_i$ to $\mathbf{q}_j$, where $\mathcal{N}_{\mathbf{Q}}(i)$ is the neighborhood of $\mathbf{p}_i$ in $\mathbf{Q}$. As $\mathbf{q}_j$ is associated with a rotation-equivariant orientation matrix $\mathbf{O}_j$, we thus define a geometric relation between $\mathbf{p}_i$ and its neighbors $\mathbf{q}_j$ in the same way as Eq. 4:

$$\mathcal{G}(\mathbf{p}_i, \mathbf{q}_j) = \left[\text{dist}(\mathbf{p}_i, \mathbf{q}_j), \text{cossim}(\mathbf{O}_j, [\vec{\mathbf{u}}_{ij}]_{j\in\mathcal{N}_{\mathbf{Q}}(i)})\right]. \tag{8}$$

The geometric relation $\mathcal{G}(\mathbf{p}_i, \mathbf{q}_j)$ is then encoded by $\delta(\cdot)$ and concatenated with neighboring point features $\mathbf{f}_j^g$ and skip-linked point features $\mathbf{f}_i^p$ from the original points. The fused

Rotation-Sensitive	input	z/z	z/SO3	SO3/SO3
PointNet (2017a)	pc	85.9	19.6	74.7
PointNet++ (2017b)	pc	89.3	28.6	85.0
PointNet++ (2017b)	pc+n	91.8	18.4	77.4
DGCNN (2019)	pc	**92.2**	20.6	81.1
Rotation-Robust	input	z/z	z/SO3	SO3/SO3
Spherical CNN (2018)	Voxel	88.9	76.9	86.9
SFCNN (2019)	pc	91.4	84.8	90.1
RI-Conv (2019)	pc	86.5	86.4	86.4
ClusterNet (2019)	pc	87.1	87.1	87.1
RI-GCN (2020)	pc	89.5	89.5	89.5
GCANet (2020)	pc	89.0	89.1	89.2
RIF (2021b)	pc	89.4	89.4	89.3
SGMNet (2021)	pc	90.0	90.0	90.0
TFN (2021)	pc	87.6	87.6	87.6
Li et al. (2021a)	pc	90.2	90.2	90.2
VN-DGCNN (2021)	pc	89.5	89.5	90.2
OrientedMP (2022)	pc	88.4	88.4	88.9
ELGANet (2022)	pc	90.3	90.3	90.3
PaRot	pc	90.9	**91.0**	**90.8**

Table 1: Classification performance on ModelNet40. "pc" and "n" denote the input datatype of raw 3D coordinates and normal, respectively.

features are passed through learnable MLPs and further aggregated by summation to update the original feature $\mathbf{f}_i^p$. The whole process can be formulated as follows:

$$\mathbf{f}_i = \sum_{j=1}^{k} \mathrm{MLP}\left([\mathbf{f}_i^p, \mathbf{f}_j^q, \delta(\mathcal{G}(\mathbf{p}_i, \mathbf{q}_j))]\right). \tag{9}$$

We can see from Eq. 9 that, unlike PointNet++, we ignore the rotation-sensitive global xyz positions of point $\mathbf{p}_i$ during the propagation process, which ensures our module to be rotation-invariant. Besides, the addition introduction of the neighboring point feature $\mathbf{f}_j^q$ further improves the feature representations.

Experiments

We evaluate our model with 3D point cloud classification and segmentation tasks, analyse the rotation robustness, compare the performance with other representative methods, and visualise the experimental results. Furthermore, we analyse the efficiency of the proposed modules and the complexity of our model. We follow the evaluation protocols of (Esteves et al. 2018): we randomly rotate training and testing objects around z-axis (z/z), rotate training objects around z-axis while implement arbitrary rotations on testing objects (z/SO3), apply arbitrary rotations to both training and testing data (SO3/SO3).

For classification, we set N_ℓ to 256 and N_g to 32. When utilising k-NN for generating local-scale patches and searching neighbours for intra-learning, we assign the number of local-scale patches k_ℓ to 64 and the number of neighbours for intra-scale learning k_{intra} to 32. Settings for segmentation are same, except that N_g and k_{intra} are changed to 64 and 16, respectively.

Method	z/SO3	SO3/SO3	Δ
PointNet (2017a)	16.7	54.7	38.0
PointNet++ (2017b)	15.0	47.4	32.4
DGCNN (2019)	17.7	71.8	54.1
RI-Conv (2019)	78.4	78.1	0.3
RI-GCN (2020)	80.5	80.6	0.1
RIF (2021b)	79.8	79.9	0.1
Li et al.* (2021a)	79.3	79.6	0.3
VN-DGCNN* (2021)	77.8	76.0	1.8
OrientedMP* (2022)	76.7	77.2	0.6
PaRot	**82.1**	**82.6**	0.5

Table 2: Classification accuracy on ScanObjectNN OBJ_BG dataset under z/SO3 and SO3/SO3. Δ denotes the absolute difference between z/SO3 and SO3/SO3. $\star$ indicates our reproduced results based on official implementations.

Shape Classification

We test the classification ability of our model on the synthetic dataset ModelNet40 (Wu et al. 2015) and the real-world dataset ScanObjectNN (Uy et al. 2019). ModelNet40 is the most commonly used dataset in point cloud analysis, which contains 12,311 pre-aligned point cloud shapes sampled from 40 categories of CAD models. In the official version, the dataset is split into 9,843 training samples and 2,468 testing samples. ScanObjectNN contains 15000 incomplete objects scanned from 2,902 real-world objects. To better explore the robustness to noise of scanned samples, we use the OBJ_BG subset which contains background noise for evaluation. We sample 1024 points from each sample as our model inputs.

In our training procedure, we introduce random rotations to each patch in the disentanglement module. During testing stage, no patch-wise rotation is applied, since disentangled features of our final model are rotation-robust, the performance will not be effected by patch-wise rotations.

We compare our model with representative models in terms of classification accuracy reported in Tables 1 and 2 for ModelNet40 and ScanObjectNN respectively. It is shown that our method achieves promising results in all three settings with a high rotation-robustness as the absolute accuracy difference between z/z, z/SO3 and SO3/SO3 is not greater than 0.5%.

Part Segmentation

For shape part segmentation task, we validate our model on the ShapeNetPart dataset (Yi et al. 2016), which contains 16,880 synthetic samples with 14,006 training and 2,874 testing data. The dataset includes 16 object categories and each category is annotated with 2 to 6 parts result in totally 50 part annotation labels. 2048 points are sampled as model input. The per-class mean intersection of union (mIoU) and the averaged mIoU of 16 classes under z/SO3 are reported and compared with other approaches in Table 3. In addition, we visualise the segmentation results of different objects under different orientation in Fig. 8. It is obvious that our model is robust in segmenting point clouds under arbitrary rotations.

Method	C.mIoU	aero	bag	cap	car	chair	earph.	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate	table
PointNet (2017a)	37.8	40.4	48.1	46.3	24.5	45.1	39.4	29.2	42.6	52.7	36.7	21.2	55.0	29.7	26.6	32.1	35.8
PointNet++ (2017b)	48.3	51.3	66.0	50.8	25.2	66.7	27.7	29.7	65.6	59.7	70.1	17.2	67.3	49.9	23.4	43.8	57.6
DGCNN (2019)	37.4	37.0	50.2	38.5	24.1	43.9	32.3	23.7	48.6	54.8	28.7	17.8	74.4	25.2	24.1	43.1	32.3
RI-Conv (2019)	75.3	80.6	80.0	70.8	68.8	86.8	70.3	87.3	84.7	77.8	80.6	57.4	91.2	71.5	52.3	66.5	78.4
GCANet (2020)	77.2	80.9	**82.6**	81.0	70.2	88.4	70.6	87.1	**87.2**	**81.8**	78.9	58.7	91.0	77.9	52.3	66.8	80.3
TFN (2021)	76.7	80.9	75.2	81.9	73.8	89.0	61.0	90.8	83.0	76.9	80.2	58.5	92.8	76.3	54.0	**74.5**	79.1
Li et al. (2021a)	74.1	81.9	58.2	77.0	71.8	**89.6**	64.2	89.1	85.9	80.7	**84.7**	46.8	89.1	73.2	45.6	66.5	**81.0**
VN-DGCNN* (2021)	75.3	81.1	74.8	72.9	73.8	87.8	55.9	**91.4**	83.8	80.2	84.4	44.5	92.8	74.6	**57.2**	70.2	78.9
PaRot	**79.2**	**82.7**	79.2	**82.3**	**75.3**	89.4	**73.9**	91.1	85.6	81.0	79.5	**65.3**	**93.9**	**79.2**	55.0	72.4	79.5

Table 3: Segmentation per class results and averaged class mIoU on ShapeNetPart dataset under z/SO3, where C.mIoU stands for averaged mIoU of 16 classes.

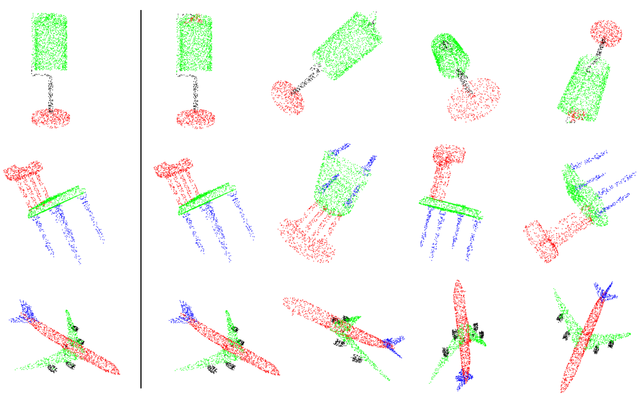

Figure 8: Visualisation of ShapeNet part segmentation result. Left most column is ground truth of a selected orientation and the second left most is the segmentation result of the model with the select orientation. The rest of columns are results of random rotated samples.

Model	Local	Intra-	Inter-	Pose Rest.		z/SO3
				orien.	position	
A	✓					84.3
B	✓	✓				87.4
C	✓		✓			70.9
D	✓	✓	✓			58.6
E	✓		✓	✓	✓	89.6
F	✓	✓		✓	✓	89.5
G	✓	✓	✓	✓		90.4
H	✓	✓	✓		✓	90.5
I	✓	✓	✓	✓	✓	91.0

Table 4: Component study results on ModelNet40 under z/SO3.

Method Analysis

Component Study. We implement an ablation study to explore the effectiveness of each module in the proposed model and report results in Table 4. We also split the geometric representation $\mathcal{G}(\mathbf{M}_m, \mathbf{M}_n)$ into two parts: $\mathrm{cossim}(\mathbf{O}_m, \mathbf{O}_n)$ and $[\mathrm{dist}(\mathbf{p}_m, \mathbf{p}_n), \mathrm{cossim}(\mathbf{O}_m, [\vec{\mathbf{u}}_{mn}]), \mathrm{cossim}(\mathbf{O}_n, [\vec{\mathbf{u}}_{mn}])]$ to study the feature used for pose restoration. Here, $\mathrm{cossim}(\mathbf{O}_m, \mathbf{O}_n)$ only contains relative orientation information and lacks positional information, while $[\mathrm{dist}(\mathbf{p}_m, \mathbf{p}_n), \mathrm{cossim}(\mathbf{O}_m, [\vec{\mathbf{u}}_{mn}]), \mathrm{cossim}(\mathbf{O}_n, [\vec{\mathbf{u}}_{mn}])]$ can detect relative position but has ambiguities for rotations around the vector $\vec{\mathbf{u}}_{mn}$.

Model A is our baseline that only extracts local-scale rotation-invariant shape content feature for classification. When we employ intra-scale learning module (A→B) without relative pose restoration operation, our model will suffer from pose information loss problem, however, it could aggregate features from a larger scale of patches and slightly improve the accuracy. This model has similar principle and performance of RI-Conv and ClusterNet. The inter-scale learning module is designed for global context exploiting. However, when adding inter-scale learning module without employing geometric relation representation (A→C), the model performance drops, as the network cannot discover the relationship between the patch-wise features and the global context. When using two learning modules (B→D), the drop of performance is larger than model A→C, since B is deeper than A and is more vulnerable to the noisy feature added at deep layers. The pose restoration strategy can preserve pose information, guide the learning process, and consistently improve the performance (B→F, C→E, D→I).

Besides, models G, H and I show that both of two parts of geometric representation can restore part of the pose information and improve the accuracy, while the full representation will restore more information and achieve the best performance.

Feature Propagation Analysis. To justify the proposed pose-aware geometric relation embedded feature propagation method, we compare the performance in terms of the mIoU over all instances (I.mIoU) and averaged mIoU of classes (C.mIoU) with the typical interpolation strategy introduced by PointNet++ (Qi et al. 2017b) in z/SO3 ShapeNet part segmentation task. The xyz coordinate positions embedded in PointNet++ are removed, otherwise it will break the rotation invariance. As shown in Table 5, our proposed propagation method outperforms interpolation strategy. Besides, the interpolation strategy weighted average the feature of nearest patches with respect to the inverse distance, patches with long distance contribute little in generating the new feature and increasing the number of neighbours cannot improve the performance. In our method, the geometric

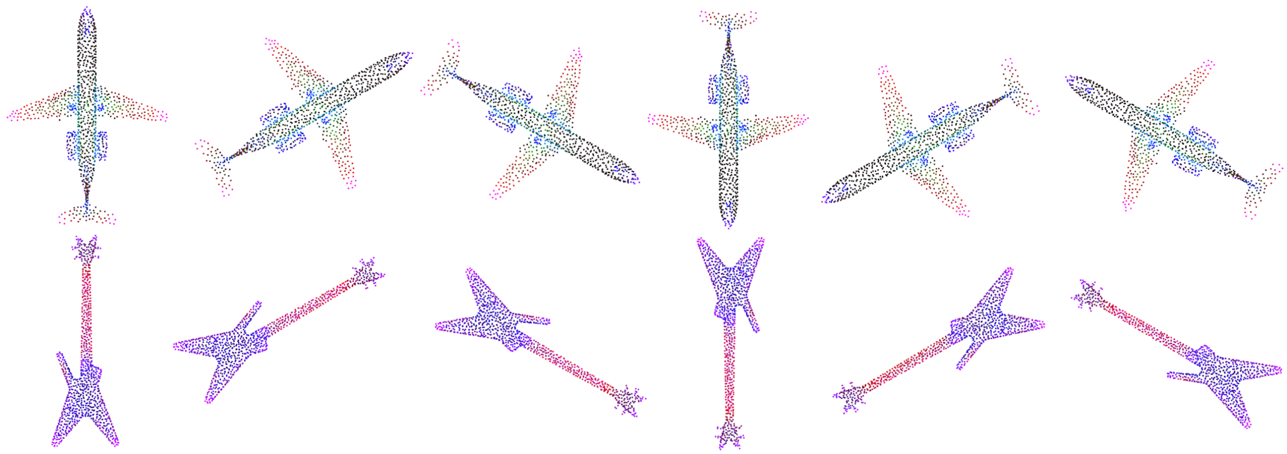

Figure 9: Visualisation of disentangled rotation-invariant features. From left to right, each sample is rotated 60° around z-axis.

	Interpolation		Pose aware	
# of neighbours	I.mIoU	C.mIoU	I.mIoU	C.mIoU
3	81.1	76.6	82.1	77.6
5	80.9	76.7	82.5	78.3
7	80.8	76.8	82.8	78.1
9	80.8	76.7	83.0	79.1
11	80.8	76.7	82.9	**79.2**
13	80.6	76.4	**83.2**	79.0

Table 5: Propagation analysis. Results of Instance mIoU and Class averaged mIoU of z/SO3 part segmentation on ShapeNetPart. The left most column determines the number of neighbour patches used for feature propagation.

relation between the target point and neighbour patches will be encoded individually, thus the performance can be further enhanced by increasing the number of neighbouring queries.

Model Complexity. Benefiting from the efficient hierarchical structure and geometric relation encoding technique, our rotation-robust model achieves high performance with a small model size and low computational cost. We avoid the calculation of dynamic graphs (Wang et al. 2019) and balance the width and depth of the network during developing procedure. Moreover, siamese procedure is not necessary during testing, thus it could be removed to further reduce the computational complexity. As shown in Table 6, the proposed model improves the accuracy by 1.5% with only 55% parameters and 45% FLOPs compared to VN-DGCNN.

Disentangled Rotation-Invariant Feature Visualisation. To examine the effectiveness of our disentanglement module in extracting consistent features for patches under arbitrary orientation, we visualise the feature responses of two different objects (i.e., aeroplane and guitar), which are rotated around the z-axis. Specifically, three channels of the disentangled rotation-invariant features which are extracted from the local branch are selected and taken as the RGB channel for visualisation. We generate 1024 local-scale patches with all points in P as reference points and disentangle with

Method	Para.	FLOPs	Acc.
PointNet++ (2017b)	1.41M	863M	85.0
DGCNN (2019)	1.72M	2449M	81.1
RI-GCN (2020)	4.19M	1237M	89.5
RIF (2021b)	2.36M	6535M	89.4
Li et al. (2021a)	2.91M	3747M	90.2
VN-DGCNN (2021)	2.77M	3183M	89.5
PaRot-training	1.55M	2091M	91.0
PaRot-testing	1.55M	1431M	91.0

Table 6: Comparisons of model size, computational complexity, and z/SO3 accuracy on ModelNet40. The *testing* version removes unnecessary auxiliary network modules.

saved best segmentation model. The result in Fig. 9 indicates that the features learned are various for patches with different shape content while invariant to rotations.

Conclusion

In this work, we propose PaRot which is a novel rotation-invariant learning model for 3D point cloud recognition. Given a point cloud, we disentangle rotation-invariant shape content features and rotation-equivariant orientations for local-scale patches and global-scale patches by introducing pairs of rotations under a siamese training procedure. To restore the rotation-sensitive pose information while maintaining the rotation invariance of learning, we compute geometric relations with patch-wise orientation matrices to represent the relative pose between patches. The geometric relations are utilised to guide the intra-scale and inter-scale feature aggregation. Following the same idea of restoring pose information with geometric relations, we further design a rotation-invariant feature propagation method which improves the segmentation accuracy of our model. Extensive experiments demonstrate the effectiveness and efficiency of our model.

References

Chen, C.; Li, G.; Xu, R.; Chen, T.; Wang, M.; and Lin, L. 2019. ClusterNet: Deep Hierarchical Cluster Network With Rigorously Rotation-Invariant Representation for Point Cloud Analysis. In *CVPR*.

Chen, J.; Kakillioglu, B.; Ren, H.; and Velipasalar, S. 2022. Why Discard if You Can Recycle?: A Recycling Max Pooling Module for 3D Point Cloud Analysis. In *CVPR*.

Chen, R.; and Cong, Y. 2022. The Devil is in the Pose: Ambiguity-free 3D Rotation-invariant Learning via Pose-aware Convolution. In *CVPR*.

Chen, W.; Jia, X.; Chang, H. J.; Duan, J.; Shen, L.; and Leonardis, A. 2021. FS-Net: Fast Shape-Based Network for Category-Level 6D Object Pose Estimation With Decoupled Rotation Mechanism. In *CVPR*.

Chen, Z.; Yang, F.; and Tao, W. 2022. DetarNet: Decoupling Translation and Rotation by Siamese Network for Point Cloud Registration. In *AAAI*.

Cohen, T. S.; Geiger, M.; Köhler, J.; and Welling, M. 2018. Spherical CNNs. In *ICLR*.

Deng, C.; Litany, O.; Duan, Y.; Poulenard, A.; Tagliasacchi, A.; and Guibas, L. J. 2021. Vector Neurons: A General Framework for SO(3)-Equivariant Networks. In *ICCV*.

Esteves, C.; Allen-Blanchette, C.; Makadia, A.; and Daniilidis, K. 2018. Learning SO(3) Equivariant Representations with Spherical CNNs. In *ECCV*.

Gu, J.; Ma, W.; Manivasagam, S.; Zeng, W.; Wang, Z.; Xiong, Y.; Su, H.; and Urtasun, R. 2020. Weakly-Supervised 3D Shape Completion in the Wild. In *ECCV*.

Gu, R.; Wu, Q.; Li, Y.; Kang, W.; Ng, W.; and Wang, Z. 2022. Enhanced Local and Global Learning for Rotation-invariant Point Cloud representation. In *MultiMedia*.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; et al. 2015. Spatial Transformer Networks. In *NeurIPS*.

Kim, S.; Park, J.; and Han, B. 2020. Rotation-Invariant Local-to-Global Representation Learning for 3D Point Cloud. In *NeurIPS*.

Li, F.; Fujiwara, K.; Okura, F.; and Matsushita, Y. 2021a. A Closer Look at Rotation-invariant Deep Point Cloud Analysis. In *ICCV*.

Li, X.; Li, R.; Chen, G.; Fu, C.; Cohen-Or, D.; and Heng, P. 2021b. A Rotation-Invariant Framework for Deep Point Cloud Analysis. In *TVCG*.

Liu, Y.; Fan, B.; Xiang, S.; and Pan, C. 2019. Relation-Shape Convolutional Neural Network for Point Cloud Analysis. In *CVPR*.

Luo, S.; Li, J.; Guan, J.; Su, Y.; Cheng, C.; Peng, J.; and Ma, J. 2022. Equivariant Point Cloud Analysis via Learning Orientations for Message Passing. In *CVPR*.

Poulenard, A.; and Guibas, L. J. 2021. A Functional Approach to Rotation Equivariant Non-Linearities for Tensor Field Networks. In *CVPR*.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *CVPR*.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *NeurIPS*.

Rao, Y.; Lu, J.; and Zhou, J. 2019. Spherical Fractal Convolutional Neural Networks for Point Cloud Recognition. In *CVPR*.

Sajnani, R.; Poulenard, A.; Jain, J.; Dua, R.; Guibas, L. J.; and Sridhar, S. 2022. ConDor: Self-Supervised Canonicalization of 3D Pose for Partial Shapes. In *CVPR*.

Sun, W.; Tagliasacchi, A.; Deng, B.; Sabour, S.; Yazdani, S.; Hinton, G. E.; and Yi, K. M. 2021. Canonical Capsules: Self-Supervised Capsules in Canonical Pose. In *NeurIPS*.

Thomas, H.; Qi, C. R.; Deschaud, J.; Marcotegui, B.; Goulette, F.; and Guibas, L. J. 2019. KPConv: Flexible and Deformable Convolution for Point Clouds. In *ICCV*.

Uy, M. A.; Pham, Q.; Hua, B.; Nguyen, D. T.; and Yeung, S. 2019. Revisiting Point Cloud Classification: A New Benchmark Dataset and Classification Model on Real-World Data. In *ICCV*.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic Graph CNN for Learning on Point Clouds. In *ACM ToG*.

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3D ShapeNets: A Deep Representation for Volumetric Shapes. In *CVPR*.

Xiang, T.; Zhang, C.; Song, Y.; Yu, J.; and Cai, W. 2021. Walk in the Cloud: Learning Curves for Point Clouds Shape Analysis. In *ICCV*.

Xu, J.; Tang, X.; Zhu, Y.; Sun, J.; and Pu, S. 2021. SGMNet: Learning Rotation-Invariant Point Cloud Representations via Sorted Gram Matrix. In *ICCV*.

Yi, L.; Kim, V. G.; Ceylan, D.; Shen, I.; Yan, M.; Su, H.; Lu, C.; Huang, Q.; Sheffer, A.; and Guibas, L. J. 2016. A Scalable Active Framework for Region Annotation in 3D Shape Collections. In *ACM ToG*.

Yu, J.; Zhang, C.; Wang, H.; Zhang, D.; Song, Y.; Xiang, T.; Liu, D.; and Cai, W. 2021. 3D Medical Point Transformer: Introducing Convolution to Attention Networks for Medical Point Cloud Analysis. In *CoRR*.

Zhang, Z.; Hua, B.; Chen, W.; Tian, Y.; and Yeung, S. 2020. Global Context Aware Convolutions for 3D Point Cloud Understanding. In *3DV*.

Zhang, Z.; Hua, B.; Rosen, D. W.; and Yeung, S. 2019. Rotation Invariant Convolutions for 3D Point Clouds Deep Learning. In *3DV*.

Zhao, C.; Yang, J.; Xiong, X.; Zhu, A.; Cao, Z.; and Li, X. 2022. Rotation Invariant Point Cloud Analysis: Where Local Geometry Meets Global Topology. In *PR*.

Zhao, H.; Jiang, L.; Fu, C.; and Jia, J. 2019. PointWeb: Enhancing Local Neighborhood Features for Point Cloud Processing. In *CVPR*.

Zhao, Y.; Birdal, T.; Lenssen, J. E.; Menegatti, E.; Guibas, L. J.; and Tombari, F. 2020. Quaternion Equivariant Capsule Networks for 3D Point Clouds. In *ECCV*.